Application Notes for PDF/X Standards

Version 4

(PDF/X-1a:2003, PDF/X-2:2003 & PDF/X-3:2003)
September 2006

Prepared by

CGATS

The Committee for Graphic Arts Technologies Standards

Administered by

NPES The Association for Suppliers of Printing, Publishing and Converting Technologies

NPES

Foreword

NPES The Association for Suppliers of Printing, Publishing and Converting Technologies performs secretariat functions for the Committee for Graphic Arts Technologies Standards (CGATS) and its subcommittees, working groups and task forces. This function includes publishing its documents. NPES it is not the author of this document, nor is it an entity that adopts any standard.

The information in this document was developed by members of CGATS Subcommittee 6 Task Force 1 (PDF/X). Although its contents were approved by ballot of the voting members of the subcommittee, this document has not been developed under the consensus process of the American National Standards Institute (ANSI), and is not an ANSI consensus document. It does not have the status of a standard or technical report, and was developed to disseminate additional information regarding the ISO 15930 series of standards on PDF/X.

At the Subcommittee's request, NPES is making this document publicly available at no cost for informational and reference purposes only.

This document does not relieve the reader of any burden imposed by law, and the reader should independently investigate and verify the performance, results, safety and efficacy of his or her own product or process, even if it follows any recommendations contained in this document. CGATS, its subcommittees, and NPES expressly disclaim liability for any damages or injury arising out of or related to the use of information contained in this document.

This document may be updated as needed by the developing CGATS subcommittee. Such updates will be identified by version number or date, and will be available on the NPES Standards Workroom.

NPES Standards Department
1899 Preston White Drive
Reston, VA 20191-4367
USA
Tel: 703-264-7200
Fax: 703-620-0994
E-mail: standards@npes.org

Contents

Application Notes for PDF/X Standards

1 Introduction

The PDF/X family of standards was, and is continuing to be, developed by ISO/TC130, Graphic technology. The goal is to provide an efficient vehicle for the exchange of combinations of vector and raster digital data, representing print ready material, within the graphic arts industry. It is a multi-part standard with several levels of compliance. The intent is that if a sender and receiver of PDF/X files use applications that have compatible levels of compliance, digital data workflows can flow seamlessly between them and be processed without surprises to achieve the rendering intended by the client.

It is important to note that while these standards ensure the unambiguous exchange of the data, they make no statement about the quality of the data contained in the files. Because such issues as raster data resolution, copydot resolution, printing conditions, annotation requirements (crop marks, register marks, etc.), pages per file, spot colour usage, allowed font types, etc. are largely market and/or publisher specific they are not specified in these standards. Rather, they are felt to be the province of trade associations and or publishers who have already begun to publish such specifications for these parameters in support of the use of the PDF/X standards. Preflight tools that can validate to both a PDF/X standard and to additional geographic or market requirements are often described as supporting “PDF/X Plus”.

In many situations simply specifying and/or validating a file as having a specific level of PDF/X compliance will be the key element in assuring an efficient workflow between multiple participants in a printing and publishing endeavor.

The family of PDF/X standards currently consists of:

- ISO 15930-1:2001, *Graphic technology — Prepress digital data exchange — Use of PDF — Part 1: Complete exchange using CMYK data (PDF/X-1 and PDF/X-1a)*; specifies the use of the Portable Document Format (PDF) v1.3 for the dissemination of complete digital data, in a single exchange, that contains all elements ready for final print reproduction. CMYK and spot color data are supported in any combination.

- ISO 15930-3:2002, *Graphic technology — Prepress digital data exchange — Use of PDF — Part 3: Complete exchange suitable for colour-managed workflows (PDF/X-3)*; specifies the use of the Portable Document Format (PDF) v1.3 for the dissemination of complete digital data, in a single exchange, that contains all elements necessary for final print reproduction. Color-managed, CMYK, gray, RGB or spot color data are supported.

- ISO 15930-4:2003, *Graphic technology — Prepress digital data exchange — Use of PDF — Part 4: Complete exchange of CMYK and spot colour printing data using PDF 1.4 (PDF/X-1a)*; specifies the use of the Portable Document Format (PDF) v1.4 for the dissemination of complete digital data, in a single exchange, that contains all elements ready for final print reproduction. CMYK and spot color data are supported in any combination.

- ISO 15930-5:2003, *Graphic technology — Prepress digital data exchange — Use of PDF — Part 5: Partial exchange of printing data using PDF 1.4 (PDF/X-2)*; specifies the use of the Portable Document Format (PDF) v1.4 for the dissemination of digital data, where all elements necessary for final print reproduction are either included or provision is made for unique identification. Color-managed, CMYK, gray, RGB or spot color data are supported.

- ISO 15930-6:2003, *Graphic technology — Prepress digital data exchange — Use of PDF — Part 6: Complete exchange of printing data suitable for colour-managed workflows using PDF 1.4 (PDF/X-3);* specifies the use of the Portable Document Format (PDF) v1.4 for the dissemination of complete digital data, in a single exchange, that contains all elements necessary for final print reproduction. Color-managed, CMYK, gray, RGB or spot color data are supported.

A file conforming to parts 1, 3, 4 or 6 of ISO must contain all the content information necessary to process and render the document, as intended by the sender. This exchange requires no prior knowledge of the sending and receiving environments and is sometimes referred to as "blind" exchange. It is platform and transport independent.

Use of files conforming to ISO 15930-2 requires prior discussion and agreement between the sender and intended receiver of the files.

Table 1 — PDF/X conformance levels

Conformance Level	Part of ISO 15930	Complete exchange	Colour-managed data permitted	Print characterization spaces supported	PDF version
PDF/X-1:2001	1	Yes	No	CMYK	1.3
PDF/X-1a:2001	1	Yes	No	CMYK	1.3
PDF/X-1a:2003	4	Yes	No	CMYK	1.4
PDF/X-2:2003	5	No	Yes	Gray, RGB, CMYK	1.4
PDF/X-3:2002	3	Yes	Yes	Gray, RGB, CMYK	1.3
PDF/X-3:2003	6	Yes	Yes	Gray, RGB, CMYK	1.4

These application notes currently only include information applicable to:

— ISO 15930-4:2003	Defines the PDF/X-1a:2003 conformance level
— ISO 15930-5:2003	Defines the PDF/X-2:2003 conformance level
— ISO 15930-6:2003	Defines the PDF/X-3:2003 conformance level

See Version 3 of these Application Notes for information applicable to:

— ISO 15930-1:2001	Defines the PDF/X-1:2001 and PDF/X-1a:2001 conformance levels
— ISO 15930-3:2002	Defines the PDF/X-3:2002 conformance level

It is anticipated that a variety of products will be developed around the PDF/X family of standards, such as readers (including viewers) and writers of PDF/X files, and products that offer combinations of these features. Different products will incorporate various capabilities to prepare, interpret and process conforming files based on the application needs as perceived by the suppliers of the products. However, it is important to note that a conforming reader must be able to read and appropriately process all files conforming to a specified conformance level.

This document is not a part of the ISO 15930 family, but is intended to supplement these standards by providing additional information and guidance. None of the information contained herein should be construed as modifying the standard. This document is intended for implementers of writers and readers of PDF/X files. It also provides information that will help workflow designers use PDF/X to advantage.

This document was developed as a cooperative effort between the technical experts of ISO/TC130/WG2/TF2 and CGATS/SC6/TF1.

To anyone who acknowledges that these Application Notes are provided "AS IS", WITH NO EXPRESS OR IMPLIED WARRANTY: permission to use, copy and distribute them for any purpose is hereby granted without fee, provided that the contents of these notes are not altered, including the NPES copyright notice tag. Neither NPES nor CGATS makes any claims or representations about the completeness of these Application Notes.

These Application Notes are subject to revision and enhancement. The most current version of this document can be found in the NPES standards workroom at http://www.npes.org/standards/tools.html. Some earlier revisions may be maintained at the same site.

Comments and suggestions should be sent to the CGATS Secretariat at standards@npes.org.

2 General

2.1 Document structure

Unless otherwise noted, the material in this section applies to all conformance levels addressed by this document, which are simply referred to as PDF/X. Sections 3, 4 and 5 provide details specific to each conformance level.

2.2 Conflict with standards

If there is a conflict between these Application Notes and any part of ISO 15930:2003, the standard will always take precedence.

2.3 Identification and conformance

A PDF file is identified by a header, the first line of which is a PDF comment that begins with “%PDF-” and is followed by a version number. While each conformance level requires that files conform to a specific version of PDF, they also state that the version number in the first line of the header is not relevant for determining PDF/X compliance.

A conforming PDF/X-1a:2003 file is identified by:

a) being a PDF file; and
b) having the **GTS_PDFXVersion** key in the Info dictionary with a value of *(PDF/X-1a:2003)*

A conforming PDF/X-2:2003 file is identified by:

a) being a PDF file;
b) having the **GTS_PDFXVersion** key in the Info dictionary with a value of *(PDF/X-2:2003)*

A conforming PDF/X-3:2003 file is identified by:

a) being a PDF file;
b) having the **GTS_PDFXVersion** key in the Info dictionary with a value of *(PDF/X-3:2003)*

In all cases “being a PDF file” means conforming to the *PDF Reference: Adobe Portable Document Format, Version 1.4 ,* as extended by *Errata for PDF Reference, third edition* (as published in the first printing, November 2001; last modified 18 June 2003). Throughout the rest of this document that combination will be referred to as the *PDF Reference*.

A conforming PDF/X file has two kinds of content:

- content that affects the final print reproduction of the composite entity;

- other content such as non-printing annotations, metadata, etc.

The PDF/X standards state that a conforming file may include valid PDF features beyond those described in the standard provided they do not affect final print reproduction of the composite entity. Implementers of PDF/X writers should be aware that features added in versions of PDF later than that defined in the *PDF Reference* might affect the final print reproduction. This includes issues such as JPEG2000 compression, 16 bit images, etc., from the PDF 1.5 specification. These new features should not be used when creating PDF/X files.

It is recommended that any features described in versions of the PDF specification earlier than 1.4, but which are not included in the *PDF Reference* itself, should not be used in PDF/X files, even if those features are not specifically prohibited by the PDF/X standards.

It is the responsibility of the producer/writer of PDF/X files to ensure compliance with the appropriate PDF/X file format.

2.4 References to PDF/X

Like other standards, ISO 15930 may be revised from time to time; therefore, references to the standard in product literature, packaging, etc., describing PDF/X compliant applications should include the year of publication; for example, ISO 15930-4:2003, or PDF/X-1a:2003.

2.5 Selecting the appropriate PDF/X conformance level for exchange

The PDF/X variant in and of itself will not identify an exchange as blind or non-blind; rather, the nature of an exchange will be governed by business discussions surrounding the exchange. Both PDF/X-1a and PDF/X-3 can be used for exchanges in which prior discussion takes place, as well as for blind exchanges. Therefore, the PDF/X-2 variant should not be used if the only determining factor is the need for a non-blind exchange.

Moreover, producers/writers of PDF/X files should choose the most restrictive conformance level that applies to the file. For example, a complete exchange using only CMYK and spot color data should be created as a PDF/X-1a file, even though either PDF/X-2 or PDF/X-3 conformance levels could be used instead. Using a less restrictive conformance level than is required may result in additional workflow processing steps, or even rejection of supplied files if a PDF/X reader is designed to process only more restrictive conformance levels.

2.6 File translation

PDF/X is an exchange standard. In some workflows, it may be appropriate for the receiver to translate PDF/X files into other file formats, provided the intended rendering of the original PDF/X file is preserved.

2.7 Output considerations for PDF/X

2.7.1 Intent of the PDF/X file’s creator

The overprint settings applied to objects within a PDF/X file reflect the intent of the PDF/X file's creator and must be preserved by applications or RIPs used in an imaging workflow. Applications and RIPs not configured to honor overprint settings contained in the PDF/X may produce unintended imaging results.

Many PDF/X workflows require that the file is translated to PostScript prior to final output. A PDF/X file may contain PDF 1.3 constructs such as **DeviceN** or **smooth shading**, which when translated to PostScript may only be understood by a PostScript Language Level 3 compatible application or device. If PostScript Language Level 2 compatibility is required, those elements that use PDF 1.3 exclusive features will have to be modified to use PostScript Level 2 operators that will accurately simulate the desired effects in order to preserve the intent of the PDF/X file's creator.

2.7.2 Validation of PDF/X workflows

Most of these application notes concentrate on the creation of conforming PDF/X files. However, once a conforming file has been created and validated, there is a chance that the file may not be interpreted correctly in subsequent processes.

Some prepress workflows may include a conversion from PDF to EPS, especially if they are using standard design tools to composite a number of partial page submissions together. It is important to ensure that the files are exported to at least PostScript Level 2, and preferably PostScript Language Level 3 EPS files. PostScript Level 1 cannot carry the overprinting information required for many print jobs. The level selected must obviously be chosen in the context of the software used later in the workflow.

The same guidelines should be followed if printing from PDF to PostScript rather than exporting.

Any applications and tools that can only process PostScript Level 1or Level 2 should now be regarded as obsolete and upgraded, since it is extremely unlikely that they can be configured to render files reliably in conformance with PDF/X.

For example, if the PDF/X file contains smooth shading and it is output as a PostScript Level 2 EPS file, the **smooth shading** construct will be lost, although the appearance of the shading may still be reproduced using other means. Unless done properly, such conversions may introduce undesirable artifacts in final output, or may lose the shading completely.

A PDF/X file may be placed on a page in a design application and printed from there to a proofer or RIP using PostScript. Most major design applications do not create separated PostScript correctly under these circumstances. Correct output is much more likely to be achieved if composite PostScript is generated and in-RIP separations are used instead.

Note that some major design applications do not generate correct composite PostScript for all aspects of elements created within that application itself, especially when spot colors are used. Some care must be taken to design and construct a page on which all elements can be printed correctly.

Many RIPs used in the graphic arts provide options to override overprint settings, e.g. ignoring all requests for overprinting. Such options should be set to match the requirements of PDF/X; refer to the RIP manufacturer for guidance. Application developers should read the *PDF Reference* and *Adobe Technical Note # 5044*.

Several groups have developed test suites in order to confirm that the PDF/X files created will be processed correctly:

For PDF/X-3, and for some aspects of PDF/X-1a, the Altona Suite is available from www.altonatestsuite.com.

The Ghent PDF Workgroup has developed a set of test patches specifically for testing workflows that include aggregation and printing of PDF/X files. these are available from www.gwg.org. These were developed from the “Kensington Suite” project, originally started by the DDAP (Digital Distribution of Advertising for Publications) and NAA (Newspaper Association of America), and now maintained by the Ghent PDF Workgroup.

Additional tools for testing compliance with PDF/X processing requirements are also available at these web sites.

2.8 Compression of PDF objects

The PDF/X standard does not specify how compression should be performed on the individual elements that are contained either directly in or as “embedded” parts of the PDF file. Any PDF stream may be compressed with any PDF-supported lossless compression technique, other than LZW, that is appropriate to the data.

The only lossy compression technique that can be used is JPEG, which can be used only with contone raster images. JBIG2 compression may not be used. The use of lossy compression may result in degradation of image quality.

LZW compression is prohibited in PDF/X. Note that some versions of Adobe Acrobat Distiller use LZW compression in the creation of thumbnails. If creation of thumbnails from Acrobat Distiller is desired, then the LZW compressed thumbnails must be recompressed using a compression method approved for PDF/X. Other applications may produce PDF files that require similar changes when converting to PDF/X.

Although all readers are required to implement all specified compression schemes, writers are not required to use compression.

2.9 Compression of entire PDF/X files

External compression can be useful to assist in identification of file corruption during transmission, or can be used to provide a mechanism for password protection of sensitive files. If external compression is to be applied to a PDF/X file, either for storage or transmission, a lossless compression method must be used. The use of lossless compression will ensure that the integrity of such a file is preserved.

The use of external compression (e.g. WINZIP or STUFFIT) in exchange requires prior understanding between sender and receiver. While it is possible to compress an entire PDF/X file using any number of available compression applications, the act of externally compressing PDF/X files violates the spirit of “blind exchange” due to the lack of compatibility between various compression/decompression applications. Therefore, any PDF/X file that has been externally compressed is no longer a conforming PDF/X file while compressed, and it should be assumed that a compliant PDF/X reader cannot read a compressed file without prior decompression.

NOTE Compressing a file that contains compressed elements may cause the file to increase in size.

2.10 Box usage and management

The inclusion of the bounding boxes, as specified in *PDF Reference*, is a required operation in the process of creating a PDF/X file.

For all PDF files, the **MediaBox** is required. Additionally, each PDF/X page shall include either an **ArtBox** or **TrimBox**, but not both. The inclusion of a **BleedBox** is optional. If a **BleedBox** is present, neither the **ArtBox** nor the **TrimBox** may extend beyond the boundaries of the **BleedBox**.

In the case where the **CropBox** is present, neither the **ArtBox** nor the **TrimBox** may extend beyond the boundaries of the **CropBox**.

The bounding boxes may be used by PDF/X-compliant pagination applications to automatically position the file in a predetermined space within a layout construct.

Within some industry workflows, both the **BleedBox** and **TrimBox** are necessary. For example, commercial, non-newspaper printing may include large numbers of pages containing bleed and trim information. It is important that boxes that represent this information be included. The accurate inclusion of the **BleedBox** and **TrimBox** will allow for the correct portion of the file to be imposed and rendered, and appropriate automation to be applied.

NOTE The use of TrimBox is recommended in preference to ArtBox.

2.11 Security and encryption of entire PDF/X files using external tools

None of the PDF/X standards covered by these application notes permit the use of PDF-based encryption. The only ISO PDF/X standard to have done so is PDF/X-1:2001, now deprecated.

If there is a need to apply encryption to protect confidential data during exchange, an external file encryption application must be used. The use of encryption in file exchange requires prior understanding between sender and receiver. Once an entire PDF/X file has been encrypted, the resulting encrypted file is no longer a conforming PDF/X file, and it should be assumed that a compliant PDF/X reader cannot read an encrypted file without prior decryption.

2.12 Application of screening

When discussing the delivery of digital material for printing, a common concern is the fidelity of the screening. Specifically, the discussion focuses on the use of screen angles, dot shapes, and screening algorithms. In a digital workflow, the receiver performs the process of screening in an effort to optimize reproduction. On occasion, the originator seeks to control the screening intent.

When the originator seeks to ensure that the receiver of the PDF/X file also uses the same screening used in the proofing of the PDF/X file, the question of how to facilitate such intent (without prior communication) is raised. The issue of screening control varies from market to market and is often dependent on the subject matter. If a supplied contract proof includes halftone screening that the customer requires, those screen parameters need to be passed along to production personnel.

Screen angles may be altered from normal specifications, in production or by the client, in order to achieve a desired result. Occasionally, the printer may need to adjust angles of various elements on a form prior to output.

While the method by which this result is best achieved is often debated, the fact remains that the PDF/X standard provides for the embedding of screening information into a PDF/X file. The PDF extended graphics state allows for specification of screening information. Although the use of the extended graphics state is optional, this is the mechanism that should be used to communicate screening information within a PDF/X file.

The ICC profile specification also has an optional screen tag, but this should not be used.

All PDF/X files should be checked for screening information as part of the preflight process and it is recommended that PDF/X readers facilitate this feature.

In a blind exchange, it may not be possible for the originator to achieve absolute control of screening. On those occasions when screen control is critical, document originators may achieve the desired results by transferring pre-screened 1-bit data prepared at the required resolution and for the desired printing conditions. This exchange requires prior discussion among the involved parties.

Specifying stochastic screening can be problematic in a blind PDF/X exchange and may require additional communication among the involved parties.

The following are guidelines for the appropriate use of screening in PDF/X workflows:

- All digital file deliveries are governed by business agreements that specify performance criteria. If screen angles are important, the client should specify that the receiver should or must comply with any screening information supplied as part of the business agreement. This, of course, must be tempered with the knowledge of the capabilities of the prepress provider, the publisher, and the printer. Many electronic workflows cannot image multiple frequencies, dot shapes, or screen angles on the same page or printed form. If such an agreement exists, the originator must specify screening information in a job ticket or accompanying business communication. Screening information may also be included inside data files as production reliability for this methodology becomes established.

- At any stage of the process, if it is not possible or feasible to process the screening as specified in the business agreement, it is the responsibility of the recipient of the file to report this to their business partners.

- If any of the components containing the screening information are copydot (or bilevel) information, care must be taken that the appropriate resolution is used when creating these files. More information on this subject is given within these application notes.

- The presence or absence of screening information has no bearing on whether the file is PDF/X compliant.

2.13 Copydot information

Copydot information is generally understood to be the file set generated from the digitization of supplied film separations so that it can be used in an all-digital workflow.

Copydot files are typically captured at high resolutions (>1000 dpi); are binary; and are not readily editable. A typical copydot file may consist of a set of five files; an independent master file [used for placement in a layout or imposition application] plus one each of the cyan, magenta, yellow, and black separations represented as bilevel data.

Regardless of the method or equipment used, caution must be exercised when creating [copydot] bilevel data. An industry “best practice” is to exactly match the resolution of the bilevel data to that of the intended imaging device. Other methods have been developed whereby data that is not at the exact resolution of the target imaging device can be imaged acceptably (e.g., creating the file set at exactly half the intended resolution).

Copydot files must be prepared for the referenced printing condition specified for the entire PDF/X file.

2.14 CT/LW Raster PDF/X

PDF/X as a file format for exchange draws its roots from the work in TIFF/IT, a standard exchange format for encoding imagery and linework from high-end color electronic prepress systems (CEPS), e.g. those made by Scitex, Hell, Eikonix, Dainippon Screen and Crosfield. TIFF/IT provides a way of encoding raster imagery for CT (continuous tone), commonly known as images, and LW, or linework, which corresponds to graphical elements other than images (text, lines, gradients). Linework is usually rendered at a higher resolution than imagery, most often at the expected resolution of the output device to provide the sharpest possible edges on graphic elements.

With the introduction of PDF/X, some early adopters chose to attempt to transform TIFF/IT into PDF/X. While the conversion of the CT or imagery components is straightforward, conversion of LW into appropriate PDF data structures is problematic. The definition of TIFF/IT LW combines an indexed-color table with pixel-level transparency, which does not have a corresponding data structure within PDF, even though both the **imageMask** object (which provides pixel-level transparency), and indexed-color images are allowed in PDF. In the absence of an indexed-color **ImageMask** object, developers are forced to choose alternative representations of LW data in PDF, any of which will be obvious to developers familiar with the PDF specification and the offset printing process. Several implementations are now operational in the market, but each has its own limitations on file size, viewing speed, or rendering speed on various RIP platforms. Ironically, while raster data is considered to be a simpler format and hence least likely to be printed differently than it appeared on proof, the complexity of conversion to PDF reduces this benefit of LW as various RIPs and PDF viewers have been known to interpret this complex data differently.

While encoding of LW data inside PDF is not prohibited by PDF/X, it is suggested that it be a transitional technology, and that developers aspire to capture linework at its earlier vector stages, rather than including resolution-dependent rasterized versions.

Note that these comments apply to data that was originally represented in CT/LW form – ‘pure’ raster PDF/X, where each page comprises a single image (either CT or screened, ‘copydot’ data) does not suffer so severely from the same problems in RIPs or PDF viewers even though they do not provide the flexibility in scaling that might be expected from a PDF/X exchange. Implementers are also cautioned to bear PDF reader image line length issues in mind when producing file creation tools.

2.15 DeviceN color space: Duotones and other uses

The **DeviceN** color space can be used to express duotone, multi-tone, and "colorized" images, and those with "bump" plates (separations intended to increase color saturation and increase the impact of an image). It can also be used for objects (especially graduated fills) constructed with multiple spot colors, a combination of spot and process colors, or a subset of the CMYK process colors. **DeviceN** color spaces using a subset of the CMYK colors can be used when a nominally CMYK object should overprint other CMYK objects.

It is also possible to encode this type of data in a PDF/X file employing multiple **Separation** color spaces.

The **Separation** color space method is inefficient for it requires the drawing of each object multiple times (once for each separation) and overprinting them. The implementation of such an overprinting method is discouraged.

2.16 Color space and OutputIntent issues

The PDF/X standards require that all data in a file be prepared for a single target printing condition. The target printing condition must be identified using an **OutputIntent**, as described in *PDF Reference*.

With **OutputIntent**, a creator of a PDF/X file communicates the printing condition for which the file was prepared. This mechanism is important in many printing environments. It can be used to preflight the file to ensure that it matches the receiver’s printing conditions. It also facilitates accurate color rendition in both hard and soft proofing.

See section 0 for details specific to PDF/X-1a, and section 0 for details specific to PDF/X-3.

2.16.1 ‘Generic’ color

Wherever possible a PDF/X file should be created for the specific print condition that it will be printed under, but there are some situations where the originator of a file may not know exactly how their job will be printed. As is the case with traditional, film-based, workflows a widely used, typical printing condition is the best choice in such a situation. In North America the typical choice in such situations is SWOP printing, which is characterized in the ICC registry as CGATS TR 001, *Graphic technology — Color characterization data for Type 1 printing.* This can be used with either CMYK files, or the output profile for device independent files. It is approximately equivalent to paper type 3 of ISO 12647-2, *Graphic technology — Process control for the manufacture of half-tone colour separations, proof and production prints — Part 2: Offset processes*, which has been characterized by FOGRA, and is also included in the ICC registry.

2.16.2 Printing condition validation

It is expected that tools such as preflight applications will be shipped with a list of the characterizations included in the ICC registry at the time that the current version was created, and that PDF/X files will be validated against that list. It is recommended that vendors provide some mechanism to allow the list of characterizations to be updated as the ICC registry evolves. The ICC registry is not constructed to allow programmatic update of such tools. Updates to lists in PDF/X tools must, therefore, be mediated through either a vendor web site (for automatic updates) or through a user interface of some form.

2.16.3 Color management notes

If color management is to be applied to a PDF/X file during rendering, either for proofing or for final output, care should be taken to apply the same processing to images and vector objects unless different rendering intents have been explicitly selected for them. This will ensure that vector objects intended to match parts of an image will do so.

2.16.4 Human-readable descriptions of output intents

ISO 15930-1:2001 and ISO 15930-3:2002 recommended that the value of the **Info** key in an output intent dictionary be used to carry be a description of the characterized printing condition in a form that will be meaningful to a human operator at the site receiving the exchanged file. The 2003 revisions amended this recommendation to use the **OutputCondition** key instead.

Note that the *PDF Reference* states that the **Info** key is required if “**OutputConditionIdentifier** does not specify a standard production condition.” While it is not clear exactly what is meant in that document by a standard production condition, the requirement that a PDF/X file also be a valid PDF file means that it is safest to use the **Info** key as well as the **OutputCondition** key in output intent dictionaries.

2.16.5 Profile selection

If there is no **RegistryName** key in the **OutputIntent**, then the embedded profile must be used to perform color-managed rendering.

When there is a **RegistryName** key and a reader is performing color-managed rendering of device dependent data in the color space of the intended printing condition (color proofing, re-targeting, etc.) then either the embedded profile (if present), or a locally held profile may be used. Such transforms only involve use of colorimetric rendering intents which are a function of the characterization and not unique to individual profiles. The recipient of a file may know and trust a locally held profile to provide a more accurate prediction of the final printed piece than using embedded profiles received from a variety of customers for such applications. Conversely, using an embedded profile will show the designer’s expected color-comparison of prints made with the embedded and local profiles may highlight potential difficulties in matching expectations.

Because the profile used to render device independent data determines gamut and tone scale compression and UCR/GCR it is important that the same profile is used for output at all stages of the workflow to ensure an accurate match between proofs and the final printed piece. The embedded profile should therefore always be used to render objects in device independent color spaces, rather than using a locally held profile.

2.16.6 Hi-fi printing conditions

The PDF 1.4 specification restricts the color space of embedded profiles to Gray, RGB, CMYK and Lab. It is therefore not possible to construct an **OutputIntent** structure that includes an embedded ICC profile to define a hi-fi print condition using more than four process colors, such as Hexachrome® [1]. There are a number of methods that may be used to carry such data in a PDF/X-2 or PDF/X-3 file:

- A hi-fi printing condition that includes colorants similar to CMYK may be represented as a CMYK space plus spot colors for the additional colorants.

- All channels of the hi-fi color space may be encoded as separate spot colors.

- If a characterization for such a printing condition were to be included in the ICC registry, a PDF/X file could reference it without requiring a profile to be embedded. Device dependent data in the file could be encoded using DeviceN color spaces.

Note that neither of the first two of these options provide sufficient colorimetric information to guarantee accurate color when printing a CMYK proof or re-targeting a PDF/X file.

2.16.7 Example OutputIntents

The keys required in an **OutputIntent** object vary slightly depending on:

- whether all print elements in the file are defined in CMYK, Gray or spot colors, or whether any device independent color spaces are used

- whether the print characterization for which the file has been prepared is included in a registry and, if so, whether that's the ICC registry at http://www.color.org.

The examples in Table 2 show the recommended keys.

[1] Hexachrome is a registered trade mark of Pantone, Inc.

Table 2 — Examples using recommended keys

	CMYK-only	Including device independent color
Characterization in ICC registry	<< /Type /OutputIntent /S /GTS_PDFX /OutputConditionIdentifier (CGATS TR 001) /RegistryName (http://www.color.org) /OutputCondition (CGATS TR 001 (SWOP)) /Info (CGATS TR 001) >> or << /Type /OutputIntent /S /GTS_PDFX /OutputConditionIdentifier (CGATS TR 001) /RegistryName (http://www.color.org) /OutputCondition (CGATS TR 001 (SWOP)) /Info (CGATS TR 001) /DestOutputProfile 100 0 R >>	<< /Type /OutputIntent /S /GTS_PDFX /OutputConditionIdentifier (CGATS TR 001) /RegistryName (http://www.color.org) /OutputCondition (CGATS TR 001 (SWOP)) /Info (CGATS TR 001) /DestOutputProfile 100 0 R >>
Characterization in another registry	<< /Type /OutputIntent /S /GTS_PDFX /OutputConditionIdentifier (Print Group 1) /RegistryName (http://www.printgroup.org) /OutputCondition (Print Group 1 (Gloss coated)) /Info (Print Group 1) /DestOutputProfile 100 0 R >>	
Characterization not in a registry	<< /Type /OutputIntent /S /GTS_PDFX /OutputConditionIdentifier (Joe’s Print) /OutputCondition (Joe’s Print – 555 264 7200) /Info (Joe’s Print – 555 264 7200) /DestOutputProfile 100 0 R >>	

2.17 Use of the Trapped key

Within the context of PDF/X, the **Trapped** key is used to indicate the trap status of the entire PDF file. The trap status of an entire file, specifically each component contained in the file, must be known when preparing the PDF/X file. The **Trapped** key is required in a conforming PDF/X file.

File senders:

- If the file is trapped, set **Trapped** to *True*.
- If the file is not trapped, and you do not want the recipient to trap it or you know that it does not require trapping, set **Trapped** to *True*.
- If the file is not trapped and you want the recipient to determine whether it requires trapping, set **Trapped** to *False*.

File recipients:

- If **Trapped** is *True*, do not apply trapping.
- If **Trapped** is *False*, apply trapping at your discretion, or according to business agreements.

The *PDF Reference* allows a third state for the **Trapped** key. The value *Unknown* is used to specify that the trapping information in the file is not known or that some, but not all, components within the file may have been trapped. The value *Unknown* is not permitted in PDF/X files.

Note that the value of the **Trapped** key must be a name object – */True* or */False* – and not a boolean.

Some recipients' workflows built around PDF/X may not be capable of trapping copydot files embedded in a PDF/X file. It is therefore recommended that source material be appropriately trapped before the creation of copydot files.

2.18 Fonts and font encodings

Historically, there have been a number of font-related problems in PDF files. Many of these problems have been addressed in the PDF/X standard by placing restrictions on font usage and embedding.

The PDF/X standard reduces the probability of font-related problems by requiring the creator to ensure that all fonts used are embedded, and that only embedded fonts and metrics are used when reading, displaying, and printing a PDF/X file.

A PDF/X conforming writer must always embed fonts, and a conforming reader must always use the embedded fonts.

There are a number of potential font problems that cannot be addressed by the PDF/X standard, as they are rooted in the PDF/X creator's environment. These potential problems are as follows:

- Fonts may be substituted without warning. Unintended font substitution may occur if the fonts are not available to the creation application when it exports a PDF/X file, generates a PostScript file, or converts a PostScript file to PDF/X.
- Some combinations of Windows printer drivers and TrueType fonts may produce undesirable results if the file is to be used for purposes other than rendering for printed output. PostScript files containing TrueType fonts may lead to un-searchable and un-editable text in PDF files.
- Windows printer drivers, in combination with the selected PPD, allow the user to specify how TrueType fonts are to be embedded in the resulting PostScript file. Typically, TrueType fonts can be included as TrueType outlines as FontType 42, CIDFontType 2 (with FontType 42 wrappers), or they may be converted to FontType 1 outlines, or bitmaps in either FontType 3 or FontType 32 format. Regardless of

the conversion performed, the fonts can be embedded in the PDF/X. The highest quality output will be obtained by retaining the fonts as TrueType (Type 42), preserving their hinting information, and this approach is recommended. It is unlikely that text produced by bitmap fonts would print at high quality on a high-resolution device.

- When using a workflow that starts by generating a PostScript file and then converts it into PDF it is recommended that all fonts used in the document are embedded in the PostScript stream if possible, and that the conversion tool be set to use those embedded fonts. If the conversion tool embeds fonts in the PDF file that were not present in the PostScript there is a chance that the wrong font could be selected if more than one version of the font is available on the computer. This is especially important when using a server-based conversion tool where the document is printed on one machine and the PostScript converted to PDF on another. For the same reason, EPS graphics containing text should have all fonts used embedded in them where possible.

Creators are warned that not all fonts can be legally embedded and that legal restrictions on embedding should be determined. Vendors creating PDF/X preflight tools or utilities to convert PostScript into PDF/X are encouraged to include checks for these issues where possible.

The PDF/X specification neither requires nor prohibits the use of font subsets in a PDF/X file. Embedding font subsets is recommended to avoid potential font name conflicts because font subsetting forces a renaming of the font within the PDF file. However, subsetting may prevent later editing of PDF text elements if font glyphs not included in the subset are required.

Some older RIPs do not fully support the CID font specification, which can lead to problems with the rendering of PDF/X files that use CID fonts. Although CID font technology was developed for double-byte fonts, some applications produce PDF files that use CID fonts for single-byte fonts. A PDF/X receiver must be prepared to process CID fonts. CID fonts are fully supported in version 2015 and above of Adobe RIPs, and in version 5.3 and above of Global Graphics' Harlequin RIP. For information on CID support in RIPs from other vendors please contact the RIP vendor.

2.19 Development of receiver requirements

As specified in the standard, “a conforming reader is a software application that shall be able to read and appropriately process all conforming PDF/X files.” The PDF/X standard has been constructed to be useful to publishers and printers. The committee acknowledges that some receivers may wish to place additional restrictions on submitted PDF/X files beyond those in the specification.

Any additional restrictions should always act in a way that further limits the options that may be used by a supplier of PDF/X files. For example, a receiver may choose to prohibit JPEG-compressed images.

Companies producing PDF/X readers and preflight-checking software must write to the complete PDF/X standard. However, companies are encouraged to include options that enable (but do not require) receiver-identified restrictions to be applied under user control. An example of such an option would be the ability to produce a warning if a JPEG-compressed image is encountered in a file.

Publishers are encouraged to review existing restrictions in their preparation specifications as they start to accept PDF/X files; many of their existing limitations may be based on historical problems in writing or reading applications that may not be applicable to PDF/X compliant tools.

The following are examples of possible receiver-imposed restrictions:

— Files must only contain a single page.

— No JPEG compression is allowed.

— No TrueType, MultipleMaster, fonts can be used.

— All copydot images must be produced at a specified resolution.

— Files with bleed requirements must include **BleedBox**.

— A control strip containing specified color patches must be placed outside the **TrimBox** (or **ArtBox)**. If **BleedBox** is also present, then the control strip must be outside that box as well.

The receiver's specifications should also identify the standard characterized printing condition for which files are to be prepared, or, if non-standard conditions are to be used, provide appropriate characterization data. This identification should include a text description, but pre-prepared ICC profiles for the specified printing condition, suitable for file preparation and inclusion in PDF/X files directly, may also be provided by the receiver of the PDF/X file.

Some industry groups are establishing preferred workflows that dictate the use of specific characterization data or specific profiles.

Receivers are also encouraged to state whether they honor screening intents in the PDF/X file that differ from their "normal" processes.

2.20 Implementation limitations

A PDF/X conforming writer or reader may have implementation-dependent limitations, but the reader should allow for all real world PDF/X files to be processed.

The following recommendations are intended to assist application developers avoid problems seen in the past:

- Minimum page sizes should be small enough for classified advertising and label printing. As a guide, current Adobe Acrobat's minimum supported page size is 3 points by 3 points.

- Maximum page sizes should be large enough for large-format imagesetters and platesetters and for poster work. As a guide, current Adobe Acrobat's maximum supported page size is 200 inches by 200 inches.

- Some applications generate PostScript, which can produce different results at different resolutions. A tool to convert PostScript to PDF/X therefore must have a nominal resolution at least as high as that of current high resolution imaging devices. The maximum resolution should not limit the maximum page size supported.

- Any limitations on the range of real numbers or object counts that may be included in a PDF/X file should be sufficiently large that they are not encountered in production work.

- Very complex clipping paths should be supported.

- Some PDF readers cannot process an **Image XObject** whose **Width** value exceeds 30,000 pixels. PDF/X writers should avoid the creation of such objects. Larger images may be tiled to achieve the same visual effect.

For further discussion, see *PDF Reference*.

2.21 Page design application limits

The process of creating a PDF/X file often involves an intermediate step of generating a PostScript file. A separate step processes the PostScript file and writes a conforming PDF/X file. There are a number of constructs used by a designer in a page makeup or drawing application that may not be carried through as expected into composite-color PostScript generated by that application. In these cases, some PDF/X generation tools may optionally be able to infer the original design intent and construct a PDF/X file that matches the designer's expectation, but most will follow the PostScript code, as written, and some aspects of the design may be lost or altered.

Similar effects may occur if a design application exports directly to PDF.

The following are examples of such problems:

- A graduated tint in a spot color may be imaged on the process plates, even though text and rules to which the same spot color has been applied image on the spot color plates as expected.
- A grayscale image or 1-bit TIFF image that has been "colorized" in a page layout application using a spot color may be imaged as process separations.
- Traps created in some page layout applications may be lost.
- The flattening of transparency effects may lead to loss of spot color information, or unwanted rasterization effects.

Vendors creating utilities to convert PostScript into PDF/X are encouraged to include checks for these and similar issues where possible.

PDF/X increases confidence that a proof made from the PDF/X file before transmission will match the final printed piece, but designers should be aware that their design intent may not be accurately carried through the creation of the PDF/X file. Final pre-transmission proofs should always be made from the PDF/X file, and not directly from the design files or intermediate PostScript. Where spot colors have been used, elements in the PDF/X file should be checked to ensure that they have been assigned to the correct separations; e.g., by proofing separations.

2.22 Document identification and metadata

The document **ID** in the **Trailer** of a PDF/X file may be regarded as unique. Adding this document **ID** to a file reference in an electronic data interchange (EDI) environment is more likely to lead to correct identification of the intended file than using only the file name, which may not be unique, and will also distinguish between revisions of the same file. Inclusion of additional file identification fields, such as the document title and modification date may also be helpful.

Document creators are strongly encouraged to follow the recommendations of the *PDF Reference* to ensure that the **ID** in the **Trailer** is likely to be unique.

Additionally, the use of the PDF version 1.4 **Metadata** key is allowed. Note that although information placed using this mechanism may be beneficial to production processes, any reader that is not PDF version 1.4 compliant may ignore this information.

Tools used to edit PDF/X files should preserve embedded metadata, and should update the XAP:VersionID field as appropriate.

2.23 File types and extensions

Many operating systems use internal information associated with the file to ensure that the correct application is opened when a file is selected, for example, by double-clicking on its icon.

Operating systems that use a “file extension” as part of the naming convention can build a connection between the extension and the processing application.

PDF/X files are PDF files. Therefore, it is recommended that PDF/X files be named with the extension used by normal PDF files (.pdf) and that their Macintosh file type should be set to ‘PDF ’.

2.24 Synchronization of base and alternate image XObjects

Alternate **image XObjects** are permitted in PDF/X. When alternate images are desired, they should always be derived from the embedded base image. In the case where alternate image(s) exist, any alteration of the base image(s) requires the re-generation of, or the immediate removal of, the alternate image(s).

This synchronous relationship between the base image and its alternate image should always be preserved.

2.25 Use of transparency

The use of partial transparency, as specified in the *PDF Reference*, is prohibited in a PDF/X file. This decision was taken for pragmatic reasons. The various products available today that will render transparent objects for output, or flatten them for further processing, do not currently produce consistent results. The aim of the PDF/X standards is to provide maximum assurance that a proof will match the final printed piece, so such inconsistency is not acceptable. It is anticipated that future revisions of the PDF/X standards will be amended to allow the use of partial transparency once shipping products have matured.

Note that the visual effect of partially transparent graphics may be achieved using techniques other than the use of the PDF 1.4 transparency keys, including pre-rendered data or flattened vector objects. The use of such techniques is allowed in a PDF/X compliant file.

2.26 Embedded PostScript

The PDF/X standards prohibit the use of **PS XObjects** and of the **PS** operator.

Note that the PDF 1.2 specification includes a mechanism for constructing an equivalent to the **PS XObject** using a **Subtype2** key in a **form XObject**. This mechanism is not included in the PDF 1.3 or 1.4 specifications, but is included in the errata document referenced by the 2003 revisions of the standard. These old-style PS XObjects should not be used in PDF/X files.

2.27 Assembling PDF/X files into a single output

Multiple PDF/X files can be assembled easily into a single compound entity for output if they have all been prepared for the same output condition.

If they have not been prepared for the same output condition the receiver must ensure that the individual files can achieve acceptable quality when transformed to the actual output condition to be used. Developers of assembly tools are encouraged to highlight such situations to the user, and/or to provide suitable color transformation options.

For additional restrictions specific to PDF/X-3 files, see also 0.

Developers and integrators may wish to explore the use of PDF/X-2 for assembly of multiple page parts supplied in PDF/X format for onward transmission to another site.

2.28 PrinterMark annotations

PrinterMark annotations may be used to draw a variety of print-related marks on the output such as crop and register marks. A PDF/X conforming reader (including viewers) may choose whether to show or print these or not, depending on the specific requirements of a particular piece of output.

It may be appropriate to include marks on a proof, for instance, but to suppress them when the same piece is output as part of an imposed flat, or placed as one element of a larger page.

In the PDF/X standards PrinterMark annotations are regarded as print elements, and the color space requirements that apply to all print elements therefore apply to them.

The PDF/X standards state that most types of annotations must fall entirely outside the **BleedBox. PrinterMark** annotations may fall inside the **BleedBox**, but must still be entirely outside the **TrimBox** or **ArtBox**.

2.29 Private keys in the Info dictionary

To avoid unexpected application problems, it is strongly recommended that all PDF/X creators follow the requirement added in PDF 1.5 that all values for private keys in the document information dictionary of a PDF/X file be ‘text strings’. This precludes the use of numbers, names etc.

3 PDF/X-1a

The following sections describe attributes specific to ISO 15930-4:2003 (PDF/X-1a:2003).

ISO 15930-4 defines a data format and its usage to permit the predictable dissemination of a compound entity to one or more locations as CMYK data and/or spot color data in a form ready for final print reproduction, by transfer of a single file.

A file conforming to ISO 15930-4 (or ISO 15930-1) must contain all the content information necessary to process and render the document, as intended by the sender. This exchange requires no prior knowledge of the sending and receiving environments and is sometimes referred to as "blind" exchange. It is platform- and transport- independent.

3.1 PDF/X-1 change history

3.1.1 Differences between PDF/X-1:1999 and PDF/X-1:2001

PDF/X-1:1999 (CGATS.12/1) was based on the *Portable Document Format Reference Manual, Version 1.2* as extended by *Technical Note #5188.* Note that this standard has now been withdrawn in deference to ISO 15930-1.

PDF/X-1:2001 was based on the *Portable Document Format Reference Manual, Version 1.3 Second Edition* as extended by *Technical Note #5413*. PDF/X-1:1999 (CGATS.12/1) was exclusively an ANSI (American National Standards Institute) standard; PDF/X-1:2001 (ISO 15930-1) is an ISO (International Organization for Standardization) standard.

The *Portable Document Format Reference Manual, Version 1.3,* introduced features such as the **DeviceN** color space, smooth shading, masked images, CID Type 2 fonts, improved overprinting (**OPM** key).

PDF/X-1:1999 required an **ICCBased**, **DefaultGray** and/or **DefaultCMYK** color space to identify the characterized printing condition for which the file was prepared. PDF/X-1:2001 used an **OutputIntent** instead, which can contain either an ICC registered print condition name or an ICC profile. An **ICCBased** color space must not be used for printing elements in PDF/X-1:2001 and PDF/X-1a:2001 files.

PDF/X-1:2001 had two distinct conformance levels (see 0), whereas PDF/X-1:1999 had only one.

PDF/X-1:2001 annotations must be located outside the **TrimBox** (or **ArtBox**) and **BleedBox**. In PDF/X-1:1999, the location of annotations was unrestricted, provided they were non-printing.

3.1.2 Differences between PDF/X-1:2001 and PDF/X-1a:2001 conformance levels

PDF/X-1:2001 supported OPI provided the high-resolution data was embedded in the PDF/X-1:2001 file. Conversely, in PDF/X-1a:2001, the presence of proxy images and OPI comments was strictly prohibited.

PDF/X-1:2001 allowed encryption, provided the encryption did not prevent the viewing or printing of the PDF/X-1 file. Conversely, PDF/X-1a:2001 could not contain an **Encrypt** dictionary.

3.1.3 Differences between PDF/X-1a:2001 and PDF/X-1a:2003 conformance levels

ISO 15930-4:2003 dropped the PDF/X-1 conformance level, leaving only PDF/X-1a. There is no longer support for embedded OPI references or for encryption in this part of the standard.

PDF/X-1a:2003 is based on PDF version 1.4, whereas PDF/X-1a:2001 was based on PDF 1.3. Note that most of the major new features of PDF 1.4 are explicitly prohibited in PDF/X-1a, including transparency, JBIG2 compression, reference XObjects and 128-bit encryption.

3.2 Advantages of PDF structures vs. OPI references

The (now deprecated) PDF/X-1:2001 conformance level included a rather specialized form of OPI. That is prohibited in all current PDF/X conformance levels.

Embedding OPI referenced files placed a significant downstream burden on the PDF/X-1-compliant reader to be able to process all of the file types shown in Table 1 of the PDF/X-1:2001 standard.

A preferred alternative is for the PDF/X file creator is to actually perform the conversion to native PDF constructs. The resulting file can be viewed in its entirety by PDF applications such as Adobe Acrobat without additional plug-ins, OPI servers, or RIPs.

This alternative relies on the creating application's ability to correctly represent the data from the legacy file formats in PDF constructs. The **ImageXObject** is one mechanism that can be used to correctly represent DCS and TIFF/IT-P1 CT data within PDF. PDF/X-compliant compression can be applied to an **ImageXObject,** making the file potentially smaller than the version using embedded file streams. Developers are cautioned that correctly and efficiently representing TIFF/IT-P1 LW data within a PDF/X file may be challenging.

3.3 Intended output condition

The characterized printing condition for which the file was prepared must be identified in the **OutputIntents** structure. In PDF/X-1a this is restricted to CMYK printing conditions (spot colors are not included in the characterization).

It is important to note that, for printing conditions included in the ICC registry, this information may be conveyed by a pointer to the entry in that registry (**OutputConditionIdentifier**) and a pointer to the ICC registry (**RegistryName**). For all other printing conditions a profile containing an AtoB1 tag (device to PCS) is required as the value of the **DestOutputProfile** key. The ICC registry may be found at http://www.color.org/.

PDF/X-1a does not include a formal method to support monochrome-only output printing conditions. Where such printing is needed PDF/X-3 may be used. However, where data space is not a critical issue, a four-color characterization may be developed where only the K channel contains device code values. The resulting profiles will be monochrome only, but compatible with PDF/X-1a restrictions.

4 PDF/X-2

The following sections describe attributes specific to ISO 15930-5:2003 - PDF/X-2.

The basic premise of PDF/X-1a and PDF/X-3 is that blind exchange is enabled (but not required). PDF/X-2, on the other hand, is based on the assumption that communication must occur between the sender and receiver of a transmission. The data carried with a PDF/X-2 file has been designed to allow as robust an automatic resolution of issues as is practicable, but some discussion of the capabilities of the receiving system will often be required before a suitable PDF/X-2 file may be constructed.

4.1 PDF/X-2 use cases

There are a number of situations where it is envisaged that PDF/X-2 may prove useful. The only common theme between these is the use of a single ‘master’ file referring to others that will be rendered in the final output; the business reasons that provide the value in that separation vary from case to case. For the purposes of this section such a workflow will be described as ‘OPI’, even though PDF/X-2 does not actually use OPI structures in PDF.

Note that there are many occasions where an OPI workflow can provide value as part of a process that leads up to the creation of PDF/X-1a or PDF/X-3 files, or to output to film or plate (e.g. to increase the response speed of design workstations). These do not automatically lead to a requirement for PDF/X-2. If an OPI workflow is resolved entirely within a single company (or a division within a larger company) then PDF/X-2 is not necessary.

PDF/X-2 adds value where a set of several files should be exchanged between companies or divisions. It can also add value where the company running a purely internal OPI workflow has little control over the names of files used in that workflow, and where the ability to resolve conclusively between files from different sources but with the same name can help to avoid the use of the wrong image.

Some examples of the situations where PDF/X-2 may be used are described below.

4.1.1 Print provider also providing scanning service

A print provider may wish to become involved early in the production of a printed piece, and may also offer scanning and retouching services. Such a provider may scan supplied images and return a low resolution ‘proxy’ image file – often called a FPO (For Position Only) file. The designer or origination company would then lay out pages using the proxy images, and submit the job as a PDF/X-2 file back to the same print service provider. The images would typically be sharpened, color corrected, retouched as necessary and separated at the service provider in parallel with the page design at the client.

In this scenario PDF/X-2 allows the same increased reliability of the design file submitted by the client that the PDF/X-1a and PDF/X-3 standards provide in single-file exchanges, while also increasing the robustness of the selection of the correct image file during final prepress processing.

The high resolution scan file selected would not normally be the same version of that file as was used to create the original proxy file, but would be a derivation of that file because of the parallel image processing and design work.

Note that it is necessary for all files in the image retouching workflow, from the initial file (often an almost raw scan) used to create the proxy image, to the final version used for image replacement, to carry sufficient metadata to allow the identification of that final version as being derived from the initial one. This may either be embedded within the image file itself or in a Digital Asset Management (DAM) system. It is not necessary that the metadata be carried as XMP at all stages in the workflow; only the files that will be selected when processing the PDF/X-2 file received from the client need the metadata in that form. The distinction between the document ID, and an ID for a version of that document must, however, always be retained.

4.1.2 Lightweight, but final form, proof file

There may be cases where a print service provider or prepress company wishes to supply a file for soft proof or approval to their clients. A graphic arts file including all of the high-resolution image data is often too large for efficient electronic transmission, and it is therefore common to use files that include only low-resolution data for this purpose.

If PDF/X-2 is used for such a proofing file, and the same file is submitted to the final prepress workflow, the possibility of any differences introduced by duplication of the creation process would be avoided. The proofing file would usually be created using low-resolution preview images taken from the proxy files, intended as placeholders only.

Sometimes the final image sharpening, color correction, retouching and separation may be performed while the client is approving the job, in which cases the high resolution scan file selected when generating plates would not be the same version of that file as was used to create the original proxy file, but would be a derivation of that file. If all image correction has been performed prior to creation of the proofing file, then the final image replacement would be performed using exactly the same file from which the proxy was created.

4.1.3 Aggregation of partial page submissions

A magazine publisher needs to aggregate a large number of submissions, from a variety of sources, into a single job file. These might include full- and partial-page ads plus editorial data. Most publishers are understandably reluctant to make any more changes than is absolutely necessary to an incoming ad file. Opening a file for editing means that the publisher must take potential responsibility for any errors that might be introduced.

A proxy file derived from each ad file may be placed in a layout with editorial data to create an aggregated job. A PDF/X-2 file generated from the aggregate may be submitted to a prepress company or press site alongside the original ad files.

The use of PDF/X-2 provides for a more reliable workflow than a simple OPI image replacement because of the additional checks that the correct ad file has been identified for each position. The final image replacement would be performed using exactly the same file from which the proxy was created.

4.2 OutputIntent issues

If a PDF/X-2 file has been prepared for a characterization that is included in the ICC registry, and does not contain any device independent color spaces, or device color spaces that don’t match that characterization, then it is not necessary to include an ICC profile in the PDF/X output intent. This is true even if such profiles are required in other PDF/X-2 or PDF/X-3 documents that it references. A profile may still be embedded, even if not required.

Where multiple, different, profiles exist in referenced files it may be necessary to process either the three component data to use a single output profile or convert all data to the intended output space prior to assembly to ensure that the correct gamut and tone compression is performed for all elements.

Preview images in proxies for externally referenced target documents are defined in the standard as being non-print elements. They may, therefore, make use of any color space (e.g. **DeviceRGB**) without requiring that a profile be included in the PDF/X-2 file PDF/X **OutputIntent** object.

4.3 Trapping in PDF/X-2

When a PDF/X-2 file set is received, some of the files may have been pre-trapped (**Trapped** = *True*), while others have not (**Trapped** = *False*). The state of the **Trapped** flag in the PDF/X-2 file only reflects whether traps have been added between elements within the PDF/X-2 file itself. It does not show whether traps between the PDF/X-2 file and its target documents, or between those target documents themselves, have been applied.

It is not possible to provide a complete and definitive guide as to how those traps should be applied. Many current prepress systems cannot guarantee that additional traps will not be created if a pre-trapped element is re-trapped. For a variety of technical and business reasons it can also be difficult to remove traps from a pre-trapped file.

The suggested workflows below may be applied in the simplest cases. Use caution when defining process steps in more complex situations.

Where there is no overlap between elements in different files:

1) Trap all target files that are not marked as already trapped.
2) Trap between objects within the PDF/X-2 (if not already trapped).
3) Assemble and output

Where there is overlap between elements in different files:

1) Supply all files as untrapped
2) Assemble all files together
3) Trap all elements to each other.

4.4 File name encoding

Unless there is a good business or technical reason to the contrary, it is recommended that file names be restricted as described in section 3.10.1 of the *PDF Reference*:

> “When creating documents that are to be viewed on multiple platforms, care must be taken to ensure file name compatibility. Only a subset of the U.S. ASCII character set should be used in file specifications: the uppercase alphabetic characters (A–Z), the numeric characters (0–9), and the underscore (_). The period (.) has special meaning in DOS and Windows file names, and as the first character in a Mac OS pathname. In file specifications, the period should be used only to separate a base file name from a file extension.”

4.5 ID keys

There are two places within a reference **XObject** structure where the **ID** of the target document may be defined: within the reference dictionary itself, and in a file specification dictionary that forms the value of the **F** key in the reference dictionary. The PDF/X-2 standard states that the first must be present. Application vendors are encouraged to ensure that an **ID** key is not also present within the file specification, or that the values of the two **ID** keys are identical.

Note that the value of the **ID** key in the reference dictionary is not required to be a direct object. Indeed, each string within the array that forms the file identifier may itself be an indirect object.

4.6 Proofing and verification

Although the PDF/X-2 standard says that a compliant rendering of a PDF/X-2 file requires that all external referenced files be present, there are situations where proofing and/or verification of PDF/X-2 files without all elements present are useful.

Any proof prints made from a PDF/X-2 file in the absence of any or all referenced files should still be rendered in accordance with the PDF/X-2 standard in all ways other than those regarding referenced files.

Vendors of proofing and validation equipment are encouraged to provide options to handle PDF/X-2 files with and without all high-resolution files being available.

4.7 Selection of the correct high resolution file

It is expected that a PDF/X-2 reader will select the target document to be used in replacing each reference **XObject** using a two phase algorithm. In the first phase, one or more candidate target documents will be found, probably by use of the file spec in the reference dictionary. In the second phase the candidate targets found in the first phase will be validated against the values of the **Metadata** and **ID** keys in the reference dictionary. Note that it is not mandatory to use two phases. Any algorithm that results in the selection of the correct target document may be used.

4.7.1 Discovery of candidate target files

The PDF/X-2 standard does not specify how a PDFD/X-2 reader should find candidate target documents. It is likely that most readers will use the same kind of techniques that are used by current OPI servers. These fall into two main categories:

a) The whole of the file spec, including a full path, is used. In some systems a mapping table allows all or part of the path to be replaced in a predetermined way, possibly to account for different paths to a server from applications running on different platforms. This method is most widely used where the proxy is created, and then the target resolved within a single site, with access to the same file server.

b) Any disk volume name and path is stripped from the supplied file spec, leaving just the file name and extension. A predetermined set of “search directories” is then scanned for files that match the required name. This method is can be appropriate when proxy generation and resolution of references are performed on different sites using different image servers.

Both of these approaches may be appropriate for use with PDF/X-2, depending on the rest of the workflow, and on where proxies are made and links resolved (see 0 – PDF/X-2 use cases).

File references in PDF files may be constructed in a variety of ways. Absolute, relative and URL-based references may be used (although it would be inappropriate to use a URL that resolves to a dictionary rather than to a single file). If necessary, relative references may be converted to absolute ones using the methods set out in section 3.10.1 of the PDF reference. URL-based references should not be used in PDF/X-2 exchanges without prior consultation between the parties.

4.7.2 Examination of candidate target documents

Once one or more candidate target documents have been identified, the PDF/X-2 reader must determine which, if any, is an appropriate file to use in replacing the reference **XObject** proxy. This is done by comparing a number of metadata items between the reference dictionary and each candidate target:

a) The first string in the **ID** in the reference dictionary against the first string in the **ID** in the trailer of the candidate target.

b) The second string in the **ID** in the reference dictionary against the second string in the **ID** in the trailer of the candidate target.

c) The **xapMM:DocumentID** property within the **xapMM:RenditionOf** element within the reference XObject Form dictionary against the **xapMM:DocumentID** property within the **Metadata** stream in the **Catalog** of the candidate target document.

d) The **xapMM:VersionID** property within the **xapMM:RenditionOf** element within the reference XObject Form dictionary against the **xapMM:VersionID** property within the **Metadata** stream in the **Catalog** of the candidate target document.

e) The **xapMM:RenditionClass** property within the **xapMM:RenditionOf** element within the reference XObject Form dictionary against the **xapMM: RenditionClass** property within the **Metadata** stream in the **Catalog** of the candidate target document.

If all five comparisons provide exact matches then the candidate target document may be regarded as the original file from which the proxy file was created.

If either or both of a) or c) provides a match, then the candidate may be regarded as a different version of the same document from which the proxy file was created.

Some workflows require an exact match to the original file, while others require a later version of the same file. See 4.1 (PDF/X-2 use cases).

The PDF/X-2 standard does not specify what action should be taken if more than one appropriate target document is identified; that decision is left to implementers. Some may choose to request input from an operator, whilst others may select the target with the latest modification date, etc.

If neither a) nor c) provides a match, the candidate target document should not be used to replace the reference XObject. If no target document can be found then a PDF/X-2 compliant rendering cannot be produced. In some cases processing should continue, whilst in others (especially final plate creation) the job should be cancelled pending operator intervention.

The XMP-based **Metadata** can provide a cleaner interface than **ID** strings, and allows for the carriage of the same data in multiple file formats, which may be important in some workflows. At present, however, a number of applications do not maintain or update XMP metadata in PDF files. Other applications do not ensure that the first string of an **ID** remains constant over successive edits to a PDF file. The combination of both **ID** and **Metadata** is therefore used in PDF/X-2 to maximize the likelihood of being able to track the relationship between the original file from which the proxy was created and a candidate target file.

4.8 Assembly of color-managed files

See 0.

4.9 Multiple page referenced documents

The PDF/X-2 standard does not place any restrictions on the documents referenced, other than that they should be PDF/X-1a, PDF/X-2 or PDF/X-3 and that they must include the appropriate metadata structures. It is therefore possible to refer to a page of a reference document that contains multiple pages.

The reference to a specific page will commonly be by page number. If a page is inserted into, or deleted from the referenced document the reference may therefore point to the wrong page. It is therefore recommended that only a workflow where exactly the same version of the referenced document is used during image replacement as was used to construct the metadata in the PDF/X-2 file in the first place be used in such cases.

4.10 Rendering of referenced files

All of PDF/X-2:2003, PDF/X-1a:2003 and PDF/X-3:2003 are based on PDF 1.4, and should be rendered according to that specification. A PDF/X-2 file may also reference PDF/X-1a:2001 and PDF/X-3:2002 files, which are based on PDF 1.3 and should therefore, at least theoretically, be rendered as such.

The only differences in the output of these older versions of PDF/X if they are rendered according to the PDF 1.4 specification will occur if the files include constructs introduced in PDF 1.4, such as transparency - when rendered as PDF 1.3 the transparency would not be applied.

The recommended workflow for PDF/X-2 is to preflight any referenced PDF/X-1a:2001 and PDF/X-3:2002 files to ensure that they do not include constructs introduced in PDF 1.4, and then to apply the same PDF 1.4 rendering conditions as are used for the rest of the PDF/X-2 data.

4.11 PDF/X-2 reader issue

A PDF/X-2 file may reference another PDF/X-2 file, and the standard states that a PDF/X-2 reader must also accept PDF/X-1a and PDF/X-3 files directly, as well as by reference from a PDF/X-2 file. If files are being submitted to a PDF/X-2 reader via a hot folder, care must be taken in selecting the files to place in that hot folder; if all files in the set are copied into the hot folder it may not be obvious to the reader which files should be opened and rendered.

The recommended approach is either to place only the ‘master’ PDF/X-2 file into the hot folder, and have the reader search for the referenced documents in other locations, or to place a job ticket or control file into the hot folder instead of any content data file.

4.12 Simplifying and resolving PDF/X-2

It is the sender’s responsibility to ensure that all necessary files will be available at the receiving site. In many cases this may best be achieved by simplifying a transfer by reducing the level of nesting (where a PDF/X-2 file references another PDF/X-2 file) used to a minimum, and possibly even resolving PDF/X-2 file sets to either PDF/X-1a or PDF/X-3.

5 PDF/X-3

The following sections describe attributes specific to ISO 15930-6:2003 (PDF/X-3).

A file conforming to ISO 15930-6 (or ISO 15930-3) must contain all the content information necessary to process and render the document, as intended by the sender. This exchange requires no prior knowledge of the sending and receiving environments and is sometimes referred to as “blind” exchange. It is platform and transport independent.

5.1 Intended output condition

The characterized printing condition for which the file was prepared must be identified in the **OutputIntents** structure. In PDF/X-3 this may be for monochrome, CMYK or RGB printing conditions (spot colors are not included in the characterization).

In PDF/X-3 any device independent color may be used alongside spot colors and device dependent process colors that match the process colors of the intended printing condition. This means that if the **OutputIntent** reflects a CMYK printing process, **DeviceGray** and **DeviceCMYK** colors may be used, whereas **DeviceRGB** colors are not allowed. Any **DeviceRGB** data must be mapped via a **DefaultRGB** color space, probably defined as **CalRGB** or **ICCBased**.

If the **OutputIntent** reflects an RGB rendering process, **DeviceRGB** colors may be used, but **DeviceCMYK** and **DeviceGray** colors may not. Any **DeviceCMYK** or **DeviceGray** data must be mapped via a **DefaultCMYK** or **DefaultGray** color space respectively, probably defined as **ICCBased**.

These rules apply to alternate and base color spaces used in **Separation**, **DeviceN**, **Indexed** and **Pattern** color spaces.

If only device dependent process colors and/or spot colors are used in a PDF/X-3 file, the same rules apply for the **OutputIntent** as with PDF/X-1a files; for printing conditions included in the ICC registry, this information may be conveyed by a pointer to the entry in that registry (**OutputConditionIdentifier**) and a pointer to the ICC registry (**RegistryName**). For printing conditions not included in the ICC registry, an output profile must be included as the value of the **DestOutputProfile** key.

If device independent colors are used in a PDF/X-3 file, even if only for the alternate color spaces, it is mandatory to embed an ICC output profile as the value of the **DestOutputProfile** key; a pointer to a registered printing condition is not sufficient.

If the **OutputIntent** indicates an RGB or CMYK rendering condition, the **AtoB1Tag** of the output profile may be used to convert device dependent values to device independent ones for soft and hard proofing. If a monochrome rendering condition is indicated, the **GrayTRCTag** may be used in the same way.

The ICC registry can be found at http://www.color.org/.

5.2 Assembling PDF/X-3 files into a single output

In PDF/X-2 (6.4.4) it states that "If the containing PDF/X-2 documents and/or one or more target PDF/X-2 or PDF/X-3 documents contain objects in device independent colour spaces, and if the profiles embedded in the **OutputIntents** in those files are not identical, then the colours in those files need to be transformed into a common data space as part of any assembly process to ensure that the correct gamut & tone compression is performed for each entity." Most commonly, this will be the color space of the intended output device. Where all profiles are identical, the files may be assembled directly, retaining device independent colors.

This same situation also applies when assembling multiple PDF/X-3 files for common output.

This is important because the profile is used to convert data from device independent color spaces to the color space of the intended output device. In addition to converting from one color space to another, this transformation also normally includes gamut and tone scale compression as well as color separation and black printer generation. Two profiles can be based on the same characterization data (same characterized printing condition) but contain significantly different tone compression (high key, low key, etc.), black generation algorithms, etc.

Those attributes were used to gain customer approval of the rendered data and therefore need to be preserved by using the profile associated with the data by the data provider. If profiles associated with different files are identical, then only one needs to be used for data transformation.

In many cases it may be determined whether two profiles are the same by performing a simple byte-for-byte comparison, but this may show them as different because of changes that have no technical significance (e.g. a change of copyright data), or because tags not used by a specific job have been omitted to reduce file size.

The two key elements that that allow a user to determine if profiles are identical are the **Profile ID** field**,** Bytes 84 to 99 of the profile header, and the **profileDescriptionTag.**

The **Profile ID field** is optional. If used, it records a checksum value generated using the MD5 fingerprinting method as defined in RFC 1321. The **profileDescriptionTag** is a required Tag in all profiles and containing invariant and localizable versions of the profile description for display.

If the **Profile ID** field is present, and the same, in multiple profiles, it provides a high degree of assurance that the profiles are the same. While not as reliable, an exact match of the contents of the **profileDescriptionTag** between profiles is a good assurance that they are the same.

Note that all of these approaches will often show two profiles as different, even though they contain identical color transforms.

See also 0.

5.3 Trapping PDF/X-3 files

The successful electronic trapping of device independent PDF/X-3 files is conditional on a number of factors.

The trapping applications available to the consumer today require the incoming data to be composite, complete, and expressed in the color space of the output device (**CMYK**, **Separation**, **DeviceN**).

Recognizing this, the only PDF/X-3 file that could be trapped accurately by either a creator or receiver is one that is devoid of any element or image that is expressed using a device-independent color space (and such files would be better represented in PDF/X-1a).

Similarly, a PDF/X-3 file containing one or more elements or images expressed in a device-independent color space would require conversion to color spaces that are readily understood by today's trapping tools, before trapping could occur.

In order to fully realize the intent and promise of a PDF/X-3 exchange, either the receiver would have to perform all color transformations prior to trapping, or all-new trapping applications would need to be developed to trap device independent color space objects and images.

The possibility of a creator producing a completely trapped, device independent color space-based, PDF/X-3 file is improbable.

The combined realities of the marketplace and manufacturing environments shared by creators and receivers alike imply that the decision to trap or not to trap device-independent color space-based PDF/X-3 files will be handled on the receiver side of a typical PDF/X-3 exchange.

Therefore, it is recommended that PDF/X-3 files containing device-independent color spaces be supplied untrapped. If the sender is certain that no trapping is required, the **Trapped** flag may be set to *True*; otherwise the **Trapped** flag should be set to *False*.