PDF association

Specification

Document management applications – Raster Image Transport and Storage

Use of ISO 32000 (PDF/raster)

Version 1.0
2017-07

PDF/R TWG

PDF Association

Neue Kantstrasse 14
14057 Berlin, Germany

Tel: +49 (0)30 39 40 50-0

Fax: +49 (0)30 39 40 50-99

E-mail: copyright@pdfa.org

Web: www.pdfa.org

TWAIN Working Group

4256 Redspire Lane
Fayetteville, NC 28306, USA

Tel: +1 910 574 6631

Email: copyright@twain.org

Web: www.twain.org

Published in Germany and the United States of America

Table of Contents

Foreword

This document describes PDF/raster, a strict subset of the PDF file format, for storing, transporting and exchanging multi-page raster-image documents, especially scanned documents. PDF/raster provides the portability of PDF while offering the core functionality of TIFF. Bitonal, grayscale and RGB images are supported. Compression options include JPEG, lossless CCITT Group 4 Fax and uncompressed.

PDF/raster was created by collaboration between the TWAIN Working Group, which originated the PDF/raster concept, and the PDF Association, which provided PDF technology expertise and perspective as well as means of communicating with the PDF software industry to ensure a diverse range of relevant viewpoints was represented.

PDF/raster is pronounced "P. D. F. Raster".

Intellectual property

This specification describes the restrictions that differentiate a PDF/raster file from a standard PDF file. Additionally, it specifies (see clause 5) that a comment is used to identify files claiming to be PDF/raster files. There is no intention herein to claim any intellectual property that is not present in the existing PDF standard, nor claim any IP that is covered therein.

Introduction

PDF/raster is intended to be a standard format for storing, transporting and exchanging scanned documents. As a subset of PDF, it takes advantage of the widespread support for viewing, printing and processing PDF files. As a narrowly restricted subset of PDF, it is much simpler to generate and interpret, allowing it to replace the TIFF and JPEG file formats for capture and delivery of scanner output.

PDF/raster imposes many restrictions on PDF content and layout, for the following benefits:

- files can be read and written without a full PDF parser or generator
- files can be created efficiently from raster images
- files can be generated using a fixed-size raster data buffer
- less than 1KB needs to be retained per page while generating a multi-page file
- images can be located and read efficiently with comparatively simple code
- PDF/raster files can be quickly and easily identified as such by software
- PDF/raster supports effective and readily available compression algorithms

PDF/raster has important advantages over the full PDF format for storing scanned documents:

- the exact original raster image data can be recovered
- a complex rendering engine is not required
- it provides a precise, well-defined target, simplifying engineering design and testing.

PDF/raster retains optional PDF security features useful for protecting content:

- encryption is allowed for implementations that need to protect document content at rest

PDF/raster retains optional PDF digital signature features useful for authenticating content:

- one or more digital signatures may be used for implementations that require verification of the document origin, authenticity, date or time of creation, and so on

PDF/raster has important advantages over TIFF and JPEG for storing scanned documents:

- Compared to TIFF, it has far fewer and simpler variants.
- Compared to TIFF, compression is simpler and better standardized and supported.
- Compared to TIFF, PDF files can be natively viewed and printed on more platforms.
- Unlike JPEG, it is natively multi-page and handles bitonal images

1 Scope

This document defines a subset of ISO 32000-1 (PDF 1.7) suitable for storage, transport and exchange of multi-page raster-image documents, including but not limited to scanned documents. Bitonal, grayscale and RGB images are supported. Compression options for image data streams include JPEG, CCITT Group 4 Fax and uncompressed.

2 Normative references

IEC 61966-2-1:1999, *Multimedia systems and equipment – Colour measurement and management – Part 2-1: Colour management – Default RGB colour space – sRGB*

ISO 15076-1:2005, *Image technology colour management – Architecture, profile format and data structure – Part 1: Based on ICC.1:2004-10*

ISO 32000-1:2008, *Document management -- Portable document format -- Part 1: PDF 1.7*

ISO 32000-2:2017, *Document management -- Portable document format -- Part 2: PDF 2.0*

ISO 19005-1:2005, *Document management – Electronic document file format for long-term preservation – Part 1: Use of PDF 1.4 (PDF/A-1)*

ISO 19005-2:2011, *Document management – Electronic document file format for long-term preservation – Part 2: Use of ISO 32000-1 (PDF/A-2)*

ISO/IEC 646:1991 *Information technology – ISO 7-bit coded character set for information interchange*

XMP: Extensible Metadata Platform, (September 2005), Adobe Systems Incorporated

NOTE This is the version of the XMP specification referenced by ISO 32000-1. Four years after the publication of ISO 32000-1 an ISO version of XMP was published as ISO 16684-1:2012, Graphic technology – Extensible metadata platform (XMP) specification – Part 1: Data model, serialization and core properties. In addition, part 2 of the ISO 16684 series defines an approach to expressing XMP schemas using Relax NG: ISO 16684-2:2014, Graphic technology -- Extensible metadata platform (XMP) -- Part 2: Description of XMP schemas using RELAX NG.

3 Terms and definitions

3.1 page image

image of one side of a physical page

3.2 physical page

a physical media object with two sides

3.3 page object

PDF object that describes a page

3.4 binary data

an ordered sequence of arbitrary bytes

3.5 unencrypted PDF/raster file

a file conforming to this PDF/raster specification that does not contain an **Encrypt** dictionary in the trailer dictionary

3.6 encrypted PDF/raster file

a file conforming to this PDF/raster specification that does contain an **Encrypt** dictionary in the trailer dictionary

4 Notation

PDF operators, PDF keywords, the names of keys in PDF dictionaries, and other predefined names are written in bold font; operands of PDF operators or values of dictionary keys are written in italic font. Some names can also be used as values, depending on the context, and so the styling of the content will be context-specific.

EXAMPLE 1 The *Sig* value for the **FT** key.

Token characters used to delimit objects and describe the structure of PDF files, as defined in ISO 32000-1:2008, 7.2.1 may be identified by their ISO/IEC 646-character name written in uppercase in bold font followed by a parenthetic two-digit hexadecimal character value with the suffix “h”.

EXAMPLE 2 **CARRIAGE RETURN** (0Dh).

Text string characters, as defined in ISO 32000-1:2008, 7.9.2 may be identified by their ISO/IEC 10646 character name written in uppercase in bold font followed by a parenthetic four digit hexadecimal character code value with the prefix “U+”.

EXAMPLE 3 **EN SPACE** (U+2002).

5 Version identification

A PDF file conforming to the PDF/raster specification is identified by one comment line near the end of the file, immediately before the last occurrence of the line in the file containing the **startxref** key. The characters between the '%' character at the beginning of the line and the end of line character shall be

PDF-raster-x.y

where 'x' – the number before the decimal point – is the major version number and 'y' – the number after the decimal point – is the minor version number.

The PDF/raster version number for PDF files conforming to this document shall be 1.0. New major versions may be incompatible with previous versions; minor versions are expected to not break existing readers.

This comment line marks the file as intended to comply with this specification.

EXAMPLE

```
trailer
<<
/Info 58 0 R
/Size 59
/Root 1 0 R
/ID
   [     <D7916DF85B0EE1998036EA145A1CE7B4>
         <D7916DF85B0EE1998036EA145A1CE7B4>
   ]
>>
%PDF-raster-1.0
startxref
177317
%%EOF
```

6 Conformance requirements

6.1 General

A conforming PDF/raster file shall conform to all requirements listed in subclauses 6.2, “PDF subset” through 6.8, “Encryption”.

6.2 PDF subset

6.2.1 General

Conformance of unencrypted and encrypted PDF/raster files only differs regarding the use of encryption. Encrypted PDF/raster files make use of encryption features introduced in ISO 32000-2, and not available in ISO 32000-1. The definition of, and the requirements for, any other feature allowed in a PDF/raster file do not differ between ISO 32000-1 and ISO 32000-2. For the sake of simplicity, all requirements for PDF/raster files with the exception of those for the use of encryption are specified on the background of ISO 32000-1.

6.2.2 Unencrypted PDF/raster files

A PDF/raster-conforming file that is not encrypted shall adhere to all requirements of ISO 32000-1 as modified by this specification.

The header shall be one of the following:

```
%PDF-1.4
%PDF-1.5
%PDF-1.6
%PDF-1.7
```

Only the following filters shall be allowed in an unencrypted PDF/raster file:

- *FlateDecode*
- *CCITTFaxDecode* (only for bitonal images)
- *DCTDecode* (only for 8-bit grayscale or RGB images)

6.2.3 Encrypted PDF/raster files

A PDF/raster-conforming file that is encrypted shall adhere to all requirements of ISO 32000-1, as modified by this specification, with the following exceptions:

- The header shall be `%PDF-2.0`.
- The file shall adhere to all requirements of ISO 32000-2, 7.6, “Encryption”, as modified by 6.8, “Encryption”, in this specification.

Only the following filters shall be allowed in an encrypted PDF/raster file:

- *FlateDecode*
- *CCITTFaxDecode* (only for bitonal images)
- *DCTDecode* (only for 8 bit grayscale or RGB images)
- *Crypt*

6.2.4 Unencrypted and encrypted PDF/raster files

All indirect references shall have a generation number equal to zero.

All indirect references shall refer to valid objects.

NOTE 1 This precludes indirect object references to a non-existent object as described in ISO 32000-1:2008, 7.3.9, "Null Object".

Stream dictionaries shall not contain a **Type** key with a value of *ObjStm*.

NOTE 2 This precludes the use of object streams described in ISO 32000-1:2008, 7.5.7, "Object streams".

6.3 Catalog dictionary

The **Catalog** dictionary shall contain the entries required by ISO 32000-1, Table 28, and shall not contain any optional entries except zero, one or more of the following entries: **Version**, **ViewerPreferences**, **PageLayout**, **PageMode**, **AcroForm**, and **Metadata**.

6.4 Metadata

6.4.1 General

The **Catalog** dictionary of a conforming file may contain the **Metadata** key whose value is a metadata stream as defined in ISO 32000-1:2008, 14.3.2.

Page dictionaries may contain the **Metadata** key whose value is a metadata stream as defined in ISO 32000-1:2008, 14.3.2. This metadata stream, if present, shall contain entries with metadata specific to the page object.

6.4.2 Document level and page level metadata streams

The document level metadata stream and page level metadata streams may use properties defined in XMP: Extensible Metadata Platform, (September 2005), or custom properties. Where custom properties are used, namespaces shall be used in such a fashion that conflicts are avoided with other entries using the same property name. Each organisation wishing to define and use its own custom properties shall define a suitable namespace based on a URL that is under the organisation's control.

EXAMPLE 1 Examples for namespaces based on which custom properties can be defined:

- http://ns.twain.org/ns/pdfraster/v1/extra_metadata
- http://ns.twain.org/ns/pdfraster/v1/some_other_fields
- http://ns.some_company.com/ns/pdf_raster/version_1/company_specific_fields

EXAMPLE 2 Properties using the same name that are based on different namespaces:

```
<rdf:Description
   rdf:about=""
   xmlns:org_a="http://ns.org_a.com/pdfraster/1.0/"
   xmlns:org_b="http://ns.org_b.com/pdfraster/1.0/"
   <org_a:JobID>ABC-123</org_a:JobID>
```



```
    <org_b:JobID>987-654-321:tre-hgf-bvc</org_b:JobID>
</rdf:Description>
```

The TWAIN Working Group provides guidance regarding metadata properties for scanned images, see www.twaindirect.org.

6.4.3 Document information dictionary

A document information dictionary may appear within a conforming file. It shall contain no entries other than **Creator**, **Producer**, **CreationDate**, and **ModDate**. Each of these entries, if present, shall be represented by the corresponding XMP property values in the document's metadata stream, if such a metadata stream is present.

Document information dictionary		**Document level metadata stream**	
Entry	**PDF type**	**Property**	**XMP type**
Creator	text string	xmp:CreatorTool	AgentName
Producer	text string	pdf:Producer	AgenName
CreationDate	date	xmp:CreateDate	Date
ModDate	date	xmp:ModifyDate	Date

Table 1: Mapping document information dictionary to corresponding XMP properties

6.5 Page objects

6.5.1 General

Each page image is represented by a PDF page object. The page object is a dictionary, and shall be constructed as mandated by ISO 32000-1:2008.

Each page object shall contain the entries required by ISO 32000-1, Table 30, and by ISO 32000-1, Table 5, and shall not contain any optional entries except zero, one or more of the following entries: **Contents**, **Rotate**, **Metadata**, **Annots**, and **PZ**.

6.5.2 Page tree nodes

Page tree nodes shall not contain any entries other than those required by ISO 32000-1, Table 29.

NOTE This provision effectively prohibits the inheritance of such entries. This also applies to the **MediaBox** key. Thus, inheritance of the **MediaBox** key is not possible in a PDF/raster file.

6.5.3 Media box

Each page object shall contain a **MediaBox** entry whose value shall be of the form [*0 0 w h*] where *w* is the width of the page and *h* is the height.

NOTE The **MediaBox** reflects the size of the page and thus the page image represented on it

prior to any rotation specified by the **Rotate** entry.

EXAMPLE An ISO A4 sized page would have a **MediaBox** value of *[0 0 595.27559 841.88976]*.

6.5.4 Annots array and digital signatures

If present, the **Annots** array in a page object shall only contain widget annotations. Such widget annotations shall have a value of *Sig* for the **FT** entry.

NOTE 1 This provision effectively limits the presence of annotations to widget annotations representing digital signatures.

For any widget annotation, the width and the height of its **Rect** entry shall be zero.

NOTE 2 This effectively prohibits the creation of a digital signature that renders a visual presentation on the page.

6.5.5 Resources dictionary

Each page object shall contain a **Resources** entry. Each page object's **Resources** dictionary shall contain an **XObject** dictionary which shall contain one or more image **XObject** resources which, for the purpose of this international standard, are called 'strips' (cf. 6.6). Strips shall be listed in a page object's **XObject** dictionary. Their order of appearance on the rendered page, from top to bottom, ignoring rotation in case the **Rotate** entry is present for the page, shall be reflected by each strip's name. The first strip on the page, if present, shall be named "strip0", and the following strips on the page, if any, shall be named "strip1", "strip2", "strip3", and so on. The **XObject** dictionary in a page object's **Resources** dictionary shall not contain any other keys.

EXAMPLE Determining the order of strips in an XObject dictionary

```
/XObject <<
   /strip2    … indirect object reference …
   /strip0    … indirect object reference …
   /strip1    … indirect object reference …
>>
```

This is valid and establishes the order as being strip0, strip1, strip2.

6.5.6 Rotation

Any page object may contain a **Rotate** entry as defined in ISO 32000-1:2008, "Table 30 – Entries in a page object". Page tree nodes shall not contain the **Rotate** key.

NOTE This provision effectively prohibits the inheritance of the **Rotate** key.

6.5.7 Contents stream

Each page object shall contain a **Contents** stream, which draws the strips of the page image contiguously to fill the **MediaBox**. The value of the **Contents** entry in a page object shall always be a single stream.

NOTE 1 This prohibits the use of an array as the value of a **Contents** key.

Each contents stream shall contain at least one **Do** operator that references a strip.

NOTE 2 This implies, that the page cannot be empty.

Only the following operators shall be present in a page object's **Contents** stream:

q

Q

cm

Do

NOTE 3 This implies that a **Contents** stream only draws the strips for a page image 'as is' – e.g. no clipping or masks are applied –, and does not draw anything else. Images can only be present in the form of image XObjects, not as inline images.

NOTE 4 While the **ri** operator is prohibited inside content streams, a rendering intent can still be set by means of an **Intent** entry in an image XObject.

6.6 Strips

6.6.1 General

Each strip shall be represented by an image XObject as described in ISO 32000-1:2008, 8.9.5, "Image Dictionaries". No entries other than **Type**, **Subtype**, **Length**, **Filter**, **DecodeParms**, **Width**, **Height**, **ColorSpace**, **BitsPerComponent** and **Intent** shall be present.

NOTE 1 The presence of the entries **Subtype**, **Width**, **Height**, **Length** is always required. The presence of the **ColorSpace** entry precludes that the image **XObject** is a mask.

Strips shall be either bitonal, grayscale or RGB images, as defined in 6.6.2, "Bitonal images", 6.6.3, "Grayscale images", and 6.6.4, "RGB images".

All the strips of a page image shall have the same value for the **Width** entry and shall all contain the same entries for **ColorSpace** and **BitsPerComponent**. The effective resolution of the strips of a page image shall be the same in the **Width** direction between all strips on the page, and shall be the same in the **Height** direction between all strips on the page.

NOTE 2 This implies that the horizontal resolution of an image XObject – regardless whether it is the only one on the page or whether it represents one of several strips – can differ from the vertical resolution.

The **Intent** entry shall either be present or be absent for all strips of a page image, and if present, shall have the same value for all strips of a page image.

All strips of a given page shall appear in the file in the order of appearance on the page, as defined in 6.5.5, "Resources dictionary".

All the strips of page N shall appear in the file before any strips of page N+1.

NOTE 3 As there will always be a risk of gap artifacts between strips when rendered to an output device, it is best to avoid use of strips except where really necessary (e.g. in low memory or high resolution conditions). For a discussion of how rendering of page content to an output device is carried out, see ISO 32000-1, "10.6.4 Scan Conversion Rules".

NOTE 4 When pages are created using strips, it is advised to use exactly one **cm** operator per invocation of the **Do** operator, and to bracket such **cm** operator and the **Do** operator inside a pair of **q**/**Q** operators, to avoid concatenation of transformation matrix calculations, which tends to increase the risk of – albeit very small – rounding errors that could lead to gap artifacts.

6.6.2 Bitonal images

Bitonal images shall be represented by an image **XObject** dictionary with *DeviceGray* or *CalGray* as the value of its **ColorSpace** entry, and *1* as the value for its **BitsPerComponent** entry.

If *CalGray* is used for a bitonal image, the **Gamma** entry shall be present in the *CalGray* colour space dictionary with a value of *2.2*.

NOTE 1 In most cases, no device independent colour space, such as **CalGray**, is used for bitonal images, and **DeviceGray** is used instead.

The **BlackIs1** entry shall not be present, or shall have a value of false. The **Decode** entry shall not be present, or shall have a value of *[0.0 1.0]*.

NOTE 2 This guarantees that a pixel value of *0* always represents black and *1* always represents white.

The **Filter** entry shall not be present, or if present, its value shall either be *null* or *CCITTFaxDecode*, in which case the otherwise optional parameter **K** entry shall be present with a value of *-1*.

6.6.3 Grayscale images

Grayscale images shall be represented by an image **XObject** dictionary with *CalGray* as the value of its **ColorSpace** entry, and *8* or *16* as the value for its **BitsPerComponent** entry.

The **Gamma** entry shall be present in the **CalGray** colour space dictionary with a value of *2.2*.

The value *0* for components in the raw image data stream shall represent black and the maximum representable value shall represent white.

In the **CalGray** colour space dictionary, the **WhitePoint** entry should be present, and the **BlackPoint** entry may be present in order to represent the scanned image as accurately as possible.

8-bit grayscale images shall have either *DCTDecode* or *null* as the value of the **Filter** entry, if present. 16-bit grayscale images shall have *null* as the value for the **Filter** entry, if present.

NOTE This implies that 8-bit images may be JPEG compressed or uncompressed, whereas 16-bit images will always be uncompressed.

6.6.4 RGB images

RGB images shall be represented by an image **XObject** dictionary with *ICCBased* or *CalRGB* as the value of its **ColorSpace** entry, and 8 or 16 as the value for its **BitsPerComponent** entry.

The ICC profile referenced by the **ICCBased** colour space should represent the sRGB colour space as defined in IEC 61966-2-1, but may represent other colour spaces where deemed necessary. The ICC profile stream dictionary may contain an **Alternate** entry which, if present, shall have a value of *DeviceRGB*.

NOTE 1 As per ISO 32000-1:2008, readers are allowed to ignore the ICC profile in an RGB ICCBased colour space and treat the image as DeviceRGB, whether or not this is specified as the Alternate colour space.

8-bit RGB images shall have either *DCTDecode* or null as the value of the **Filter** entry, if present. 16-bit RGB images shall have *null* as the value for the **Filter** entry, if present.

NOTE 2 This implies that 8-bit images may be JPEG compressed or uncompressed, whereas 16-bit images will always be uncompressed.

6.7 Incremental updates

It is expected that there will hardly ever be a need to incrementally update PDF/raster files. Rather, a PDF/raster file would always be rewritten completely in case it needs to be modified. It is only when applying digital signatures that the need to apply an incremental update to an existing PDF/raster file can arise. This is especially true when a digital signature already has been applied – only by adding a second digital signature in the form of an incremental update it is possible to maintain the integrity of the PDF/raster file and the already applied digital signature.

Accordingly, PDF/raster file shall not include incremental updates unless such incremental updates have been applied exclusively in order to represent digital signatures applied to the PDF/raster file.

6.8 Encryption

Encryption may be used in a PDF/raster file. If a file is encrypted:

- It shall include an **Encrypt** dictionary in the trailer dictionary.
- The encryption dictionary shall specify the Standard security handler (cf. ISO 32000-2:2017, 7.6.4. “Standard security handler”), and use the AES algorithm and a key length of 256.
- The value of the **V** key in the encryption dictionary shall be *5*.

Readers may support decryption of encrypted PDF/raster files, depending on their needs. If a reader that does not support encrypted PDF/raster files is asked to read an encrypted PDF/raster file, whatever error it returns or reports shall indicate specifically that the failure is due to the reader not supporting encrypted PDF/raster files, or not supporting the particular encryption encountered.

The associated generation, communication and storage of encryption keys and/or passwords is outside the scope of this specification.

Annex A: Application Notes

(informative)

These notes are not part of the definition of PDF/raster. They are to guide developers in interpreting and implementing the standard to achieve the greatest success in storing and communicating raster images.

A. 1. Scan order vs. orientation

It is expected that for efficiency and simplicity, scanners will write page data into PDF/raster files in 'scan order' i.e. the first row of data received from the leading edge of the physical page will be written out as the first row of strip0. The scanner can use the **Rotate** entry for a page object to specify the 'true' or desired page orientation for viewing and printing.

If a scanner has the resources, it has the option to rotate pages into viewing orientation (first row => visually top row) before writing them – adjusting the **Rotate** attribute to *0*, of course. This saves some work and time for readers.

A. 2. Calculating the MediaBox

It is recommended that writers derive the MediaBox values using the pixel dimensions of the scan and the intended pixels per inch (ppi), assuming that UserUnit has a value of 1.0:

$$\text{‹}width\ of\ MediaBox\text{›} = \frac{72 \times \text{‹}number\ of\ horizontal\ pixels\text{›}}{\text{‹}horizontal\ resolution\ in\ pixels\ per\ inch\text{›}}$$

$$\text{‹}height\ of\ MediaBox\text{›} = \frac{72 \times \text{‹}number\ of\ vertical\ pixels\text{›}}{\text{‹}vertical\ resolution\ in\ pixels\ per\ inch\text{›}}$$

The value of the **MediaBox** would then be written as:

[0 0 ‹width of MediaBox› ‹height of MediaBox›]

EXAMPLE

- DIN A 4 size is defined as 210 mm by 297 mm
- 210 mm is equivalent to 595.275590551181 pt
- For the purpose of this specification a width of 210 mm would be rounded to 595.27559
- Accordingly 297mm - equivalent to 841.889763779528 - would be rounded to 841.88976 pt
- Thus (assuming **UserUnit** is equal to 1), the MediaBox for a portrait DIN A 4 size page would be written as:
  `/MediaBox [0 0 595.27559 841.88976]`
- A Letter size page is defined as 8.5 inch by 11 inch
- 8.5 inch is equal to 612 pt, and 11 inch is equal to 792 pt
- Thus (assuming **UserUnit** is equal to 1), the MediaBox for a portrait letter size page would be written as:
  `/MediaBox [0 0 612 792]`

A. 3. Reconstructing resolution

The resolution of a page image can be reconstructed from the **MediaBox** and strip dimensions:

$$\text{‹horizontal resolution in ppi›} = \frac{72 \times \text{‹width of first strip›}}{\text{‹width of MediaBox›}}$$

$$\begin{aligned}\text{‹total height›} &= \text{‹height of first strip›} \\ &\quad + \text{‹height of second strip›} + \ldots + \text{‹height of last strip›}\end{aligned}$$

$$\text{‹vertical resolution in ppi›} = \frac{72 \times \text{‹total height›}}{\text{‹height of MediaBox›}}$$

In the range from 25 to 4000, it is recommended to round these ppi values to the nearest tenth (0.1).

NOTE If the **Rotate** key for a page is present with a value of 90 or 270 (degrees) the two ppi values need to be interchanged to obtain the ppi values of the viewed image.

A. 4. Value ranges

Following ISO 32000-1:2008, it is recommended that a PDF/raster file does not use any of the following:

- any integer greater than 2147483647 or less than -2147483648.
- any real number outside the range of +/-3.403 x 10^38.
- any real number closer to zero than +/-1.175 x 10^-38.
- any string longer than 32767 bytes.
- any name longer than 127 bytes.
- more than 8388607 indirect objects.
- any q/Q pairs nested deeper than 28 levels.
- any page boundaries less than 3 units or greater than 14400 units in either direction.
- a value of PZ less than 8 (percent) or greater than 6400 (percent)

A. 5. PDF/A-conforming PDF/raster files

It is possible to create PDF/raster files that also conform to PDF/A. To achieve this result, at least the following aspects need to be taken into account:

- If the colour spaces for all strips are device independent colour spaces it is not necessary to include an **OutputIntent**. For bitonal images, PDF/raster allows both device dependent **DeviceGray** and device independent **CalGray**. In order to save the need to include an **OutputIntent** it is thus necessary to use **CalGray** for bitonal images.
- PDF/A requires the presence of document level XMP metadata, including PDF/A part number (e.g. ‘1’ for PDF/A-1) and conformance level (e.g. “B” for conformance level B).
- For certain entries present in the document information dictionary it can be necessary to mirror them in the document level XMP metadata. As the presence of entries in the document information dictionary is not required by PDF/A, it is usually easier to only include metadata entries – if any – in the document-level XMP metadata.

- As the functionality in PDF/raster files is relatively limited, only PDF/A-1b or PDF/A-2b conformance are possible. For the purposes of PDF/raster, the difference between PDF/A-1b and PDF/A-2b is very minor.

As PDF/A effectively prohibits encryption, only unencrypted PDF/raster files can also conform to PDF/A.

A. 6. Encryption of PDF/raster files

Encryption of PDF/raster is supported through security handlers as described in ISO 32000-2:2017 7.6 "Encryption" with the following constraint: For the **V** key in the encryption dictionary only the value *5* can be used. A value of *5* for **V** permits the specification of crypt filters with a key length of 256 bits (32 bytes).

The "Algorithm 1.A: Encryption of data using the AES algorithms" as described in ISO 32000-2 7.6.3.2 has to be used.

The main implication is that Algorithm 1.A uses the starting key directly, and does not modify the key at all. Algorithm 1.A is used only with the AES algorithm and 256-bit keys.

The **R** (revision) key in the *Additional encryption* dictionary as described in ISO 32000-2:2017, Table 21 is required to have value of *6* if the document is encrypted with a **V** value of *5*. In that case, the Standard Security handler has to behave as described in ISO 32000-2, 7.6.4, "Standard security handler".

It is up to each reader implementation to support or not support decryption of encrypted PDF/raster files. If a reader that does not support encrypted PDF/raster files is asked to read an encrypted PDF/raster file, whatever error it returns or reports shall indicate specifically whether the failure is due to the reader not supporting encrypted PDF/raster files, or whether it is due to not supporting the particular encryption method encountered.

The generation, exchange, entry and storage of encryption keys and/or passwords is outside the scope of this specification.

A. 7. Approximate sRBG using CalRGB

In some cases where the sRGB colour space accurately characterises colour information it can be desirable to use a more compact colour characterisation in the form of a **CalRGB** colour space, even when sacrificing some accuracy. A **CalRGB** colour space that approximates the sRGB colour space can be defined as shown in the example below.

EXAMPLE

```
[
   /CalRGB
   <<
          /Matrix
                 [
                        0.412384 0.212646  0.0193176
                        0.35759  0.715164  0.119171
```



```
                        0.180496 0.0721893 0.950546
                ]
        /WhitePoint
                [
                        0.950455 1 1.08905
                ]
        /BlackPoint
                [
                        0 0 0
                ]
        /Gamma
                [
                        2.2 2.2 2.2
                ]
    >>
]
```

A. 8. PDF/raster example

The example below shows the source code for a minimalistic sample file, consisting of one JPEG-compressed RGB page image using an ICC profile as its colour characterisation.

EXAMPLE

```
%PDF-1.4
%âãÏÓ
1 0 obj
<<
    /Type/Pages
    /Count 1
    /Kids
            [
                    5 0 R
            ]
>>
endobj

2 0 obj
<<
    /Type/Catalog
    /Metadata 4 0 R
    /Pages 1 0 R
>>
endobj

3 0 obj
<<
    /CreationDate    (D:20160923145104-05'00)
    /Creator (PDF/raster sample file creator 1.0)
    /Producer        (PDF/raster sample file producer 1.0)
    /Title   (PDF/raster sample file)
>>
endobj

4 0 obj
<<
    /Type/Metadata
    /Length 928
    /Subtype/XML
>>
stream
<?xpacket begin="" id="W5M0MpCehiHzreSzNTczkc9d"?>
<x:xmpmeta xmlns:x="adobe:ns:meta/">
    <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
            <rdf:Description rdf:about="" xmlns:dc="http://purl.org/dc/elements/1.1/">
                    <dc:title>
                            <rdf:Alt>
                                    <rdf:li xml:lang="x-default">PDF/raster sample file</rdf:li>
                            </rdf:Alt>
```



```
                </dc:title>
            </rdf:Description>
            <rdf:Description rdf:about="" xmlns:xap="http://ns.adobe.com/xap/1.0/">
                <xap:CreateDate>2016-09-23T14:51:04-05:00</xap:CreateDate>
                <xap:CreatorTool>PDF/raster sample file creator 1.0</xap:CreatorTool>
            </rdf:Description>
            <rdf:Description rdf:about="" xmlns:pdf="http://ns.adobe.com/pdf/1.3/">
                <pdf:Producer>PDF/raster sample file producer 1.0</pdf:Producer>
            </rdf:Description>
            <rdf:Description rdf:about="" xmlns:xapMM="http://ns.adobe.com/xap/1.0/mm/">
                <xapMM:DocumentID>uuid:42646CE2-2A6C-482A-BC04-030FDD35</xapMM:DocumentID>
            </rdf:Description>
    </rdf:RDF>
</x:xmpmeta>
<?xpacket end="w"?>
endstream
endobj

5 0 obj
<<
    /Type/Page
    /Contents 8 0 R
    /MediaBox
            [
                0
                0
                612
                792
            ]
    /Parent 1 0 R
    /Resources
            <<
                /XObject
                        <<
                                /strip0 7 0 R
                        >>
            >>
>>
endobj

6 0 obj
<<
    /N 3
    /Filter /FlateDecode
    /Length 2575
>>
stream
    ... ICC profile (compressed binary data) ...
endstream
endobj

7 0 obj
<<
    /Type /XObject
    /Subtype /Image
    /Width 850
    /Height 1100
    /ColorSpace
            [
                /ICCBased 6 0 R
            ]
    /BitsPerComponent 8
    /Filter /DCTDecode
    /Length 9 0 R
>>
stream
    ... JPEG image data (compressed binary data) ...
endstream
endobj

8 0 obj
<<
    /Length 34
>>
stream
612 0 0 792 0 0 cm
/strip0 Do
endstream
endobj

9 0 obj
```



```
176934
endobj

xref
0 9
0000000000 65535 f
0000000015 00000 n
    ... cross reference table with entries for all indirect objects ...
0000004559 00000 n
0000181674 00000 n
trailer
<<
    /Size 9
    /Root 2 0 R
    /Info 3 0 R
    /ID
             [
                      <45df5e844f0d47e7ecc063efd84fa47a>
                      <45df5e844f0d47e7ecc063efd84fa47a>
             ]
>>
%PDF-raster-1.0
startxref
181756
%%EOF
```

