Application Notes for PDF/X Standards

Version 3 — September 2002

Prepared by:

CGATS

The Committee for Graphic Arts Technologies Standards

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. This 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 the same level 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 are expected to establish specifications for these parameters in support of the use of the PDF/X standards.

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 15929, Graphic technology — Prepress digital data exchange — Guidelines and principles for development of PDF/X standards
- ISO 15930-1, Graphic technology — Prepress digital data exchange — Use of PDF — Part 1: Complete exchange using CMYK data (PDF/X-1 and PDF/X-1a)
- ISO 15930-2, Graphic technology — Prepress digital data exchange — Use of PDF — Part 2: Partial exchange (PDF/X-2)
- ISO 15930-3, Graphic technology — Prepress digital data exchange — Use of PDF — Part 3: Complete exchange suitable for colour managed workflows (PDF/X-3)

ISO 15930-1 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.

ISO 15930-2 defines a data format and its usage to permit the predictable dissemination of a compound entity to one or more locations as CMYK data, spot color data or data using device independent color spaces, as well as any combination thereof, in a form ready for final print reproduction, with some elements provided separately from the PDF/X-2 file.

ISO 15930-3 defines a data format and its usage to permit the predictable dissemination of a compound entity to one or more locations as CMYK data, spot color data or data using device independent color spaces, as well as any combination thereof, in a form ready for final print reproduction, by transfer of a single file.

A file conforming to ISO 15930-1 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.

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

These application notes currently only include information applicable to:

- ISO 15930-1:2001 (Published)
  - Defines the PDF/X-1:2001 and PDF/X-1a:2001 conformance levels
- ISO 15930-3:2002 (Published)
  - 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.

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.

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

2 General

2.1 Document structure

Unless otherwise noted, the following material applies to all conformance levels addressed by this document, which are simply referenced as PDF/X. These Application Notes will address all parts of the ISO 15930 standard, including present standards and future additions and revisions to those standards.

2.2 Conflict with standards

If there is a conflict between these Application Notes and the PDF/X standard, 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. For conformance with PDF/X-1:2001, PDF/X-1a:2001 and PDF/X-3:2002, the version number should be 1.3. However, since any file conforming to an earlier version of PDF also conforms to version 1.3, an application that processes PDF/X files must also accept files with any of the following headers:

%PDF1.0
%PDF1.1
%PDF1.2

All versions of all parts of the ISO 15930 define the PDF version upon which they are based.

A conforming PDF/X-1:2001 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-1:2001*);
c) may have the GTS_PDFXConformance key in the Info dictionary with a value of (*PDF/X-1:2001*).

A conforming PDF/X-1a:2001 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:2001*);
c) having the GTS_PDFXConformance key in the Info dictionary with a value of (*PDF/X-1a:2001*)

A conforming PDF/X-3:2002 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-3:2002)

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 standard states 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 *Portable Document Format Reference Manual, Version 1.3, Second Edition*, as extended by Adobe Technical Note #5413, might affect the final print reproduction. Therefore, any such added 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 1.3, but which are not included in version 1.3 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.

Tools used to edit PDF/X files should preserve embedded metadata.

2.4 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.

2.5 Output considerations for PDF/X

2.5.1 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 **SmoothShading**, 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.

2.5.2 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 1 should now be regarded as obsolete and upgraded – it is extremely unlikely that they can be configured to render files reliably in conformance with PDF/X. It is also unlikely that applications and tools that can process only PostScript Level 2 can be configured to be conforming and upgrades should be seriously considered.

For example, if the PDF/X file contains smooth shading and it is output as a PostScript Level 2 EPS file, the **SmoothShading** 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 Manual and Adobe Technical Note # 5044.

In order to confirm that the PDF/X files created will be processed correctly, DDAP (Digital Distribution of Advertising for Publications) and NAA (Newspaper Association of America) have developed a suite of test files and their associated documentation. These test files, referred to as the "Kensington Suite," should be run by anyone who is intending to create, receive or process PDF/X files.

The Kensington Suite will be made available at no charge on the websites of the two organizations, namely http://www.ddap.org/ and http://www.NAA.org/, as well as at http://www.npes/standards/tools.html.

2.6 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 15930-1:2001, PDF/X-1a:2001 or PDF/X-3:2002.

2.7 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. The use of lossy compression may result in degradation of image quality.

LZW compression is prohibited in PDF/X. Note that current 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.

2.8 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 (like 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 – 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.

2.9 Box usage and management

The inclusion of the bounding boxes as specified in *Portable Document Format Reference Manual, Version 1.3, Second Edition*, 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**.

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

The only PDF/X standard that permits the use of encryption is PDF/X-1:2001.

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 – it should be assumed that a compliant PDF/X reader cannot read an encrypted file without prior decryption.

2.11 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. For PDF/X-1, both the DCS and EPS file formats also have the capability to contain specific screening information.

NOTE: Above references to DCS and EPS apply only to PDF/X-1, not to PDF/X-1a, or PDF/X-3. TIFF 6.0 and TIFF/IT-P1 file formats do not allow for the exact specification of screening.

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 prescreened 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.

2.12 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.

2.13 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.

2.14 Color space 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 referenced in Adobe Technical Note #5413.

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 sections 3.8 for details specific to PDF/X-1, PDF/X-1a, and 5.2 for details specific to PDF/X-3.

2.15 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 *Portable Document Format Reference Manual, Version 1.3, Second Edition* 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.

2.16 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.

2.17 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.

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 expected to establish 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.

2.18 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 recommendations below 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.

- Many PDF readers cannot process an **Image XObject** whose **Width** value exceeds 32,000. 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 the appropriate Portable Document Format specification version for the conformance level.

2.19 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.

Examples of such problems are:

- 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.

2.20 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 Manual 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.

2.21 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 '.

2.22 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.

2.23 Use of transparency

The PDF 1.4 specification adds a number of features to PDF, including support for partial transparency. Objects are marked as transparent by the addition of new keys to various objects in the PDF file, especially in **ExtGState** objects and **Image XObjects**. Some combinations of values for these keys indicate partial transparency, while other combinations denote "null transparency" – where objects are fully opaque as is the case for all objects in files compatible with PDF version 1.3 or earlier.

A PDF/X compliant reader processing a file that includes a **GTS_PDFXVersion** key identifying it as a PDF/X compliant file should ignore any keys that are not defined in version 1.3 of the PDF specification or in the ISO 15930 International Standard. They should be treated as private extensions that do not affect the final rendered representation. This means that the presence of keys and values described in version 1.4 of the PDF specification for representing non-null transparency does not prevent a file from being PDF/X compliant.

In such a case there is a potential for significantly different rendering of the file by a PDF/X-compliant application compared to what would be generated by applications in common use that are compatible with PDF version 1.4. This is obviously an undesirable result, therefore:

- PDF/X-compliant writers are recommended not to include the keys used for controlling transparency in PDF version 1.4.

- Tools designed to verify or correct PDF/X compliance are recommended to treat the presence of keys and values that would indicate non-null transparency to a PDF 1.4 reader as a condition requiring correction. Key/value combinations indicating null transparency may be accepted.

- Tools that render PDF/X files must render the file according to the PDF 1.3 specification and ignore any implication of partial transparency, rendering all objects as fully opaque.

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.

A file should be regarded as including non-null transparency if any of the keys shown in Table 1 is present in an **ExtGState** object with any value other than that shown in the table.

Table 1

Key	Value
BM	Normal or Compatible
CA	1.0
ca	1.0

A file containing the **SMask** key in an **ExtGState** object or **Image XObject** should also be regarded as containing non-null transparency

2.24 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 and should not be used in PDF/X files.

2.25 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 5.3.

3 PDF/X-1 and PDF/X-1a

The following sections describe attributes specific to ISO 15930-1:2001 (PDF/X-1:2001 and PDF/X-1a:2001).

ISO 15930-1 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 15930-1 (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.

3.1 Embedded OPI files

PDF/X-1 allows the use of OPI, but PDF/X-1a does not.

The PDF/X standard is designed to facilitate the exchange of complete files. In other words, all component files and resources that are contained within a single PDF/X file constitute a complete "page." This "page" can represent a single page of a product such as a book, or an advertisement that might be used at a later stage in the production process as a component of a page.

Within the PDF/X specification, it is possible to define the whole page in the PDF format (within the PDF/X boundaries); however, to allow easier integration of non-PDF files, the notion of Open Prepress Interface (OPI) references to embedded files was introduced.

In normal prepress usage, OPI refers to the mechanism/process by which a low-resolution, proxy image of a high-resolution page component is used in the creation process. The implementation of OPI expedites the creation and processing of a page layout file. At the final output stage of the process, the page layout file is delivered to an OPI server, where proxy images are replaced by their high-resolution counterparts, resulting in a complete file. While OPI itself is not limited to raster data, it should be noted that all OPI referenced, embedded files in PDF/X-1 are restricted to raster data.

In PDF/X-1:2001, this concept has been broadened to the point where the low-resolution proxy is present within the PDF/X file. This proxy references an embedded high-resolution file that is also within the body of the PDF/X file, as opposed to referencing the high-resolution file externally [on an OPI server].

The purpose of encoding the file in this manner is to allow the encapsulation of all high-resolution files within the body of the PDF/X file without having to encode them in the PDF format. This may be particularly applicable in workflows that include file types such as copydot and TIFF/IT-P1.

The receiver of a PDF/X file may choose to process the entire file dynamically [as is], or may develop a mechanism that strips out the high-resolution data, relocating it on an OPI server for subsequent OPI processing.

Within a PDF/X-1 file, OPI objects should be included only when there is an associated embedded file object.

Using embedded files in PDF/X-1 implies the need for appropriate processors for each embedded file type.

Conversely, in PDF/X-1a:2001, the presence of proxy images and OPI comments is strictly prohibited.

3.2 Advantages of PDF Structures vs. OPI References

Embedding OPI referenced files places 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 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 Use of PDF Security and encryption

It is possible to exchange a PDF file that uses **Standard** encryption as described in *Portable Document Format Reference Manual, Version 1.3, Second Edition*, to limit the permissions of the receiver. For example, the sender may wish to restrict the ability to edit the file. The receiver would then be required to contact the sender for permissions to go beyond viewing and printing.

The PDF/X-1 standard requires that the rights to view and print a PDF/X-1 file must always be granted.

In PDF/X-1a, however, any use of the **Encrypt** dictionary is prohibited.

3.4 File type registered names

The registered names for EPS and DCS in Table 1 of ISO 15930-1:2001, section 6.5, use underscores between the second and third parts of the name. All other registered names use a hyphen (minus sign).

3.5 Miscellaneous implementation issues

Subsequent to the completion and approval of PDF/X-1:2001 it was discovered that EPS and DCS files created using some versions of Adobe Photoshop include the stroke operator within the prolog. This particular usage does not impact the final print reproduction. Embedding the EPS or DCS file in a PDF file produces a file that does not conform to the PDF/X standard because the stroke operator is prohibited in embedded EPS and DCS files. (See 6.10 of PDF/X-1:2001.)

Had it been known that some raster EPS and DCS image files use vector-painting operators, in a way that does not affect final print reproduction, it is likely that such usage would have been allowed. It would have also been treated in the same way as valid PDF features that do not affect final print reproduction.

Other such anomalies might be found as PDF/X becomes broadly implemented. In these instances, it is desirable to have an application embedding EPS or DCS files as part of creating a PDF/X file remove the offending operator.

3.6 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.* PDF/X-1:2001 is based on the *Portable Document Format Reference Manual, Version 1.3 Second Edition* as extended by *Technical Note #5413*. Note that this standard has now been withdrawn in deference to ISO 15930-1:2001. 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 uses 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 has two distinct conformance levels 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.

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

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

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

3.8 Intended output condition

The characterized printing condition for which the file was prepared must be identified in the **OutputIntents** structure. In PDF/X-1 and 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 www.color.org.

Neither PDF/X-1 nor PDF/X-1a includes 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-1 and PDF/X-1a restrictions.

4 PDF/X-2

A future version of this application note will contain information on the PDF/X-2 standard.

5 PDF/X-3

The following sections describe attributes specific to ISO 15930-3:2001 - PDF/X-3.

ISO 15930-3 defines a data format and its usage to permit the predictable dissemination of a compound entity to one or more locations as CMYK data, spot color data or data using device independent color spaces, as well as any combination thereof, in a form ready for final print reproduction, by transfer of a single file.

A file conforming to ISO 15930-3 (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.

5.1 Use of PDF Security and encryption

It is possible to exchange a PDF file that uses **Standard** encryption as described in *Portable Document Format Reference Manual, Version 1.3, Second Edition*, to limit the permissions of the receiver. For

example, the sender may wish to restrict the ability to edit the file. The receiver would then be required to contact the sender for permissions to go beyond viewing and printing.

In PDF/X-3, however, any use of the **Encrypt** dictionary is prohibited.

5.2 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, **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** colors may not. Any **DeviceCMYK** data must be mapped via a **DefaultCMYK** color space, 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-1 and 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/.

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

If device-independent color data is used in PDF/X-3 files, the profile included in the **OutputIntent** of each file must be compared to those in all other files to be assembled together. Where all profiles are identical, the files may be assembled directly, retaining device independent colors. If different profiles are used, then colors must be transformed to the output device color space prior to assembly to ensure that the correct gamut and tone compression is performed for each entity. See also 2.25.

