INTERNATIONAL STANDARD

ISO 23504-1

First edition
2020-07

Corrected version
2020-09

Document management applications — Raster image transport and storage —

Part 1: Use of ISO 32000 (PDF/R-1)

Applications de gestion de documents — Transport et stockage des images tramées —

Partie 1: Utilisation de l'ISO 32000 (PDF/R-1)

Reference number
ISO 23504-1:2020(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: 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.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

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. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 171, *Document management applications*, Subcommittee SC 2, *Document file formats, EDMS systems and authenticity of information*.

This corrected version of ISO 23504-1:2020 incorporates the following corrections:

— Angled brackets inserted around 'total height' in the numerator of the second formula in A.4;

— ']' added to the line before '/Whitepoint' in A.8.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

This document describes PDF/R (Raster), a strict subset of the PDF file format, for storing, transporting and exchanging multi-page raster-image documents, especially scanned documents and photographs. PDF/R 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.

This document describes the restrictions that differentiate a PDF/R file from a standard PDF file. Additionally, it specifies (see Clause 5) that a comment is used to identify files claiming to be PDF/R 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.

PDF/R 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/R 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;

— images can be located and read efficiently with comparatively simple code;

— PDF/R files can be quickly and easily identified as such by software;

— PDF/R supports effective and readily available compression algorithms.

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

— the 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/R retains optional PDF security features useful for protecting content:

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

PDF/R 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/R 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.

PDF/R was created by collaboration between the TWAIN Working Group, which originated the PDF/R 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.

Document management applications — Raster image transport and storage —

Part 1: Use of ISO 32000 (PDF/R-1)

1 Scope

This document defines a subset of ISO 32000 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

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements 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 32000-1:2008, *Document management — Portable document format — Part 1: PDF 1.7*

ISO 32000-2[1)]:2020, *Document management — Portable document format — Part 2: PDF 2.0*

3 Terms and definitions

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

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at http://www.electropedia.org/

3.1
page image
image of one side of a *physical page* (3.2)

3.2
physical page
physical media object with two sides

3.3
unencrypted PDF/R file
file conforming to this PDF/R specification that does not contain an **Encrypt** dictionary in the trailer dictionary

3.4
encrypted PDF/R file
file conforming to this PDF/R specification that does contain an **Encrypt** dictionary in the trailer dictionary

1) Under preparation. Stage at the time of publication: ISO DIS 32000-2.

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[2)] 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/R 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 comment shall be:

%PDF-raster-x.y

where

"x" (the digit before the decimal point) is the major version number

"y" (the digit after the decimal point) is the minor version number

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

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

EXAMPLE

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


2) Under preparation. Stage at the time of publication: ISO/IEC DIS 10646.

6 Conformity requirements

6.1 General

A conforming PDF/R file shall conform to all requirements listed in 6.2, "PDF subset" to 6.8, "Encryption".

6.2 PDF subset

6.2.1 General

Conformity of unencrypted and encrypted PDF/R files only differs regarding the use of encryption. Encrypted PDF/R 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/R file do not differ between ISO 32000-1 and ISO 32000-2. For the sake of simplicity, all requirements for PDF/R 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/R files

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

The header shall be one of the following:

— "`%PDF-1.4`";

— "`%PDF-1.5`";

— "`%PDF-1.6`";

— "`%PDF-1.7`".

NOTE If the contents of the file are inconsistent with the version number in the header processing results will be implementation dependent.

No filters other than the following shall be used in an unencrypted PDF/R file:

— *FlateDecode*;

— *CCITTFaxDecode* (only for bitonal images);

— *DCTDecode* (only for 8-bit grayscale or RGB images).

6.2.3 Encrypted PDF/R files

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

— the header shall be "`%PDF-2.0`";

— the file shall adhere to all requirements of ISO 32000-2:2020, 7.6, "Encryption", as modified by 6.8, "Encryption", in this document.

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

— *FlateDecode*;

— *CCITTFaxDecode* (only for bitonal images);

— *DCTDecode* (only for 8 bit grayscale or RGB images);

— *Crypt*.

6.2.4 Unencrypted and encrypted PDF/R files

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

All objects referred to be indirect references shall be listed.

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 32000-1:2008, Table 28. It 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 for which the value is a metadata stream as defined in ISO 32000-1:2008, 14.3.2.

Page dictionaries may contain the **Metadata** key for which the 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 ISO 16684-1:2019 (XMP)[5] 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 organization wishing to define and use its own custom properties shall define a suitable namespace based on a URL that is under the organization'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[1].

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**.

6.4.4 XMP Metadata

If an XMP metadata stream is present, each of the entries in the document information dictionary, shall be represented by the corresponding XMP property value. Table 1 indicates the mapping between document information dictionary and XMP properties.

Table 1 — Mapping document information dictionary to corresponding XMP properties

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

6.5 Page objects

6.5.1 General

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

Each page object shall contain the entries required by ISO 32000-1:2008, Table 30, and shall contain one **Contents** entry, and shall not contain any optional entries except zero, one or more of the following entries: **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 32000-1:2008, 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/R file.

6.5.3 Media box

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

NOTE 1 The **MediaBox** is defined in default user space coordinate units with a default value of 1/72 inch (see ISO 32000-1:2008, 8.3.2.3, “User space”).

NOTE 2 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].

See Annex A.3, "(informative) Calculating the MediaBox" for a detailed example.

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 that, for the purpose of this document, are called "strips" (see 6.6, "Strips"). 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 shall be named "strip0". 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. See A.2, "Scan order versus orientation" for a possible use case.

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

6.5.7 Content stream

Each page object shall contain the **Contents** key, with a value that is a content stream which draws the strips of the page image contiguously to fill the **MediaBox**. Each strip's **Width** direction shall be parallel with the width direction of the media box. Each strip's effective width shall be scaled to the exact width of the media box. Each strip shall be positioned fully inside the media box. 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 content stream shall contain at least one **Do** operator that references a strip.

NOTE 2 This implies that the page cannot be empty.

A page object's content stream shall contain only the following operators.

— **q**;

— **Q**;

— **cm**;

— **Do**.

NOTE 3 This implies that a content 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.

The presence of the entries **Subtype**, **Width**, **Height**, **Length** is always required. The absence of the **ImageMask** key and the absence of the *JPXDecode* filter (as defined in 6.2.2, "Unencrypted PDF/R files", and 6.2.3, "Encrypted PDF/R files") imply that the presence of the **ColorSpace** entry is also always required.

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. See Annex A.4, "(informative) Reconstructing resolution", for calculations how to reconstruct the image resolution.

NOTE 1 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. If present, it 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.

As there will always be a risk of gap artefacts between strips when rendered to an output device, a writer should not use multiple strips for one page image 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 32000-1:2008, 10.6.4 "Scan Conversion Rules".

When pages are created using strips, writers should use exactly one **cm** operator per invocation of the **Do** operator, and should 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 artefacts.

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, if present, shall have a value of *false*. The **Decode** entry, if present, 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, if present, shall have a value of *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. The maximum representable value shall represent white.

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

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

NOTE This implies that 8-bit images can 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 (ISO 15076-1:2010[4]) referenced by the **ICCBased** colour space should represent the sRGB colour space as defined in IEC 61966-2-1[3], but may represent other colour spaces where deemed necessary. The ICC profile stream dictionary shall contain an **Alternate** entry which shall have a value of *DeviceRGB*. See Annex A.8, "(informative) Approximate sRBG using CalRGB", for a simple sRGB approximation example.

NOTE 1 As per ISO 32000-1:2008, readers can ignore the ICC profile in an RGB ICCBased colour space.

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

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

Annex A.9, "(informative) PDF/R example", presents an example showing the structure of PDF/R files.

6.7 Incremental updates

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

NOTE In general, a PDF/R file is rewritten. It is only when applying digital signatures that the need to apply an incremental update to an existing PDF/R 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 is it possible to maintain the integrity of the PDF/R file and the already applied digital signature.

6.8 Encryption

Encryption may be used in a PDF/R 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 (see ISO 32000-2:2020, 7.6.4, “Standard security handler”), use the AES algorithm and have a key length of 256.

— the value of the **V** key in the encryption dictionary shall be *5*.

The associated generation, communication and storage of encryption keys and/or passwords is outside the scope of this document. See A.7, "Encryption of PDF/R files", for implementation notes.

Annex A
(informative)

Application notes

A.1 General

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

A.2 Scan order versus orientation

It is expected that, for efficiency and simplicity, scanners will write page data into PDF/R 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.3 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*:

$$\langle width\ of\ MediaBox\rangle = \frac{72\times\langle number\ of\ horizontal\ pixels\rangle}{\langle horizontal\ resolution\ in\ pixels\ per\ inch\rangle}$$

$$\langle height\ of\ MediaBox\rangle = \frac{72\times\langle number\ of\ vertical\ pixels\rangle}{\langle vertical\ resolution\ in\ pixels\ per\ inch\rangle}$$

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 595.275 590 551 181 pt.

— For the purpose of this document, a width of 210 mm would be rounded to 595.275 59 pt.

— Accordingly 297 mm (equivalent to 841.889 763 779 528 pt) would be rounded to 841.889 76 pt.

— Thus the **MediaBox** for a portrait DIN A4-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 the **MediaBox** for a portrait Letter-size page would be written as:

```
/MediaBox [0 0 612 792]
```

A.4 Reconstructing resolution

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

$$\langle horizontal\ resolution\ in\ ppi\rangle = \frac{72\times\langle width\ of\ \ first\ strip\ in\ pixels\rangle}{\langle width\ of\ \ MediaBox\ in\ points\rangle}$$

$$\langle total\ height\rangle = \langle height\ of\ \ first\ strip\ in\ pixels\rangle + \langle height\ of\ \ second\ strip\ in\ pixels\rangle + \ldots + \langle height\ of\ \ last\ strip\ in\ pixels\rangle$$

$$\langle vertical\ resolution\ in\ ppi\rangle = \frac{72^{*}\langle total\ height\rangle}{\langle height\ of\ \ MediaBox\ in\ points\rangle}$$

In the range from 25 to 4 000, it is recommended to round these ppi values to the nearest tenth (0,1).

If the **Rotate** key is present in a page dictionary with a value of 90 or 270 (degrees), the two ppi values are interchanged to obtain the ppi values of the viewed image.

A.5 Value ranges

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

— any integer greater than 214 748 364 7 or less than –214 748 364 8;

— any real number outside the range of ±3.403 × 10^38;

— any real number closer to zero than ±1.175 × 10^–38;

— any string longer than 32 767 bytes;

— any name longer than 127 bytes;

— more than 838 860 7 indirect objects;

— any q/Q pairs nested deeper than 28 levels;

— any page boundaries less than 3 units or greater than 14 400 units in either direction;

— a value of **PZ** less than 8 (per cent) or greater than 6 400 (per cent).

A.6 PDF/A-conforming PDF/R files

It is possible to create PDF/R files that also conform to PDF/A[2]. 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/R allows both device dependent **DeviceGray** and device independent **CalGray**. In order to not require an **OutputIntent** it is necessary to use **CalGray** for bitonal images.

NOTE 6.3 in this document prohibits the inclusion of an output intent.

— PDF/A requires the presence of document level XMP metadata, including PDF/A part number (e.g. “1” for PDF/A-1) and conformity level (e.g. “B” for conformity level B).

— For certain entries present in the document information dictionary it might 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/R files is relatively limited, only PDF/A-1b or PDF/A-2b conformity is possible. For the purposes of PDF/R, the difference between PDF/A-1b and PDF/A-2b is very minor.

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

A.7 Encryption of PDF/R files

It is up to each reader implementation to support or not support the decryption of encrypted PDF/R files.

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

A.8 Approximate sRBG using CalRGB

In some cases where the sRGB colour space accurately characterizes colour information it can be desirable to use a more compact colour characterization 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.9 PDF/R 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 characterization.

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/R sample file creator 1.0)
   /Producer(PDF/R sample file producer 1.0)
   /Title   (PDF/R 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/R 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/R 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/R 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
```