BS ISO 32000-2:2020

BSI Standards Publication

Document management — Portable document format

Part 2: PDF 2.0

Documents
Document Sharing Hub

Full document details: BS ISO 32000-2:2020

National foreword

This British Standard is the UK implementation of ISO 32000-2:2020. It supersedes BS ISO 32000-2:2017, which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee IDT/1, Document Management Applications.

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

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

© The British Standards Institution 2020
Published by BSI Standards Limited 2020

ISBN 978 0 539 01020 6

ICS 35.240.30; 37.100.99

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

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 December 2020.

Amendments/corrigenda issued since publication

Date	Text affected

Documents
Document Sharing Hub

Full document details: BS ISO 32000-2:2020

INTERNATIONAL STANDARD

ISO 32000-2

Second edition
2020-12

Document management — Portable document format —

Part 2: PDF 2.0

Gestion de documents — Format de document portable —

Partie 2: PDF 2.0

Reference number
ISO 32000-2:2020(E)

Documents
Document Sharing Hub
© ISO 2020

Full document details: BS ISO 32000-2:2020

COPYRIGHT PROTECTED DOCUMENT

© ISO 2020

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

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

© ISO 2020 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO 32000-2:2020

Contents Page

Foreword vii

Introduction viii

1 Scope 1

2 Normative references 2

3 Terms and definitions 7

4 Notation 15
4.1 General 15
4.2 Established notations 15

5 Version designations 17

6 Conformance 18
6.1 General 18
6.2 Conforming PDF documents 18
6.3 PDF processors 18

7 Syntax 20
7.1 General 20
7.2 Lexical conventions 21
7.3 Objects 24
7.4 Filters 34
7.5 File structure 53
7.6 Encryption 71
7.7 Document structure 96
7.8 Content streams and resources 110
7.9 Common data structures 114
7.10 Functions 123
7.11 File specifications 132
7.12 Extensions dictionary 141

8 Graphics 145
8.1 General 145
8.2 Graphics objects 145
8.3 Coordinate systems 149
8.4 Graphics state 156
8.5 Path construction and painting 169
8.6 Colour spaces 177
8.7 Patterns 219

© ISO 2020 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO 32000-2:2020

8.8 External objects ... 253
8.9 Images ... 254
8.10 Form XObjects ... 270
8.11 Optional content ... 276

9 Text ... 293
9.1 General ... 293
9.2 Organisation and use of fonts ... 293
9.3 Text state parameters and operators ... 300
9.4 Text objects ... 306
9.5 Introduction to font data structures ... 311
9.6 Simple fonts ... 313
9.7 Composite fonts ... 327
9.8 Font descriptors ... 343
9.9 Embedded font programs ... 351
9.10 Extraction of text content ... 355

10 Rendering ... 360
10.1 General ... 360
10.2 Raster output device native colour ... 361
10.3 CIE-Based colour to device colour ... 361
10.4 Conversions among device colour spaces ... 361
10.5 Transfer functions ... 364
10.6 Halftones ... 366
10.7 Scan conversion details ... 382
10.8 Rendering for separations ... 385

11 Transparency ... 387
11.1 General ... 387
11.2 Overview of transparency ... 387
11.3 Basic compositing computations ... 389
11.4 Transparency groups ... 402
11.5 Soft masks ... 414
11.6 Specifying transparency in PDF ... 415
11.7 Colour space and rendering issues ... 425

12 Interactive features ... 437
12.1 General ... 437
12.2 Viewer preferences ... 437
12.3 Document-level navigation ... 441
12.4 Page-level navigation ... 458
12.5 Annotations ... 465

© ISO 2020 – All rights reserved
Document Sharing Hub
Full document details: BS ISO 32000-2:2020

12.6 Actions ... 506
12.7 Forms ... 528
12.8 Digital signatures ... 567
12.9 Measurement properties ... 595
12.10 Geospatial features ... 601
12.11 Document requirements ... 606

13 Multimedia features ... 614
13.1 General ... 614
13.2 Multimedia ... 614
13.3 Sounds ... 637
13.4 Movies ... 638
13.5 Alternate presentations ... 640
13.6 3D Artwork ... 642
13.7 Rich media ... 700

14 Document interchange ... 713
14.1 General ... 713
14.2 Procedure sets ... 713
14.3 Metadata ... 714
14.4 File identifiers ... 719
14.5 Page-piece dictionaries ... 719
14.6 Marked content ... 720
14.7 Logical structure ... 722
14.8 Tagged PDF ... 745
14.9 Repurposing and accessibility support ... 796
14.10 Web capture ... 802
14.11 Prepress support ... 814
14.12 Document parts ... 834
14.13 Associated files ... 838
Annex A (informative) Operator Summary ... 844
Annex B (informative) Operators in Type 4 Functions ... 848
Annex C (informative) Advice on maximising portability ... 850
Annex D (normative) Character sets and encodings ... 853
Annex E (normative) Extending PDF ... 877
Annex F (normative) Linearized PDF ... 879
Annex G (informative) Linearized PDF access strategies ... 901
Annex H (informative) Example PDF files ... 905
Annex I (normative) PDF versions and compatibility ... 936
Annex J (informative) XObject comparison ... 938

© ISO 2020 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO 32000-2:2020

Annex K (normative) XFA forms 944
Annex L (normative) Parent-child relationships between the standard structure elements in the standard structure namespace for PDF 2.0 947
Annex M (informative) Differences between the standard structure namespaces 973
Annex N (informative) Best practice for halftones 974
Annex O (normative) Fragment identifiers 977
Annex P (informative) An algorithm to determine the actual blending colour space of a transparency group 980
Annex Q (normative) Method for determining transparency on a page 982
Bibliography 984

© ISO 2020 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 32000-2:2020

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 on 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 the following URL: www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 171, *Document management applications*, Subcommittee SC 2, *Application issues*, in collaboration with Technical Committee ISO/TC 130, *Graphic technology*.

This second edition cancels and replaces the first edition (ISO 32000-2:2017), which has been technically revised.

A list of all the parts of ISO 32000 can be found on the ISO website. Changes from previous parts and editions are listed in the Introduction (clauses 0.3 and 0.4).

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.

© ISO 2020 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO 32000-2:2020

Introduction

0.1 PDF

PDF enables users to exchange and view electronic documents easily and reliably, independent of the environment in which they were created or the environment in which they are viewed or printed.

At the core of PDF is an advanced imaging model derived from the PostScript®[1] page description language. This PDF Imaging Model enables the description of text and graphics in a device-independent and resolution-independent manner at a complete, precise and professional level. Unlike PostScript, which is a programming language, PDF is based on a structured binary file format that is optimised for high performance in interactive viewing.

PDF includes objects such as annotations and hypertext links that are not part of the page content itself but are useful for interactive viewing and document interchange. PDF also includes data structures such as tagged PDF, XMP and an associated files mechanism, that are useful for document management and content reuse.

PDF files can be created natively in PDF form, converted from other electronic formats. Since PDF supports a wide range of image and compression technologies, it is a suitable format for documents digitised from paper, microform, or other hard copy formats. Businesses, governments, libraries, archives and other institutions and individuals around the world use PDF to represent considerable bodies of important information. Since its introduction in 1993, aided by the explosive growth of the Internet, PDF has become widely used for the electronic exchange of documents.

There are several specific applications of PDF that have evolved in which limiting the use of some features of PDF while requiring the use of others, enhances the usefulness of PDF. The following International Standards address specialised uses of PDF:

- PDF/X (ISO 15930) is the industry standard for the intermediate representation of printed material in electronic prepress systems for conventional printing applications.
- PDF/A (ISO 19005) is the industry standard for the archiving of digital documents.
- PDF/UA (ISO 14289) is the industry standard for accessible PDF documents and processors.
- PDF/E (ISO 24517) provides a mechanism for representing engineering documents and exchanging engineering data.
- PDF/VT (ISO 16612-2 and ISO 16612-3) is for high volume printing of personalised documents including variable data.
- ISO 19593 describes a method for storing data in a PDF file that correspond to the processing steps of printed products (such as cutting, folding, glueing, Braille, printed white, and printed varnish).
- ISO 21812 describes how document part metadata in a PDF file can be used to communicate the intended appearance of print products and their components.

As corporations, government agencies, and educational institutions streamline their operations by

[1] PostScript® is a trademark of Adobe Systems Incorporated. This information is given for the convenience of users of this document and does not constitute an endorsement by ISO of the product named.

© ISO 2020 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 32000-2:2020

replacing paper-based workflows with electronic exchange of information, the impact and opportunity for the application of PDF will continue to grow at a rapid pace.

PDF, together with software for creating, viewing, printing and processing PDF files in a variety of ways, fulfils a set of requirements for electronic documents including:

- preservation of document fidelity independent of the device, platform, and software,
- merging of content from diverse sources — Web sites, word processing and spreadsheet programs, scanned documents, photos, and graphics — into one self-contained document while maintaining the integrity of all original source documents,
- an extensible metadata model at the document and object level,
- collaborative editing of documents from multiple locations or platforms,
- digital signatures to certify authenticity,
- security and permissions to allow the creator to retain control of the document and associated rights,
- accessibility of content to those with disabilities,
- extraction and reuse of content for use with other file formats and applications, and
- electronic forms to gather and/or represent data within business systems.

0.2 ISO 32000 and PDF

PDF was developed and specified by Adobe Systems Incorporated beginning in 1993 and continuing until 2007 when ISO 32000-1 was first prepared. The Adobe Systems version PDF 1.7 was the basis for ISO 32000-1. The ISO 32000 series has been created as a multi-part document, of which this is Part 2. This allows future parts to be created without rendering ISO 32000, or applications based on it, obsolete. See clause 5, "Version designations" for how the version numbers of PDF (1.0, 1.1, 1.2, […] 2.0) relate to one another.

The primary purpose of this document is to define well-formed PDF documents (conforming PDF files).

In carefully specifying what constitutes a well-defined PDF document, it is natural to describe why a particular feature is to be included in the PDF file and what effect it is designed to have on PDF processing software. So, although the primary objective of this document is to describe the content of conforming PDF files, it also serves secondary purposes of defining exactly how a PDF component is constructed, suggesting why a producer might choose to use the various PDF constructs, as well as what behaviour is elicited from software consuming that PDF file. The choice of which specific set of features a particular PDF processor supports is not specified.

PDF files represent electronic documents. Over time, it was natural to add features that take advantage of PDF's nature, and the power of computer viewing devices. The size of the PDF documentation has more than quadrupled since its first introduction, and the number of features that a PDF processor is expected to support has grown to be large.

0.3 Changes introduced in ISO 32000-2:2017

Starting with ISO 32000-2:2017 (PDF 2.0) the term "conforming reader" is no longer used. The terms "interactive PDF processor", "PDF reader" and "PDF writer" are used instead, and have a conditional

© ISO 2020 – All rights reserved

Documents Document Sharing Hub

Full document details: BS ISO 32000-2:2020

conformance definition. See 6, "Conformance" for further discussion of this change.

This document includes many changes from ISO 32000-1:2008, however only significant new features are marked as being new in PDF 2.0.

PDF 2.0 includes the following new features:

- 7.6.7, "Unencrypted wrapper document"
- 8.6.5.9, "Use of black point compensation";
- 12.5.6.24, "Projection annotations";
- 12.8.3.4, "CAdES signatures as used in PDF";
- 12.8.4, "Long term validation of signatures";
- 12.8.4.3, "Document Security Store (DSS)" and 12.8.5, "Document timestamp (DTS) dictionary";
- 12.10, "Geospatial features";
- 13.7, "Rich media" annotations;
- 14.7.4, "Namespaces" for tagged PDF;
- 14.9.6, "Pronunciation hints";
- 14.12, "Document parts";
- 14.13, "Associated files";
- Support for PRC (see 13.6, "3D Artwork");
- Support for UTF-8.

PDF 2.0 adds many new capabilities to existing features in PDF, including:

- Transparency and blend mode attributes for annotations;
- Stamp Annot intent;
- Polygon/Polyline real paths;
- 256-bit AES encryption;
- ECC-based certificates;
- Unicode-based passwords;
- Document requirement extensions;
- New value for tab order of fields and annotations;
- Page-level OutputIntents;
- Referenced (external) OutputIntents;
- Thumbnails for embedded files;
- Halftone Origin (HTO);
- Measurement & Point Data for image & form XObjects;
- L (length) key for inline image data;
- Viewer preferences enforcement (of print scaling);

© ISO 2020 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO 32000-2:2020

- 3D measurements;
- GoToDp action;
- RichMediaExecute action;
- Extension to GoTo and GoToR to support linking to a specific structure element;
- Extension to Signature Field Locks and Signature Seed Values;
- Extensions to 3D viewing conditions, incl. transparency;
- Ref (reference) structure element property;
- PageNum and Bates artifact types;
- New list types for structured lists;
- “Short” (short name) attribute for table header cells
- Extensions to OutputIntents (MixingHints and SpectralData).

The following clauses have been substantially rewritten for PDF 2.0:

- 7.4.7, "JBIG2Decode filter";
- 10.1 – 10.3, "Rendering";
- 11, "Transparency";
- 12.8, "Digital signatures";
- 14.3, "Metadata";
- 14.8, "Tagged PDF";
- 14.9, "Repurposing and accessibility support".

PDF 2.0 includes many important corrections, extensions and clarifications for existing features, including:

- Corrections for many typing errors including bad symbols and truncated formulae.
- Updates and changes in normative references and the bibliography.
- Improved cross referencing for clauses, tables and figures within this document.
- Clarification for processing dashed and degenerate lines, clarification for processing text objects and blending colour spaces within the transparency framework, clarifications and enhancements for annotation appearances, stamp annotations extension and polyline annotation enhancement.
- Strengthened encryption including introduction of elliptic curve cryptography, more control over forms tab ordering, enforced viewer preferences, rich text, improvements to digital signatures for long term signatures, 3D viewing improvements including 3D projections, revised blend formulae for **ColorBurn** and **ColorDodge**, additional structure tags to improve accessibility, requirement for metadata streams to be XMP and support for hyperlinks in rich text.
- Clarification for PDF version numbering, resource inheritance, required and optional signature dictionary **SubFilter** keys, artifacts, developer-defined extensions, word breaking and page sizes, which file to show when first opening a collection, scope of header attributes, precedence of CID font widths, when a **CIDToGIDMap** is used with Type 2 CID fonts, deprecate sound and movie actions and annotations in favour of newer methods, rendering

© ISO 2020 – All rights reserved

AVA Documents Document Sharing Hub

Full document details: BS ISO 32000-2:2020

intent and ImageMask, precedence of Type 1 encoding methods, the wording used to define delimiters with respect to << and >>, Identity CMaps and CIDFonts, a special case when closing and filling a path, that clipping follows filling rules and that operating on an undefined path generates an error.

- Clarification and terminology improvements among Type 1, TrueType, CFF and OpenType fonts; thumbnails for embedded files.
- Specification of XFA used for rich text in annotations.
- The rewrite of 14.8, "Tagged PDF" includes clarification of the parent/child relationships between tags, simplifies and extends the standard tag set, and adds the use of namespaces for custom tag sets (see also 14.7.4, "Namespaces" for new namespace functionality).

Some features present in earlier versions of PDF have been deprecated in PDF 2.0, including:

- XFA (incl. NeedsRendering);
- Movie, Sound and TrapNet annotations;
- Movie and Sound actions;
- Info dictionary;
- Assistive technology restrictions via DRM;
- ProcSet;
- OS-specific file specifications;
- OS-specific additions to Launch actions;
- Names for XObjects;
- Names for Fonts;
- Arrays of Blend Modes;
- Alternate Presentations;
- Open prepress interface (OPI);
- CharSet (For Type 1 fonts);
- CIDSet (for CID fonts);
- Prepress viewer preferences (ViewArea, ViewClip, etc.);
- NeedAppearances;
- adbe.pkcs7.sha1;
- adbe.x509.rsa_sha1;
- Encryption of FDF files;
- Suspects flag in MarkInfo dictionary;
- UR signatures;
- Transfer functions in the graphics state.

0.4 Changes introduced in ISO 32000-2:2020

ISO 32000-2:2020 includes additional corrections and clarifications as indicated in this clause. Precise locations of key changes are also indicated by the text string “(2020)”. In addition, F.3 "Linearized PDF

© ISO 2020 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO 32000-2:2020

document structure" and Annex L "Parent-child relationships between the standard structure elements in the standard structure namespace for PDF 2.0" have been significantly updated. PDF character collections were also updated as follows:

- Adobe-Japan1-6 becomes Adobe-Japan1-7;
- Adobe-CNS1-6 becomes Adobe-CNS1-7;
- Adobe-KR-9 is introduced;
- Adobe-Korea1-2 & Adobe-Japan2-0 were deprecated;
- Adobe-GB1-5 remains unchanged.

ISO 32000-2:2020 also makes several important Normative References changes due to various documents being withdrawn or obsoleted. Additionally some previous Normative References have been moved to the Bibliography:

- ISO 3166-1 is now an undated reference – see 7.9.2.2.2, "Text string language escape sequences";
- The ISO/IEC 14492 dated reference for JBIG2 was updated to the 2019 edition – see 7.4.7, "JBIG2Decode filter";
- The ISO/IEC 14496-22 dated reference for the Open Font format was updated to the 2019 edition – see 9.6.3, "TrueType fonts";
- ISO 15076-1:2010 dated reference for ICC.1 can be supplemented by the Errata list and approved revisions available from the ICC website (http://color.org/icc_specs2.xalter) – see 8.6.5.5, "ICCBased colour spaces";
- The ISO/IEC 15444-1 dated reference for JPEG 2000 was updated to the 2019 edition – see 7.4.9, "JPXDecode filter";
- The ISO/IEC 19444-1 dated reference for XFDF was updated to the 2019 edition;
- This document requires that RFC 3454 (“stringprep”) and RFC 4013 (“SASLprep”) continue to be used to maintain backward compatibility even though these RFCs are marked as obsolete by IETF;
- RFC 6234 (US Secure Hash Algorithms) is replaced by FIPS PUB 180-4;RFC 2083 (Portable Network Graphics) is replaced by ISO/IEC 15948:2004 for PNG predictors – see 7.4.4.4, "LZW and Flate predictor functions";
- ISO/DIS 21757-1 replaces several Adobe, ECMA and ISO publications related to ECMAScript in PDF 2.0 – see 12.6.4.17, "ECMAScript actions";
- This document makes explicit reference to ECMA-363 U3D 3rd edition and not the latest U3D 4th edition;
- PDF character collections are now all referenced to GitHub repositories;
- The TrueType Reference is now undated.

An attempt is being made to keep copies of all references without copyright restrictions available for free download on the following website: https://www.pdfa.org/iso-32000-normative-references/.

© ISO 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 32000-2:2020

Documents
Document Sharing Hub

Document management — Portable document format —

Part 2:
PDF 2.0

IMPORTANT — The electronic file of this document contains colours which are considered to be useful for the correct understanding of the document. Users who need a paper copy of this document will therefore benefit from using a colour printer.

1 Scope

This document specifies a digital form for representing electronic documents to enable users to exchange and view electronic documents independent of the environment in which they were created or the environment in which they are viewed or printed. It is intended for developers of software that creates PDF files (PDF writers), software that reads existing PDF files and (usually) interprets their contents for display (PDF readers), software that reads and displays PDF content and interacts with the computer users to possibly modify and save the PDF file (interactive PDF processors) and PDF products that read and/or write PDF files for a variety of other purposes (PDF processors). (PDF writers and PDF readers are more specialised classifications of interactive PDF processors and all are PDF processors).

This document does not specify the following:

- specific processes for converting paper or electronic documents to the PDF file format;
- specific technical design, user interface implementation, or operational details of rendering;
- specific physical methods of storing these documents such as media and storage conditions;
- methods for validating the conformance of PDF files or PDF processors;
- required computer hardware and/or operating system.

© ISO 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 32000-2:2020

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 or corrigenda) applies.

ISO 3166-1, *Codes for the representation of names of countries and their subdivisions – Part 1: Country codes*.

ISO/IEC 8824-1, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation*

ISO/IEC 10646, *Information technology – Universal Coded Character Set (UCS)*

ISO/IEC 10918 (all parts), *Information Technology – Digital Compression and Coding of Continuous-Tone Still Images: Requirements and guidelines* (informally known as the JPEG standard, for the Joint Photographic Experts Group, the ISO group that developed the standard)

ISO/IEC 14492:2019, *Information technology – Lossy/lossless coding of bi-level images*

ISO/IEC 14496-22:2019, *Information technology – Coding of audio-visual objects — Part 22: Open Font Format*

ISO 14739-1, *Document management – 3D use of Product Representation Compact (PRC) format — Part 1: PRC 10001*

ISO 15076-1:2010, *Image technology colour management – Architecture, profile format and data structure — Part 1: Based on ICC.1:2010*

ISO/IEC 15444-1:2019, *Information technology — JPEG 2000 image coding system: Core coding system*

ISO/IEC 15444-2:2004, *Information technology — JPEG 2000 image coding system: Extensions*

ISO/IEC 15948:2004, *Information technology — Computer graphics and image processing — Portable Network Graphics (PNG): Functional specification*

ISO 16684-1, *Extensible metadata platform (XMP) specification — Part 1: Data model, serialization and core properties*

ISO 17972-4, *Graphic technology — Colour data exchange format (CxF/X) — Part 4: Spot colour characterisation data (CxF/X-4)*

ISO 18619, *Image technology colour management – Black point compensation*

ISO 19162, *Geographic information – Well known text representation of coordinate reference systems*

ISO 19444-1:2019, *Document management – XML Forms Data Format – Part 1: Use of ISO 32000-2 (XFDF 3.0)*

ISO/DIS 21757-1, *Document management — ECMAScript for PDF — Part 1: Use of ISO 32000-2 (PDF 2.0)*

IEC 61966-2-1 ed1.0 (1999-10), *Multimedia systems and equipment — Colour measurement and*

© ISO 2020 – All rights reserved
Documents
Document Sharing Hub

Full document details: BS ISO 32000-2:2020

management — Part 2-1: Colour management — Default RGB colour space — sRGB (with Amendment 1 IEC 61966-2-1-am1 ed1.0 (2003-01))

ANSI X9.62-2005, *Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA), American National Standards Institute*

INCITS 4-1986 (R2017), *Information Systems - Coded Character Sets - 7-Bit Standard Code for Information Interchange (7-Bit ASCII)"*

PostScript Language Third Edition, (February, 1999), Adobe Systems Incorporated[1]

Adobe Glyph List, Adobe Systems Incorporated[2]

Adobe Glyph List for New Fonts, Adobe Systems Incorporated[2]

Adobe PDF Signature Build Dictionary Specification v.1.4, (March 2008), Adobe Systems Incorporated[1]

Adobe TIFF Revision 6.0, Final, (June 1992), Adobe Systems Incorporated[1]

Adobe Type 1 Font Format, Version 1.1, (February 1993), Addison-Wesley, ISBN 0-201-57044-0[1]

Adobe XML Architecture, XML Forms Architecture (XFA) Specification, version 3.3, (January 2012), Adobe Systems Incorporated[1]

Adobe-Japan1-7 Character Collection for CID-Keyed Fonts, https://github.com/adobe-type-tools/cmap-resources

Adobe-GB1-5 Character Collection for CID-Keyed Fonts, https://github.com/adobe-type-tools/cmap-resources

Adobe-CNS1-7 Character Collection for CID-Keyed Fonts, https://github.com/adobe-type-tools/cmap-resources

Adobe-Korea1-2 Character Collection for CID-Keyed Fonts, https://github.com/adobe-type-tools/cmap-resources

Adobe-Japan2-0 Character Collection for CID-Keyed Fonts, https://github.com/adobe-type-tools/cmap-resources

Adobe-KR-9 Character Collection for CID-Keyed Fonts, https://github.com/adobe-type-tools/cmap-resources

Adobe Technical Note #5014, *Adobe CMap and CID Font Files Specification, Version 1.0*, (8 October 1996), Adobe Systems Incorporated[1]

Adobe Technical Note #5015, *Type 1 Font Format Supplement*, (May 1994) , Adobe Systems Incorporated[1]

Adobe Technical Note #5116, *Supporting the DCT Filters in PostScript Level 2*, (November 1992), Adobe

[1] These documents can be found at the PDF Association at https://www.pdfa.org/iso-32000-normative-references/ as well as via the Adobe Systems Incorporated Web site http://www.adobe.com/.

[2] These documents can be found at https://github.com/adobe-type-tools/agl-aglfn.

© ISO 2020 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO 32000-2:2020

Systems Incorporated[1]

Adobe Technical Note #5176, *The Compact Font Format Specification, version 1.0*, (December 2003), Adobe Systems Incorporated[1]

Adobe Technical Note #5177, *The Type 2 Charstring Format*, (March 2000), Adobe Systems Incorporated[1]

Adobe Technical Note #5620, *Portable Job Ticket Format, Version 1.1*, (April 1999), Adobe Systems Incorporated[1]

Adobe Technical Note #5660, *Open Prepress Interface (OPI) Specification, Version 2.0*, (January 2000), Adobe Systems Incorporated[1]

FIPS PUB 180-4, *Secure Hash Standard (SHS)*, (August 2015), Federal Information Processing Standards, https://doi.org/10.6028/NIST.FIPS.180-4 [3]

FIPS PUB 186-4, *Digital Signature Standard, describes DSA signatures*, (July 2013), Federal Information Processing Standards, https://doi.org/10.6028/NIST.FIPS.186-4 [3]

FIPS PUB 197, *Advanced Encryption Standard (AES)*, (November 2001), Federal Information Processing Standards, https://doi.org/10.6028/NIST.FIPS.197 [3]

RFC 1321, *The MD5 Message-Digest Algorithm*, (April 1992), Internet Engineering Task Force (IETF), https://dx.doi.org/10.17487/rfc1321 [4]

RFC 1950, *ZLIB Compressed Data Format Specification, Version 3.3*, (May 1996), Internet Engineering Task Force (IETF), https://dx.doi.org/10.17487/rfc1950 [4]

RFC 1951, *DEFLATE Compressed Data Format Specification, Version 1.3*, (May 1996), Internet Engineering Task Force (IETF), https://dx.doi.org/10.17487/rfc1951[4]

RFC 2045, *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*, (November 1996), Internet Engineering Task Force (IETF), https://dx.doi.org/10.17487/rfc2045 [4]

RFC 2046, *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types*, (November 1996), Internet Engineering Task Force (IETF), https://dx.doi.org/10.17487/rfc2046 [4]

RFC 2315, *PKCS #7: Cryptographic Message Syntax Version 1.5* (March 1998), Internet Engineering Task Force (IETF), https://dx.doi.org/10.17487/rfc2315 [4]

RFC 3161, *Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)*, (August 2001), Internet Engineering Task Force (IETF), https://dx.doi.org/10.17487/rfc3161 [4]

RFC 3454, *Preparation of International Strings ("stringprep")*, (December 2002), Internet Engineering Task Force (IETF), https://dx.doi.org/10.17487/rfc3454 [4]

RFC 3986, *Uniform Resource Identifier (URI): Generic Syntax*, (January 2005), Internet Engineering Task Force (IETF), https://dx.doi.org/10.17487/rfc3986 [4]

[3] These FIPS documents are also available from NIST at http://www.nist.gov/itl/fipscurrent.cfm.

[4] These documents are also available as Internet Engineering Task Force RFCs at https://www.ietf.org/

© ISO 2020 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO 32000-2:2020

RFC 4013, *SASLprep: Stringprep Profile for User Names and Passwords*, (February 2005), Internet Engineering Task Force (IETF), https://dx.doi.org/10.17487/rfc4013 [4]

RFC 5035, *Enhanced Security Services (ESS) Update: Adding CertID Algorithm Agility*, (August 2007), Internet Engineering Task Force (IETF), https://dx.doi.org/10.17487/rfc5035 [4]

RFC 5280, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile,* (May 2008), Internet Engineering Task Force (IETF), https://dx.doi.org/10.17487/rfc5280 [4]

RFC 5480, *Elliptic Curve Cryptography Subject Public Key Information*, (March 2009), Internet Engineering Task Force (IETF), https://dx.doi.org/10.17487/rfc5480 [4]

BCP 47 (RFC 5646), *Tags for Identifying Languages*, (September 2009), Internet Engineering Task Force (IETF), https://dx.doi.org/10.17487/rfc5646 [4]

RFC 5652, *Cryptographic Message Syntax (CMS)*, (September 2009), Internet Engineering Task Force (IETF), https://dx.doi.org/10.17487/rfc5652 [4]

RFC 5755, *An Internet Attribute Certificate Profile for Authorization*, (January 2010), Internet Engineering Task Force (IETF), https://dx.doi.org/10.17487/rfc5755 [4]

RFC 5816, *ESSCertIDv2 Update for RFC 3161*, (March 2010), Internet Engineering Task Force (IETF), https://dx.doi.org/10.17487/rfc5816 [4]

RFC 6960, X.509 *Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP*, (June 2013), Internet Engineering Task Force (IETF), https://dx.doi.org/10.17487/rfc6960 [4]

RFC 7231, *Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content*, (June 2014), https://dx.doi.org/10.17487/rfc7231

RFC 8018, *PKCS #5: Password-Based Cryptography Specification Version 2.1*, (January 2017), Internet Engineering Task Force (IETF), https://dx.doi.org/10.17487/rfc8018 [4]

RFC 8118, *The application/PDF Media Type, Internet Engineering Task Force*, (March 2017), Internet Engineering Task Force (IETF), https://dx.doi.org/10.17487/rfc8118 [4]

ETSI EN 319 122-1 V1.1.1 (2016-04) *Electronic Signatures and Infrastructures (ESI); CAdES digital signatures; Part 1: Building blocks and CAdES baseline signatures*, European Telecommunications Standards Institute[5]

ETSI EN 319 122-2 V1.1.1 (2016-04) *Electronic Signatures and Infrastructures (ESI); CAdES digital signatures; Part 2: Extended CAdES signatures*, European Telecommunications Standards Institute[5]

ETSI EN 319 142-1 V1.1.1 (2016-04) *Electronic Signatures and Infrastructures (ESI); PAdES digital signatures; Part 1: Building blocks and PAdES baseline signature*, European Telecommunications Standards Institute[5]

ETSI EN 319 142-2 V1.1.1 (2016-04) *Electronic Signatures and Infrastructures (ESI); PAdES digital signatures; Part 2: Additional PAdES signatures profiles*, European Telecommunications Standards Institute[5]

[5] These documents are available from http://www.etsi.org

© ISO 2020 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO 32000-2:2020

EPSG *Geospatial Coordinate System Reference Codes*, administered by the International Association of Oil and Gas Producers (OGP), http://www.epsg.org

Extensible Markup Language (XML) 1.0, Fifth Edition, November 2008, World Wide Web Consortium (W3C), https://www.w3.org/TR/xml/

Accessible Rich Internet Applications (ARIA) 1.1, December 2017, World Wide Web Consortium (W3C), https://www.w3.org/TR/wai-aria-1.1/

Mathematical Markup Language (MathML) Version 3.0, 2nd Revision, April 2014, World Wide Web Consortium (W3C), https://www.w3.org/TR/MathML3/

Japanese Industrial Standard (JIS) X 4051:2004

PANOSE Classification Metrics Guide, (February 1997), Hewlett-Packard Corporation, *https://monotype.github.io/panose/pan1.htm*

Pronunciation Lexicon Specification (PLS) Version 1.0, October 2008, World Wide Web Consortium (W3C), https://www.w3.org/TR/pronunciation-lexicon/

ITU Recommendation T.4: Standardization of Group 3 facsimile terminals for document transmission, International Telecommunication Union (ITU), https://www.itu.int/rec/T-REC-T.4/en

ITU Recommendation T.6: Facsimile coding schemes and coding control functions for Group 4 facsimile apparatus, International Telecommunication Union (ITU), https://www.itu.int/rec/T-REC-T.6/en

Standard ECMA-363, Universal 3D File Format, 3rd Edition (U3D), (June 2006), Ecma International

Synchronized Multimedia Integration Language (SMIL 3.0), December 2008, World Wide Web Consortium (W3C), http://www.w3.org/TR/SMIL3/

TrueType Reference Manual, Apple Inc., https://developer.apple.com/fonts/TrueType-Reference-Manual

© ISO 2020 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO 32000-2:2020

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 http://www.iso.org/obp
- IEC Electropedia: available at http://www.electropedia.org/

3.1
approval signature
one or more digital signatures applied to a document, each one allowing the detection of changes to the document and which confirms a signer of the document

3.2
array object
one-dimensional collection of *objects* (3.44) arranged sequentially and implicitly numbered starting at 0

3.3
ASCII American Standard Code for Information Interchange
widely used convention for encoding a specific set of 128 characters as binary numbers defined in INCITS 4-1986 (R2017)

3.4
ASN.1
means of reference to ISO/IEC 8824-1

3.5
binary data
sequence of bytes

3.6
boolean objects
either the keyword **true** or the keyword **false**

3.7
byte
group of 8 binary digits (an 8-bit value) which collectively can be configured to represent one of 256 different values

3.8
CAdES
means of reference to ETSI EN 319 122-1 V1.1.1 and ETSI EN 319 122-2 V1.1.1

3.9
certification signature
one and only one digital signature applied to a document which allows the detection of changes to the

© ISO 2020 – All rights reserved

AVA Documents Document Sharing Hub

Full document details: BS ISO 32000-2:2020

document and which confirms the origin or the author of the document and that is able to allow or to block specified actions to be performed when using conformant *signature handlers* (3.61) or *PDF processors* (3.49)

3.10
character
numeric code representing an abstract symbol according to some defined character encoding rule

Note 1 to entry: There are three manifestations of characters in PDF, depending on context:

- A PDF file is represented as a sequence of 8-bit *bytes* (3.7), some of which are interpreted as character codes in the *ASCII* (3.3) character set and some of which are treated as arbitrary *binary data* (3.5) depending upon the context.
- The contents (data) of a string or *stream object* (3.63) in some contexts are interpreted as character codes in the PDFDocEncoding, UTF-8 or UTF-16BE character set.
- The contents of a string within a PDF *content stream* (3.12) in some situations are interpreted as character codes that select *glyphs* (3.29) to be drawn on the page according to a character encoding that is associated with the text font.

3.11
character set
defined set of *characters* (3.10) each assigned a unique character value

3.12
content stream
stream object whose data consists of a sequence of instructions describing the graphical elements to be painted on a page

3.13
cross reference table
data structure that contains the *byte* (3.7) offset start for each of the *indirect objects* (3.33) within the file

3.14
developer
entity, including individuals, companies, non-profits, standards bodies, open source groups, etc., developing standards or software to use and extend this document

3.15
deprecated
a part of ISO 32000 that should not be written into a PDF 2.0 document, and should be ignored by a *PDF processor* (3.49)

Note 1 to entry: In some cases variations on these restrictions on continued use of a deprecated feature are explicitly stated in this document.

Note 2 to entry: Implementers are cautioned that some features that are deprecated in this document could have tighter constraints placed on them, or even be removed completely, in a later version of ISO 32000, or in subset standards such as PDF/X (ISO 15930), PDF/A (ISO 19005), PDF/E (ISO 24517),

© ISO 2020 – All rights reserved
Documents
Document Sharing Hub

Full document details: BS ISO 32000-2:2020

PDF/VT (ISO 16612-2 and ISO 16612-3) and PDF/UA (ISO 14289).

3.16
dictionary object
associative table containing pairs of objects, the first object being a *name object* (3.39) serving as the key and the second object serving as the value and may be any kind of object including another dictionary

3.17
direct object
object that has not been made into an *indirect object* (3.33)

3.18
document part
set of related pages and/or related sets of pages

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

3.19
document part hierarchy
hierarchical data structure that specifies the organisation of *document parts* (3.18)

3.20
document part metadata
metadata associated with a *document part* (3.18)

3.21
DSA
Digital Signature Algorithm, and means of reference to FIPS PUB 186-4

3.22
electronic document
electronic representation of a page-oriented aggregation of text, image and graphic data, and metadata useful to identify, understand and render that data, that can be reproduced on paper or displayed without significant loss of its information content

3.23
end-of-line marker
EOL marker
one or two *character* (3.10) sequence marking the end of a line of text, consisting of a CARRIAGE RETURN character (0Dh) or a LINE FEED character (0Ah) or a CARRIAGE RETURN followed immediately by a LINE FEED

3.24
FDF file
file conforming to the Forms Data Format containing form data or annotations that may be imported into a PDF file

Note 1 to entry: See 12.7.8, "Forms data format".

© ISO 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 32000-2:2020

3.25
filter
optional part of the specification of a *stream object* (3.63), indicating how the data in the stream should be decoded before it is used

3.26
font
identified collection of graphics that may be *glyphs* (3.29) or other graphical elements

3.27
font program
software program written in a special-purpose language, such as the Type 1, TrueType, or OpenType *font* (3.26) format, that is understood by a specialized font interpreter

3.28
function
special type of object that represents parameterised classes, including mathematical formulas and sampled representations with arbitrary resolution

3.29
glyph
recognizable abstract graphic symbol that is independent of any specific design

[SOURCE: ISO/IEC 9541-1:2012, 3.12]

3.30
graphics state
top of a push down stack of the graphics control parameters that defines the current global framework within which the graphics operators execute

3.31
ICC profile
ISO 15076-1:2010 or any of the ICC.1 specifications

3.32
ICC specification
cross-platform profile format for the creation and interpretation of colour data and means of reference to ISO 15076-1:2010 or any of the ICC.1 specifications

3.33
indirect object
object (3.44) that is labelled with a positive *integer object* (3.34) number followed by a non-negative integer generation number followed by keyword **obj** and having keyword **endobj** after it

3.34
integer object
mathematical integers with an implementation specified interval centred at 0 and written as one or more decimal digits optionally preceded by a sign

© ISO 2020 – All rights reserved
Documents
Document Sharing Hub

Full document details: BS ISO 32000-2:2020

3.35
interactive PDF processor
special case of a *PDF processor* (3.49) that responds to interactive user controls defined in this document

3.36
JPEG
common image compression data format and means of reference to ISO/IEC 10918

3.37
job definition
information that specifies the production requirements and workflow of a unit of work involving purposing PDF content to one or more messaging channels

3.38
job ticket
electronic specification of process control for print production in either a published or proprietary format

3.39
name object
atomic symbol uniquely defined by a sequence of *characters* (3.10) introduced by a SOLIDUS (/) (2Fh), but the SOLIDUS is not considered to be part of the name

3.40
name tree
similar to a dictionary that associates keys and values but the keys in a name tree are strings and are ordered

3.41
null object
single *object* (3.44) of type null, denoted by the keyword **null**, and having a type and value that are unequal to those of any other object

3.42
number tree
similar to a dictionary that associates keys and values but the keys in a number tree are integers and are ordered

3.43
numeric object
either an *integer object* (3.34) or a *real object* (3.56)

3.44
object
basic data structure from which PDF files are constructed and includes these 9 types: array, boolean, dictionary, integer, name, null, real, stream and string

3.45
object reference

© ISO 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 32000-2:2020

object (3.44) value used to allow one object to refer to another; that has the form "<n> <m> **R**" where <n> is an *indirect object* (3.33) number, <m> is its generation number and **R** is the uppercase letter R

3.46
object stream
stream that contains a sequence of PDF *objects* (3.44), except those of **Type** *stream*

3.47
PAdES
means of reference to ETSI EN 319 142-1 V1.1.1 and ETSI EN 319 142-2 V1.1.1

3.48
PDF imaging model
derived from the PostScript page description language and enables the description of high quality text and graphics in a device-independent and resolution-independent manner

3.49
PDF processor
software, hardware or any other active agent that writes, reads, updates or otherwise processes a PDF file which conforms to this document and does so in a manner that conforms to this document

3.50
PDF file
contiguous stream of *binary data* (3.5) that conforms to this document

3.51
PDF reader
PDF processor (3.49) that reads a PDF file

3.52
PDF writer
PDF processor (3.49) that writes a PDF file

3.53
PKCS #7
means of reference to Internet RFC 2315

3.54
PRC
means of reference to ISO 14739-1

3.55
RDF
means of reference to RDFa Core 1.1

3.56
real object
approximate mathematical real numbers, but with limited range and precision and written as one or more decimal digits with an optional sign and a leading, trailing, or embedded PERIOD (2Eh) (decimal point)

© ISO 2020 – All rights reserved
Documents
Document Sharing Hub

Full document details: BS ISO 32000-2:2020

3.57
rectangle
specific *array object* (3.2) used to describe locations on a page and bounding boxes for a variety of *objects* (3.44) and written as an array of four numbers giving the coordinates of a pair of diagonally opposite corners, typically in the form [ll_x ll_y ur_x ur_y] specifying the lower-left x, lower-left y, upper-right x, and upper-right y coordinates of the rectangle, in that order

3.58
resource dictionary
associates resource names, used in *content streams* (3.12), with the resource *objects* (3.44) themselves and is organised into various categories (e.g., Font, ColorSpace, Pattern)

3.59
running text
body of text, as distinct from headings, footnotes, diagrams, callouts and other non-textual material

3.60
SHA
Secure Hash Algorithms, and means of reference to FIPS 180-4

3.61
signature handler
software module that implements a specific form of signing and/or augmentation and/or verification of digital signatures

3.62
sRGB
means of reference to IEC 61966-2-1 ed1.0 (1999-10) with Amendment 1 IEC 61966-2-1-am1 ed1.0 (2003-01)

3.63
stream object
dictionary followed by zero or more *bytes* (3.7) bracketed between the keywords **stream** and **endstream**

3.64
string object
series of *bytes* (3.7) (unsigned integer values in the range 0 to 255) in which the bytes are not *integer objects* (3.34), but are stored in a more compact form

3.65
TIFF
means of reference to Adobe TIFF Revision 6.0

3.66
UCS
means of reference to ISO/IEC 10646

3.67
U3D

© ISO 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 32000-2:2020

means of reference to ECMA-363, Universal 3D File Format, 3rd Edition (U3D)

3.68
white-space character
characters (3.10) that separate PDF syntactic constructs such as names and numbers from each other; white-space characters are HORIZONTAL TABULATION (09h), LINE FEED (0Ah), FORM FEED (0Ch), CARRIAGE RETURN (0Dh), SPACE (20h), NULL (00h); (see "Table 1 — White-space characters" in 7.2.3, "Character set")

3.69
XFDF
means of reference to ISO 19444-1:2019, *Document management – XML Forms Data Format – Part 1: Use of ISO 32000-2 (XFDF 3.0)*

3.70
XMP
means of reference to ISO 16684-1

Note 1 to entry: In addition to ISO 16684-1, Adobe Systems Inc. also publish *Adobe XMP: Extensible Metadata Platform, Part 2: Additional Properties* and *Adobe XMP: Extensible Metadata Platform, Part 3: Storage in Files*.

3.71
XMP packet
structured wrapper for serialised XML metadata and means of reference to ISO 16684-1 that can be embedded in a wide variety of file formats

Note 1 to entry: In addition to ISO 16684-1, Adobe Systems Inc. also publish *Adobe XMP: Extensible Metadata Platform, Part 2: Additional Properties* and *Adobe XMP: Extensible Metadata Platform, Part 3: Storage in Files*.

© ISO 2020 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 32000-2:2020

4 Notation

4.1 General

PDF operators, PDF keywords, the names of keys in PDF dictionaries, and other predefined names are written in bold font; words that denote operands of PDF operators or values of dictionary keys are written in italic font. An italic font is also used to introduce key concepts or reference specific terms of importance.

NOTE 1 If a word or phrase is used as both a keyword and a value, the notation used in this document will default to italic font unless doing so would cause undue confusion.

NOTE 2 For backwards compatibility, words denoting predefined names, operands or values and other specific terms of importance use US English spelling.

Token characters used to delimit objects and describe the structure of PDF files, as defined in 7.2, "Lexical conventions", shall be identified by their INCITS 4-1986 (R2017) (ASCII 7-bit USA codes) character name written in upper case followed by a parenthetic two digit hexadecimal character value with the suffix "h".

Ellipses (...) are used within PDF examples to indicate omitted detail. Pairs of ellipses are also used to bracket comments about such omitted detail.

Characters in text streams, as defined by 7.9.2, "String object types", are identified by their Unicode character name written in uppercase in font followed by a parenthetic four digit hexadecimal character code value with the prefix "U+" as shown in the example in this clause:

EXAMPLE EN SPACE (U+2002)

4.2 Established notations

Due to the historical development of PDF and the requirement to remain backward compatible with existing PDF files, the notations used for keys and numbers in PDF files use United States English conventions, while the rest of the language in this document conforms to Oxford spelling in most cases. For example, within a PDF you will encounter the key **Alternate** whereas Oxford spelling would use the term Alternative. Likewise PDF files use the key **Color** rather than the Oxford spelling Colour. Exceptions to this are found where switching between the two spellings could cause confusion, such as with the term *artifact* and the word *artefact*. In cases such as these, this document will use the spelling of the term of art throughout the entirety of the document rather than switching between two spellings. Real numbers that contain a decimal radix separator use the PERIOD (2Eh) character and not the COMMA (2Ch) as used in Oxford spelling. Numbers do not include spaces.

Any use of the terms *UCS* or *UCS-2* throughout this document shall be inferred as referring to ISO/IEC 10646.

Any use of the term *JPEG* throughout this document shall be inferred as referring to compressed image data which is in accordance with ISO/IEC 10918 (all parts and amendments). ISO/IEC 10918 is informally known as the JPEG standard, for the Joint Photographic Experts Group, the ISO group that developed the standard.

Any use of the term *XFDF* throughout this document shall be inferred as referring to data which is in

© ISO 2020 – All rights reserved

AVA Documents Document Sharing Hub

Full document details: BS ISO 32000-2:2020

accordance with ISO 19444-1:2019.

Any use of the term *RDF* throughout this document shall be inferred as referring to data which is in accordance with *RDFa Core 1.1, Third Edition*.

Any use of the term *sRGB* throughout this document shall be inferred as referring to data in accordance with IEC 61966-2-1 ed1.0 (1999-10) with Amendment 1 IEC 61966-2-1-am1 ed1.0 (2003-01).

Any use of the phrases *Group 3, Group 4,* or *CCITT facsimile* shall be inferred as referring to data in accordance with *ITU Recommendation T.4: Standardization of Group 3 facsimile terminals for document transmission* and *ITU Recommendation T.6: Facsimile coding schemes and coding control functions for Group 4 facsimile apparatus*, International Telecommunication Union (ITU).

Any use of the phrase *TIFF* throughout this document shall be inferred as referring to data in accordance with *Adobe TIFF Revision 6.0, Final, (June 1992)*.

Any use of the term *XMP* throughout this document shall be inferred as referring to data in accordance with ISO 16684-1.

Any use of the term *Adobe-Japan1* throughout this document shall be inferred as referring to data in accordance with Adobe-Japan1-7 Character Collection for CID-Keyed Fonts.

Any use of the term *Adobe-GB1* throughout this document shall be inferred as referring to data in accordance with Adobe-GB1-5 Character Collection for CID-Keyed Fonts.

Any use of the term *Adobe-CNS1* throughout this document shall be inferred as referring to data in accordance with Adobe-CNS1-7 Character Collection for CID-Keyed Fonts .

Any use of the term *Adobe-Korea1* throughout this document shall be inferred as referring to data in accordance with Adobe-Korea1-2 Character Collection for CID-Keyed Fonts. Adobe-Korea1 was deprecated in this document (2020).

Any use of the term *Adobe-Japan2* throughout this document shall be inferred as referring to data in accordance with Adobe-Japan2-0 Character Collection for CID-Keyed Fonts. Adobe-Japan2 was deprecated in this document (2020).

Any use of the term *DSA* throughout this document shall be inferred as referring to the digital signature algorithm defined by *FIPS PUB 186-4.*

Any use of the term *PKCS #7* throughout this document shall be inferred as referring to cryptographic data in accordance with *Internet RFC 2315.*

Any use of the term *SHA* throughout this document shall be inferred as referring to the Secure Hash Algorithms defined in *FIPS 180-4*.

Any use of the term *CAdES* throughout this document shall be inferred as referring to data in accordance with *ETSI EN 319 122-1 V1.1.1* and *ETSI EN 319 122-2 V1.1.1*.

Any use of the term *PAdES* throughout this document shall be inferred as referring to data in accordance with *ETSI EN 319 142-1 V1.1.1* and *ETSI EN 319 142-2 V1.1.1*.

© ISO 2020 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO 32000-2:2020

5 Version designations

For the convenience of the reader of this document, the PDF versions in which various features were introduced are provided informatively within the document. The first version of PDF was designated PDF 1.0 and was specified by Adobe Systems Incorporated in the PDF Reference 1.0 document published by Adobe and Addison Wesley in June 1993. Since then, PDF has gone through seven Adobe revisions designated as: PDF 1.1, PDF 1.2, PDF 1.3, PDF 1.4, PDF 1.5, PDF 1.6 and PDF 1.7. All non-deprecated features defined in a previous PDF version were also included in the subsequent PDF version.

ISO 32000-1 defines a PDF version matching 1.7. This document defines a new version of PDF designated 2.0. This document is also suitable for interpretation of files made to conform to any of the previous Adobe PDF specifications 1.0 through 1.7 and ISO 32000-1.

Throughout this specification, in order to indicate at which point in the sequence of versions a feature was introduced, a notation with a PDF version number in parenthesis (e.g., (PDF 1.3)) is used. Thus if a feature is labelled with (PDF 1.3) it means that PDF 1.0, PDF 1.1 and PDF 1.2 were not specified to support this feature whereas all versions of PDF 1.3 and greater were defined to support it.

© ISO 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 32000-2:2020

6 Conformance

6.1 General

This clause describes conformance with this document.

6.2 Conforming PDF documents

Conforming PDF files shall adhere to all requirements of this document. A conforming PDF file is not obligated to contain any feature other than those explicitly required by this document.

NOTE The proper mechanism by which a file can identify itself as being a PDF file of a given version level is described in 7.5.2, "File header".

6.3 PDF processors

6.3.1 General

A PDF processor is any software, hardware or any other active agent that writes, reads, updates or otherwise processes a PDF file which conforms to this document, and does so in a manner that conforms to this document. Two specialised classifications of PDF processors are PDF writers, which create PDF documents, and PDF readers, which interpret PDF documents. The purpose of a PDF reader can be to display or print a document, or to extract data from a document. An interactive PDF processor interacts with a user while processing the PDF file and as a consequence may read or write PDF files. In this document, when it is important to distinguish what is written into a PDF file, the more specialised term PDF writer will be used, when it is important to distinguish the act of reading from the contents of a PDF file, the term PDF reader will be used and when it is important to distinguish the act of user interaction while processing a PDF file, the term interactive PDF processor will be used. Otherwise, the more general term PDF processor will be used.

Some PDF processors may also choose to support fragment identifiers (https://www.w3.org/DesignIssues/Fragment.html) which are added at the end of a URI to provide a reference to subordinate content within the target of the URI. These fragment identifiers anchor onto specific content or characteristics of the PDF document to which the URI refers, and represent either an action to be performed on the document when it is opened, or an object within the document (see Annex O, "Fragment identifiers").

6.3.2 Conformance of PDF processors

6.3.2.1 General

If a PDF writer creates a PDF file, the file created shall conform to this document. If a PDF writer adds or amends objects in a pre-existing PDF file then the added or amended elements shall conform to the file format requirements of this document, and shall be consistent with related existing elements.

With the exception of linearized PDF files, all PDF files should be read using the trailer and cross-reference table as described in 7.5, "File structure". Linearized files should be read as specified in Annex F, "Linearized PDF" and Annex G, "Linearized PDF access strategies". Reading a non-linearized file in a serial manner is not reliable because of the way objects are to be processed after an

© ISO 2020 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO 32000-2:2020

incremental update. A PDF processor shall respect and honour standard security (see 7.6.4, "Standard security handler") when it opens a document for processing.

Each PDF processor chooses which subsets of PDF functionality to support. For each subset that the processor chooses to support, the processor shall comply with the applicable provisions in this document.

NOTE The subset standards for PDF, such as PDF/X (ISO 15930), PDF/A (ISO 19005), PDF/E (ISO 24517), PDF/VT (ISO 16612-2) and PDF/UA (ISO 14289), define the term "conforming reader" because they restrict the processing of documents conforming to those standards to strict processing rules. PDF 2.0 is highly generalized to accommodate various processing needs, so the notion of a “conforming reader” is not useful for this document.

6.3.2.2 PDF processors providing rendering

When a PDF processor renders a PDF page, it shall render the page contents as defined in this document. When doing so, the PDF processor shall respect the default (or any other user-specified) optional content definitions (8.11, "Optional content"), if any. A PDF processor shall also render the appropriate appearance stream for all annotations (12.5.5, "Appearance streams") which have appearance streams designated for this purpose as indicated by the annotation flags (see 12.5.3, "Annotation flags"), unless otherwise instructed.

6.3.2.3 Interactive PDF processors

An interactive PDF processor shall follow all the requirements in 6.3.2.2, "PDF processors providing rendering" for rendering. In addition, such a processor shall support all of the interactive aspects of optional content (8.11, "Optional content").

© ISO 2020 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO 32000-2:2020

7 Syntax

7.1 General

This clause covers everything about the syntax of a PDF file at the object, file, and document level. It sets the stage for subsequent clauses, which describe how the contents of a PDF file are interpreted as page descriptions, interactive navigational aids, and application-level logical structure.

PDF file syntax is best understood by considering it as four parts, as shown in "Figure 1 — PDF file components":

- *Objects*. A PDF file is a data structure composed from a small set of basic types of data objects. Subclause 7.2, "Lexical conventions" describes the character set used to write objects and other syntactic elements. Subclause 7.3, "Objects" describes the syntax and essential properties of the objects. Subclause 7.3.8, "Stream objects" provides complete details of the most complex data type, the stream object. Subclause 7.4, "Filters" introduces stream filters primarily used to make PDF files smaller by applying compression technology to stream data.
- *File structure*. The PDF file structure determines how objects are stored in a PDF file, how they are accessed, and how they are updated. This structure is independent of the semantics of the objects. Subclause 7.5, "File structure" describes the file structure. Subclause 7.6, "Encryption" describes a file-level mechanism for protecting a document's content from unauthorised access.
- *Document structure*. The PDF file structure specifies how the basic object types are used to represent components of a PDF file: pages, fonts, annotations, and so forth. Subclause 7.7, "Document structure" describes the overall document structure; later clauses address the detailed semantics of the components.
- *Content streams*. A PDF content stream contains a sequence of instructions describing the appearance of a page or other graphical entity. These instructions, while also represented as objects, are conceptually distinct from the objects that represent the document structure and are described separately. Subclause 7.8, "Content streams and resources" discusses PDF content streams and their associated resources.

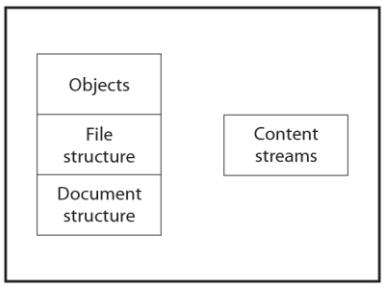

Figure 1 — PDF file components

In addition, this clause describes some data structures, built from basic objects, that are so widely used that they can almost be considered basic object types in their own right. These objects are covered in: 7.9, "Common data structures"; 7.10, "Functions"; and 7.11, "File specifications". Subclause 7.12, "Extensions dictionary" describes a dictionary used to identify developer-defined extensions to this document that are used in the file.

© ISO 2020 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO 32000-2:2020

NOTE Variants of PDF's object and file syntax are also used as the basis for other file formats. These include the Forms Data Format (FDF), described in 12.7.8, "Forms data format", and the Portable Job Ticket Format (PJTF), described in Adobe Technical Note #5620, *Portable Job Ticket Format*.

Annex H, "Example PDF file" presents several examples showing the structure of PDF files.

7.2 Lexical conventions

7.2.1 General

At the most fundamental level, a PDF file is a sequence of bytes. These bytes can be grouped into tokens according to the syntax rules described in subclauses 7.2.2, "Representation" through 7.2.4, "Comments". One or more tokens are assembled to form higher-level syntactic entities, principally objects, which are the basic data values from which a PDF file is constructed.

A PDF file is represented as a sequence of 8-bit bytes, some of which are interpreted as character codes in the ASCII character set and some of which are treated as arbitrary binary data depending upon the context.

The contents (data) of a string or stream object in some contexts are interpreted as character codes in the PDFDocEncoding, UTF-8 or UTF-16 character set as described in 7.9.2, "String object types" and 7.3.8, "Stream objects" respectively.

The contents of a string within a PDF content stream in some situations are interpreted as character codes that select glyphs to be drawn on the page according to a character encoding that is associated with the text font as described in clause 9, "Text".

7.2.2 Representation

A non-encrypted PDF file can be entirely represented using byte values corresponding to the visible printable subset of the ASCII character set defined in INCITS 4-1986 (R2017), plus white-space characters. However, a PDF file is not restricted to the ASCII character set; it may contain arbitrary bytes, subject to the following considerations:

- The tokens that delimit objects and that describe the structure of a PDF file shall use the ASCII character set. In addition all the reserved words and the names used as keys in PDF standard dictionaries and certain types of arrays shall be defined using the ASCII character set.
- The data values of strings and streams objects may be written either entirely using the ASCII character set or entirely in binary data. In actual practice, data that is naturally binary, such as sampled images, is usually represented in binary for compactness and efficiency.
- A PDF file containing binary data shall be transported as a binary file rather than as a text file to ensure that all bytes of the file are faithfully preserved.

NOTE 1 A binary file is not portable to environments that impose reserved character codes, maximum line lengths, end-of-line conventions, or other restrictions.

NOTE 2 In this clause, the usage of the term character is entirely independent of any logical meaning that the value can have when it is treated as data in specific contexts, such as representing human-readable text or selecting a glyph from a font.

7.2.3 Character set

The PDF character set is divided into three classes referred to as regular, delimiter, and white-space

© ISO 2020 – All rights reserved

AVA Documents Document Sharing Hub

Full document details: BS ISO 32000-2:2020

characters. This classification enables the grouping of characters into tokens including separating syntactic constructs such as names and numbers from each other. The rules defined in this subclause apply to all characters in the file except within strings, streams, and comments.

NOTE 1 See 7.3.4, "String objects", 7.3.8, "Stream objects" and 7.2.4, "Comments".

PDF syntax treats any sequence of consecutive white-space characters, not inside of a string or stream, as one character. The characters that are considered white-space characters are shown in "Table 1 — White-space characters" and can be used to separate syntactic constructs such as names and numbers from each other. In all other contexts, PDF syntax treats any sequence of consecutive white-space characters as one character.

Table 1 — White-space characters

Decimal	Hexadecimal	Octal	Name
0	00	000	Null (NUL)
9	09	011	HORIZONTAL TAB (HT)
10	0A	012	LINE FEED (LF)
12	0C	014	FORM FEED (FF)
13	0D	015	CARRIAGE RETURN (CR)
32	20	040	SPACE (SP)

The CARRIAGE RETURN (0Dh) and LINE FEED (0Ah) characters, also called newline characters, shall be treated as end-of-line (EOL) markers. The combination of a CARRIAGE RETURN followed immediately by a LINE FEED shall be treated as one EOL marker. EOL markers may be treated the same as any other white-space characters. However, sometimes an EOL marker is required — that is, preceding a token that shall appear at the beginning of a line.

NOTE 2 The examples in this document use a convention that arranges tokens into lines. However, the examples' use of white-space for indentation is purely for clarity of exposition and need not be included in practical use.

The delimiter characters (,), <, >, [,], /, and % are special (LEFT PARENTHESIS (28h), RIGHT PARENTHESIS (29h), LESS-THAN SIGN (3Ch), GREATER-THAN SIGN (3Eh), LEFT SQUARE BRACKET (5Bh), RIGHT SQUARE BRACKET (5Dh), SOLIDUS (2Fh) and PERCENT SIGN (25h), respectively). They delimit syntactic entities such as arrays, names, and comments. The delimiter characters { and } (LEFT CURLY BRACE (7Bh) and RIGHT CURLY BRACE (7Dh)) are additional delimiter characters within Type 4 PostScript calculator functions (see 7.10.5 "Type 4 (PostScript calculator) functions"). In addition, the double character constructs << (3C3Ch) and >> (3E3Eh) are used to delimit dictionary objects (see 7.3.7, "Dictionary objects"). Any of these delimiters terminates the entity preceding it and is not included in the entity. Delimiter characters are allowed within the scope of a string when following the rules for composing strings; see 7.3.4.2, "Literal strings". The leading (of a string does delimit a preceding entity and the closing) of a string delimits the string's end. "Table 2 — Delimiter characters" shows the delimiter characters.

© ISO 2020 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO 32000-2:2020

Table 2 — Delimiter characters

Glyph	Decimal	Hexadecimal	Octal	Name
(	40	28	050	LEFT PARENTHESIS
)	41	29	051	RIGHT PARENTHESIS
<	60	3C	074	LESS-THAN SIGN
>	62	3E	076	GREATER-THAN SIGN
[	91	5B	133	LEFT SQUARE BRACKET
]	93	5D	135	RIGHT SQUARE BRACKET
{	123	7B	173	LEFT CURLY BRACKET
}	125	7D	175	RIGHT CURLY BRACKET
/	47	2F	057	SOLIDUS
%	37	25	045	PERCENT SIGN

All characters except the white-space characters and delimiters are referred to as regular characters. These characters include bytes that are outside the ASCII character set. A sequence of consecutive regular characters comprises a single token. PDF syntax is case-sensitive; corresponding uppercase and lowercase letters are distinct.

7.2.4 Comments

Any occurrence of the PERCENT SIGN (25h) outside a string or inside a content stream (see 7.8.2, "Content streams") introduces a comment. The comment consists of all characters after the PERCENT SIGN and up to but not including the end-of-the-line marker. A PDF processor should ignore comments. PDF processors shall treat comments as single white-space characters for the purposes of lexical conversion (see 7.2, "Lexical conventions"). That is, a comment separates the token preceding it from the one following it.

EXAMPLE The PDF syntax fragment in this example is syntactically equivalent to just the tokens abc and 123.

```
abc%comment (/%) blah blah blah
123
```

Comments (other than the %PDF–n.m and %%EOF comments described in 7.5, "File structure") have no semantics defined by this document. PDF writers do not need to preserve comments.

NOTE 1 Using comments to communicate information to PDF processors is discouraged. The preferred and documented mechanism is to use PDF names (see Annex E, "Extending PDF").

NOTE 2 This clause was clarified in this document (2020).

© ISO 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 32000-2:2020

7.3 Objects

7.3.1 General

PDF syntax includes nine basic types of objects: boolean values, integers, real numbers, strings, names, arrays, dictionaries, streams, and the null object.

Objects may be labelled so that they can be referred to by other objects. A labelled object is called an indirect object (see 7.3.10, "Indirect objects").

Each object type, their method of creation and their proper referencing as indirect objects is described in 7.3.2, "Boolean objects" through 7.3.10, "Indirect objects".

7.3.2 Boolean objects

Boolean objects represent the logical values of true and false. They appear in PDF files using the keywords **true** and **false**. See 4, "Notation", for more information regarding the notation of these keywords throughout this document.

7.3.3 Numeric objects

PDF syntax provides two types of basic numeric objects: integer and real. Integer objects represent mathematical integers. Real objects represent mathematical real numbers. The range and precision of numbers may be limited by the internal representations used in the computer on which the PDF processor is running; Annex C, "Advice on maximising portability", gives these limits for typical implementations.

An integer shall be written as one or more decimal digits optionally preceded by a sign. The value shall be interpreted as a signed decimal integer and shall be converted to an integer object.

EXAMPLE 1 Integer objects

```
123   43445 +17   -98   0
```

A real value shall be written as one or more decimal digits with an optional sign and a leading, trailing, or embedded PERIOD (2Eh) (decimal point). The value shall be interpreted as a real number and shall be converted to a real object. A real number shall not be present when an integer is expected. Wherever a real number is expected, an integer may be used instead. For example, it is not necessary to write the number 1.0 in real format; the integer 1 is sufficient.

EXAMPLE 2 Real objects

```
34.5  -3.62 +123.6   4. -.002 0
```

A PDF writer shall not use the PostScript language syntax for numbers with non-decimal radices (such as 16#FFFE) or in exponential format (such as 6.02E23).

Throughout this document, the term *number* refers to an object whose type can be either integer or real.

© ISO 2020 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO 32000-2:2020

7.3.4 String objects

7.3.4.1 General

A string object shall consist of a series of zero or more bytes. The maximum length of a string used in the computer on which the PDF processor is running may be subject to implementation limits; see Annex C, "Advice on maximising portability".

String objects shall be written in one of the following two ways:

- As a sequence of literal characters enclosed in parentheses () (using LEFT PARENTHESIS (28h) and RIGHT PARENTHESIS (29h)); see 7.3.4.2, "Literal strings"
- As hexadecimal data enclosed in angle brackets <> (using LESS-THAN SIGN (3Ch) and GREATER-THAN SIGN (3Eh)); see 7.3.4.3, "Hexadecimal strings"

NOTE In many contexts, conventions exist for the interpretation of the contents of a string value. This subclause defines only the basic syntax for writing a string as a sequence of bytes; conventions or rules governing the contents of strings in particular contexts are described with the definition of those particular contexts.

Subclause 7.9.2, "String object types" describes the encoding schemes used for the contents of string objects.

7.3.4.2 Literal strings

A *literal string* shall be written as an arbitrary number of characters enclosed in parentheses (LEFT PARENHESIS (28h) and RIGHT PARENTHESIS (29h)). Any characters may appear in a string except unbalanced parentheses and the backslash (REVERSE SOLIDUS (5Ch)), which shall be treated specially as described in this subclause. Balanced pairs of parentheses within a string require no special treatment.

EXAMPLE 1 The following are valid literal strings:

```
(This is a string)
(Strings can contain newlines
and such.)
(Strings can contain balanced parentheses ()
and special characters ( * ! & } ^ %and so on) .)
(The following is an empty string .)
()
(It has zero (0) length.)
```

Within a literal string, the REVERSE SOLIDUS is used as an escape character. The character immediately following the REVERSE SOLIDUS determines its precise interpretation as shown in "Table 3 — Escape sequences in literal strings". If the character following the REVERSE SOLIDUS is not one of those shown in "Table 3 — Escape sequences in literal strings", the REVERSE SOLIDUS shall be ignored.

Table 3 — Escape sequences in literal strings

Sequence	Meaning
\n	LINE FEED (0Ah) (LF)

© ISO 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 32000-2:2020

Sequence	Meaning
\r	CARRIAGE RETURN (0Dh) (CR)
\t	HORIZONTAL TAB (09h) (HT)
\b	BACKSPACE (08h) (BS)
\f	FORM FEED (0Ch) (FF)
\(	LEFT PARENTHESIS (28h)
\)	RIGHT PARENTHESIS (29h)
\\	REVERSE SOLIDUS (5Ch) (Backslash)
\ddd	Character code ddd (octal)

A PDF writer may split a literal string across multiple lines. The REVERSE SOLIDUS (5Ch) (backslash character) at the end of a line shall be used to indicate that the string continues on the following line. A PDF processor shall disregard the REVERSE SOLIDUS and the end-of-line marker following it when reading the string; the resulting string value shall be identical to that which would be read if the string were not split.

EXAMPLE 2

```
(These \
two strings \
are the same.)
(These two strings are the same.)
```

An end-of-line marker appearing within a literal string without a preceding REVERSE SOLIDUS shall be treated as a byte value of (0Ah), irrespective of whether the end-of-line marker was a CARRIAGE RETURN (0Dh), a LINE FEED (0Ah), or both.

EXAMPLE 3

```
(This string has an end-of-line at the end of it.
)
(So does this one.\n)
```

The *ddd* escape sequence provides a way to represent characters outside the printable ASCII character set.

EXAMPLE 4

```
(This string contains \245two octal characters\307.)
```

The number *ddd* may consist of one, two, or three octal digits; high-order overflow shall be ignored. Three octal digits shall be used, with leading zeros as needed, if the next character of the string is also a digit.

EXAMPLE 5

the literal

© ISO 2020 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO 32000-2:2020

```
(\0053)
```

denotes a string containing two characters, \005 (Control-E) followed by the digit 3, whereas both

```
(\053)
```

and

```
(\53)
```

denote strings containing the single character \053, a plus sign (+).

Since any 8-bit value may appear in a string (with proper escaping for REVERSE SOLIDUS (backslash) and unbalanced PARENTHESES) this \ddd notation provides a way to specify characters outside the ASCII character set by using ASCII characters only. However, any 8-bit value may appear in a string, represented either as itself or with the \ddd notation described.

When a document is encrypted (see 7.6, "Encryption"), all of its strings are encrypted; the encrypted string values contain arbitrary 8-bit values. When writing encrypted strings using the literal string form, the PDF writer shall follow the rules described. That is, the REVERSE SOLIDUS character shall be used as an escape to specify unbalanced PARENTHESES or the REVERSE SOLIDUS character itself. The REVERSE SOLIDUS may, but is not required, to be used to specify other, arbitrary 8-bit values.

7.3.4.3 Hexadecimal strings

Strings may also be written in hexadecimal form, which is useful for including arbitrary binary data in a PDF file. A hexadecimal string shall be written as a sequence of hexadecimal digits (0–9 and A–F or a–f) encoded as ASCII characters and enclosed within angle brackets (using LESS-THAN SIGN (3Ch) and GREATER-THAN SIGN (3Eh)).

EXAMPLE 1

```
<4E6F762073686D6F7A206B6120706F702E>
```

Each pair of hexadecimal digits defines one byte of the string. White-space characters (see “Table 1 — White-space characters”) shall be ignored. The empty string can be represented in a hexadecimal string by a LESS-THAN SIGN (3Ch) immediately followed by a GREATER-THAN SIGN (3Eh).

If the final digit of a hexadecimal string is missing — that is, if there is an odd number of digits — the final digit shall be assumed to be 0.

EXAMPLE 2

```
<901FA3>
```

is a 3-byte string consisting of the characters whose hexadecimal codes are 90, 1F, and A3, but

```
<901FA>
```

is a 3-byte string containing the characters whose hexadecimal codes are 90, 1F, and A0.

7.3.5 Name objects

Beginning with PDF 1.2 a name object is an atomic symbol uniquely defined by a sequence of any characters (8-bit values) except null (character code 0). Uniquely defined means that any two name objects that, after all escaping is expanded (see below), and the resulting sequences of bytes are not an exact binary match denote different objects. Atomic means that a name has no internal structure; although it is defined by a sequence of characters, those characters are not considered elements of the

© ISO 2020 – All rights reserved

AVA Documents Document Sharing Hub

Full document details: BS ISO 32000-2:2020

name.

Even though a name may contain any characters except null, in order to express a name following the general rules for parsing PDF files, when written into a PDF file names should be encoded as described in the succeeding paragraphs.

When writing a name in a PDF file, a SOLIDUS (2Fh) (/) shall be used to introduce a name. The SOLIDUS is not part of the name but is a prefix indicating that what follows is a sequence of characters representing the name in the PDF file and shall follow these rules:

a) A NUMBER SIGN (23h) (#) in a name shall be written by using its 2-digit hexadecimal code (23), preceded by the NUMBER SIGN.

b) Any character in a name that is a regular character (other than NUMBER SIGN) shall be written as itself or by using its 2-digit hexadecimal code, preceded by the NUMBER SIGN.

c) Any character that is not a regular character shall be written using its 2-digit hexadecimal code, preceded by the NUMBER SIGN only.

NOTE 1 There is not a unique encoding of names into the PDF file because regular characters can be coded in either of two ways.

No token delimiter (such as white-space) occurs between the SOLIDUS and the encoded name. White-space used as part of a name shall always be coded using the 2-digit hexadecimal notation.

Regular characters that are outside the range EXCLAMATION MARK(21h) (!) to TILDE (7Eh) (~) should be written using the hexadecimal notation.

The token SOLIDUS (a slash followed by no regular characters) introduces a unique valid name defined by the empty sequence of characters.

NOTE 2 The examples shown in "Table 4 — Examples of literal names" and containing # are not valid literal names in PDF 1.0 or 1.1.

Table 4 — Examples of literal names

Syntax for Literal name	Resulting Name
/Name1	Name1
/ASomewhatLongerName	ASomewhatLongerName
/A;Name_With-Various***Characters?	A;Name_With-Various***Characters?
/1.2	1.2
/$$	$$
/@pattern	@pattern
/.notdef	.notdef
/Lime#20Green	Lime Green
/paired#28#29parentheses	paired()parentheses
/The_Key_of_F#23_Minor	The_Key_of_F#_Minor

© ISO 2020 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO 32000-2:2020

Syntax for Literal name	Resulting Name
/A#42	AB

In PDF syntax, literal names shall always be introduced by the SOLIDUS character (/), unlike keywords such as **stream**, **endstream**, and **obj**.

NOTE 3 This document follows a typographic convention of writing names without the leading SOLIDUS when they appear in running text and tables. For example, Type and FullScreen denote names that would actually be written in a PDF file (and in code examples in this document) as /Type and /FullScreen.

The maximum length of a name used in the computer on which the PDF processor is running may be subject to an implementation limit; see Annex C, "Advice on maximising portability". The limit applies to the number of characters in the name's internal representation. For example, the name /A#20B has three characters (A, SPACE, B), not six.

As stated above, name objects shall be treated as atomic within a PDF file. Ordinarily, the bytes making up the name are never treated as text to be presented to a human user or to an application external to a PDF processor. However, occasionally the need arises to treat a name object as text, such as one that represents a font name (see the **BaseFont** entry in "Table 109 — Entries in a Type 1 font dictionary"), a colourant name in a **Separation** or **DeviceN** colour space, or a structure type (see 14.7.3, "Structure types").

In such situations, the sequence of bytes making up the name object should be interpreted according to UTF-8, a variable-length byte-encoded representation of Unicode in which the printable ASCII characters have the same representations as in ASCII. This enables a name object to represent text virtually in any natural language, subject to the implementation limit on the length of a name.

NOTE 4 PDF syntax does not prescribe what UTF-8 sequence to choose for representing any given piece of externally specified text as a name object. In some cases, multiple UTF-8 sequences can represent the same logical text. PDF name objects are considered distinct objects if, after all escaping is expanded, the resulting sequences of bytes are not an exact binary match.

7.3.6 Array objects

An *array object* is a one-dimensional collection of objects arranged sequentially. Unlike arrays in many other computer languages, PDF arrays may be heterogeneous; that is, an array's elements may be any combination of numbers, strings, dictionaries, or any other objects, including other arrays. An indirect reference (see 7.3.10, "Indirect objects") shall be used to refer to a stream object, and may also be used to refer to any other kind of object. An array may have zero elements.

An array shall be written as a sequence of objects enclosed in SQUARE BRACKETS (using LEFT SQUARE BRACKET (5Bh) and RIGHT SQUARE BRACKET (5Dh)).

EXAMPLE

```
[549 3.14 false (Ralph) /SomeName]
```

PDF syntax directly supports only one-dimensional arrays. Arrays of higher dimension can be constructed by using arrays as elements of arrays, nested to any depth.

© ISO 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 32000-2:2020

7.3.7 Dictionary objects

A dictionary object is an associative table containing pairs of objects, known as the dictionary's entries. The first element of each entry is the key and the second element is the value. The key shall be a name (unlike dictionary keys in the PostScript language, which may be objects of any type). All keys shall be direct objects. The value may be any kind of object, including another dictionary. An indirect reference (see 7.3.10, "Indirect objects") shall be used as the value to refer to a stream object, and may also be used to refer to any other kind of object. A dictionary entry whose value is **null** (see 7.3.9, "Null object") shall be treated the same as if the entry does not exist. (This differs from the PostScript language, where null behaves like any other object as the value of a dictionary entry.) The maximum number of entries in a dictionary used in the computer on which the PDF processor is running may be subject to implementation limits; see Annex C, "Advice on maximising portability". A dictionary may have zero entries.

NOTE 1 The statement requiring dictionary keys to always be direct objects was added in this document (2020).

The entries in a dictionary represent an associative table and as such shall be unordered even though an arbitrary order may be imposed upon them when written in a file. That ordering shall be ignored.

Multiple entries in the same dictionary shall not have the same key.

A dictionary shall be written as a sequence of key-value pairs enclosed in double angle brackets (<<...>>) (using LESS-THAN SIGNs (3Ch) and GREATER-THAN SIGNs (3Eh)).

EXAMPLE

```
<</Type /Example
         /Subtype /DictionaryExample
         /Version 0.01
         /IntegerItem 12
         /StringItem (a string)
         /Subdictionary <<
               /Item1 0.4
               /Item2 true
               /LastItem (not !)
               /VeryLastItem (OK)
               >>
>>
```

NOTE 2 Do not confuse the double angle brackets with single angle brackets (< and >) (using LESS-THAN SIGN (3Ch) and GREATER-THAN SIGN (3Eh)), which delimit a hexadecimal string (see 7.3.4.3, "Hexadecimal strings").

Dictionary objects are the main building blocks of a PDF file. They are commonly used to collect and tie together the attributes of a complex object, such as a font or a page of the document, with each entry in the dictionary specifying the name and value of an attribute. By convention, the **Type** entry of such a dictionary, if present, identifies the type of object the dictionary describes. In some cases, a **Subtype** entry (sometimes abbreviated **S**) may be used to further identify a specialised subcategory of the general type. The value of the **Type** or **Subtype** entry shall always be a name. For example, in a font dictionary, the value of the **Type** entry shall always be *Font*, whereas that of the **Subtype** entry may be *Type1*, *TrueType*, or one of several other values.

The value of the **Type** entry can almost always be inferred from context. The value of an entry in a page's font resource dictionary, for example, shall be a font object; therefore, the **Type** entry in a font

© ISO 2020 – All rights reserved
AVA Documents Document Sharing Hub
Full document details: BS ISO 32000-2:2020

dictionary serves primarily as documentation and as information for error checking. The **Type** entry shall not be required unless so stated in its description; however, if the entry is present, it shall have the correct value. In addition, the value of the **Type** entry in any dictionary, even in private data, shall be either a name defined in this document or a registered name; see Annex E, "Extending PDF", for details.

7.3.8 Stream objects

7.3.8.1 General

A stream object, like a string object, is a sequence of bytes. Furthermore, a stream may be of unlimited length, whereas a string shall be subject to an implementation limit. For this reason, objects with potentially large amounts of data, such as images and page descriptions, shall be represented as streams.

NOTE 1 This subclause describes only the syntax for writing a stream as a sequence of bytes. The context in which a stream is referenced determines what the sequence of bytes represents.

A stream shall consist of a dictionary followed by zero or more bytes bracketed between the keywords **stream** (followed by newline) and **endstream**:

dictionary
`stream`
...Zero or more bytes...
`endstream`

All streams shall be indirect objects (see 7.3.10, "Indirect objects") and the stream dictionary shall be a direct object. The keyword **stream** that follows the stream dictionary shall be followed by an end-of-line marker consisting of either a CARRIAGE RETURN and a LINE FEED or just a LINE FEED, and not by a CARRIAGE RETURN alone. The sequence of bytes that make up a stream lie between the end-of-line marker following the **stream** keyword and the **endstream** keyword; the stream dictionary specifies the exact number of bytes. There should be an end-of-line marker after the data and before **endstream**; this marker shall not be included in the stream length. There shall not be any extra bytes, other than white-space, between **endstream** and **endobj**.

NOTE 2 Without the restriction against following the keyword **stream** by a CARRIAGE RETURN alone, differentiating a stream that uses CARRIAGE RETURN as its end-of-line marker and has a LINE FEED as its first byte of data from one that uses a CARRIAGE RETURN–LINE FEED sequence to denote end-of-line is not possible.

Alternatively, beginning with PDF 1.2, the bytes may be contained in an external file, in which case the stream dictionary specifies the file, and any bytes between **stream** and **endstream** shall be ignored by a PDF processor.

"Table 5 — Entries common to all stream dictionaries" lists the entries common to all stream dictionaries; certain types of streams may have additional dictionary entries, as indicated where those streams are described. The optional entries regarding filters for the stream indicate whether and how the data in the stream shall be transformed (decoded) before it is used. Filters are described further in 7.4, "Filters".

7.3.8.2 Stream extent

Every stream dictionary shall have a **Length** entry that indicates how many bytes of the PDF file are

© ISO 2020 – All rights reserved

AVA Documents Document Sharing Hub
Full document details: BS ISO 32000-2:2020

used for the stream's data. (If the stream has a filter, **Length** shall be the number of bytes of encoded data.) In addition, most filters are defined so that the data shall be self-limiting; that is, they use an encoding scheme in which an explicit end-of-data (EOD) marker delimits the extent of the data. Finally, streams are used to represent many objects from whose attributes a length can be inferred. All of these constraints shall be consistent.

EXAMPLE An image with 10 rows and 20 columns, using a single colour component and 8 bits per component, requires exactly 200 bytes of image data. If the stream uses a filter, there needs to be enough bytes of encoded data in the PDF file to produce those 200 bytes. An error occurs if **Length** is too small, if an explicit EOD marker occurs too soon, or if the decoded data does not contain 200 bytes.

Streams shall also not contain too much data, with the exception that there may be an extra end-of-line marker in the PDF file before the keyword **endstream**.

Table 5 — Entries common to all stream dictionaries

Key	Type	Value
Length	integer	(*Required*) The number of bytes from the beginning of the line following the keyword **stream** to the last byte just before the keyword **endstream**. (There may be an additional EOL marker, preceding **endstream**, that is not included in the count and is not logically part of the stream data.) See 7.3.8.2, "Stream extent", for further discussion.
Filter	name or array	(*Optional*) The name, or an array of zero, one or several names, of filter(s) that shall be applied in processing the stream data found between the keywords **stream** and **endstream**. Multiple filters shall be specified in the order in which they are to be applied. NOTE It is not recommended to include the same filter more than once in a **Filter** array.
DecodeParms	dictionary or array	(*Optional*) A parameter dictionary or an array of such dictionaries, used by the filters specified by **Filter**, respectively. If there is only one filter and that filter has parameters, **DecodeParms** shall be set to the filter's parameter dictionary unless all the filter's parameters have their default values, in which case the **DecodeParms** entry may be omitted. If there are multiple filters and any of the filters has parameters set to nondefault values, **DecodeParms** shall be an array with one entry for each filter in the same order as the **Filter** array: either the parameter dictionary for that filter, or the null object if that filter has no parameters (or if all of its parameters have their default values). If none of the filters have parameters, or if all their parameters have default values, the **DecodeParms** entry may be omitted.
F	file specification	(*Optional; PDF 1.2*) The file containing the stream data. If this entry is present, the bytes between **stream** and **endstream** shall be ignored. However, the **Length** entry should still specify the number of those bytes (usually, there are no bytes and **Length** is 0). The filters that are applied to the file data shall be specified by **FFilter** and the filter parameters shall be specified by **FDecodeParms**.
FFilter	name or array	(*Optional; PDF 1.2*) The name of a filter to be applied in processing the data found in the stream's external file, or an array of zero, one or several such names. The same rules apply as for **Filter**.

© ISO 2020 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO 32000-2:2020

Key	Type	Value
FDecodeParms	dictionary or array	(*Optional; PDF 1.2*) A parameter dictionary, or an array of such dictionaries, used by the filters specified by **FFilter**, respectively. The same rules apply as for **DecodeParms**.
DL	integer	(*Optional; PDF 1.5*) A non-negative integer representing the number of bytes in the decoded (defiltered) stream. This value is only a hint; for some stream filters, it may not be possible to determine this value precisely.

7.3.9 Null object

The null object has a type and value that are unequal to those of any other object. There shall be only one object of type null, denoted by the keyword **null**. An indirect object reference (see 7.3.10, "Indirect objects") to a nonexistent object shall be treated the same as a null object. Specifying the null object as the value of a dictionary entry (7.3.7, "Dictionary objects") shall be equivalent to omitting the entry entirely.

7.3.10 Indirect objects

Any object in a PDF file may be labelled as an indirect object. This gives the object a unique object identifier by which other objects can refer to it (for example, as an element of an array or as the value of a dictionary entry). The object identifier shall consist of two parts:

- A positive integer object number. Indirect objects may be numbered sequentially within a PDF file, but this is not required; object numbers may be assigned in any arbitrary order.
- A non-negative integer generation number. In a newly created file, all indirect objects shall have generation numbers of 0. Non-zero generation numbers may be introduced when the file is later updated; see 7.5.4, "Cross-reference table" and 7.5.6, "Incremental updates"

Together, the combination of an object number and a generation number shall uniquely identify an indirect object.

The definition of an indirect object in a PDF file shall consist of its object number and generation number (separated by white-space), followed by the value of the object bracketed between the keywords **obj** and **endobj**.

EXAMPLE 1 Indirect object definition

```
12 0 obj
(Brillig)
endobj
```

Defines an indirect string object with an object number of 12, a generation number of 0, and the string value *Brillig*.

The object may be referred to from elsewhere in the file by an indirect reference. Such indirect references shall consist of the object number, the generation number, and the keyword **R** (with white-space separating each part):

```
12 0 R
```

© ISO 2020 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 32000-2:2020

NOTE (2020) Any object outside of an object stream (see 7.5.7, "Object streams") can consist solely of an object reference. PDF syntax thus permits chains of such objects.

Beginning with PDF 1.5, indirect objects may reside in object streams (see 7.5.7, "Object streams"). They are referred to in the same way; however, their definition shall not include the keywords **obj** and **endobj**, and their generation number shall be zero.

An indirect reference to an undefined object shall not be considered an error by a PDF processor; it shall be treated as a reference to the null object.

EXAMPLE 2 If a file contains the indirect reference 17 0 R but does not contain the corresponding definition then the indirect reference is considered to refer to the null object.

Except where documented to the contrary, any object value may be a direct or an indirect reference; the semantics are equivalent.

EXAMPLE 3 The following shows the use of an indirect object to specify the length of a stream. The value of the stream's **Length** entry is an integer object that follows the stream in the file. This allows applications that generate PDF files in a single pass to defer specifying the stream's length until after its contents have been generated.

```
7 0 obj
      <<Length 8 0 R>>                              %An indirect reference to object 8
stream
      BT
         /F1 12 Tf
         72 712 Td
         (A stream with an indirect length) Tj
      ET
endstream
endobj

8 0 obj
   77                                               %The length of the preceding stream
endobj
```

7.4 Filters

7.4.1 General

Stream filters are introduced in 7.3.8, "Stream objects". An option when reading stream data is to decode it using a filter to produce the original non-encoded data. Whether to do so and which decoding filter or filters to use are specified in the stream dictionary.

EXAMPLE 1 If a stream dictionary specifies the use of an **ASCIIHexDecode** filter, an application reading the data in that stream will need to transform the ASCII hexadecimal-encoded data in that stream in order to obtain the original binary data.

A PDF writer may encode data in a stream (for example, data for sampled images) to compress it or to convert it to a portable ASCII representation (or both). A PDF reader shall invoke the corresponding decoding filter or filters to convert the information back to its original form.

The filter or filters for a stream shall be specified by the **Filter** entry in the stream's dictionary (or the **FFilter** entry if the stream is external). Filters may be cascaded to form a pipeline that passes the stream through two or more decoding transformations in sequence. For example, data encoded using LZW and ASCII base-85 encoding (in that order) shall be decoded using the following entry in the stream dictionary:

© ISO 2020 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO 32000-2:2020