Prepress Workflow Advancements Utilizing PDF/X-4 and the Adobe PDF Print Engine

Jason Lisi[*] and Christopher Smyth[**]

Keywords: Automation, digital, ICC, graphics, imaging, JDF, output, PDF, premedia, prepress, process, production, publication, publishing, rendering, RGB, standardization, standards, workflow

Abstract

Current prepress workflow processes have evolved significantly due to improvements to PostScript technologies; however, inherent limitations with PostScript still remain. Modern releases of Adobe's Portable Document Format (PDF) include features that are not supported in a PostScript environment, and consequently, native PDF prepress workflows and new PDF standards have been developed to take further advantage of advancements in PDF architecture.

It wasn't until the release of ISO standard PDF/X-1a:2001 (ISO 15930-1) that the PDF/X standard gained uptake amongst application developers. Based on PDF 1.3, PDF/X-1a:2001 provided accuracy and predictability, while maintaining some degree of flexibility within the prepress workflow. However, as authoring software continues to grow in complexity, features have been added that are not easily supported in a PDF/X-1a:2001 file or traditional PostScript RIPs, such as native vector transparency and ICC color managed workflows. The need to flatten transparency and convert to CMYK so early in the workflow limits flexibility and repurposing of the PDF file and can result in unpredictable errors in output. PDF/X-4:2008, which became an ISO standard (15930-7) in 2008, offers many features that can overcome the limitations inherent in PDF/X-1a:2001, specifically when integrated with a native PDF workflow, such as the Adobe PDF Print Engine launched by Adobe in 2006.

Both the Adobe PDF Print Engine and the PDF/X-4:2008 standard are still in the early stages of their respective product life cycles, and mainstream adoption has not yet occurred; consequently, this research project was conducted to test the feasibility of using these combined technologies for existing large-scale production workflows, specifically magazine publishing.

[*]Ryerson University, School of Graphic Communications Management
[**]PreMedia Consultant, Toronto, Canada

In this two-stage research project, the end usability of both PDF/X-1a:2001 and PDF/X-4:2007 (draft) files was tested. It should be noted here that PDF/X-4:2007 is a draft specification of the ratified 2008 standard which was available as an Adobe PDF Preset in Creative Suite 3, hereafter referred to as PDF/X-4:2007. Custom test pages, as well as version 2 of the Ghent Output Suite, were used to contrast and compare output results created from running PDF/X-1a:2001 and PDF/X-4:2007 files through workflows based on a Configurable PostScript Interpreter (CPSI) RIP and Adobe PDF Print Engine (APPE) RIP.

In phase one of the research PDF/X-1a:2001 and PDF/X-4:2007 files were created as a direct export from Adobe InDesign CS3 and were processed at three different magazine production facilities, all using a Kodak Prinergy 4.0 workflow system. The PDF/X-1a:2001 files rendered differently for each of the three test teams, depending on the job processing parameters set within the RIP. None of the PDF/X-1a:2001 files output as was intended in the original files, regardless of whether the CPSI or APPE RIP was used. The PDF/X-4:2007 processed with the CPSI RIP also produced erratic and inconsistent results for all output teams. In contrast, the PDF/X-4:2007 files processed with the APPE RIP reproduced consistently for all teams, with the output matching the intended result in the file. The PDF/X-4:2007 files processed through the APPE showed a clear improvement in the reproducibility of vector-based transparent objects, especially when spot color objects were placed in transparent vignettes, and then mapped to process as part of the RIP workflow parameters.

Phase two testing narrowed down the test parameters to focus on the discrepant results found in phase one. In particular, it was discovered that a PDF/X-1a:2001 file would produce much different results based on where in the workflow Pantone colors were mapped to process. Further testing was done with regard to the effect of color mapping on transparent Pantone objects in the flattened PDF-X/1a:2001 files, as well as the effect of different overprint settings set within process template of the RIP. The findings in phase two confirmed that the application of color mapping at different stages of the workflow had significant affect on the outcome when PDF files are flattened prior to output but had little consequence for the non-flattened PDF/X-4:2007 files processed through an APPE RIP.

Based on the results of both phases of testing, it was concluded that PDF/X-4:2007 files processed through the Adobe PDF Print Engine could offer measurable workflow improvements over flattened PDF/X-1a:2001 files processed using a conventional CPSI (PostScript) RIP, especially when vector transparency with spot color objects come into play. In addition, it was determined that the expanded color managed color spaces could be very useful for printers that are looking to implement a late binding color workflow to better take advantage of mixed output capabilities, such as traditional offset and digital print.

Introduction

The initial testing and research summarized in this paper was initiated by Magazines Canada, a national non-profit trade association in Canada that advocates consumer, cultural, specialty, professional, and business media magazines in Canada (http://www.magazinescanada.ca).

The Magazines Canada Technical Standards Sub-Committee (TSC) develops and maintains the digital Magazine Advertising Canadian Specification (dMACS), material specifications, as well as supports user education and industry best practices. As part of the dMACS specification document, the Magazines Canada TSC has endorsed PDF/X-1a:2001 as an acceptable format for digital ad material submission for blind exchange, and advocates its use as a reliable and consistent format for digital ad submission for Canadian Magazine publishers (Magazines Canada, 2009).

Overview of the Research Project

In late 2007 and early 2008, the Magazines Canada TSC commissioned a small working group to investigate the feasibility of adopting PDF/X-4 as a standard for digital file submission for magazine ads in Canada, as either a replacement for PDF/X-1a:2001, or an alternative to it. The TSC concluded that such a study was warranted based on three key factors: (1) With Adobe Creative Suite 3, PDF/X-4 became a selectable option for PDF export; (2) More and more printers were gaining access to Adobe PDF Print Engine RIPs through system upgrades; and (3), Increasing design complexity was beginning to challenge the PDF/X-1a:2001 file format and traditional PostScript workflow.

The working group, which consisted of magazine publishers, magazine printers, and graphic communications educators, outlined key areas of concern that needed to be studied. In particular, the working group was interested in analyzing if PDF/X4 files, when processed using the Adobe PDF Print Engine (APPE), circumvented some of the more common issues that can affect output quality of files in a more traditional workflow, including, but not limited to:

- Vector-based transparency and problematic transparency flattening;
- The use of Adobe Smart Objects with multiple tiers of embedding within native application files;
- Vector-based transparencies that blend from a spot color to a process color that has had transparency applied to it;
- Complex object layering and the effect of layer groups;
- The use of non CMYK images and graphics;
- In-RIP color conversion and flattening issues.

Background on PostScript and PDF

A brief overview of the workings of PostScript and PDF, with regard to final output for print, development, technology, and capabilities, will help build a framework for appreciating the industry need to develop a new standard, PDF/X4. As well, a review should help with understanding the perspective of the work group's research and testing of the standard, used in conjunction with a native PDF renderer.

Page Description Language

A Page Description Language (PDL) is a language that describes the final appearance of a printed page in an abstract format, rather than only using bitmapped data.

PostScript

Released in 1984 Adobe's PostScript Level 1 was unique from other solutions at that time, in that it was not tied to a specific output device. According to Adobe, it was designed to "provide a uniform way to represent visual elements on any raster device" (Adobe Systems Incorporated, 1988 p. 6). PostScript was removed from "the level of rasterization to ensure true device independence" (Adobe Systems Incorporated, 1988 p. 54).

Adobe also took a different approach for their business model in that Original Equipment Manufacturers (OEMs) could license a PostScript interpreter and use it to build an output device. Adobe also provided open access to the specifications, which allowed others to design supporting software (prepressure.com, 2009a). These factors, combined with the other key innovations at the time, saw PostScript emerge as the de facto standard for output in the print industry.

PostScript Imaging Model

The PostScript language is designed for two purposes: "it provides an imaging model for describing and printing complex text and graphics, and it is a complete and general programming language" (Adobe Systems Incorporated, 1988, pp 57). While about "one-third of the PostScript language is devoted to graphics," "the remainder makes up an entirely general computer programming language" (Adobe Systems Incorporated, 1985 p. 4).

An imaging model can be considered as an abstract concept, through which graphics are rendered. It is a set of rules that are used by output devices. A sophisticated imaging model "enables applications to describe the appearance of pages containing text, graphical shapes, and sampled images in terms of abstract graphical elements rather than directly in terms of device pixels" (Adobe Systems Incorporated, 2006 p. 34). This kind of high-level detachment "frees application software from having to make device-specific rendering decisions" (Adobe Systems Incorporated, 1988, p. 6) and is what allows PostScript to be device independent.

Specifically, PostScript page content is specified in terms of straight lines and cubic Bézier curves, utilizing a Cartesian plane coordinate system of 'x' and 'y' pairs. This vector flexibility allows for "arbitrary page transformation such as scaling and rotating, and also allows the file to be output at a variety of resolutions" (Adobe Systems Incorporated, 1985). The PostScript imaging model also uses a concept of a 'Current Page,' on which PostScript draws content. To begin with, the current page is blank; PostScript uses painting

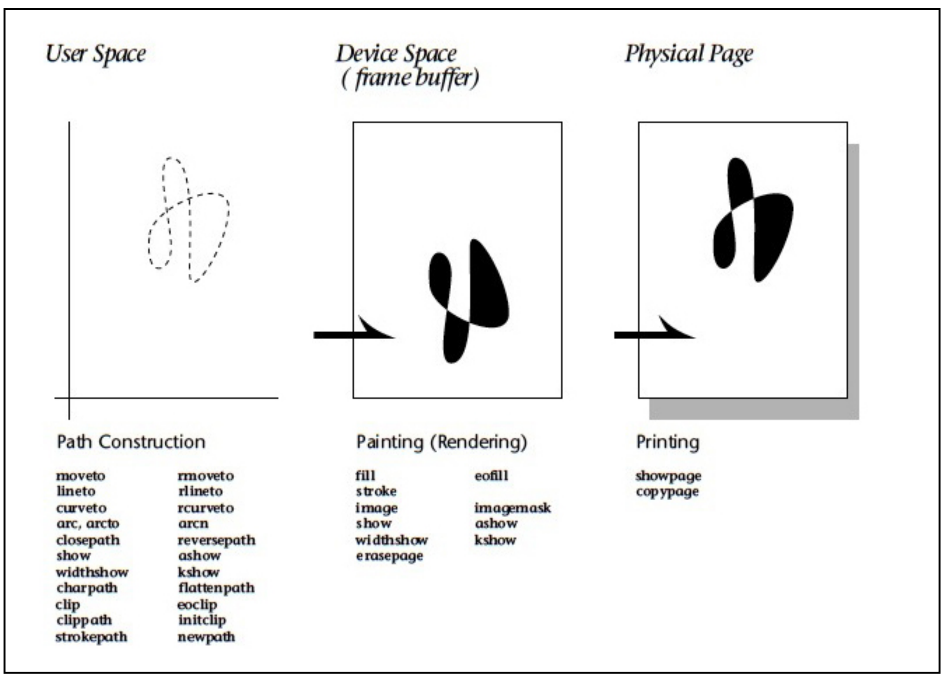


Figure 1. *PostScript Imaging Model, (Source Adobe Systems Incorporated, 1988, p.49).*

operators (such as fill or stroke) to "place marks on the current page, each of which completely obscures marks that they may overlay" (Adobe Systems Incorporated, 1985, p. 10). This is a significant aspect of PostScript since painting is opaque in the PostScript imaging model (Adobe Systems Incorporated, 1985). Any new marks knock out whatever was underneath it; consequently, PostScript cannot directly output pages created with partial transparency effects applied.

PostScript as a Programming Language

PostScript is an imperative type programming language, which means that a PostScript file defines a sequence of commands for the interpreter to resolve (Adobe Systems Incorporated, 1988). PostScript is interpreted and stack based, which allows files to be of varying lengths and complexities (Adobe Systems Incorporated, 1988).

This is important to note, because, a "showpage" command is used in the program, which triggers final output, but only after the entire page has been read. There is no random access to page contents in a multipage document; all pages must be processed in sequence in order to determine any one pages final appearance (Adobe Systems Incorporated, 1999). PostScript was designed for output, not interactivity.

Further Development of PostScript

PostScript Level 2 was released in 1991, and included several improvements: improved speed and reliability (caching reusable content), support for in-RIP separations, image decompression (for example, JPEG images could be rendered by a PostScript program), support for composite fonts, and improved screening algorithms Adobe Systems Incorporated, 1991).

PostScript was further developed, and in 1998 Adobe released PostScript 3. An important feature in this version was output support for more than 256 gray levels per color (12-bit screening allowed for up to 4096 gray levels per color). This helped address visible banding in blends. PostScript 3 also offered improved support for in-RIP separations (DeviceN), as well as support for PDF files (Adobe Systems Incorporated, 1999).

Portable Document Format (PDF) and the PDF Imaging Model

Adobe released the Portable Document Format (PDF) in 1993. It was designed as a tool to allow people to exchange electronic documents independent of the original authoring environment, portable across all platforms and operating systems (Adobe Systems Incorporated, 1999).

The PDF is a subset of the PostScript Page Description Language; however, unlike PostScript, PDF is not a programming language. It provides resolution independence, but it also includes a document structure that supports navigation within the file.

This structure also allows content to be included as objects—for example, annotations and external links—and be cataloged into a cross-reference table, which is included at the end of a file (Adobe Systems Incorporated, 1999). As such, a PDF file can be compared to a database, allowing for direct access to each object, and each page of a PDF document is independent of the others (Adobe Systems Incorporated, 1999). This lets the content objects be accessed randomly by a software reader, as opposed to PostScript, where objects need to be processed sequentially (Adobe Systems Incorporated, 1997). While the mechanisms for the imaging model are similar, PDF significantly differs from PostScript in the following ways (Adobe Systems Incorporated, 2006):

- PDF has a defined file structure that allows an application to access parts of a document in arbitrary order.
- PDF does not include programming language features such as procedures, variables, and control constructs.
- Font metrics are included in PDF files for accurate viewing and output
- A PDF file may contain non-imaging data, such as hyperlinks and logical structure information for document interchange."

Further Development of PDF

In 1996, Adobe released Acrobat 3.0 and the PDF 1.2 specifications. PDF 1.2 was the first version of PDF that was generally considered suitable for a commercial prepress environment, because it included support for: Open Prepress Interface (OPI), CMYK color space, spot colors, as well as halftone functions and overprint information.

Commercially available in 1999, Acrobat 4.0 and PDF 1.3 extended support for the print industry by including support for color management, International Color Consortium (ICC) profiles, as well as DeviceN color spaces (Scribus, 2008).

The original imaging model of PDF was opaque, similar to the PostScript imaging model. In 2001 Adobe released Acrobat 5 and PDF 1.4, which included an important key function transparency. In 1993, Acrobat 6 and PDF 1.5 brought another key feature, Optional Content Groups (layers) (Adobe Systems Incorporated, 2003).

Transparency and Its Relationship to File Production for Print

Transparency in layout applications is the ability to control the opacity of an object so that it is translucent, or semi opaque, this allows any objects beneath it to be visible (Adobe Systems Incorporated, 2007a). This is done through a process where objects can be overlaid, or "composited," with the previously existing contents of the page; "producing results that combine the colors of the object and its backdrop according to their respective opacity characteristics" (Adobe Systems Incorporated, 1999). Partial transparency is always simulated at some level by mixing colors. The simulated effect is achieved by applying a variety of blending effects, which softens the edges of an object by smoothly fading the object from opaque to transparent (Adobe Systems Incorporated, 2007a).

Flattening Transparency

Transparency must be "resolved" before final output for print because screened output can't be translucent for offset printing.

At its simplest, flattening converts all overlapping areas in a stack of transparent objects (atomic regions), plus all text and graphics that interact with transparency, into smaller opaque regions (complexity regions) that simulate the appearance of the original transparent. Flattening cuts apart transparent art to represent overlapping areas as discrete pieces that are either vector objects or rasterized areas. As artwork becomes more complex (mixing images, vectors, type, spot colors, overprinting, and so on), so do the flattening and its results (Adobe Systems Incorporated, 2007a).

In layout authoring applications, the "object stacking order," or how the objects interact, can significantly impact how software resolves or flattens the transparency. This can create significant differences between on-screen renderings and final output. Generally accepted best practices dictate that text and spot colors should be kept at the top of the stacking order, which will help keep them from being unnecessarily affected by flattening. Spot colors can further pose a challenge when used with transparency, since the flattening process can add overprint instructions to appropriate spot-colored atomic regions.

In the case of two objects to which process colors are applied, the intersection of divided atomic regions is a single process color object. With spot colors, overprint is required to separate the colors correctly, even if no overprint has been manually configured. This is significant for accurate output as workflow software can be configured to reset overprints and knockouts, which can have implications for predictable output.

Transparency Output in a PostScript Workflow

The output of Transparency in Adobe CPSI workflow has led to some documented reproduction concerns (Adobe Systems Incorporated, 2007b).

- Spot colors may display colors on process plates or convert to process.
- Transparency flattening can include the process of executing the overprint attribute manually assigned to spot objects. When this occurs,

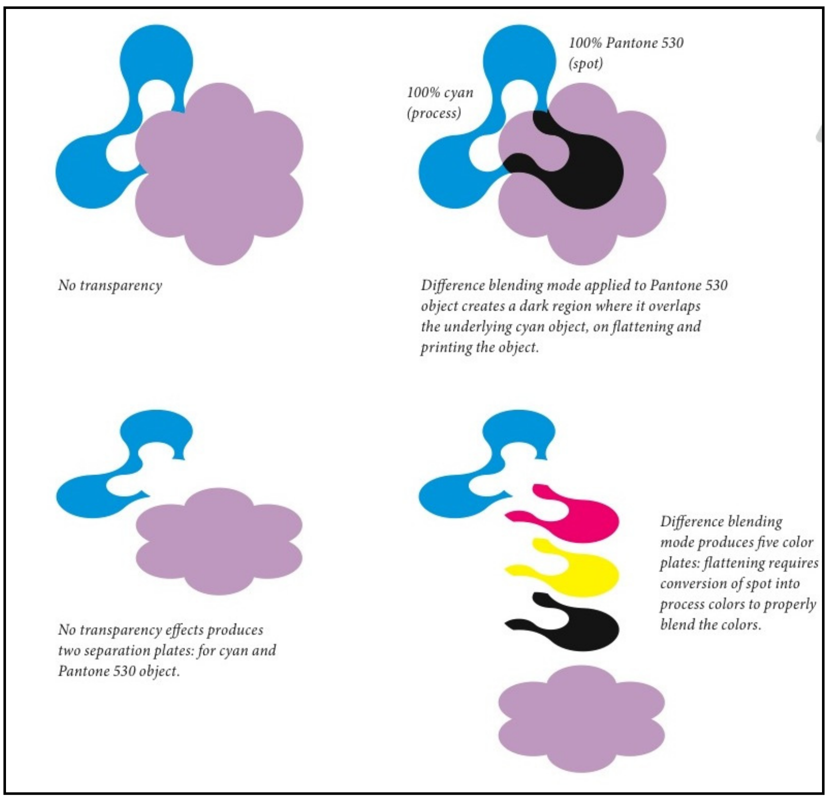


Figure 2. *Four Color Process With Spot Transparency Flattening (Source Adobe Systems Incorporated, 2007a, p. 16)*

overprinting instructions are not preserved after flattening; however, the objects look correct when printed because the overprint is taken into consideration when transparency is flattened.

- Vector objects may get rasterized at a resolution that's too low for the output device.
- Artifacts may appear along the edges of atomic regions.
- Hairlines and strokes may fatten. While this is generally understood to occur on lower-resolution output devices (preliminary proofs), it can cause delays while the anomalies are investigated.
- Type may be converted to filled strokes, thickening characters. Again, this problem usually occurs on low-resolution devices. It is less of a problem on high-resolution devices.
- Transparency and OPI Open Prepress Interface (OPI) is not compatible with transparency. Low-resolution images must be replaced with high-resolution images before flattening. If they aren't replaced and low-resolution images are flattened, the results will be low resolution.

The transparency effects within PDF files were developed using extensions to the PDF language; however, this also meant that PDF applications designed for earlier versions of the specification might display and output significantly different results than applications that could fully support transparency (Adobe Systems Incorporated [2007a]). Adobe designed transparent elements to appear opaque in earlier versions of PDF. Unfortunately this led to situations where files were output, and reproduced, that were different from the intended end result. This had a significant impact on the print industry because "when PDF files are used to prepare work for professional printing, transparency issues could cause millions of printed copies to be incorrect, and have to be destroyed" (Adobe Systems Incorporated [2007a]).

Optional Content Groups (OCG); "Layers"

PDF version 1.5 introduced the concept of "Layers" to PDF files; however, the concept of layers in PDF files is not the same as layers in layout authoring applications. Referred to as Optional Content Groups (OCG) in PDF, PDF layers were designed to provide a mechanism to incorporate related optional content into one PDF file. For example, OCGs could have useful benefits for Publishers and Catalogers for language or regional versioning, Packaging workflows for versions, dielines, printer's marks, and folding information, or in variable data workflows for promotional marketing materials (Adobe Systems Incorporated, 2003).

Layers in the application file can be "preserved" when creating a PDF 1.5 file, becoming Acrobat layers, or optional content groups (OCG). However, Adobe

InDesign does not currently support this for exporting a PDF/X-4 file, since PDF/X-4 is based on PDF 1.4. With a PDF/X-4 file, the application layers are processed into an Optional Content Configuration Dictionary (OCCD). An OOCD is a set of OCGs in one fixed group (Prepressure.com, 2009b). This approach was taken in an effort to adhere to the "blind exchange" philosophy of PDF/X files. The layers could not be unambiguous, or the receiver would have to contact the supplier to determine which layers should be output. An OCCD PDF/X-4 file could allow for automated processing by the receiver.

The Adobe PDF Print Engine and the Implications of a PDF RIP

The differences between PostScript and PDF also mean that a PDF file generally must be translated, or converted, before being output through a PostScript workflow. This can be a challenge with regard to varying PostScript levels.

Recently, native PDF interpreters have been developed to take advantage of the additional functionalities of the PDF imaging model not supported in the PostScript model. Global Graphic's "Eclipse Release" of their Harlequin was available in 2002. Adobe introduced their Adobe PDF Print Engine (APPE) in 2006, and an updated version in 2008.

The major benefits to a native PDF interpreter based workflow are support for more advanced PDF functions such as transparency and layers, as well as fewer overall interpretation errors through less file conversions. A native PDF renderer should allow for a late binding workflow, where transparency can remain unresolved, or native, until imaging. This should improve processing time, reduce output errors and limit "time and materials wasted in troubleshooting, proofing, and reprinting in the production process" (Adobe Systems Incorporated, [2007b p. 2]).

Evolution of Premedia Workflows

With the 1990s' advent of computer-to-plate (CTP) technologies in offset printing, the traditional film-based image carrier to the plate became an electronic file. This freedom from manufacturing constraints led to a redistribution of responsibility in the supply chain, as the newer technologies and systems were adopted. However, this was sometimes accomplished without the support of traditional prepress knowledge and techniques.

A variety of file formats emerged to replace analog film in the prepress workflow:

- Application files, from specific versions, and different platforms;
- CEPS (Color Electronic Prepress Systems) data;

- CT/LW (Continuous tone /Linework);
- TIFF/IT-P1 (Tag image File Format for Image Technology) files;
- PostScript, and variants (EPS, DCS);
- PDF, and derivative PDF/X variants.

The different file formats had varying levels of output predictability, and often an inverse relationship with flexibility. Rasterized formats, such as TIFF/IT-P1 files, left little to interpretation and offered almost the same stability and reliability of film; however, they required specialized (and relatively expensive) software to create and consume and offered no flexibility for last-minute changes. Application files offered much greater flexibility but, due to issues such as fonts and support art, less stability.

Customers and suppliers began to develop concerns about digital file submission for print. There was confusion over whose responsibility it now became to create the final files for press, as well as who assumed responsibility to generate final proof and verify content, as well as the often-contentious costs associated with reproduction concerns. Communication and education about file format requirements and prepress considerations became a significant part of the process for publishers and printers.

Preflighting Files

The unpredictable quality of digital files submission for print led to the development of a new industry channel, "Preflight," with software and processes to verify material before it entered the production stream. Preflight evolved and changed in an effort to address common file submission problems. It also highlighted the potential benefits the potential benefits of a standardized file format: revenue protection, avoiding lost opportunity costs, and improved efficiencies through further processing automation

Publishing as a Case Study

Publishers serve two audiences: their readers and their advertisers. In order to help deliver the advertisers' messages to the readers, the advertising files must be incorporated into the workflow. These files could come from a variety of national and international advertising agencies, as well as from smaller localized accounts. This spectrum of suppliers encompasses a varying level of understanding and application of knowledge in premedia workflow.

Advertising files would have to be verified and processed into a format that could be used with common desktop publishing applications. The full-page advertising files must be combined with the editorial files to create printer forms, and partial page ads have to be inserted into editorial pages. The continuing pressures of accelerated production schedules reinforced the need to have an accurate and reliable file format that could be easily integrated into the existing digital production workflow. The development of PDF/X-1a:2001 offered a standard file format that could be successfully used and offer the desired end results.

A blind exchange is designed to be one in which the parties have no requirement to exchange or communicate technical requirements in advance, other than their agreement on a standard. This improves efficiencies and final quality by reducing process and output errors. Suppliers could standardize their internal workflows to produce one kind of file; receivers could build processes around incorporating a stable, predictable file format. An accredited, open, standard has benefits over a proprietary specification in that it is developed through a collaborative process. After Adobe released the PDF language into the public domain, "it became feasible to work with Adobe to extend the format to meet the needs of the graphic arts and to then develop a standard that defined appropriate usage of the format in graphic arts applications" (McDowell 1998 p. 23). Using PDF as the basis for a standard for graphic arts has a key benefit in that it would be based on a file format that was readily understood and supported in the supply chain.

Standard File Formats: PDF/X-1

PDF/X-1 was developed based on work started by American National Standards Institute's (ANSI) Committee for Graphic Arts Technologies Standards (CGATS) in the late 1990s and later published by the ISO in 2001. The format is a subset of the PDF, with some key attributes to ensure that the file could be used for blind exchange and offer stable, predictable output after processing, including:

- Embedding of fonts;
- CMYK with specific compression (only FLATE and RunLength lossless compressions);
- Output intent specified (SWOP, FOGRA, etc.);
- Trapping key must be identified;
- The definition of a bleed, trim, and art-box;
- No OPI support;
- No multimedia support;
- No document encryption.

Originally based on PDF 1.2, the standard was updated to PDF 1.3 (which was released in 1999) for X1-a: 2001. Publishers, such as Time Inc. in particular, were early adopters of PDF/X-1 because of the clear benefits across the supply chain (Bailey 2005). PDF/X-1a is now a very common approach to solving problems of production file reliability and customer education within the magazine publishing industry, as "PDF/ X-1a workflows near ubiquity for some publishers" (Pettas, 2008 ¶ 1). The files are easily created and exchanged within common desktop publishing applications, with support for PDF/X-1 in Adobe Creative Suite, as well as QuarkXPress.

Other PDF/X Formats

The ISO group also published other variants of PDF/X, in addition to X1. PDF/X-3:2002 was published in 2002 and is very similar to PDF/X-1, except that PDF/X-3 supports RGB and CIELAB color spaces and ICC based workflows. However, a color-managed workflow can pose significant challenges for a blind file exchange, moreso with supplied contract color proofs. It requires the file receiver to pay particular attention when processing files to ensure that the correct steps are taken to honor and apply profiles. This can be further complicated if the file creator hasn't necessarily taken the same steps with proof creation to ensure reliable output (Universal Photographic Digital Imaging Guidelines [UPDIG] Coalition, 2008). In general, the specific use of PDF/ X-1 or PDF/X-3 in publishing seems to depend on geographic location; PDF/X-1 is preferred in North America, while PDF/X-3 is supported in some European based workflows (Bailey, 2005).

In addition, other PDF/X standards were developed to meet specific needs of industry channels. While in general "the goal throughout the various parts of ISO 5930 has been to maintain the degree of flexibility required while minimizing the uncertainty" (ISO, 2008 p. v), the use of a standard file format for exchange assumes a certain level of understanding of the attributes by the parties involved in the exchange. With PDF/X, the responsibilities are on the supplier to ensure settings are correct and on the receiver (processor) to ensure those settings are honored.

PDF/X Testing and Research Conducted

Magazines Canada is a national, non-profit trade association advocating consumer, cultural, specialty, professional, and business media magazines in the country. The association concentrates on government affairs, services to the advertising trade, circulation marketing, professional development, and direct-to-retail distribution.

The Technical Standards Sub-committee of Magazines Canada develops and maintains Magazine Advertising Canadian Specifications (dMACS) material specifications as well as supports user education and best practices in the industry.

Magazines Canada's Technical Standards sub-Committee created a working group to research, test, and evaluate support for the PDF/X-4 standard through member publisher's existing CPSI based workflows, as well as potential workflows based on the APPE. This mandate was driven by the ongoing development of the PDF/X-4 and the ability to select it as an option in Adobe Creative Suite version 3 applications. In addition, many of the major publishers and printers in the Canadian market were gaining direct access to the APPE.

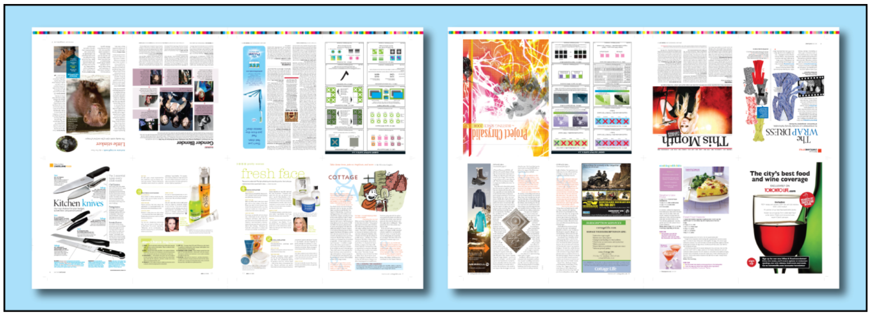
The testing was based on Adobe Creative Suite 3 applications, as well as workflow software running APPE version 1. Although Creative Suite 4 and APPE version 2 are now both commercially available, neither were commercially available at the beginning of testing. The intent of the testing was to determine if the dMACS should be updated to include support for PDF/X-4.

Based on initial discussions, the working group devised a 16-page Adobe InDesign test form that would be used to investigate the results of processing PDF/X-1a:2001 and PDF/X-4 files through both a standard CPSI RIP and an Adobe PDF Print Engine (APPE) RIP. It should be noted that this testing used an Adobe draft of the PDF/X-4:2008 standard which was available as an Adobe PDF Preset in Creative Suite 3, hereafter referred to as PDF/X-4:2007. Please refer to the Validity of Research Methodology Discussion later in the paper for more information about the PDF/X-4 draft specification.

It was decided that the test pages should be a combination of live production pages (i.e., "live" pages from magazines) and designed test pages and targets. To that end, one full-page house ad, eight editorial-only pages, and three editorial pages with placed PDF partial ads were supplied by two different magazine publishers to use for testing. Two of the test pages consisted of selected test targets from the Ghent Output Suite version 2.0, one test page was a fabricated mock editorial page that contained an RGB image and vector transparency, and the last test page was a mock magazine cover with advanced design complexity.

The group selected the specific Ghent Output Suite V.2.0 targets based on discussions about the practical possibility of accepting a file format that supported embedded ICC profiles. It was decided it would be useful to test each site's individual workflows, in an effort to clearly identify what areas would need to be addressed to support this, as well as well to comprehensively test some of the targets for a variety of overprint conditions. Both of these are of particular importance to publishers who need to combine third-party- supplied

Figure 3. 16 Page Test Form Used in Round One Testing.



PDF/X-1a and PDF/X-3 files (partial page advertisements) with editorial pages (Hartsock, 2007).

Summary of Test Form Pages

Page 1: Chrysalid Cover

The Chrysalid cover was the most complex page of the test suite, incorporating several challenging elements:

- A layered Photoshop file placed in InDesign. The layered Photoshop file consisted of:
    - Two spot channels (172C and 604C)
    - An Illustrator file with a linked TIFF file, complex vector objects, and text placed as a smart object in a layer with blending mode set to multiply
    - A layered Photoshop placed as a smart object in a layer with blending mode set to multiply
- Two rounded text frames created in InDesign with a stroke of 55K set to multiply with 29% opacity
- Main heading text created with an Open Type font and color set to PMS 172C
- Sub heading text outlined and filled with a gradient that went from PMS 172 to 100Y 10K. The object was set to 82% opacity and placed over a box of 100Y 10K set to 56% opacity

Page 2: Ghent Output Suite V. 2.0 Page 1 of 2

Page two consisted of a mosaic of Ghent Output Suite 2.0 test targets placed in InDesign as PDF files:

- GWG 1.0 – CMYK Overprint Test
- GWG 1.1 – CMYK Overprint Mode
- GWG 2.0 – Spot to CMYK Overprint
- GWG 3.0 – Gray Overprint Patch

- GWG 3.1 – Gray Image Overprint
- GWG 4.0.1 – White Overprint Patch
- GWG 5.0 – Font Substitution
- GWG 5.1 – Font Subset and Substitution
- GWG 6.0 – Use of Shadings
- GWG 6.1 – Use of Shadings

Page 3: Nature Scrapbook – "Little Stinker"

Page three consisted of a real-life magazine editorial page created in InDesign that was modified to include an RGB image with an embedded profile, as well as vector-based transparency applied to a custom text frame in InDesign (object set to multiply at 100%).

Page 4: Style Icon – "Wrap Dress"

Page four was an editorial page with multiple close-cropped images.

Page 5: Toronto Life – "Wine Guide"

Page five was a full-page house ad for a magazine website.

Page 6: How to Buy – "Kitchen Knives"

Page six was an editorial page with multiple close-cropped images.

Page 7: Cottage Life – "Cottage Q & A"

Page seven was an editorial page with multicolor type and limited graphics.

Page 8: Cottage Life – "Cottage Q & A"

Page eight was an editorial page with a 1/3 vertical ad placed in InDesign as a PDF.

Page 9: Cottage Life – "Cottage Q & A"

Page nine was an editorial page with two different ads placed in InDesign as a PDF.

Page 10: Pretty Woman – "Fresh Face"

Page ten was an editorial page with multiple close cropped and uncropped images.

Page 11: Pretty Woman – "Fresh Face"

Page eleven was an editorial page with multiple close cropped and uncropped images.

Page 12: Cooking With Kids – "Strawberry Lime Cooler"

Page twelve was an editorial page formatted as a recipe with text and images.

Page 13: Theatre – "This Month"

Page thirteen was an editorial page with text and a simple image layout.

Page 14: In Concert – "Gender Bender"

Page fourteen was an editorial page with multiple images in a grid layout.

Page 15: In Concert – "Gender Bender"

Page fifteen was an editorial page with a 1/3 vertical ad placed in InDesign as a PDF.

Page 16: Ghent Output Suite V. 2.0 Page 2 of 2

Page sixteen consisted of a mosaic of more Ghent Output Suite 2.0 test targets placed in InDesign as PDF files:

- GWG 7.0 – Use of embedded profile in output intent
- GWG 7.1 – Use of embedded profile in output intent
- GWG 7.2 – Use of embedded profile in output intent
- GWG 8.0.1 – DeviceN Support (6 colors)
- GWG 8.1 – DeviceN Support (5 colors)
- GWG 8.2 – DeviceN Support (4 colors)
- GWG 9.0 – Font Support
- GWG 10.0 – Spot to CMYK conversions
- GWG 11.0 – Use of Default CMYK color space
- GWG 12.1 – Black Overprint/Knockout

Round One Testing

Research Design and Testing Methodology

In the initial round of testing, the 16-page InDesign test form was exported to PDF from InDesign twice: once using the PDF/X-1a:2001 Adobe PDF Preset, and once using the PDF/X-4:2007 (draft) Adobe PDF Preset (non-flattened). These two groups of PDF files were each given to two separate magazine publisher/printer test teams to output using their Kodak Prinergy 4 workflows. Each PDF file was processed through the Prinergy workflow twice: once using the standard CPSI RIP and once using the APPE RIP. Each output team used their standard in-house workflow settings that they would normally use to process magazine pages. This round of testing resulted in a total of 64 refined pages per output team across four different jobs:

- 16 PDF/X-1a:2001 pages processed with the CPSI RIP;
- 16 PDF/X-1a:2001 pages processed with the APPE RIP (no transparency flattening);
- 16 PDF/X-4:2007 pages processed with the CPSI RIP;
- 16 PDF/X-4:2007 pages processed with the APPE RIP (no transparency flattening).

To accurately represent the basic four-color workflow of national magazine pressruns, each team ran the files through the RIP with color mapping so that any spot colors in the PDF files were converted to process at the RIP stage. Each output team processed the test pages, output them to hard copy proof, and analyzed them for accuracy and consistency.

Results and Observations: Round One

After this initial round of testing, several observations were noted:

- Complex vector-based transparencies consistently reproduced as intended for files that were saved as non-flattened PDF/X-4 files, providing those files were processed using the APPE with no transparency flattening applied. This seemed to indicate that PDF/X-4 could be considered a viable alternative to PDF-X1a, providing they were used in a workflow where native vector-based transparencies are not flattened and the files are processed natively with the APPE.
- Complex vector-based transparencies did not reproduce as intended for files that were saved as PDF/X-1a files, regardless of whether those files were RIPped using the APPE or CPSI RIP. Not only did the files not reproduce as intended, the errors were inconsistent across sites.
- Complex vector-based transparencies did not reproduce as intended for files that were saved as PDF/X-4 and processed using the CPSI RIP. Interestingly, the results, although incorrect, were different than the results obtained with the PDF/X-1a files, suggesting that flattening done at the RIP stage produces different results than flattening done at the PDF creation stage.
- In general, the two pages of Ghent test targets did not reproduce as expected by either of the test teams, regardless of the PDF file format or RIP processing employed. This was a not unexpected result, as the participating sites were not currently using end-to-end ICC workflows. It was interesting to note that the knockouts and overprints returned different results between the sites. For a detailed breakdown of the Ghent target results from round one testing, please see **Table A1 in Appendix A**.
- The twelve standard editorial pages supplied for testing reproduced correctly as both PDF/X-1a:2001 and PDF/X-4:2007 files, on both the CPSI and APPE RIPs. This result was considered by the working group to be a positive outcome; it demonstrated that a that a PDF/X-4/APPE workflow could be used for the existing magazine production workflows, since it was able to produce the same results for pages that were known to pass through a PDF/X-1a/CPSI workflow without issue.

TABLE 1: SUMMARY OF RESULTS: ROUND ONE TESTING				
Test Page	PDF/X-1a CPSI	PDF/X-1a APPE	PDF/X-4 CPSI	PDF/X-4 APPE
01 (cov)	Fail	Fail	Fail	Pass
02 (Ghent)	Fail	Fail	Fail	Fail
03 (Stinker)	Fail	Fail	Fail	Pass
04 (Wrap Dress)	Pass	Pass	Pass	Pass
05 (Wine Guide)	Pass	Pass	Pass	Pass
06 (Knives)	Pass	Pass	Pass	Pass
07 (Q & A)	Pass	Pass	Pass	Pass
08 (Q & A)	Pass	Pass	Pass	Pass
09 (Q & A)	Pass	Pass	Pass	Pass
10 (Fresh Face)	Pass	Pass	Pass	Pass
11 (Fresh Face)	Pass	Pass	Pass	Pass
12 (Lime Cooler)	Pass	Pass	Pass	Pass
13 (This Month)	Pass	Pass	Pass	Pass
14 (Gender Bender)	Pass	Pass	Pass	Pass
15 (Gender Bender)	Pass	Pass	Pass	Pass
16 (Ghent)	Fail	Fail	Fail	Pass

Based on the findings of round one testing, the working group concluded that a second round of testing was necessary to determine why discrepancies occurred when flattening was employed and why the results varied dependant on the PDF format. The group also wanted to further explore the relationship between transparency flattening and color mapping of color and how the sequencing of events changes the outcome behavior. Last, the working group wanted to further explore the results of the Ghent targets to better understand the discrepancies in and between the sites that were observed in round one testing, especially with regards to overprint states.

Round Two Testing

Research Design and Testing Methodology

The working group decided that the second round of testing should focus on the four test pages that did not reproduce correctly on the first round of testing:

Page 1: Chrysalid Cover

The cover remains a valid test target due to its unique and varied transparency, the use of Pantone colorants in a non-traditional blend space, and the complexity of embedded objects with varying

Page 2: Ghent Output Suite V. 2.0 Page 1 of 2

After the initial results in round one testing, the working group decided to use the Ghent targets as a measurement of process control to determine the impact of modifying existing workflow parameters in an attempt to modify overprint behaviors.

Page 3: Nature Scrapbook – "Little Stinker"

This page remained a valid test target because the vector transparency that extends across both an RGB and CMYK image did not reproduce properly in a flattened workflow. Further investigation into why this occurred was warranted.

Page 4: Ghent Output Suite V. 2.0 Page 2 of 2

Similar to page two, this page was left in the test as a measure of the effect of changing workflow parameters and the result it has on color conversions, especially when Pantone colors are mapped to process prior to the RIP workflow versus the conversion being done in-RIP.

***Figure 4.** Four-Page Test Form Used in Round Two Testing.*

For round two testing, some key changes to methodology were made. In particular, the working group felt it best to reduce variables in the workflow, and work towards a common "job" configuration in the RIP. This would help to focus on the upstream and output processes. Consequently, workflow parameters were standardized between test teams to ensure the files would be processed identically between sites. By doing so, procedural implementation could be dismissed for any differences in results reported between sites. The other significant difference in testing for round two was that control workflow parameter modifications were tested to determine if outcomes could be altered significantly by adjusting in-RIP parameters.

Test parameters for round two testing:

- 4 PDF/X-1a:2001 pages processed with the CPSI RIP, no process modifications;
- 4 PDF/X-1a:2001 pages processed with the CPSI RIP, no color mapping (spot to process);
- 4 PDF/X-1a:2001 pages processed with the CPSI RIP, CMYK overprint turned on in Prinergy;
- 4 PDF/X-4:2007 pages processed with the CPSI RIP, no process modifications;
- 4 PDF/X-4:2007 pages processed with the CPSI RIP, no color mapping (spot to process);
- 4 PDF/X-4:2007 pages processed with the CPSI RIP, CMYK overprint turned on in Prinergy;
- 4 PDF/X-1a:2001 pages processed with the APPE RIP, no process modifications;
- 4 PDF/X-1a:2001 pages processed with the APPE RIP, no color mapping (spot to process);
- 4 PDF/X-1a:2001 pages processed with the APPE RIP, CMYK overprint turned on in Prinergy;
- 4 PDF/X-4:2007 pages processed with the APPE RIP, no transparency flattening, no process modifications;
- 4 PDF/X-4:2007 pages processed with the APPE RIP, no transparency flattening, no color mapping (spot to process);
- 4 PDF/X-4:2007 pages processed with the APPE RIP, no transparency flattening, CMYK overprint turned on in Prinergy.

Results and Observations: Round Two

One of the most significant discoveries in round two testing was that PDF/X-1a test files behaved quite differently depending on when in the workflow color mapping occurred:

- Vector-based transparent objects with spot colors did not reproduce as intended when the RIP was instructed to convert spot colors to process in the PDF-X/1a files.
- Vector-based transparent objects with spot colors did not reproduce as intended when the RIP was instructed to retain spot colors in the PDF-X/1a files.
- When the same PDF/X-1/a file was saved from InDesign with Spot to Process selected, the file behaved more predictably, and in most cases the file reproduced correctly on both the CPSI and APPE RIP.

Based on the round two testing, several key observations were noted:

- PDF/X-4 remained a consistent and predictable file format for complex files when used in conjunction with the APPE providing native transparency is retained until output; however, PDF/X-4 files that were flattened as part of the workflow did not render accurately regardless of workflow parameters set.
- When spot colors are used on transparent vector objects, especially spot to CMYK vignettes, there can be significant variability with regard to how those transparent object will be processed based on:
  - Whether or not the file will be flattened prior to output;
  - Whether or not color mapping will be employed;
  - What stage in the workflow color mapping is initiated (e.g., at the PDF creation stage or in the RIP).
- Despite modifications to overprint setting in the RIP, the test targets with the Ghent Output Suite 2.0 targets still did not output as intended in any of the round two testing scenarios.

TABLE 2: COMBINED SUMMARY OF RESULTS: ROUND TWO TESTING				
Testing Parameter	**Page 1 (Cover)**	**Page 2 (Ghent)**	**Page 3 (Stinker)**	**Page 4 (Ghent)**
PDF/X-1a:2001 – CPSI				
No RIP modifications	Fail	Fail	Fail	Fail
No Color Mapping	Fail	Fail	Fail	Fail
CMYK O/P On	Fail	Fail	Fail	Fail
PDF/X-4:2007 – CPSI				
No RIP modifications	Fail	Fail	Fail	Fail
No Color Mapping	Fail	Fail	Fail	Fail
CMYK O/P On	Fail	Fail	Fail	Fail
PDF/X-1a:2001 – APPE				
No RIP modifications	Pass	Fail	Pass	Fail
No Color Mapping	Pass	Fail	Pass	Fail
CMYK O/P On	Pass	Fail	Pass	Fail
PDF/X-4:2007 – APPE				
No RIP modifications	Pass	Fail	Pass	Fail
No Color Mapping	Pass	Fail	Pass	Fail
CMYK O/P On	Pass	Fail	Pass	Fail

Results and Discussions

Transparency, Spot Colors, and Flattening

In both rounds of testing done for this study, the cover page (page one) and the mock editorial page (page three) failed to refine properly when vector-based transparency was flattened as part of the workflow. In the case of the PDF/X-1a files, flattening occurred when the file was created. For the PDF/X-4 files, flattening occurred as part of the RIP workflow process when files were processed using the CPSI RIP. To understand why the issues occurred, it is important to expand on how the Adobe transparent imaging model works and how different color spaces interact with this model.

The Adobe transparent imaging model changes the dynamic of how color is derived on a page. The stacking hierarchy of objects is till important, since this object order will be used to calculate color (Adobe refers to the stacked objects as a transparency stack). With the transparent imaging model, "objects are arranged from bottom to top in the order in which they are specified. The color of the page at each point is determined by combining the colors of all enclosing objects in the stack according to compositing rules defined by the transparency model" (Adobe, 2006, p. 514). The complexity of this computation is increased when one considers that each object is drawn with an associated backdrop, and that backdrop will usually consist of other stacked objects previously defined. All the while this is taking place, color is being calculated at each unique point using a blend mode, which is a function of the object's color and the resulting object backdrop. Different objects within a stack can have different blend modes, resulting in a wide variety of color option combinations.

Within the Adobe transparent imaging model, color space is an important consideration, and plays a significant part in the outcome of the blending color space, as seen in Adobe's basic compositing formula:

Equation 1. *Adobe Basic Compositing Formula (Adobe, 2006, p. 517).*

$$C_r = \left(1 - \frac{\alpha_s}{\alpha_r}\right) x C_b + \frac{\alpha_s}{\alpha_r} x \left[\left(1 - \alpha_b\right) x C_s + \alpha_b x B\left(C_b, C_s\right)\right]$$

Where :

C_b = Backdrop Color

C_s = Source Color

C_r = Result Color

α_b = Backdrop Alpha

α_s = Source Alpha

α_r = Result Alpha

$B\left(C_b, C_s\right)$ = Blend Function

The importance of color space in this equation is stated in the sixth edition of the Adobe Systems Incorporated, 1999 Manual:

> The compositing formula shown above is actually a vector function: the colors it operates on are represented in the form of *n*-element vectors, where *n* is the number of components required by the color space in which compositing is performed. The *i*th component of the result color C_r is obtained by applying the compositing formula to the *i*th components of the constituent colors C_b, C_s, and $B(C_b,C_s)$. The result of the computation thus depends on the color space in which the colors are represented. For this reason, the color space used for compositing, called the blending color space, is explicitly made part of the transparent imaging model. When necessary, backdrop and source colors are converted to the blending color space before the compositing computation (p. 518).

The blending color space supports various device and ICC color spaces and have a unique relationship with spot colors. Blending can be done on spot colors, allowing transparency to be applied; however, spot colors are not converted to a blending space unless they are first reverted to an alternate color space (Adobe, 2006). This can produce undesirable results when attempting to reconcile transparency for the purpose of flattening. When a file that has transparent spot objects combined with not spot objects in a stack is flattened, it can be difficult to simulate the many possible result colors that will result within that blend space, since the spot colors must be dealt with separate from the blend space. Since the act of rasterization in itself can lead to loss of information (Adobe, 2006), it could be argued that flattening a file with spot color transparency has more chance of rendering incorrectly than a file that has transparency that only affects CMYK objects. This rationalization would explain why the cover test form used for this project did not output correctly when saved as a PDF/X-1a:2001 file, or as a PDF/X-4:2007 file that was flattened, since in both cases flattening occurred prior to spot color mapping. In contrast, if the same cover is saved as a PDF/X-1a:2001 file that has been instructed to convert colors to process, the spot colors are mapped to process, allowing them to be included as part of the blending space, and consequently allowing for a more accurate result. This explains why the cover, when output in this manner, rendered correctly.

An advantage of PDF/X-4, when used with the Adobe PDF Print Engine, is that transparency does not need to be resolved until final output. At the point that transparency needs to be resolved in an APPE workflow, several important aspect of the output are known and can be used to create more stable results on output. Variables such as the number of and output resolution are known and can be incorporated into the final transparency reconciliation. The research conducted for this paper proved that complex transparency stacks were reconciled accurately and consistently when that reconciliation occurred at the end of the workflow, just prior to output.

Figure 5. *Comparison of flattening results on page one of the test form. Note that both the PDF/X-1a and PDF/X-4 files rendered incorrectly when flattening was applied.*

Results of the Ghent Target Pages

The test forms used for this research included several tests from the Ghent Output Suite V.2.0. The rationale behind including these patches was to use them as a means of process control to identify and analyze the effect of workflow process configurations on the outcome of the tests.

This accomplished what the targets were designed for: identify specific workflow concerns that would have to be investigated and addressed before a workgroup could consider adopting an ICC-based process. It was beyond the scope of the TSC's research group to resolve each concern raised. Standardization on one specific "job" configuration for the software reduced variables and offered the participants a common base to investigate individual workflows separately. For detailed breakdown of the Ghent Test results please see Table A1 in Appendix A.

The different overprint modes (OPM 0 and OPM 1) in the Ghent targets refer to "standard overprint mode" and "Illustrator overprint mode." The difference is in how 0% tints of CMYK colored vector objects are handled. In OPM 0, an object with a 0% tint value in any CMYK color will produce a knockout of the corresponding color underneath it (referred to as "foreground ink wins"). In Illustrator overprint mode (OPM 1), the tint value 0 is neutral: it is ignored (Adobe Systems Incorporated, 2006). This is important for publishers to ensure that all overprints in supplied advertising material are being processed accurately.

Validity of Research Methodology

The research done for this paper utilized a draft specification of the PDF/X-4 format (referred to as PDF/X-4:2007 in Adobe Creative Suite 3) and version one of the Adobe PDF Print Engine. PDF/X-4:2008 and version two of the Adobe PDF Print Engine were not commercially available at the start of this research.

There is one major difference between the draft version of PDF/X-4 used in this research and the ratified ISO standard worth noting here. In the draft version PDF/X-4 objects could be ICC tagged the same as the final color rendering intent. This was explicitly disallowed in the final published spec, in an effort to prevent significant output differences in color management and overprint. However, the handling for ICC RGB, transparency, and spot color were not affected by the change in the specification (Dov Isaacs, personal communication, February 16, 2010).

The authors feel that the results of testing done with the draft specification of PDF/X-4 are valid because the changes between the draft and final standard have no impact on the research done or its findings.

As well, although version two of the Adobe PDF Print Engine is now commercially available, the most significant changes between the two RIP versions have to do with support for digital printing (Dov Isaacs, personal communication, February 16, 2010). Since this study was focused on conventional web offset printing, the results obtained by running test files through the first APPE should still be considered valid.

Conclusions

After significant testing and analysis, it was determined by the working group that PDF/X-4 shows significant promise to address issues with transparency handling in existing workflows. The workgroup had identified different processes at publishers for addressing files supplied with unresolved transparency; some refused the files, others “flattened” them internally, and accepted the responsibility. The PDF/X-4 files tested, when processed with the APPE RIP, overcame many of the reoccurring problems that are inherent in the rigid structure of a pre-flattened workflow.

However, the results also clearly demonstrated the need for detailed internal process reviews, before the participating organizations could support ICC-enabled workflows, especially with a blind exchange.

Based on the findings of this research, the authors feel there are two key advantages of using the PDF/X-4 format with the Adobe PDF Print Engine that should be considered in relation to magazine production and printing in general:

- Native transparency support, allowing content to remain at the highest level of extraction for as long as possible, and
- Color managed ICC workflow support, allowing printers to consider late binding workflows for multiple output devices.

When you examine PDF/X-4 and APPE separately, each has its own advantages worthy of note. PDF/X-4 has the advantage of flexibility through support for layers groups, color managed ICC workflows, and the ability to support native vector transparency. While it is true that all of these things can be done in a PDF without the PDF/X-4 specification, what PDF/X-4 adds is the reliability and accuracy of a standard. This is very important for any printer that needs predictability in files submitted. Even though a PDF/X-4 file is flexible and versatile, it conforms to a predetermined set of criteria. This gives the printer a reasonable assurance that the file will behave in a predictable manner under specific conditions, which is extremely important in a blind exchange workflow.

The Adobe PDF Print Engine is a new imaging model and can output files without the need to flatten transparency. This is a considerable advantage over traditional CPSI RIPS, especially when transparency involves complexity caused by color mapping spot colors to CMYK.

In summary, the results of this testing indicate that significant workflow advancements can be realized when the PDF/X-4 standard file format is combined with a native PDF workflow like the Adobe PDF Print Engine, when compared to current best practices that use pre-flattened PDF files and/or a PostScript imaging model when the files in question use advanced features of transparency. It also offers a potential foundation for publishers and printers wishing to explore an ICC-based workflow.

Areas of Further Study

Analysis of Processing Time

Processing speed is of obvious importance for workflow efficiency; however, the research that was completed focused on the visual results of output across multiple facilities, not individual pages or processing times of specific workflow configurations. Since the tests were done at multiple locations, it was decided that contextual variables, such as specific hardware and software configurations and capabilities, as well as relative volume (load), would limit the value of processing speed comparisons. Nevertheless, all test teams reported subjective improvements in overall output speeds when utilizing a native PDF workflow. This can likely be attributed to less file conversions, as well as resolving transparency at output, not flattening it in a conversion process. For future consideration, it would be worth developing a test to determine if a measurable difference processing time exists between different workflow/file configurations in an attempt to analyze which attributes of a file increase processing time in each of the different scenarios.

Late-binding Workflows

A workflow using PDF/X-4:2008, processed through a native PDF rendered system, addresses some of the clear limitations of a PDF/X-1:2001-based PostScript workflow; transparency output issues can be resolved as late as possible with a higher degree of predictability and accuracy.

One clear benefit of an ICC-based workflow is the potential ability to send non-CMYK data to either conventional offset or non-impact digital presses (Bailey 2005). This flexibility would help both file creators and receivers reduce the amount of communication and conversion steps required to exchange files and output accurate reproductions. This could allow for much greater versatility with regard to last-minute decisions about output devices and can make splitting jobs between conventional and digital printing much more feasible.

While publishers are in a position to benefit from this type of late-binding format, it is recommended that they explore this on closed loop editorial workflows before extending into the general supplied advertising workflow. For example, Hearst Magazines is currently using PDF/X-4 for its editorial workflow, but according to Hearst's Director of Technology Ken Pecca, using the system for advertising material at this point would be a "huge liability" (Pettas, 2008 ¶ 5).

PDF/X-4:2010 And Optional Content Groups (Layers)

The current process of handling optional content groups (OCGs) in PDF/X-4:2008 limits the file format's ability to be adopted in other workflows, especially packaging. Third-party software vendors have created solutions to extend this functionality, providing the capability to create and manipulate layers within a PDF/X-4 file. One example is pdfToolbox by Callas Software (http://www.callassoftware.com). The fact that third-party vendors are creating these tools indicates a need in industry to have control and flexibility over OCGs in PDF/X-4 files (Nias, 2009).

According to Dov Isaacs, Principal Scientist at Adobe Systems Incorporated, the PDF/X-4 standard is in the process of being updated, with the updated standard being referred to as PDF/X-4:2010. The primary difference in the updated specification is improved support for layers (Dov Isaacs, personal communication, February 16, 2010). As noted, while the existing PDF/X-4 standard is based on PDF 1.6, Adobe's current implementation of PDF/X-4 in their software is based on PDF 1.4. This limits any access to optional content once the file has been created. The updated standard is in direct response to industry feedback, specifically requesting more flexibility for layers. PDF/X-4:2010 will include selectively for layers, as well as overall updated documentation within the specification itself.

While this flexibility may serve the needs of some specific industry channels, it could prove to be at odds with the blind exchange utilized by efficient publishing advertising workflows; unambiguous layers would require direct communications between sender and receiver. Once the PDF/X-4:2010 update is available, there would be opportunity to explore how the new optional content support is deployed, perhaps through a supplier-facing web-based portal, and determine possible impact and implementation for the publishing industry.

Ghent Test Suite

The Ghent Workgroup (http://www.gwg.org) is developing an update to their test suites, which will be based on PDF/X-4. This should provide a valuable analytical tool for parties interested in evaluating the adoption of a PDF/X-4-based workflow, as well as vendors requiring a development resource. There is opportunity to research how existing workflow methodologies process PDF/X-4 files by using the new Ghent test suite to see if changes to existing parameters have to be considered to take full advantage of PDF/X-4 as a standard for magazine production.

PDF/X-VT

In January 2010 PDF/VT (announced in early 2008) passed ISO voting. The emerging variable-data market could benefit from PDF/VT (a subset of PDF/X-4) that has been designed for that purpose. Members of the development group include Adobe, Callas Software, Global Graphics, and Ricoh. Publication is expected in late 2010. As an area of further study, the relationship between PDF/X-4 and PDF/VT could be assessed, and the benefits of PDF/VT as a bridge between static and variable-data publishing for magazines could be explored.

Acknowledgments

The authors would like to acknowledge the support of the following people and organizations for their contributions to this research paper:

- Ryerson University, The Faculty of Communication and Design and the School of Graphic Communications Management, for providing equipment, facilities, and materials to complete the testing, as well as financial assistance to facilitate the presentation of these findings at the 2010 TAGA Technical Conference;
- Magazines Canada and its members for supporting the research and participating in field testing;
- Leonard Rosenthol and Dov Isaacs, Adobe Systems Incorporated, for their consultation and interpretation of the PDF/X-4:2008 specification and the Adobe PDF Print Engine;
- Nicola Kidd, Scott Millward, and Ahmed Sagarwala, Ryerson '08, for their significant contribution in designing the initial test pages used to conduct this study.

References

Adobe Systems Incorporated. 1985. *Postscript Language Tutorial and Cookbook*, Addison-Wesley Publishing Company, Inc.

Adobe Systems Incorporated. 1988. Postscript Language Program Design, Adobe Systems Incorporated.

Adobe Systems Incorporated. 1991. *PostScript Language Reference, second edition,* Addison-Wesley Publishing Company.

Adobe Systems Incorporated. 1997. *PDF for Prepress Workflow and Document Delivery*. Retrieved February 11, 2010 from http://www.adobe.com/products/postscript/pdfs/PDFps3.pdf.

Adobe Systems Incorporated. 1999. *PostScript Language Reference, third edition.* Addison-Wesley Publishing Company .

Adobe Systems Incorporated. 2003. *PDF Reference, fourth edition: Adobe Portable Document Format Version 1.5*. Retrieved October 20, 2009 from https://www.adobe.com/devnet/pdf/pdfs/PDFReference15_v6.pdf.

Adobe Systems Incorporated. 2006. *PDF Reference, sixth edition: Adobe Portable Document Format Version 1.7*. Retrieved December 20, 2009 from https://www.adobe.com/devnet/pdf/pdfs/PDFReference15_v6.pdf.

Adobe Systems Incorporated. 2007a. *A Designer's Guide to Transparency for Print Output.* Retrieved December 30, 2009 from www.adobe.com/designcenter/creativesuite/articles/cs3ip_transguide.pdf.

Adobe Systems Incorporated. 2007b. *Transparency in Adobe Applications: A Print Production Guide*. Retrieved December 30, 2009 from www.adobe.com/designcenter/creativesuite/articles/cs3ip_printtrans.pdf.

Bailey, Martin. 2005. *PDF/X Frequently Asked Questions*. Retrieved December 20, 2009 from http://www.globalgraphics.com/pdfs/products/pdfx-faq.pdf.

Hartsock, Nettie. 2007, April. Ghent PDF Workgroup announces Ghent Output Suite 2.0. *Planet PDF*. Retrieved January 24, 2010 from http://www.planetpdf.com/creative/article.asp?ContentID=Ghent_PDF_Workgroup_announces_Ghent_Output_Suite_2&gid=7487.

ISO International Standard Organization. 2008. *15930-7 Graphic technology — Prepress digital data exchange using PDF — Part 7* Switzerland, International Standard Organization.

Magazines Canada. 2009. *Production Services: dMACS*. Retrieved January 20, 2010 from http://www.magazinescanada.ca/dmacs.php?cat=dmacs.

McDowell, David. 1998. "DDAP, TIFF/IT-P1, PDF/X-1 What's It All About?" *The Prepress Bulletin,* March/April 1998.

Nias, Simon. 2009, March. "Callas claims PDF/X-4 first with layers capability in pdfToolbox." *Printweek.com*. Retrieved January 20, 2010 from http://www.printweek.com/RSS/Premedia/News/890246/Callas-claims-PDF-X-4-first-layers-capability-pdfToolbox.

Pettas, Joanna. 2008. Hearst Implements PDF/X-4 Workflow System. Retrieved November 22, 2009 from http://www.foliomag.com/2008/hearst-implements-pdf-x-4-workflow-system.

Prepressure.com. 2009. *The History of Postscript*. Retrieved February 11, 2010 from http://www.prepressure.com/postscript/basics/history/2.

Prepressure.com. 2009. *The PDF/X-4 file format*. Retrieved February 12, 2010 from http://www.prepressure.com/pdf/basics/pdfx-4.

Scribus. 2008. *PDF for Prepress Workflow and Document Delivery*. Retrieved December 4, 2009 from http://www.scribus.net/?q=faq/pdfintro.

Universal Photographic Digital Imaging Guidelines Coalition. 2008. *Universal Photographic Digital Imaging Guidelines v.4.0: Image Receiver Guidelines*. Retrieved January 21, 2010 from http://updig.org/pdfs/updig_image_receivers_guidelines_v40.pdf.