XPS FAQ

January 2008

1 Introduction

This document is designed to answer your questions about Microsoft®'s new XPS page description language. Our goal in writing this is to provide a comprehensive set of information that's useful for many different people: from those supporting or purchasing office printers, to those involved in developing or administering document management systems (and other products that store, manipulate or display documents), to those working in or developing products for professional printing environments.

Some of the answers in this FAQ are relatively technical, so we've tried to structure the document to balance the technical knowledge required to understand an answer against the level required to ask the question in the first place. In other words, if you don't know what the question means, you probably don't need the answer!

We expect that most people will dip into this document for reference, rather than reading it all sequentially. As a result, you may not see the definition of a term before you come across a reference to it. To simplify your search we've collected terms and acronyms in section 13.

This text is copyright Global Graphics Software. If you'd like to reproduce it in whole or in part; on the web, in a magazine or elsewhere; in the original English or in translation; please contact Global Graphics. Permission will usually be granted very quickly and easily for any use that will help to spread accurate information.

Thanks, and we hope this proves useful to you.

Martin Bailey, CTO, Global Graphics

(martin.bailey@globalgraphics.com)

2 Contents

 Copyright © Global Graphics Software Limited, 2007. All Rights Reserved

3 What is XPS, and where does it come from?

"XPS" stands for "XML Paper Specification". It is one aspect of the new graphics architecture for handling screen display and printing in Microsoft's Windows Vista™. Vista was released to corporate customers in November 2006, and was put on general release in January 2007.

The new architecture is designed to make it far easier to develop applications with attractive and graphically rich displays on screen, and to seamlessly translate those into the print stream and document sharing.

The design does not, however, limit XPS to only being implementable by Microsoft, or limit third party implementations to run on Windows Vista. Third party applications are already available on older versions of Windows and on Macintosh and Linux, for instance.

3.1 What components make up the new architecture?

(This section contains terminology that is useful to fully understand the reasoning behind some of the conclusions offered in this FAQ, but it is not necessary to grasp all of the technical detail).

The whole new graphics architecture is encapsulated in an API called Windows Presentation Foundation (WPF).

A WPF application uses a language called XAML in order to instruct the operating system to draw items on the screen.

Microsoft provides a framework, with supporting development tools, to generate code that makes XAML calls. This is .NET 3[1].

The terms "WPF application" and ".NET 3 application" are used more or less interchangeably to refer to programs that have been written to be compliant with WPF, which currently means almost exclusively that they've been written using .NET 3. Which word is used depends largely on whether the speaker is emphasizing the programming language used to develop the program, or the graphical functionality it provides.

3.2 Where does XPS fit into this?

The XAML language supports all of the kind of things that you need in a user interface, including animation and text that can wrap to fit within a defined area, as well as a rich set of graphical elements including images and vector graphics. There are occasions, especially for document sharing and printing, when those are not required. The capability for text reflow can actually be a disadvantage in those situations because it introduces the possibility that a document will appear differently when viewed or printed on different devices.

XPS is essentially a sub-set of XAML designed for a fixed page format. Everything that you need to represent a document containing one or more pages is retained in XPS, including all the text, graphics and some metadata; the rest has been discarded. That sub-set is then stored as an XML file, wrapped in a ZIP archive (see 11.1 for more technical detail on XPS file structure). When a WPF application prints, the XAML instructions it uses to describe the page are written into XPS by the print subsystem.

XPS is designed for use in three ways within Windows Vista:

- As a spool file format: the format in which documents are submitted to each printer driver in the print system. A printer driver can then convert that spool file format into the page description language (PDL) required by the printer, can render it into a raster (bitmap) and pass that on to the printer itself, or can simply pass the XPS directly through to a device that natively supports XPS. Note that the phrase "spool file" does not necessarily mean that the file is actually spooled as a file on disk before transmission to the printer driver.

[1] .Net 3.0 was released alongside Vista; .Net 3.5 was released in November 2007, in parallel with development of Vista SP1.

Copyright © Global Graphics Software Limited, 2007. All Rights Reserved

- As a printer command language. A number of printers are being released that directly support the XPS format in the same way that they might support the PCL or PostScript® languages.
- As a document sharing format: allowing a static, electronic representation of the document to be shared for viewing, collaborative development, proofing or remote printing.

It would be easy to conclude that only a WPF application can make or print XPS files at this stage, but that's not the case. Any Windows application may be written to submit XPS directly to an XPSDrv printer driver, and any application on any operating system can be written to create XPS files, simply by following the published specification.

For more technical details of the print system, see 11.4.

3.3 How does XPS control how a job is printed?

Alongside the XPS specification itself, Microsoft have defined a "print ticket" that is designed to carry a variety of data about how the job should be printed. It's used in combination with aspects of the Windows Vista print system that allows applications to ask for details of the capabilities of a specific printer.

The functionality controlled by the print ticket is very much in line with that offered by IPP and similar specifications aimed at office printing. The Print Ticket specification includes all of the capabilities offered by print configuration interfaces in earlier versions of Windows. Those previous versions, however tend to duplicate data structures set in different dialogs which can lead to user confusion; Print Tickets and WPF rationalize this into a more coherent and consistent structure and user interface. Many configuration items may be set for the job as a whole, for an individual document within the job, or for an individual page.

Print Tickets may also be extended by individual vendors to allow control of printer-specific features.

For office printing the Print Ticket specification and the print system around it in Windows Vista is a significant step forward. It's not, however, designed for use in graphic arts environments. For more discussion on this point see 9.3.

3.4 What was wrong with the old Windows graphics architecture?

Applications running on Windows versions up to XP and 2003 Server used a graphical interface called GDI[2] for drawing onto the screen and for printing[3]. It is not usable as a sharable document format.

GDI was designed over 20 years ago, and has proven to be insufficient for the requirements of current user interface design or printing environment. XAML calls allow a much richer and higher level access to drawing commands, making the design and implementation of graphically rich user interfaces and documents much easier.

GDI has limited graphics capabilities by comparison with XPS, and with other modern PDLs such as PDF and PostScript®. It does not support smooth gradients directly (although images or a large number of simple objects with varying colors may be positioned to approximate the effect). It also does not support the concept of either semi-transparent objects or masked images (where a single color in the image is marked as transparent).

As a result, printing of documents created in applications that offer the option to use graduations or transparency at the design stage can be very inefficient and very slow because the application must work around the limitations of GDI.

[2] Strictly, Windows XP uses "GDI+" rather than GDI, but the differences are relatively small, and GDI+ is not universally viewed as an improvement.

[3] It's commonly said that GDI is also a spool file format, which is almost, but not quite true. The GDI commands used to draw a page are written to an EMF spool file on disk in pre-Vista versions of Windows. The relationship between GDI and EMF is rather similar to that between the subset of XAML that refers to static, two dimensional graphics and XPS; one is a set of commands that can be used by a computer program to draw the graphics, the other is a file or data stream format that represents them.

 Copyright © Global Graphics Software Limited, 2007. All Rights Reserved

Perhaps most noticeably, GDI is restricted to transmitting data in RGB; it does not support the CMYK space used on many color devices, or even the single Black colorant used on monochrome devices. It definitely doesn't allow spot colors (used in professional print), device independent colors (e.g. those based on ICC color profiles) or hi-fi and photo-ink device color spaces (used in many ink-jet devices).

Many of these limitations can be overcome. When printing to devices that include PostScript language compatible interpreters, some applications generate the PostScript language to describe each page themselves instead of relying on the printer driver to do so. That PostScript can then be passed through the print system directly to the printer. This technique is widely implemented by applications used in the professional printing space, but not by the vast majority of applications designed for office use. In a similar way vendors of some photo printers provide applications (or plugins for common photo manipulation applications) that send data around the GDI print subsystem to avoid the GDI limitations. These strategies are inherently limited in that the application and the printer driver need to be written to explicitly work with each other, reducing flexibility and user choice.

Finally, the GDI interface passed responsibility for providing print configuration user interfaces to both the printer driver and to the application from which the user wished to print. That can lead to confusion. If you enter a request for two copies of a document in the application's print dialog, and another for three copies in the printer driver's dialog, you may end up with two, three or even six copies; the result will depend on the specific combination of application and printer driver. The Print Ticket support built into the XPS print system is designed to simplify print configuration both for software developers and for users, and to avoid this confusion.

3.5 Is XPS a 'standard'?

The XPS specification as it currently stands was written by Microsoft, in consultation with a small number of other expert companies. In June of 2007 they passed over rights to the specification to Ecma International, an internationally recognized standards-setting body. Ecma intends to produce a standard based on XPS (see www.ecma-international.org/news/PressReleases/Ecma%20creates%20TC46.htm). The work in Ecma is being handled by a new technical committee, chaired by Martin Bailey of Global Graphics: TC46. For progress details, see www.ecma-international.org/memento/TC46.htm.

3.6 Is XPS only relevant to Windows Vista?

The WPF/XAML/.NET 3/XPS architecture is intimately built into the core of Windows Vista. Support for almost all of the same XPS creation and viewing functionality can be added to existing Windows XP and Windows 2003 Server installations with a free download: the "Microsoft .NET Framework 3 Redistributable Package". Version 3.0 of .Net was released concurrently with Vista; version 3.5 was released in November 2007.

Microsoft Office 2007 on Windows XP, 2003 Server and Vista can export directly to XPS (and PDF) by using a free plugin downloaded from www.microsoft.com; Microsoft have stated that they will provide an equivalent for Office 2008 running on Macintosh when that ships in January 2008.

Microsoft have also made statements about XPS viewers being made available for non-Windows platforms, although they have not announced either a date or the form that they will take[4].

A number of third party solutions are already available on other platforms, including older versions of Windows, Macintosh and Linux; although to date the majority of these are not mainstream applications.

So XPS is not only relevant to Windows Vista, but it's clear that the vast majority of XPS creation in the short term will be tied to either Windows Vista or Office 2007.

[4] The currently shipping versions of Microsoft SilverLight (previously Windows Presentation Foundation Everywhere, WPF/e) for Linux and Macintosh enable XAML documents to be viewed on those platforms, but do not support XPS.

Copyright © Global Graphics Software Limited, 2007. All Rights Reserved

3.7 How might XPS change in the future?

A discussion of future changes may seem odd when the XPS specification has only recently been published and the primary tools for creating files are only just released, but there are two kinds of change that should be anticipated. The first of these is to the format itself.

One possible reason for vendors to amend the XPS that their products support would be the approval of XPS as a standard through Ecma (see 3.5). Virtually no standardization effort results in a simple rubber-stamping of a submitted specification, so it's likely that the standardized XPS will not be identical to the XPS implemented in Windows Vista, although the first Ecma version is likely to be very similar. Exactly how each vendor will update their tool sets to accommodate such differences will presumably depend on the extent of the changes required and the timing of that standardization with respect to their planned release schedule. For Microsoft, that would mean how it matches against of the following version of Windows. Ecma's schedules tend to be rapid, at least in comparison to other international standardization bodies, which may well mean that approval as a standard will fit well with the requirements of the next Windows release anyway.

Microsoft has indicated that Windows 7 (code-name Vienna) is likely to be released in 2010[5]. This kind of schedule is in the same ball-park as that of PDF which has historically been upgraded at roughly two year intervals, although that has been shortened slightly for the last revision. Of course, it's also possible to envisage new versions of XPS associated with service packs or other minor upgrades from Microsoft; it's hard to judge at this stage exactly what plans Microsoft may have.

The second way in which XPS will change over time does not require changes to the specification, but is more a result of the evolution of the XPS creation ecosystem. In the short term, a large proportion of XPS files will be created using the Vista printing subsystem from non-WPF applications. The print subsystem components are set out in detail in 11.4, but there's one simple rule that doesn't require in-depth technical understanding: printing from a non-WPF application means that documents are subject to most of the restrictions of the GDI format (see 3.4). Therefore, in the short term most XPS documents can be expected to be RGB only, and not to contain transparent or smooth shading elements. This is not to say that a reasonable approximation to the creator's intended visual appearance won't be maintained, only that the appearance will be stored in the file using different structures that do not maintain the same ability to take maximum advantage of the quality of a wide variety of displays and printers.

Some XPS files even in the short term should be less restricted because the applications creating them use direct export through their own XPS creation code, including those made using the "Save as XPS" plugin for Microsoft Office 2007. Other applications with direct export capabilities are expected to appear over time. Printing from an application capable of generating XPS directly will be able to take full advantage of all of the features of the format.

3.8 How does XPS compare to "GDI Direct" or "TrueImage"?

Microsoft has announced new page description languages and printer control languages before. In the early 1990's they launched TrueImage; a few laser printers shipped with TrueImage RIPs, but it was soon dropped.

A key message to take from the relationship between XPS and the underlying graphical architecture of Windows Vista (WPF, XAML, .NET 3) is that XPS is integral to the whole design. It may be the most obviously identifiable part of the whole to everyone other than programmers, but it's just the tip of a tightly connected iceberg. This is in sharp contrast to TrueImage, which was very much a bolt-on, and driven almost entirely within the print driver group within Microsoft.

A number of lower cost printers, mostly designed for the SoHo market, render the GDI spool file in pre-Vista versions of Vista to a raster, and then pass that raster on to the printer over the network, or a parallel or USB cable. These devices are often called "GDI direct" printers or WinPrinters. Exactly the same approach can be used under Vista by rendering the XPS spool file to a raster and delivering that to the device. See 8.3 for more on this subject.

[5] Comments at Microsoft Partner Conference, Denver, July 2007.

 Copyright © Global Graphics Software Limited, 2007. All Rights Reserved

XPS was explicitly designed from the ground up to act as a printer control language as well as a spool file format. Its use for document sharing will also serve to increase adoption rates over a wider population of users. In combination with the enhanced graphical capabilities of XPS in comparison to GDI, these issues mean that it is suitable for use in larger workgroup and production printers, where GDI Direct never achieved significant success.

3.9 What is Metro?

Metro is the code name that was used for what is now referred to as XPS for the first few years of its development.

4 How does XPS compare with PDF?

On the face of it, both XPS and PDF appear to be designed for very similar roles; both are static representations of paged documents that can be shared and then printed. This document is going to concentrate on the practical implications of the formats and the products available to work with them rather than on the question of whether the two were designed to compete.

The capabilities of the two specifications are roughly similar. Both are certainly above the threshold that makes them capable of representing the rich graphics that designers increasingly wish to use in their work, and to share and print those graphics accurately and efficiently.

The differentiating factors between the two therefore come down to the infrastructure and ecosystems around them. It's important to differentiate between what restrictions the XPS or PDF specifications themselves impose, and what restrictions are only present because applications have not yet been written to perform a function; development of new applications around XPS is expected to be rapid because of the very large market available to successful vendors.

4.1 What are PDF's advantages over XPS?

PDF is already installed on a huge number of workstations, and is used as the document sharing format of choice in many organizations. Adobe® claim more than 500 million distributions of various versions of Acrobat or Adobe Reader, although it's not clear how many of those are duplicates to the same person, e.g. in order to access successive versions, or supplied on disks for other applications to support user documentation. Of course, it's only the reading software that's free for PDF; Adobe further claim that "Acrobat has been adopted by more than 5 million users", but don't clarify whether 'user' means a single workstation, or an enterprise. It seems likely that perhaps a few tens of millions of copies of Acrobat Professional and Acrobat Standard have been sold. In addition, there are a large number of applications from other vendors that can also make PDF, including the Macintosh operating system and products based on Jaws PDF Technology. In contrast, the installed base of XPS creation tools is somewhat smaller, but growing rapidly (see 4.2).

PDF was first launched over ten years ago and a rich and mature tool set has built up around it, for collaborative editing, form-filling, printing, archiving and a variety of other needs. XPS is, of course, a new format, and that richness and maturity is therefore not yet available.

In legal terms, PDF has been accepted as a valid format for digital submission of documentation to many government agencies and private organizations. In some cases the additional specifications that such organizations layer on top of a simple request to submit a PDF file are well written and largely guarantee the delivery of a static, readable and precisely reproducible document. Other agencies have some way to go to complete their specifications, and there is a risk that files may not be usable, e.g. because a symbolic font that is vital to understand a graphic has not been embedded. An ISO standard is already available that, although primarily aimed at archiving, provides a very solid foundation for reliable document submission; see en.wikipedia.org/wiki/PDF/A. It seems probable that a number of agencies and organizations are evaluating XPS for similar submission guidelines, and XPS has a number of advantages from day 1; it's not possible, for instance, to create a valid XPS file that doesn't include all of the fonts necessary to reproduce it correctly.

Copyright © Global Graphics Software Limited, 2007. All Rights Reserved

A variety of PDF creation and modification tools and helpful white papers are available to ensure compliance with regulations requiring accessibility of data to all, such as Section 508 in the USA and the equivalent in other jurisdictions. While many individual PDF files are not accessible, there is work ongoing to develop an International Standard based on PDF to define compliance and best practices; see pdf.editme.com/PDFUA. One of the underlying technologies requirements for support of Section 508 is the ability to tag the structure of text flows, associations between graphics and text, etc. This also allows for reflow on demand, allowing documents to be adjusted for viewing on devices with very different form factors, such as mobile devices. While the XPS specification does make provision to include this data in an XPS file, tools and best practice documentation don't yet seem to be available to use it.

The PDF specification and applications built around it provide rich support for interactive features such as post-it notes, forms filling and collaborative editing. Support for these in current XPS applications is almost non-existent.

PDF viewers are available on a wider variety of platforms than those for XPS. Adobe supplies viewers for many versions of Windows; Macintosh; Symbian, Palm OS and Windows Mobile for mobile devices; Linux; HP-UX; AIX and Sun Solaris[6]. Third party readers extend the operating system support even further. At the time of writing, XPS viewers are only available on Windows Vista, XP and 2003 Server.

Finally, PDF is widely accepted and trusted in both the document management and graphic arts communities. That trust is built on years of experience, learning which tools and techniques work to give a reliable and predictable output, and which do not. Those same communities have not yet had the exposure to XPS to allow them to build the same level of trust.

4.2 What are XPS's advantages over PDF?

In contrast with PDF's widespread installed base, XPS is starting from a small toe-hold, but is likely to vastly exceed the number of copies of tools that can make PDF in very short order. Virtually every copy of Windows Vista[7] will include the capability to make XPS for free. Microsoft announced in Late October 2007 that 88 million copies had been shipped, meaning that they will come in at the low end of the 100-200 million copies by the end of 2007 range predicted by them in early January. Even so, this is reportedly twice the number of copies of Windows XP sold in the first year after its release.

Looking a little further ahead, Microsoft also predicted[8] that roughly two thirds of PCs could be running Vista by 2010[9]. This means that the number of copies of XPS creation tools in the field may pass that of PDF creation tools some time in 2009. Of course, there's no guarantee that the users of the computers in which Windows Vista is installed will use that functionality … or even know that it's there!

At the same time, every one of those machines will also have the pre-installed capability to view XPS documents, which may mean roughly equal numbers of viewer-equipped machines for XPS and PDF, with the balance moving always towards XPS over time as Vista is used on a larger and larger percentage of the world's computers.

The fact that the creation and reading tools are a standard part of the Windows platform will make it much easier for IT departments within large organizations. They will not need to install and support additional software, they will not need to track licenses, and it will be included in one of their single largest maintenance contracts, avoiding the need to manage an additional contract or option in an existing one.

---

[6] Adobe's mobile and UNIX readers lag behind those for other platforms in PDF version support at the time of writing; it appears that they have chosen to allow other suppliers such as Quickoffice to develop for these platforms, sometimes including Adobe technology in their products.

[7] As a result of pressure from the EU, Microsoft has made it possible for individual OEM partners to remove the XPS creation capability from copies of Windows Vista that they pre-install on PCs. It seems very unlikely, however, that many "stripped-down" copies, if any, will actually be sold; why would a PC manufacturer disadvantage themselves in that way?

[8] Lyra Conference, Jan 2007.

[9] That may seem a long way away, but it's well within the product life-span of most software applications and printer models that are in the planning stages of development today.

 Copyright © Global Graphics Software Limited, 2007. All Rights Reserved

XPS creation and viewing can also be installed on Windows XP and Server 2003 for free, but the number of users doing so is likely to be very small in comparison to Vista installations[10].

Being based on XML and ZIP, there is a huge array of off-the-shelf tools that can be used by developers to help build XML creation and modification applications[11]. Many of those tools are open source, and license-free. In addition, Vista includes APIs that enable relatively easy support for opening and displaying XPS files. That means that creating XPS applications is relatively cheap and easy. When that is combined with the high expected number of users, which implies a large market potential for well-designed tools, it is expected that the next couple of years will see an explosion of new products for handling XPS onto the market.

Perhaps surprisingly, XPS even has some advantages over PDF for the graphic arts market. PDF allows documents to be created where the correct appearance is dependent on a specific font being available, but that font does not need to be embedded within the file. Thus, a user viewing or printing it on a system where that font is not available will not see the page as intended. XPS specifies that all fonts used in a document are embedded within the file. That means that the file size may be a little larger, but avoids the potential for some very costly mistakes. This is one of the key requirements of specifications and standards such as PDF/X and PDF/A, which are layered on top of PDF in order to achieve predictable rendering.

A second requirement in PDF/X and PDF/A is that all colors used be completely characterized, rather than using device color spaces such as CMYK or RGB. As is implied by the term "device color space", the actual color resulting from the use of numbers in RGB or CMYK depends on the device on which they are output[12]. By 'characterizing' them, in other words, by saying which device they are intended to represent (typically by tagging them with an ICC color profile), the output can be made more predictable and uniform across multiple devices. Baseline PDF does not require that all colors are characterized, while XPS does, making it rather similar to PDF/X-3 and PDF/A in this respect.

XPS also includes support for higher precision, wide gamut color than PDF does. PDF was, for many years, limited to 8 bits per pixel[13], which can lead to severe banding in some images, especially where one flat color should blend smoothly into a rather similar one. PDF version 1.5[14] increased that limit to 16 bits per pixel[15], which would be sufficient for virtually all images, except that a vanishingly small proportion of applications have implemented this. XPS supports images at 32 bits per pixel through its inclusion of the HD Photo image format, and what is intended to be a simpler way to color-manage data in 16 bits per pixel in the scRGB color space. It remains to be seen how widely these will be adopted, but the availability of tools in the Windows Vista tool kits will make development of applications significantly easier and faster and therefore tend to move people in that direction. See also 11.2.

For the corporate environment, many of Microsoft's applications wrap very extensive digital rights management (DRM) support around XPS, by using their rights management system (RMS). This includes the potential for very flexible control of access to documents by individual, by time and by location. Much of the practical application of this potential comes through integration of RMS support into products such as Microsoft Office SharePoint Server (MOSS) 2007. It's important to note, however, that the DRM is not a part of the XPS specification itself. It is a part of the implementation decisions made by Microsoft, and it may not be possible to extend it into third party solutions running on non-Windows platforms. Baseline PDF does have some DRM support and Adobe designed their PDF applications to be extensible so a rich selection of third-party solutions are available to provide enhanced DRM. Which approach will win out

---

[10] There had been some rumors that SP3 for Windows XP might include the .NET 3 framework, which would greatly increase the proportion of Windows XP users with all of the infrastructure available for handling XPS files. These rumors now seem to have been inaccurate.

[11] Tools to render XPS for viewing or print, or for conversion to other formats, remain relatively difficult, because the process of reading the XPS component parts is only a small part of the complete process.

[12] See www.colorwiki.com/wiki/The_Color_of_Toast for a good explanation of device color spaces.

[13] 256 levels of color in each channel of an image

[14] Published alongside Adobe Acrobat 6.

[15] 64 thousand levels per channel

Copyright © Global Graphics Software Limited, 2007. All Rights Reserved

will largely depend on corporate use of products like MOSS that are tied into the Microsoft RMS for reasons other than RMS. If they are used anyway, then it is likely that DRM around XPS will be much easier and cheaper for IT departments to implement than equivalent protection for documents in other formats such as PDF.

In parallel with DRM issues, corporate clients who have selected solutions such as MOSS for other reasons will find additional benefits to XPS, such as pre-installed support for thumbnails and metadata reporting of XPS documents without requiring third-party solutions to be installed.

4.3 How do XPS and PDF file sizes compare?

XPS is based on XML, and XML is well known to be a verbose format, which can lead to very large files. Does this mean that XPS files will always be larger than a PDF file of the same document?

The XML of an XPS file is wrapped into a ZIP archive structure and in the process has compression applied to it, usually flate. Flate (like other compression formats such as LZW) works by recognizing repeated patterns of data; the more sequences are repeated within the data; the more it can be compressed. XML's verbosity is as a result of using many sequences many times, which means that it compresses very well inside ZIP. As a result, our prediction is that XPS files should be roughly equal in size to an equivalent PDF file. The result will vary with the efficiency of the tools used to build the files, both in the original XML generation, and in the flate compression. Right now, with a relatively immature tool set, XPS files tend to be somewhat larger than equivalent PDFs, but we expect that to change with time.

This is very much in line with Adobe's experience in developing Mars (see 4.7). Jim King of Adobe has said that they expect Mars files to be roughly the same size as current PDF, but that their prototype tools are currently making files a little larger[16].

4.4 How and why is file validation important?

One of the less obvious problems that some users and vendors have encountered is that a large number of PDF files that have been made do not fully conform to the specification[17]. Most of the ways in which files are non-compliant are very minor, and do not cause any problems; but some can lead to situations where software cannot read the files correctly and to finger-pointing between the vendors of the software involved. The most common tool employed by users and many smaller vendors to check a PDF file is Adobe Reader, but that was not designed as a validation tool. There is a clear business case for a reader application to display as much of the information in a document as possible without unduly worrying the user, at least outside of specialist markets such as archiving, legal or professional print. This applies even if the file does not conform strictly to the specification.

Unfortunately it appears that there is no complete PDF file compliance tool available at the time of writing, and the best commonly available tool for validation has been so poorly publicized that it's almost unknown[18].

Over many revisions of the PDF specification and versions of software to make and read PDF this has resulted in many products that make broken files. The vendors of those applications don't feel a need to correct them because the files that they make open in Acrobat and therefore "must be correct". The knock-on effect of that is that all other PDF readers must also accept those bad PDF files; to do otherwise would invite criticism from users who don't generally care about the cause of an error, they just know that they can't do what they want to be able to do, and that "the file can't be to blame because it opens in Acrobat".

[16] "The Future of PDF and Flash" class in the "PDF University: PDF for Document Publishing" course at XPlor Document University, Miami, March 2 2007.

[17] This includes PDF files created by Adobe's own applications.

[18] The pre-flight profiles in Acrobat Pro from version 7 onwards include one named "Report PDF syntax issues", which validates content streams and resources. It does not validate embedded ICC profiles or fonts, and, more importantly, does not validate the lowest level formatting of the file, because it relies on the same PDF parser as the page display code.

 Copyright © Global Graphics Software Limited, 2007. All Rights Reserved

As a result workflows are less robust because you cannot guarantee that a file that can be opened in one application will open in another.

In contrast, the XML within XPS files may be validated against appropriate schemas (made publicly available by Microsoft) using off-the-shelf XML tools. In addition, the Windows Vista DDK includes a tool named "IsXPS"[19], which is explicitly written to validate conformance to the specification. Unfortunately a reasonable proportion of non-compliance issues with respect to PDF files are due to errors in embedded fonts (especially TrueType), or ICC profiles that do not fully comply with the respective specifications. The same is likely to occur with XPS files, and IsXPS does not validate such enclosures, but it's still a good step in the right direction.

Microsoft have also made several different XPS reference renderers available: their own XPS viewers within Windows Vista and the .NET 3 framework; a second Microsoft viewer that's part of a kit called "XPS Essentials" for on-screen viewing; and the "Reference Raster Image Processor (RIP)" from Global Graphics, for printing[20]. While there are a few known issues in all of these with respect to rigid adherence to the XPS specification, they are probably in roughly the same ball-park of accuracy as the Adobe Reader is for PDF files.

For formal conformance testing Microsoft also supply an additional viewer in the Windows Vista DDK, called LooksGood. This can create raster files on disk rather than only showing documents on screen, allows the resolution of the output to be defined and allows anti-aliasing to be disabled[21]. LooksGood is based on the same core rendering code as the .Net 3 viewer.

It remains to be seen how commercial pressure to display files that do not strictly follow the specification will affect future versions of XPS products. Our hope is that the ready availability of file validation tools will reduce the number of products making bad files, and therefore allow continuing close validation by the readers. If nothing else, the independent conformance validation that they provide will simplify the resolution of disputes over the cause of compatibility problems.

4.5 What about PDF Packages?

Version 1.7 of PDF introduced a new concept; that of "PDF packages", that allow multiple files to be packaged for delivery together, whether they are formatted in PDF or not. One of those documents can be nominated to be displayed when the package is opened in a PDF viewer, making it suitable for use as a delivery note or manifest. This may become a useful method of working where a group of documents must be submitted to a government agency or other similar body.

An XPS file can be used in a similar way, because of the way it's constructed using the Open Packaging Conventions (see 11.1). In fact that ability has been a requirement for the XPS specification from quite early in the process, but applications to make use of this are not yet widely available.

4.6 How open are XPS and PDF?

There has been some discussion in a variety of forums about just how "open" PDF is, often in contrast with Microsoft's Office document formats[22].

[19] WinDDK\6000\tools\print\<*platform*>\IsXPS

[20] Available from http://www.microsoft.com/whdc/device/print/RRIP.mspx. Note that this reference RIP does not contain all of the functionality of Global Graphics' commercial products, and represents only the state of XPS rendering in those products in December 2006. Later development and optimizations mean that current commercial products are many times faster.

[21] Note that disabling anti-aliasing of text requires adjustment of ClearType preferences.

[22] At least the formats used prior to Office 2007, when the Office Open XML formats were introduced.

Copyright © Global Graphics Software Limited, 2007. All Rights Reserved

Both PDF and XPS are 'proprietary' formats, in that they were designed, and are owned (and copyright) by single, commercial concerns. On the other hand, both are about as 'open' as you can get within that constraint:

- The specifications for both have been published and are available free of charge on the respective companies' web-site.
- Both Adobe and Microsoft have made public statements around intellectual property and especially around the use of their patents for the purposes of implementing products using the formats. In this respect the Microsoft statement is slightly broader, in that it specifically grants licenses for reading and writing XPS, while the Adobe statement restricts the license for one of its patents only to vendors creating PDF files and not reading them[23].
- No fee is payable to, and no permission is required from, either Adobe or Microsoft by a third party implementing products around the formats[24].
- Both vendors have submitted their formats for adoption by international standards bodies. Adobe has been cooperating with AIIM[25] in submitting PDF to ISO[26], where it's completed a "Fast track" ballot as ISO 32000. The results indicate widespread international support, but are not totally conclusive at this time. Microsoft has submitted XPS to Ecma for standardization (see 3.5).

4.7 What about Mars?

Mars is an Adobe technology project that uses the conceptual structures of a PDF file, but encodes them using XML, ZIP and (an extended version of) SVG. Using these foundation pieces counters some of the advantages of XPS, by making it possible to develop Mars writers and readers using a variety of off-the-shelf tools, but at the cost of some loss of backward compatibility.

At the time of writing Adobe has not published an adoption path or schedule for Mars; it has been described as an exploration of a possible direction for future versions of PDF.

4.8 Is there a conclusion?

It's clear that the environments for XPS and PDF have different strengths and weaknesses, and that the implementation rate and availability of applications for XPS will change radically over the next few years.

It seems likely that both formats will establish their strongholds and that any vendor, service provider or user who is interacting significantly with multiple organizations will need to be able to handle both.

4.9 What about PostScript?

This section has concentrated on comparisons between PDF and XPS because that's something that many people have raised as an important issue. When you consider XPS use in the context of printing, however, it's equally valid to make a comparison against PostScript.

Both are good at handling graphically rich pages, including a variety of color spaces, vector graphics and images and smooth graduations. XPS extends that further than PostScript in a number of ways, however, including support for:

- transparency
- high-precision wide gamut color; often described as "high dynamic range" (HDR).

---

[23] This document is not an authoritative statement of either Microsoft or Adobe's legal notices. See http://go.microsoft.com/fwlink/?LinkId=52369, and http://partners.adobe.com/public/developer/support/topic_legal_notices.html

[24] Clearly, a fee might be payable if the third party were to license libraries or source code from Adobe, Microsoft or anyone else; the lack of a fee refers to the third party working from the published specification alone.

[25] www.aiim.org

[26] www.iso.org

 Copyright © Global Graphics Software Limited, 2007. All Rights Reserved

The standard Print Ticket support in XPS is more complete than that in PostScript. While both are extensible, the use of standard tags leads to greater portability between devices (see also 3.3 and 9.3).

Both formats are streamable, meaning that the first pages of the job can be processed and printed before the last pages of the job have been produced, assuming that the rest of the environment within which they are used can process jobs in that way[27]. This reduces "first page out" time on desktop printers and enables transactional printing.

Both formats can be generated programmatically, e.g. by using database report generators, but with different strengths and weaknesses. Postscript generation can be coded entirely within the report generator, while XPS generation typically requires access to external tools to apply the ZIP container format. On the other hand competent PostScript programmers can be hard to find.

5 Why will XPS be adopted in the corporate arena?

In many large corporate environments virtually every user has a copy of Microsoft Office (or an equivalent from another vendor). For a long time the document sharing format(s) of choice within an organization has been the native format of the authoring suite in use. Unfortunately, the use of Word documents in this way has some significant disadvantages, including the potential for changed line-ends and page breaks as text reflows for a variety of reasons.

That's not to say that it's the wrong thing to use in all cases. If you're sharing a document for collaborative editing, then sharing a format that everyone can edit is obviously the road to follow. There is, however, a significant difference between the requirements for exchange of authoring documents and submission or distribution of fixed, final form pages.

Other issues, such as the exchange of documents using fonts that are not actually embedded within the documents, are less important in a closed environment where the IT department ensures that all computers have the corporate fonts required installed. As soon as documents are sent outside the company, however, this can start to cause problems.

At a time when increased corporate and government transparency is being legislated for in many countries worldwide, there are many who believe that it makes sense for organizations to standardize on a reliable exchange format for document sharing, and to use that for the majority of documents, whether they are initially intended for external visibility or not.

The question then, is: which format?

Until recently, the clear choice was PDF, or perhaps a specification layered on top of PDF such as PDF/A. With the release of Windows Vista and the availability of XPS, the question is thrown open again; possibly even for those organizations that are some way down the road in implementing PDF adoption.

One of the important factors is that XPS has the potential to significantly reduce the load on the IT department, because the capability for XPS creation and viewing will be supplied pre-installed on new computers. There's no need to perform additional installations or upgrades, to track software licenses or to manage maintenance contracts.

XPS creation capabilities can also represent a significant cost saving. While Adobe Reader is free, PDF creation is not (at least through the products and services that a corporate or government customer is likely to consider). Even with volume discounts on some of the less expensive PDF creation tools, an installation of tens of thousands of workstations is not cheap.

On the other hand, PDF technology is available across more platforms, and only a small proportion of workstations worldwide will be converted to Windows Vista in 2007. Installation of .NET 3 on Windows XP or Server 2003 is another installation for the IT department to do, albeit one with no license to track and no purchase fee to be paid.

In addition, while both PDF and XPS are more than adequate to provide a reliable and efficient exchange format for static documents, the PDF specification and the applications available around it are currently

[27] XPS must be deliberately created as streamable in order to be used in this way.

Copyright © Global Graphics Software Limited, 2007. All Rights Reserved

better placed for interactive features such as forms filling, and for collaborative development and approvals of documents.

As a result, it's likely that different organizations will make different decisions, with some opting for PDF and others for XPS.

Even a partial penetration of the corporate document management space will result in a huge opportunity for software vendors. It is likely that the gap in interactive and collaborative capabilities between PDF and XPS applications will close within a few years as new XPS-based products are released.

As a result, there will be pressure on government agencies to accept XPS files whenever digital document submissions are allowed. Some agencies (especially in the USA) are already well down the road in accepting PDF, while others have yet to make a real start on their procedures. Those who have not yet started may find it easier to accept a file format that they can read and print without needing to buy additional software. That's doubly true when it's likely to be slightly more reliable out of the box; for PDF the agency must define additional restrictions in order to avoid receiving files that do not include all of the fonts required, for instance, while valid XPS files already meet that goal. Agency-imposed restrictions like that one require more expertise at the agency to define in the first place, and incur significantly more support costs to explain to those who submit files that do not follow them. If you can just say "send me an XPS file" rather than "send me a PDF file, but remember that you must make sure that the 'embed all fonts' check-box is ticked'", there is a clear win for the agency.

Away from their own internal requirements, many organizations send out documents to their customers, prospects and constituents, including brochures, statements, white papers and product manuals. It's now very common to find a PDF file on a CD instead of a printed manual when you buy a new piece of equipment or software program. Somewhere in the package there are usually instructions as to how to open the PDF file, and how to get a reader for it if you don't already have one. Sometimes those instructions are printed, but often (especially when provided with software) they are in the form of an HTML page. So why do you need instructions telling you how to open a PDF file, but not on opening the HTML file that tells you how to open the PDF file? It's because virtually every computer already has a web browser installed that will correctly display the HTML when you double-click on it, but the publisher cannot rely on the reader having an up-to-date PDF viewer.

As Vista spreads across the marketplace, it will be more and more reasonable to assume that people will have an XPS reader. Expect to start to see user manuals and brochures made available both in PDF and XPS initially, and then to drift slowly but surely towards delivery only in XPS.

6 How should a vendor implement XPS in a document management system?

The clear implication of section 5 is that document management system vendors should be evaluating when and how they might support XPS within their product ranges.

Such systems fall into two broad categories:

- Systems that store documents in a variety of formats, and therefore provide functionality that is applicable to all of them, often with the ability to convert between PDLs for onward delivery or for printing.
- Systems that convert all documents on import into a single format, thus limiting the incremental work required for each new format to an import component.

Both types of system require conversion technology; the first also requires a variety of analysis technology, which may include generation of thumbnails, page previewing, metadata management, text extraction & indexing, search or modification, e.g. to add page numbers.

Since XPS is based on XML and ZIP (see 11.1), the first steps of developing this conversion, analysis and modification functionality are relatively easy for competent programmers. The difficulty of providing full support for XPS varies with the functionality required and may require some in-depth graphics expertise, for instance to convert from XPS to a more limited PDL such as PostScript while correctly flattening any transparent elements.

Tools to assist in this process are available both from Microsoft and from third-party vendors.

 Copyright © Global Graphics Software Limited, 2007. All Rights Reserved

7 How does XPS fit into office printing?

Windows Vista supports both the GDI and XPSDrv print subsystems[28] (see 11.4 for more technical detail). Many (but not all) printer drivers from previous versions of Windows will continue to work under Windows Vista.

Exactly how GDI and XPSDrv printer drivers will compare to each other depends on whether the program being printed from is capable of making XPS for print or not. All WPF applications can do so, but other applications can be written to make XPS for print as well[29]. Printing from an XPS-capable application to an XPSDrv printer driver should be fast, and has the potential to achieve higher print qualities, especially for photo printing, smooth gradients and transparent objects.

When printing from a non- XPS-capable application the output quality achievable is largely limited by the restrictions of the GDI specification (see 3.4); it makes little difference whether the printer driver is GDI or XPSDrv.

In any environment where XPS files are shared for viewing users will sometimes want to print them as well. If that's done by opening the file in an XPS viewer and printing from there, then both GDI and XPSDrv printer drivers will work more or less equally well, unless the document includes graphical elements that cannot be represented in GDI, as discussed above. In many cases, however, it may be desirable to automate the printing of documents by submitting them directly to the printer, rather than by opening them in a viewer first. In order to do that efficiently the printer must use an XPSDrv printer driver.

Printing from Windows versions before Vista can sometimes be confusing, as decisions about device configuration can be set differently in the print dialogs for both the application and the printer driver. Under Windows Vista, this is much improved, and WPF applications unify the data displayed to avoid confusion.

8 How should a vendor implement XPS for office printing?

8.1 What are the resource requirements for XPS?

It's easy to consider XPS as just another PDL when considering how it should be implemented in an office printing environment, but it does have some unusual characteristics that you should understand before designing a printing solution.

Over time printer control languages and PDLs have become more and more graphically rich. PCL 4, 5 and 6 are relatively simple. PostScript raised the bar somewhat, especially PostScript LanguageLevel 3, adding support for a wide variety of font types and color spaces, and for smooth graduations. That is as complex as any PDL widely implemented within embedded printer controller boards.

PDF includes roughly the same complexity of fonts and color spaces, together with smooth shading as PostScript. It is simpler to the extent that PDF is a static representation of a page, without requiring a complete programming language to be implemented, as it is for PostScript. On the other hand, general handling of PDF documents requires that the whole file be available before starting to process it, which often implies a need for a hard disk on the controller.

Recent versions of PDF also include support for transparency, which makes significant demands on memory and processing power in order to composite multiple graphical elements together in often very complex ways.

The benefits of native PDF processing on the controller board in a printer are relatively low. When printing from a Macintosh running MacOS/X the native format within the print stream is PDF and that can be sent directly to the device, without additional conversion. On the other hand, Macintosh users make up a very small proportion of the printer marketplace, and there is no standard delivery mechanism for

[28] As do Windows XP and Server 2003 when the .NET 3 framework is installed on them

[29] Vista Photo Gallery is a case in point; a non-WPF application that nevertheless creates XPS for print directly, primarily to enable it to take advantage of the support for HDR images in an XPS print path.

Copyright © Global Graphics Software Limited, 2007. All Rights Reserved

desktop printing of PDF from a Windows computer. A printer driver would need to deliberately convert from GDI (or XPS) in order to send PDF to the printer[30].

The relatively small benefits of native PDF support on a printer, combined with the relatively high costs of doing so mean that native PDF support is not common. Some vendors have provided utilities that allow a user to send PDF files directly to the printer. Other vendors have provided printer drivers that create PDF, but those are almost exclusively intended for creation of a PDF file on disk, rather than sending it direct to a printer.

The graphical complexity of XPS is roughly comparable with that of current versions of PDF, although it has one advantage in that it is easier to make streamable XPS files so that the printer controller can consume and render them without requiring that they be entirely available before starting. This means that XPS first-page-out time is reduced in comparison to that for a PDF sent over the network to the device, but also means that native XPS support does not necessarily require a hard disk in the printer, while PDF does in order to handle all but the smallest files.

A PDF implementation more or less requires that the printer vendor supplies fonts for use when a file is submitted that does not have all of the required fonts embedded in the document itself. In contrast, an XPS file always has all of the fonts it needs embedded, so a printer that supports only XPS does not require any font licensing[31].

So the cost of an XPS implementation is likely to be a little less than that of a PDF implementation; the big difference is in the benefit. Because XPS is the format used within the new print system in Windows Vista, and a very easy way to take advantage of the increased quality potential of prints from WPF applications, there is a high demand for printers that can process XPS directly.

In terms of resource usage, full support for XPS will require more memory and slightly faster processors than PostScript in order to run at the same speed. Fortunately this can be offset when the XPS is provided direct from an XPS printer driver by sharing the processing between the printing workstation and the embedded controller; see "scalable consumption" in 8.3 below.

8.2 Which print subsystem should I use?

Windows Vista supports both the "Version 3" GDI printer drivers that have been used in Windows for the last 20+ years, and new "XPSDrv" printers that are built around the capabilities of the XPS format (for more details, see 11.4). A printer vendor therefore has to answer two questions, each with multiple choices, to determine how to support a printer under Vista. The first is whether to implement an XPSDrv driver, or a GDI (Version 3) one?

- *Continue using the GDI printer driver*

  The printer will work as well as it used to in the past, but will not take advantage of the opportunity of increasing the quality of the output when used with XPS-capable applications.

- *Implement an XPSDrv printer driver alongside a GDI printer driver*

  This route allows the printer to run with maximum efficiency, by using the XPSDrv printer driver with XPS-capable applications, and the GDI driver with non- XPS-capable applications. The cost of that efficiency, however, is user confusion (and therefore increase support costs): which printer driver should they install, and how do they decide which to use when printing from different applications?

[30] Conversion from GDI to PDF would never lead to a PDF file containing live transparency, of course, but then there's no real benefit over the use of PostScript.

[31] This benefit is reduced if the XPS is implemented alongside PostScript and/or PCL, of course.

 Copyright © Global Graphics Software Limited, 2007. All Rights Reserved

- *Implement only an XPSDrv printer driver*

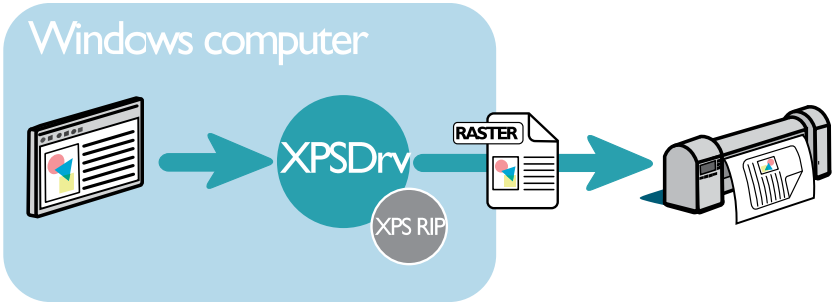
  The printer will take maximum advantage of the print quality achievable when printing from XPS-capable applications, and will do so at maximum speed[32].

One important non-technical issue related to this question is which of the routes described above is compatible with certification for the Windows Vista Premium logo program. Until June 1st 2007, a printer could be logo compliant if it shipped with only a GDI printer driver. It must now have an XPSDrv driver to conform to the logo requirements[33]. In both cases, any XPSDrv print driver supplied must comply with a number of additional requirements. See www.microsoft.com/whdc/winlogo/hwrequirements.mspx for the official Microsoft documentation.

8.3 How should my printer driver interact with the printer controller?

The second of the two questions, is how the printer driver should interact with the controller board embedded within the printer itself. This decision has no bearing on whether or not the printer and its associated driver are eligible for the Windows Vista Premium logo.

- *Render in the print pipeline*



When driving a printer with a relatively low print speed it is common that the printer driver render the pages to a raster that can be streamed to the controller within the device itself. This approach is used most often for SoHo devices that are directly connected to a single computer (e.g. over USB), but is becoming more and more common for network connected devices intended for use by small workgroups as network bandwidths increase and price pressure builds in that sector. The major advantage is in cost; a controller that accepts only pre-generated raster data can be built more cheaply than one that includes a renderer for one or more PDLs. The disadvantages are that rendering on the host computer may slow down the user interface unacceptably (increasing the "return to application" time), and that raster transmission over networks can be slow, or can reduce the remaining available bandwidth to too great an extent in a shared environment. It is these disadvantages that restrict the host renderer model to relatively low speed devices.

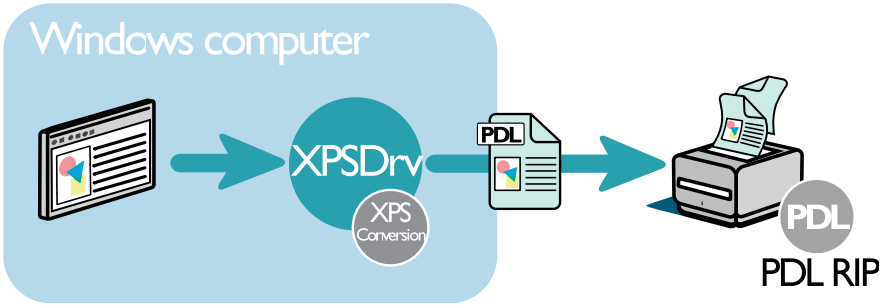
The construction of a complete sequence of operations before and after rendering in the print subsystem can be done in an XPSDrv printer driver even more easily than it can for GDI because of the ease of constructing pipelines of filters. On the other hand many GDI printer drivers make use of the Microsoft-supplied GDI renderer in order to rasterize pages; Microsoft has not supplied an equivalent XPS renderer for use in XPSDrv print drivers, although they are available from third-party vendors.

[32] It's likely that a vendor would wish to provide a GDI print driver alongside the XPSDrv one even in this case, for use on earlier versions of Windows. A clearly documented selection of which printer driver to use for each operating system, or an installer that can make that decision automatically, avoids the user confusion described above.

[33] Requirement IMAGING-0010 in the Windows Logo Program Device Requirements documentation

Copyright © Global Graphics Software Limited, 2007. All Rights Reserved

- *Convert to another PDL*



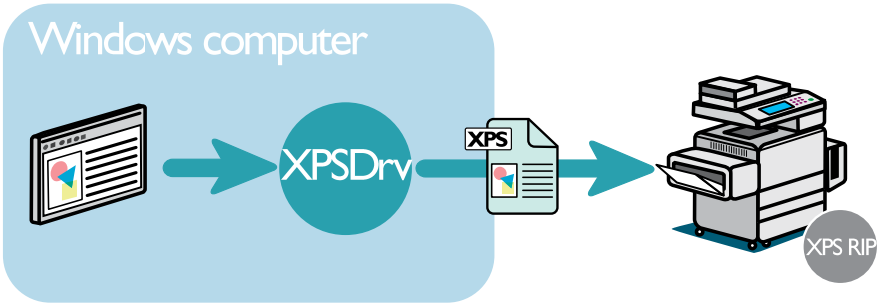
When used with a printer that supports one or more PDLs such as PostScript or PCL on the controller, the printer driver can convert from XPS to one of those PDLs and then transmit the result across the network to the printer. This can be a very efficient way of implementing XPS for devices that are already installed, or for devices that are still in development, but that are advanced too far to allow the addition of native XPS interpreting.

The disadvantages of this approach are that XPS is far richer than most pre-existing PDLs (see 11.1) so a naïve conversion may be somewhat inefficient or may occasionally introduce unacceptable artifacts. A conversion of an XPS document containing transparency, for instance, requires flattening to an opaque representation in order to convert it to PostScript or PCL. This solution is probably most useful for those situations where the controller within the device cannot be easily amended.

One approach to generating PDLs from printer drivers that has been used in the past has been to rasterize the whole page, and to send that wrapped as the desired PDL. This can be used for PCL, and PostScript, amongst others, and can sometimes be a useful short-term approach to minimize development time if pre-written components can be re-used. In the longer term consideration should be given to the potential short-comings of this method, which include:

- increased load on the workstation (increased return to application time)
- increased bandwidth requirements on the network (slowing print delivery, or reducing performance for other users)
- increased resource requirements and cpu load on the controller board (increasing board costs or reducing print speed)
- the difficulty of accessing interactive PDL features on the device, such as use of forms.

- *Render XPS natively within the printer controller*

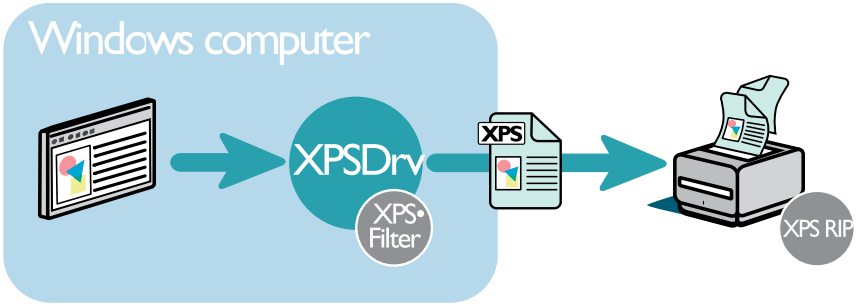


A third option is to include support for XPS rendering on the controller board within the printer itself, either on its own, or alongside other PDLs. This method ensures that the printer driver is only required to perform the minimum processing, thus ensuring rapid return to application for the user. It also ensures maximum advantage can be taken of the potential for increased print quality when printing from XPS-capable applications; and enables other delivery routes to the printer, such as driverless printing, output from memory cards and USB drives and pull printing.

Copyright © Global Graphics Software Limited, 2007. All Rights Reserved

A potential disadvantage is that XPS is a richer PDL than those with which most printer vendors have had prior experience. It may, therefore, require more resources in terms of memory or processing power on the embedded controller, which may translate into a higher cost.

- *Render XPS natively, with assistance from Scalable Consumption*

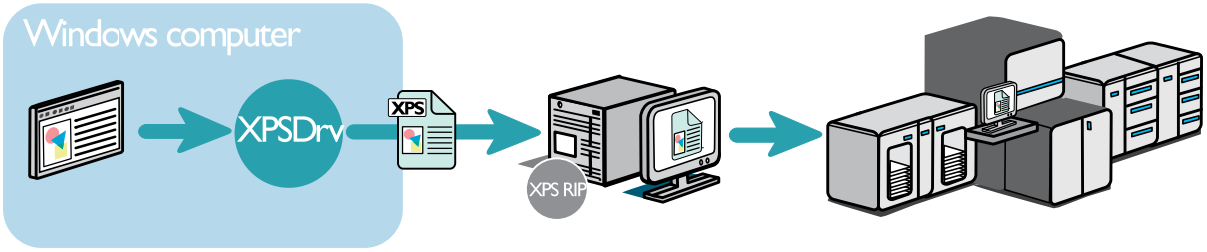


This option allows the printer vendor to take the advantages of native XPS rendering on the printer controller board, while greatly reducing the risk that a controller with resources at the level required for PostScript or PCL might fail to print a complex XPS file (see 8.1 for more on the resource requirements for XPS).

A component is included within the filters in the XPSDrv print pipeline that is capable of recognizing structures that the print controller may not be able to handle. That component simplifies them, while retaining the same visual appearance. Thus, a scalable consumption component could selectively flatten graphical elements that use transparency in an XPS document, and reconstruct the document using the flattened elements rather than the original ones.

Many printer vendors sell a variety of different devices for which the optimal selections include several, or even all of these options; perhaps their SoHo devices will render within the printer controller, while mid-range networked office printers will use scalable consumption and their larger convenience devices will include hard disks and sufficient RAM to handle XPS rendering completely within the controller. Any devices already in the field, or those that will be launched within the next year and are therefore too far down the development path to have XPS added on the controller, may be supported using conversion in the printer driver.

Some printer vendors also sell digital production presses and/or wide format printers with a server-based rendering solution or DFE which should also be considered when defining an overall strategy with respect to XPS. In these cases the DFE is likely to migrate towards native XPS support (as shown in the diagram below) in parallel with continued existing support for other PDLs. In the short term conversion in the print driver to another PDL that is already supported by the DFE may be a suitable approach (see above).



Selection of a single technology for all of these devices may be helpful in maximizing output consistency between them while minimizing development resource requirements.

8.4 How does XPS support influence product differentiation?

Vendors considering the implementation of any new technology, such as XPS, must usually consider how doing so will influence their ability to differentiate their products from those of their competitors.

In the short term the simple fact of supporting XPS will be an advantage for some sales, but it will, over time, become an expected part of the product comparison charts. In other words, lack of XPS support in certain segments of the market will become an active disadvantage.

Copyright © Global Graphics Software Limited, 2007. All Rights Reserved

When it comes to how XPS is supported, clearly there will be variations in the speed of implementations and in the print or display quality achieved, in terms of color fidelity, smoothness of graduated tints, etc. As a well defined specification, for which validation tools are readily available (see 4.2), compatibility is another defining goal, and anything that deviates from that compatibility will often (quite rightly) be regarded as inferior.

Product differentiation will continue to be achieved through value-add functionality, such as scanning, authentication, accounting links etc.

9 How and why will XPS be adopted for professional printing?

The professional printing industry is in a state of transition at the moment, expanding out from the use of conventional presses with plates to add the capability of digital printing, with a variety of hybrid solutions. Some sectors of print will continue to use conventional presses, whilst others are in the process of switching to digital output.

At the same time communications and publishing in general are no longer the almost exclusive domain of print; additional channels appear to be added almost every day. Publishers, media buyers and advertisers need to consider all of those channels and to select the balance that will work best for them.

As a part of this the traditional print company is moving towards becoming a more general communications fulfillment or full service provider. The interface between client and provider is becoming closer to increase efficiency, and in an attempt to avoid a client switching providers.

It is in this context that the advent of XPS must be evaluated.

One of the effects of XPS (and the underlying WPF and XAML) is that it makes it far easier for a software vendor to create applications that allow the user to design and create very rich graphical documents. It has been said that WPF actually makes it harder to design an application with an old-style user interface than a rich one, and that richness will feed through directly into the documents created by those applications.

In addition, the XPS specification requires that all fonts are embedded within an XPS file, and that all colors are characterized (e.g. by tagging with an ICC profile). In many ways this places it on a par with PDF/X-3 in terms of reliability of rendering[34].

This means that documents created on the desktop, using tools that are either supplied pre-installed with the operating system, or that can be installed for free, are likely to become both graphically richer, and yet may be printed more reliably than has been common in the past. It increases the potential level of sophistication in a standard desktop print pipeline to near commercial print quality.

The same document format may be used across the board in a large organization, from spreadsheets to letters, to items that will be printed professionally, including manuals, newsletters and brochures.

9.1 How will corporate use affect XPS adoption in the graphic arts?

Section 4.9 discusses how and why XPS is likely to be adopted in a large organization. This section will make the assumption that at least some will decide to standardize on XPS in their corporate document sharing systems. That leads to an assumption that the average office user will be able to print XPS files from their desk (see 7).

But it's as we move away from the desktop that the implications for graphic arts start to appear, probably in two steps:

- *CRD and in-plant*

  Once the organization's document management system is storing large numbers of XPS documents it's inevitable that the corporate reproduction department (CRD) or in-plant print department will be asked to print them whenever a larger number of copies are required. Initially that may be done using some of the same methods that printing from the desktop would use, such as by opening the files in a

[34] This does not necessarily translate directly into printability; total area coverage, image resolution, type size etc may still be outside the capabilities of any particular print process, just as they could be for a PDF/X file.

 Copyright © Global Graphics Software Limited, 2007. All Rights Reserved

viewer and printing from there, but the constraints on costs and staff time will lead inescapably to a desire to handle files in a more automated way.

That desire will lead to pressure on the vendors of document management systems (especially those directly related to print management) and of light production printers and digital production presses to process XPS files within those systems directly. A first step may be for those systems to be able to convert from XPS into a PDL that they already have the capability to process as required, perhaps to PDF. Native handling of XPS within those products is, however, likely to lead to increased performance and fidelity.

- *Copy shops, print-for-pay and commercial print*

  From time to time a CRD will need to send out jobs to an outside print service provider. That may be for overflow work or a job that needs to be processed by distribution and print rather than in-house printing, perhaps for marketing collateral to be printed near a large exhibition to avoid transporting it long distances. In those instances where the CRD or in-plant is actually a division of a print service provider operating on-site under contract this will be even more likely.

  If the customer usually prints XPS documents in the CRD, there will be a natural desire to send exactly the same XPS documents to the external print provider … and an expectation that any print provider who wants repeat work from the same client will be able to process them.

  If we assume that products will be extended to include XPS processing for use in CRDs and in-plants, then that same functionality will also usually be available for all other marketplaces into which the same product is sold. Initially that's likely to apply to those print providers that use the same equipment as CRDs, which means copy shops, print-for-pay and commercial printers using digital presses and small-format offset presses.

  Of course, one of the reasons for outsourcing print may be to gain access to equipment that the CRD does not run itself. That may include wide format printers for point of sale or exhibition graphics, or larger offset presses for longer run materials, amongst others. The availability of products to process XPS in a CRD may not directly translate into availability for these devices. Many print vendors, however, source key parts of software within their systems from OEM providers, which means that development for the CRD marketplace is likely to lead to a shorter lead time on availability for other devices once demand starts to rise.

9.2 Does XPS make it easier to handle Publisher and PowerPoint files?

Many copy shops and small commercial print companies receive a proportion of their work as Word, Publisher and PowerPoint files, created using Microsoft Office. Despite some undoubted strides in the quality of output from Publisher over the last couple of versions, it can still be difficult to prepare these files to make plates from them.

All Microsoft Office 2007 applications, including PowerPoint and Publisher, can save files as XPS by using a "Save as XPS" plug-in that's available for download free from Microsoft's web site[35]. You might therefore think that customers using those applications would be able to create files for submission to the print provider that are guaranteed to embed all the required fonts, and to ensure that all colors are characterized and therefore ensure comparable results across a variety of printer technologies.

Unfortunately, customers who design work with PowerPoint and Publisher tend to be relatively unsophisticated in relation to printing and computer use. There is also a rather small overlap between these customers and the designers in large organizations who may have a formal training program for XPS creation because of a decision to use XPS as the document sharing format of choice. As a result, it remains more likely that such clients will continue to supply files as PowerPoint and Publisher documents, with the concomitant issues around supply of the fonts used.

[35] Microsoft's initial plan had been to include the functionality in the base install, but that was changed in response to comments from Adobe.

Copyright © Global Graphics Software Limited, 2007. All Rights Reserved

Even worse, there is no option to save as XPS from Publisher while still retaining colors in CMYK or spots, even if the document has been created in that color mode. The XPS file created will always be in RGB (the other Office programs, such as Word and PowerPoint, can only create documents in RGB in the first place).

Fortunately, alongside the "Save as XPS" plug-in for the Office programs, there is also a "Save as PDF" plug-in. Unlike saving to XPS, "Save as PDF" offers the option to save the document for commercial printing, and selecting that will result in a composite color PDF file that retains the CMYK and spot color data, although correct use of this option is still far from easy.

Many of the applications in a print provider's workflow will already process PDF files perfectly well, so it's clear that the "Save as PDF" plug-in offers significant advantages over saving as XPS. The print provider must take a little care still, because that only saves as a baseline PDF file, without guaranteeing that all fonts be embedded, and without including characterization data for the colors.

9.3 How should XPS job tickets be handled in prepress?

One of the differences between office printing (direct from the desktop to the printer) and graphic arts work (with an operator or technician specifically tasked to do print work) is in how job configuration information stored within a job should be handled. A PDL such as PostScript, for instance, can include instructions to set the resolution at which output is produced, or the screen frequency to be used for halftones. In a graphic arts environment those instructions can be useful … but usually only if they are originated at some step in the print provider's own workflow. Acting on settings included in print jobs by the customer who provided the job often leads to processing errors or unusable output.

As a result many pre-press workflows, including RIPs and DFEs, have a variety of switches that allow specific pieces of configuration data to be ignored, depending on whether the print provider's own workflow will be used to set each one, or whether the customer settings could reach the output device.

The same will inevitably be true when processing XPS files, but to an even greater extent. The primary focus of the configuration settings defined in the XPS Print Ticket specification is office printing. Thus duplexing, page rotation and stapling are well represented, but items aimed at graphic arts production are not. Even where Print Ticket items address issues that are relevant for graphic arts workflows, such as color management, it would be dangerous in many cases to allow those into a production workflow, because they were almost certainly set to control a printing device at the customer's site rather than the production press.

Until more experience is gained with the Print Ticket items seen within XPS files submitted to prepress departments it's probably safest to ensure that most of them are completely ignored by prepress workflows.

9.4 What other implications does XPS have for pre-press capabilities?

Amongst the other transitions occurring in the graphic arts at the moment is a slow drift towards submission of jobs in device independent color spaces, often referred to as "three color workflows". This is being driven by a number of factors, including smaller file sizes, a desire to use the same image data across multiple publication channels (many of which are primarily RGB), and a desire to shift the responsibility for correctly separating work to CMYK nearer to the press.

As more and more graphic arts establishments add digital presses alongside conventional ones these companies are slowly becoming more confident that color management can produce the desired output, because digital presses are usually driven with color management enabled.

The remaining uncertainties around color management for production work are largely down to inconsistencies between the way different applications embed color management data within PDLs, and the way that different prepress systems interpret that color management data. Over the last few years considerable effort has been put into the development of specifications that clarify what applications and prepress systems should do. The most successful of these to date has been the PDF/X-3 standard.

In a workflow where jobs are submitted to the print provider as PDF/X-3 files, and where those are processed in the production workflow as specified in the standard, predictable color is being achieved

 Copyright © Global Graphics Software Limited, 2007. All Rights Reserved

consistently and reliably. The most important weakness in this approach is that a PDF/X-3 file can usually be processed without obvious warnings or errors through a prepress workflow that is designed for baseline PDF, and that does not conform to the requirements of the PDF/X-3 standard. As a result the output may not match the designer's requirements, often in subtle ways that may evade detection in the print provider's quality assurance procedures.

The technical capabilities of the XPS and PDF/X-3 specifications are rather similar in this respect; all colors used in the file must be fully characterized. The main technical difference is that PDF/X-3 is slightly more complete in that the tone and gamut compression and black generation settings used by the file's creator are passed to the print provider for use in final production. This should lead to an even closer match between the customer's pre-submission proof and the final printed piece. On the other hand, there is less ambiguity in an XPS workflow; there is no 'baseline' file format to confuse it with.

Thus PDF/X-3 has the potential for a marginally better match to the client's expectations when used in well controlled and configured workflows. XPS, on the other hand, is less likely to go seriously wrong when the workflow is slightly less well controlled. One is a single malt, while the other is a good blend!

9.5 When will XPS be used for high-end print?

The high cost of mistakes in printing work such as Fortune 500 annual reports, or magazine ads in Vogue means that the printers involved in such products are naturally cautious and conservative. Customers and ad agencies are equally aware of the risks, and normally follow the advice of their print partners in selecting the file formats for delivery. It's unlikely that we'll see artwork for this kind of printing delivered in XPS in the near future because a change in the choice of format, however good the new format might be, carries higher risks than staying with methods that have been proven in practice

This contrasts starkly with less high profile print work, where managing costs and customer retention are the highest priority for a print provider. Keeping costs to a minimum would normally work in favor of a conservative attitude to file formats. When a large customer says they want to deliver something different, however, the printer often has no choice but to comply, and may gain a short- to medium-term competitive advantage by proactively supporting such formats.

9.6 Is XPS useful for the data center and transactional printing?

Just like the graphic arts, data center printing is in transition. There are increasing demands for higher quality and richer graphics, including full color. Driving factors behind this include increasing use of web-to-print solutions for personalized output, and the desire to develop "transpromo" by carrying advertising in documents such as statements that would previously have been tightly focused on their primary purpose.

As a result much of the work that would have historically been performed in a data center is now tending towards the use of similar equipment to CRDs and in-plant print groups. In measuring market sector growth many commentators don't even regard this as being data center printing at all, but record it as falling within the CRD, or use an entirely new category instead.

One aspect of this kind of change must be to evaluate the data format used to transmit data from the database to the printer's digital front end (DFE).

Historically that has often been AFP and metacode for the longest runs, with some use of PPML, proprietary meta-content formats, PostScript and PCL in short- to medium-runs. AFP and metacode do not provide the graphical richness that more and more clients are demanding, however, without the use of pre-rendered graphics or references to files in other formats such as PDF.

More recently some test runs have been done using PDF in order to carry rich, full-color graphics on statements. Those test runs were reported as successful, but great care must be taken to use PDF in this way. As a general rule the whole PDF file must be available in order to read it, which means that it's very

Copyright © Global Graphics Software Limited, 2007. All Rights Reserved

hard to stream, that is to start rendering and printing before the whole file has been produced[36]. Some have suggested a "chunking" approach, where a series of small PDF files are produced. Each chunk would then be printed in turn, allowing processing of the first page without waiting for the very last page of the whole job to be generated.

A second complication with using PDF in this way is that it's not based on any standardized underlying format and therefore needs specialized tools in order to create it. That requires some significant effort to put together appropriate software to produce PDF from a database report generator.

A number of other formats could conceivably be used in this context, perhaps the most obvious being SVG (probably placed through PPML). That provides for rich vector graphics in full color (albeit only in sRGB), and allows references to external files for inclusion of images. SVG is based on XML, which means that a combination of database report generation and off-the-shelf tools makes it relatively simple to construct SVG on-the-fly. Unfortunately, almost all SVG printing products involve a conversion to PDF; this can be a useful way to make PDF for printing, but only at the cost of requiring the chunking approach described above.

Very similar issues affect the use of other XML-based formats such as XSL-FO, although there are a small number of products using XSLT and XSL-FO directly, allowing use of custom XML grammars for printing.

XPS has many of the same advantages as SVG, in that it's based on widely used standards and specifications for which a large number of off-the-shelf tools are available (see 11.1). This means that data center staff can prepare database report generation and other programmatic file creation systems relatively easily.

In addition XPS can be constructed in such a way that it can be streamed from the producer to the consumer without too much trouble; in other words, so that the first pages of the job can be printed before the XPS for the last pages is produced.

Printing solutions for XPS will be appearing anyway for a variety of digital production presses as a result of the requirements of CRDs, in-plants and print-for-pay establishments (see 9.1). It's a relatively small step from there to believe that DFEs supporting XPS will also appear for transactional printing, given the apparent suitability of the format for that use.

If your choice of digital production press for transactional printing does not support XPS natively, you can still migrate the print job generation phase toward XPS and convert from that into PDF, PostScript, PCL or whatever format the print DFE can accept. This approach also enables the use of common job generation code for a variety of different job types, and when driving a range of different presses with different PDL capabilities, but at the cost of an additional conversion before the job is rendered, and of reducing the ability to stream the job smoothly.

9.7 What about personalization and other variable data?

Another current market in transition, personalized print finally seems to be growing towards the potential that has been predicted for it as organizations achieve adequate databases of information and the willingness and understanding to use it.

Just like transactional and data center printing (see 9.6), the choice of data format to pass from generator to consumer is very important, but this is one area where no clear leader has emerged, despite the development of PPML which was intended to allow a simplification of the choice.

The advantages of XPS for variable data print are very similar to those for transaction streaming:

- Support for rich graphics (both vector and image)

[36] The PDF specification does include sections that appear to have been designed to allow PDF files to be streamed if they were specifically created with that in mind. That option does not appear to have been taken up at all widely, though, perhaps because of the lack of streaming PDF creators?

 Copyright © Global Graphics Software Limited, 2007. All Rights Reserved

- Relatively simple programmatic construction because of the underlying ZIP and XML structure of XPS
- Support for streaming

Also, just as for transactional work, a workflow for variable data can leverage the richness and relative ease of implementation of XPS either by the use of native XPS support in the production press' DFE, or by conversion to a variety of other PDLs.

9.8 Will XPS have any impact on kiosk printing or web-to-print?

Kiosk printing can only be a success if it is very easy for an unsophisticated user to reliably obtain correct output without human intervention. That's one of the primary reasons why only photo printing has had any real success in kiosks to date. There are simply too many things that can go wrong if you start to envisage a typical walk-in copy-shop customer plugging in a USB stick containing document files.

It might have been expected that PDF could solve this dilemma, but many occasional customers (as opposed to corporate users supplying repeat business) do not have the software required to make PDF files. In addition, to avoid issues with missing fonts etc, they would need to be provided with additional instructions on how the PDF files should be created, e.g. as PDF/X. The more complex the instructions required, the more often they will not be followed completely correctly. A kiosk that must reject a significant proportion of the files submitted to it will not make a profit.

XPS addresses both of those issues, to at least some extent. Every user of Vista will have access to XPS creation software. In addition, every valid XPS file has all fonts embedded with no additional instructions required for the user. It remains to be seen whether sufficient occasional users of copy shops will be aware of the XPS creation capability on their machines, and will understand what is meant by even the simpler level of instruction that might be available on a kiosk machine.

Many of the same issues affect web-to-print systems aimed at the same largely unsophisticated market. Those systems are becoming successful for photo-printing, and they have one advantage over kiosks for document printing in that instructions can be presented to the user whilst they are sitting at their own computer and preparing the files for submission. It's also possible to perform automatic and more or less immediate pre-flighting for printability, allowing any problems to be brought to the customer's attention while they are in a position to make corrections and re-submit.

Many questions still remain: Is there a sufficient demand for kiosk or web-to-print services for document printing as opposed to higher margin products such as photographs, photo books and specialty items such as mouse mats and mugs? After all, most people can now print at home. Kiosks and web-to-print are used primarily to avoid the time required to print large numbers of photographs, or to access features that most home printers don't provide, such as higher quality photo printing, large format, unusual substrates etc. What is the unique selling point for document printing? Perhaps it could be binding?

If there is a potential demand, then we may yet see printing kiosks in supermarkets, enabled through XPS.

9.9 How should a vendor implement XPS for professional print?

Products for processing incoming files for digital front ends (DFEs) on digital presses or for prepress workflows in conventional or hybrid print environments tend to fall into one of two categories:

a) Tightly integrated server solutions where configurations are created ahead of time and then applied automatically to all jobs submitted through a specific input channel. These systems usually emphasize the efficiency of automated processing and offer little or no opportunity for operator interaction on an individual job once it's been submitted.

b) Solutions comprising multiple components, often installed on separate computers. These solutions often include one process that imports a job into the system and then allow (or require) for significant operator intervention on that job before and/or during its passage through the other components.

The optimal approach to short- and medium-term implementation of XPS will vary between systems, but it's possible to lay out options for the two classes of workflow described above.

Copyright © Global Graphics Software Limited, 2007. All Rights Reserved

Tightly integrated solutions are typically built around RIP products. Conceptually they can be described as interpreting the PDL data and storing it in some form of "display list" or "document object model" (DOM) in memory. The display list is then modified as required for trapping, imposition, color management and the other processes that are to be applied, and is then rendered to the raster format required by the output device.

In the short term it would clearly be possible to add a module upstream of the integrated part of the workflow that would take an XPS file and convert it to something that the RIP can already process, such as PostScript or PDF. PDF would be a better choice than PostScript in this situation because its graphic capabilities are rather similar to those of XPS so the need for complex conversions such as flattening of transparent elements would be avoided.

Such a conversion is, however, an additional and potentially quite complex, step in the workflow, adding extra processing time and the potential for conversion errors. If the RIP were capable of natively consuming XPS by reading it directly into the display list, that would obviate the need for an extra process.

Component solutions, on the other hand, are collections of a number of applications bound together in a framework. In most such products each job that is currently in the system is stored in the same format, which may be a PDL in common use (often PDF), or may be a format proprietary to that specific system or vendor. Each of the applications that perform separate processes such as trapping or color management must be able to read, update and write jobs in the storage format selected.

If you're adding support for a new PDL into this kind of system it doesn't really make sense to modify all of the component processes to be able to handle the new PDL. It would be much less work to extend only the single module that is used to import new jobs into the system … or to add another module that can import jobs in the new PDL alongside the previous import component. There is a clear requirement here that all supported PDLs be convertible into the format used for storage between the other components without loss of quality or imaging artifacts. Fortunately virtually all graphical elements in XPS can be represented in PDF, which opens the door to the use of XPS import filters in most component workflow systems. The two most obvious areas where complete fidelity cannot necessarily be maintained are in high precision wide gamut color images, and the use of Windows Color System (WCS) as an alternative to ICC-based color management. Even in these cases, however, adequate approximations are possible if handled with care. There are problems with WCS for professional print (especially on conventional presses) that may render that issue moot anyway, see 11.3.

9.10 What should a print service provider do about XPS?

There are several messages to take away from the predictions set out above. The most important is that owners and managers of print providers in some sectors of the industry should each consider when their particular mix of customers will be asking for XPS support, and how they will respond. It's likely that a well-timed and well-implemented XPS plan will be at least a short-term advantage, and may open doors to new business if sold proactively.

A second message is that only some organizations will choose to standardize on XPS, while others will continue to use PDF. Outside of large organizations the current mix of preferences, from application data files to PDF to XPS will continue to be submitted for printing, and most print providers will continue to need to support those formats. When printing from those applications, especially from those that were not designed for high-end professional printing and do not include direct export of PDF, the best route will probably continue to be to print via PostScript. In other words, your prepress workflow may well need to be able to handle PostScript, PDF and XPS.

10 How do I …

This section sets out the routes to perform commonly required processes. In passing it therefore gives some details on what tools are available to achieve those ends.

 Copyright © Global Graphics Software Limited, 2007. All Rights Reserved

10.1 How do I make XPS files for sharing?

Under Windows Vista any application can create XPS files for sharing simply by printing to the "Microsoft XPS Document Writer" (MXDW) printer instance that is pre-installed with Vista itself.

Exactly the same facility is available under Windows XP and Windows 2003 Server after downloading the free ".NET 3 Framework" from www.microsoft.com[37].

If you're using Microsoft Office 2007, running under Windows Vista, XP or Server 2003, you can download the "Save as XPS" plugin for free from www.microsoft.com. The plugin allows you to export documents from all of the Office applications, including Word, PowerPoint and Publisher, to XPS. This method has some advantages over printing through MXDW in that it retains some live transparency and smooth shading, rather than simplifying everything to the constraints of the GDI to XPS conversion pathway. On the other hand, the current "Save as XPS" plugin for Office tends to create larger and less efficient files than those from MXDW.

Microsoft has stated that an equivalent will be available for Office 2008 on Macintosh when that's released.

A number of other vendors are now shipping tools that create XPS, albeit mainly in niche markets to date. Perhaps the most important of these is AutoDesk, who have updated their AutoCAD product to save files that combine their internal data with viewable (and printable) XPS views of each drawing.

As Vista rolls out it's expected that more applications will be released with either the ability to export directly to XPS, or that are based on WPF, which makes it very easy for developers to incorporate rich graphics within the XPS produced when printing through MXDW.

10.2 How do I view XPS files?

Under Windows Vista XPS files can be opened in the XPS Viewer that's accessed automatically through Internet Explorer[38]. If you download and install the free ".NET 3 framework" and update to Internet Explorer 7, exactly the same functionality is available on Windows XP and 2003 Server.

Under Windows XP and 2003 Server a separate viewer is also available in a different free download from their web site, the "XPS Essentials pack"[39].

Microsoft has also said that they will provide viewers for older versions of Windows and for Macintosh and some flavors of UNIX and Linux[40]. Early indications that this might be provided in Microsoft Silverlight (formerly known as WPF/everywhere or WPF/e) now seem to be unlikely, and no other public statements appear to have been forthcoming.

Any widespread adoption of XPS for document sharing within the corporate sphere is likely to lead to the development and release of additional viewing applications with more advanced collaboration and file modification abilities; it will simply be too rich a market for software vendors to ignore.

10.3 How do I print using XPS?

Printing using XPS under Windows Vista, XP or Server 2003 is very similar to creating an XPS file (see 10.1), but you select a printer rather than the "Microsoft XPS Document Writer" pseudo-printer. If the printer that you're outputting to is supplied with an XPSDrv printer driver, then you'll be sending the data

---

[37] Version 3.0 or 3.5.

[38] Normally, double-clicking on an XPS file will cause it to be opened in Internet Explorer, but there is a minor quirk if you have an alternative browser (such as FireFox) installed. Double clicking an XPS file will open that browser instead, and it probably won't be able to do anything with the XPS file. You can either start Explorer and open the file from there, or manually change the association of the .XPS extension in the Vista control panels. We expect a fix from Microsoft for this at some point.

[39] www.microsoft.com/whdc/xps/viewxps.mspx

[40] Andy Simonds' blog (then XPS program manager at Microsoft), Oct 31, 2005. No longer available on-line.

Copyright © Global Graphics Software Limited, 2007. All Rights Reserved

as XPS, at least as far as the printer driver itself. In other words, you print using XPS in exactly the same way that you'd print using GDI under a previous version of Windows.

If you're writing tools to stream XPS to a device, then you can feed that XPS directly into the Windows print subsystem if the printer you're sending to has an XPSDrv print driver associated with it. If the printer includes native XPS support (or conversion within its DFE), you could also send the XPS directly through the relevant port or use whatever other method it supports to receive a print stream.

10.4 How do I print an XPS file that somebody has sent me?

If you've been sent a document for sharing, or if you're a print provider or document services provider who's been sent an XPS file, you can print that file using a variety of techniques.

If you're running Windows Vista, XP or Server 2003, the simplest approach (and one that does not require any software purchases) is to open the file in an XPS Viewer (see 10.2) and print it from there. Surprisingly, however, opening XPS in an XPS viewer and printing through an XPS-based print subsystem does perform some on-the fly conversion as part of the process. It's not yet clear whether that can introduce any unwanted imaging artifacts in the resulting output, although it seems likely to be safe.

If you want to send the XPS file to a printer, you have a printer driver for the device installed on your computer, know that is an XPSDrv printer driver, and don't mind using command lines rather than a graphical user interface you can open a command prompt and use 'copy' or 'type'[41] to send the file directly into the print pipeline, by typing this:

```
type <filename> <printername>
```

Replace <*filename*> with the name of the XPS file (including the .XPS extension, if there is one), and <*printername*> with the name of the printer.

In a professional print environment you may wish to increase the automation of your print workflow. A potential route to that, without requiring upgrading your prepress equipment or the DFE on your digital press, would be to add a tool to convert from XPS into a PDL that your equipment already supports, such as PDF or PostScript. That conversion tool could be provided by the vendor of your prepress/DFE equipment, or it could be from a third party. It could be part of the print workflow, or it could be part of a document management system upstream from the print-specific steps.

As described in 11.5, conversion like this can be a reasonable thing to do, especially as one step in a migration towards full support of a new PDL. In each case, however, you have to weigh the benefits of the use of conversion against the costs, which can include reduced performance and less certainty of fidelity.

Both office users and printing professionals can also use a printer, prepress system or DFE that consumes XPS natively, avoiding the potential problems associated with conversion.

11 Can you tell me more about the technicalities of XPS?

11.1 What's inside an XPS file?

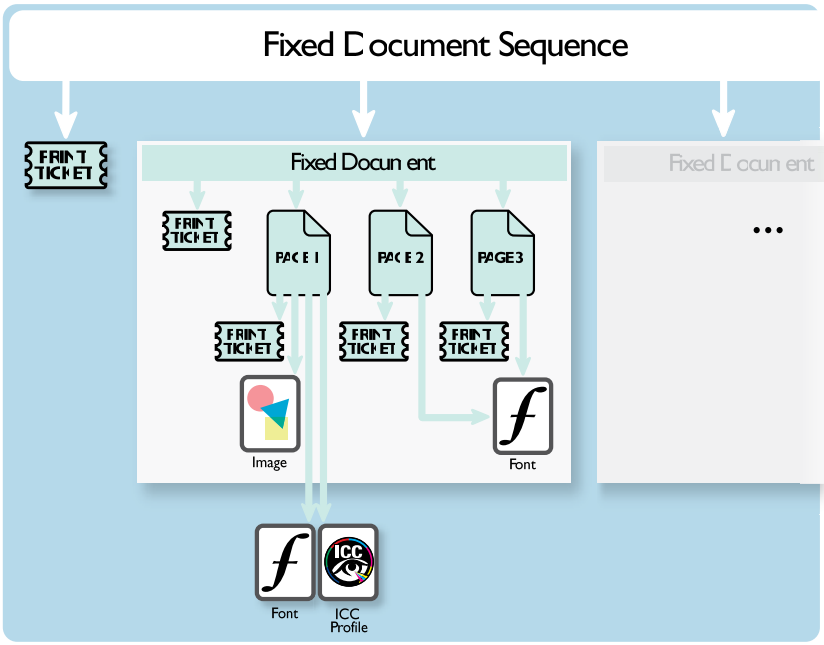
An XPS file is a container designed to carry a variety of components, including vector graphics and images, and the resources that they require, such as fonts and color spaces. In addition there is provision for annotations and metadata.

Each page within the file is stored as a separate XML stream (a "FixedPage") in order to facilitate modifications. Pages are grouped into documents (FixedDocument), each of which is another separate XML stream which references the FixedPage streams making up that document, and any resources and metadata associated with them. The documents are then grouped into the job as a whole using a FixedDocumentSequence XML stream. All of these components are bundled within one ZIP archive, so the user sees only a single file[42]. This use of a ZIP archive, and the structuring of components within it is

[41] 'type' has proven to be consistently more reliable than 'copy', especially for larger files.

 Copyright © Global Graphics Software Limited, 2007. All Rights Reserved

specified by the Open Packaging Conventions (OPC), which is the same container specification as that used for documents from Microsoft Office 2007 (with different contents, of course).



Each page, document and file (job) can also have a print ticket associated with it. Printing controls at the page level generally override those at the document level, and those at the document level override those at the job level.

The intermediate FixedDocument layer within this hierarchy allows pages that form part of different documents to be bundled into a single file for ease and reliability of transmission and sharing, while retaining their logical separation. In addition, it allows different print controls, such as requests for binding, to be applied to each of those logical documents.

Much of the foundation of XPS is based on a variety of pre-existing standards and specifications, including:

- XML – used for vector graphics in a very similar way to SVG, and for metadata and the overarching structure and cross-referencing of the parts of the file.
- ZIP – used to package all of the parts of the document together, and to compress the data where appropriate[43].
- ICC color profiles – to carry color management specifications
- OpenType fonts – for font data, including support for Latin fonts, CJK[44] or "double byte" fonts and other scripts such as Arabic, Hebrew and Cyrillic.
- JPEG, PNG, TIFF for images

---

[42] If you rename an .XPS file to have a .ZIP extension you can open it with any ZIP tool, including the Windows Explorer in recent versions of Windows, and see the overall structure inside it.

[43] XML has a reasonably well deserved reputation for being verbose, leading to large files. The flate compression most commonly used in ZIP is very efficient at compressing data with repeating sequences of bytes, and that's exactly what a typical XML file is built from. The use of ZIP to aggregate and compress the various components of the document therefore means that XPS files should be roughly the same size as a PDF file containing the same graphical data (see **Error! Reference source not found.**).

[44] "Chinese, Japanese, Korean" a common acronym for scripts containing thousands of individual glyphs.

Copyright © Global Graphics Software Limited, 2007. All Rights Reserved

In addition to these pre-existing specifications, XPS adds:

- Windows Color System (WCS) – see 11.3.
- HD Photo – see 11.2.
- The XPS XML grammars themselves – for content, metadata and the document backbone.
- The PrintTicket XML grammar

All of the XML grammars used are designed to be extensible; an implementer can add new metadata keys, print controls and even new graphical elements, although care must, of course, be taken with such extensions as only products from one or a small set of vendors will support them.

11.2 What's HD Photo?

HD Photo is a new image file format that's designed to carry high-dynamic range images, allowing high-precision, wide-gamut graphics to be encoded. It was originally named Windows Media Photo (WM Photo), and the name changed only shortly before Vista shipped, which means that both names may be found in current documentation.

HD Photo combines two features:

- A modern image compression algorithm, which is capable of compressing images to smaller sizes than JPEG compression can do for the same quality; Microsoft claim that HD Photo files are typically around half the size of equivalent JPEG files. In this sense HD Photo is rather similar to JPEG2000.
- The capability to retain richer colors with more tonal detail than JPEG can. JPEG is limited to eight bits of data per color, per pixel, allowing 256 different color levels to be recorded, while HD Photo can record up to 32 bits per color, per pixel, retaining more color levels than can be usefully recorded by any current input device. In this sense it's closer to the camera RAW files created by digital SLRs, or to Adobe's Digital NeGative format (DNG), although Microsoft does not envisage the same conversion step that's an important part of any RAW image workflow for photography. Instead the flexibility inherent from retaining full image data is retained throughout the image life-cycle.

In December 2007, Microsoft released the final version of a plugin for Adobe PhotoShop to open and save HD Photo images[45]. In July 2007 Microsoft announced that HD Photo had been submitted to the Joint Photographic Experts Group (JPEG) for adoption as an international standard: JPEG-XR; that submission has now been approved, and the committee is now actively working on the standardization.

Several vendors of technology used in digital cameras have announced that they will support it.

If HD Photo lives up to its promise of smaller file sizes or increased image quality for the same file size, when compared to JPEG then that may well lead to adoption in compact digital cameras. While it's unclear that HD Photo has technical advantages over JPEG2000, the ease of implementation of HD Photo in amateur photo processing and management applications running under Windows Vista is likely to tip the balance in its favor.

Whether HD Photo can make any inroads against camera RAW or DNG in the digital SLR space is far less clear at this point.

11.3 What's WCS?

WCS is an appearance-modeling approach to color as opposed to the color management based on ICC color profiles, as used by most current color management systems. As such, it may be very suitable for use for display and print in non color-critical applications, where a pleasing rendition is the goal.

[45] The first beta version was made available in March 2007.

 Copyright © Global Graphics Software Limited, 2007. All Rights Reserved

In theory appearance modeling could also provide improved color rendition for professional print, avoiding some of the uncertainty about exactly where in a color transform any gamut and tone-scale compression occurs, or retaining the black channel through a CMYK to CMYK transform, for instance.

Unfortunately, WCS also suffers from some drawbacks in relation both to professional print, and to the better quality color printers that are becoming common for home photo printing:

- In many professional print environments the total area coverage (TAC) of ink laid down when printing is deliberately limited, either to reduce costs, or to avoid physical problems such as paper stretch. This is also important for some home and office inkjets, where printing areas with solid color in all inks can lead to ink dripping across the print. This kind of limitation is not possible with WCS.
- There's no support for devices using more than 4 inks, such as those used for packaging, for some high-quality professional print applications, on many digital production presses, and on home photo printers.

A more detailed technical analysis of WCS, written by Steve Upton, President of Chromix, is available at www.colorwiki.com/wiki/Vista's_New_Color_Management_System_-_WCS.

In the Vista implementation each WCS profile is wrapped inside an ICC profile that should produce a roughly equivalent result in most cases if it's used instead of the WCS profile. This may prove to be an advantage, allowing products that don't directly support WCS to generate acceptable output. On the other hand, it may prove to be yet another cause of difficult-to-explain differences in color output between different tools and printers; a problem that has plagued color management since it was first invented.

11.4 How is the print system organized under Windows Vista

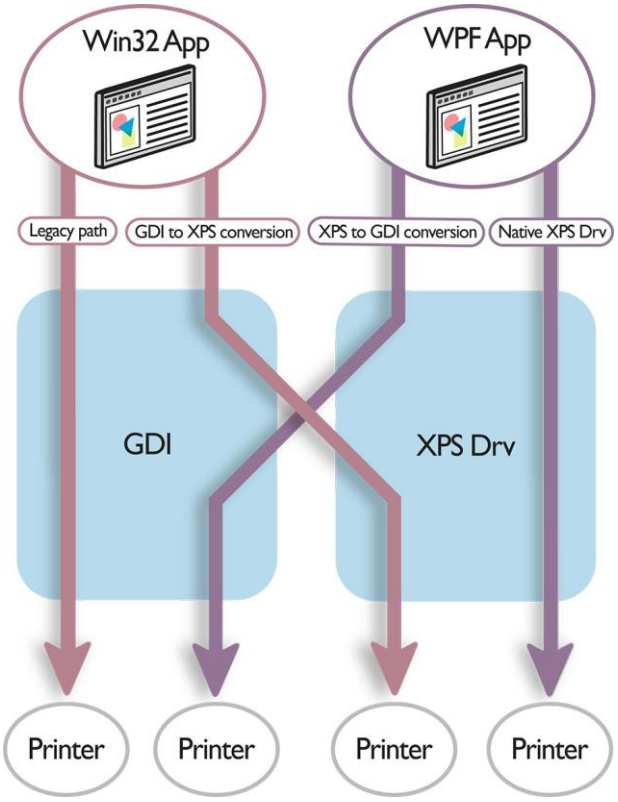
Windows Vista supports both the "Version 3" GDI printer drivers that have been used in Windows for the last 20+ years, and new "XPSDrv" printers that are built around the capabilities of the XPS format. Exactly how these are called depends on:

- whether the application from which you are printing is an XPS-capable application (probably a WPF application, built using .NET 3), or is an older "Win32" application
- what kind of driver is installed for the printer to which the job will be sent; whether that's GDI or XPSDrv.

All print jobs sent from a non-XPS-capable application will initially be converted into GDI. If there is a mismatch between the application and the printer driver, the job will be automatically converted into the other print pathway; the user does not need to do anything special in order to allow this to happen.

All jobs that pass through the GDI print pathway at any stage will be more or less limited to the capabilities of that pathway, including a restriction to RGB. Thus the potential for higher quality direct print, such as smoother shading, direct selection of black ink/toner, high-precision wide-gamut color and the highest quality rendering of transparency, is only achievable when printing from an XPS-capable application to a printer driven through an XPSDrv driver. XPS files saved (rather than printed) from a Win32 application through product-specific export filters such as those available for Microsoft Office 2007 can also achieve that same potential if they are later printed through an XPSDrv-supported device.

Copyright © Global Graphics Software Limited, 2007. All Rights Reserved



Windows XP and Windows Server 2003 only support Win32 applications and the GDI print pathway out of the box. By installing the .NET 3 framework[46], these operating systems can support WPF applications and XPSDrv print drivers in the same way as Windows Vista.

11.5 What's wrong with document conversion?

A number of the sections above make references to the use of document conversion, in printing, or as part of a document management and sharing strategy. It's clear that there are many use cases where conversion is an attractive option, and can have some significant benefits, especially the ability to leverage existing systems and resources.

Any conversion, however, can also have disadvantages. The most common being:

- A possibility of introducing artifacts, and therefore reducing the fidelity of any later processing of the document, whether for printing, viewing on-screen or analysis such as text extraction and indexing.

  In a print pipeline this possibility can be minimized if the conversion process is configured to precisely match the capabilities of the RIP for which the conversion is being made, including resolution, device color space and bit-depth, screening etc.

- The introduction of another step in the process almost inevitably either reduces performance or requires additional resources.

  If the conversion replaces some other process rather than adding to it (as might be the case, for instance, when a document is imported into a document management system) then this issue may not arise.

- A conversion step may, in some cases, prevent the streaming of documents, leading to greatly reduced throughput or longer first-page-out time in some print environments.

These costs must always be balanced against the benefits that the use of conversion provides:

[46] a free download from www.microsoft.com

 Copyright © Global Graphics Software Limited, 2007. All Rights Reserved

- Potentially reducing time-to-market in comparison with native support across all components of a product range, making it attractive as part of a migration strategy for a vendor.
- Compatibility with pre-existing equipment, making it attractive for use with solutions embedded within hardware which can be more problematic to upgrade or extend, or by third-party solutions providers who cannot directly influence the installed products.
- Potentially reduced development and testing requirements when adding new functionality, leading to reduced costs.
- Consistency with the architecture of a system that already standardizes all documents or graphics into a single format and then allows a wide variety of analysis, manipulation and processing of files in that format.

12 What other resources about XPS are available?

Microsoft publishes:

- An overview of XPS at www.microsoft.com/whdc/xps/default.mspx.
- "XPS for end users" at www.microsoft.com/whdc/xps/xpsusers.mspx.
- "XPS for IT professionals" at www.microsoft.com/whdc/xps/xpsitpro.mspx.
- XPS for application and hardware developers at www.microsoft.com/whdc/xps/xpsappdevs.mspx.
- A list of tools to view and generate XPS at www.microsoft.com/whdc/xps/viewxps.mspx
- "XPS and Color Printing Enhancements in Windows Vista" at www.microsoft.com/whdc/xps/vista_print.mspx.
- The XPS, OPC and HD Photo specifications etc at www.microsoft.com/whdc/xps/downloads.mspx.
- "Reference Raster Image Processor (RIP)" (provided by Global Graphics) at www.microsoft.com/whdc/device/print/RRIP.mspx[47].

Quality Logic provides XPS implementation test suites and tools, and XPS training; see www.qualitylogic.com/xps/xps_test_tools.html.

"Windows Vista's XPS more than just a PDF competitor" by Greg Schultz at articles.techrepublic.com.com/5100-10877-6102213.html.

This FAQ and a number of white papers are available from Global Graphics at www.globalgraphics.com/xps/index.html.

13 Terminology

This section provides a number of brief definitions for terms and acronyms used throughout the FAQ.

Avalon A code name used for XAML, now obsolete

CMYK A four-channel color space based on Cyan, Magenta, Yellow and Black.

DFE Digital Front End. The server on which the majority of software directly associated with a digital production press runs.

DRM Digital Rights Management. A method or system to restrict what the recipient of a file can do with it based on policies set by the publisher or supplier of that file.

Ecma An international standards body. Microsoft's Office Open XML specification and Adobe's ActionScript were both standardized through Ecma. www.ecma-international.org/

[47] Note that this reference RIP does not contain all of the functionality of Global Graphics' commercial products, and represents only the state of XPS rendering in those products in December 2006. Later development and optimizations mean that current commercial products are many times faster.

Copyright © Global Graphics Software Limited, 2007. All Rights Reserved

GDI Graphics Device Interface, used by pre-Vista versions of Windows to carry data for presentation on-screen and in the print pipeline.

ICC International Color Consortium. An organization defining specifications for color management and color profiles. See www.color.org.

HD Photo A new image file format defined by Microsoft that is capable of storing images with a very large dynamic range because of the very high bit depths and compressions supported. Previously named WMPhoto.

IPP Internet Printing Protocol. See www.pwg.org/ipp/

Metro A code-name used for XPS, now obsolete

MXDW Microsoft XPS Document Writer. A printer instance pre-installed on Windows Vista that can be used to create XPS documents by printing from any application. Also available on Windows XP and Server 2003 by installing the ".NET 3 Framework" which can be downloaded from www.microsoft.com.

.NET 3 A framework developed by Microsoft, with supporting development tools, to generate code based on XAML. Pronounced "dot net 3". See 3.1.

.NET 3 Framework An add-on for Windows XP and Server 2003 that provides the environment for .NET 3 applications to run, and that includes the XPSDrv print sub-system, the MXDW printer and XPS viewing in Internet Explorer.

PCL Printer Control Language; a page description language defined by Hewlett-Packard.

PDF Portable Document Format; a page description language defined by Adobe Systems.

PDF/A A family of international standards based on a subset of PDF aimed at providing reliable formats for long-term storage of page-based documents, but also useful for short-term exchange of documents within and between organizations.

PDF/X A family of international standards based on a subset of PDF aimed at providing reliable formats for the exchange of page-based documents for printing, especially in the professional print sector.

PDL Page Description Language; a format such as PDF, PostScript and XPS that fully describes a representation of a document containing one or more pages in a way that can be understood by a printer. Modern PDLs usually contain additional metadata beyond that required for printing.

PostScript A page description language defined by Adobe Systems

Print Ticket The information within an XPS file that's designed to control the printing process, e.g. by requesting double-sided printing.

Raster Graphical data made up of pixels. This may be an image created by scanning, screen-grabbing or a digital camera, or the output of a RIP or renderer in preparation for sending the data to a printer.

RGB A three-channel color space based on Red, Green and Blue.

RMS Rights Management System. A system used to implement DRM. This is the preferred Microsoft acronym.

Section 508 US legislation aimed at ensuring that information is accessible and usable by all, including those who use assistive technology such as text-to-speech solutions

SoHo "Small office, Home office". Used in this FAQ to denote printers that are often connected via USB or parallel connections rather than being networked. Print jobs are also often sent to them as rasters which are generated from the GDI or XPS print stream inside the printer driver, rather than a PDL being passed on to the controller within the printer.

SVG Scalable Vector Graphics. An XML-based format for representing vector graphics. Scalable Vector Graphics; see www.svg.org.

Vector Graphical data formed of lines, fills, text etc, as opposed to raster (image) data.

 Copyright © Global Graphics Software Limited, 2007. All Rights Reserved

WCS Windows Color System, see 11.3

WHQL Windows Hardware Quality Labs; see www.microsoft.com/whdc/GetStart/default.mspx

Windows Presentation Foundation See WPF

WinFX A code-name used for WPF, now obsolete.

WMPhoto Windows Media Photo. The original name for HD Photo. Both names are in current use, but HD Photo will become more popular. See HD Photo.

WPF Windows Presentation Foundation. The new graphics architecture in Windows Vista. The code-name 'WinFX' was used for several years before the WPF release name was announced in 2006. See 3.1.

XAML A declarative markup used to describe the appearance and behavior of objects on the screen. Pronounced "zammul". The code-name "Avalon" was used before the final public name was selected. See 3.1.

XML the eXtensible Markup Language, a foundation format widely used for storage and communication of computer readable data structures. It forms the basis for Microsoft's new Office Open XML document formats, and or JDF, PPML and a lot of data handling on web sites.

XPSDrv The subsystem under Windows Vista that processes XPS files for printing.

XSL-FO XML Stylesheet Language, Formatting Objects; a (relatively primitive) page description language, defined in XML.

XSLT XML Stylesheet Language Transformation. A specification describing how an XML file may be programmatically manipulated, e.g. to convert from one XML grammar to another.

14 Who wrote the answers in this FAQ?

All of the answers in this FAQ have been written by staff at Global Graphics (www.globalgraphics.com). In 2003 Microsoft selected Global Graphics to provide consultancy on the XPS specification and XPS proof of concept development services. We also developed a print reference implementation RIP for Microsoft to assist IHVs and ISVs in the evaluation of quality printed output. This is available for download from the Microsoft website. Over the same time we've also been discussing XPS at length with a wide variety of software and hardware vendors, leading to "early adopter" programs for electronic document and printing technology, leading to the availability of several commercial products and the signing of a number of contracts with major partners.

The questions in this FAQ have arisen from those discussions, and out of conversations with many end-users, consultants and commentators.

In parallel with our XPS developments, we're also an important provider of printing and conversion technologies for page description languages including PostScript and PDF. If this FAQ appears overly pro-XPS it's because we believe that a clear exposition of the benefits of XPS has been lacking to date, not because we believe that XPS is universally better than other PDLs. As explained above we believe that XPS, PDF and PostScript all have their individual strengths and weaknesses, and expect them to develop and be used in parallel with each other for some time to come.

Global Graphics is also a member of a number of related standards organizations including CIP4 (developing JDF), CGATS and ISO task forces working on PDF/X and PDF/A, PODi, the ICC and the Ghent PDF Workgroup[48].

It's almost inevitable that the philosophy and conclusions set out in this FAQ are the same as those that have informed decisions around our own product line-up and architectures. Realizing that, we've tried very hard to present a balanced viewpoint, both across PDLs and in as vendor-neutral a way as possible.

[48] One of the authors of this document, Martin Bailey, is a former CEO of CIP4, chairs the Ecma XPS technical committee, CGATS and the PDF/X task forces within CGATS and ISO, and is the primary UK representative to the ISO PDF/A and PDF task forces.

Copyright © Global Graphics Software Limited, 2007. All Rights Reserved

The technology experts for open document and print solutions

www.globalgraphics.com

Global Graphics Software Inc.
31 Nagog Park, Suite 315
Acton, MA 01720, USA
Tel: +1 (978) 849 0011
Fax: +1 (978) 849 0012

Global Graphics Software Ltd
2nd Floor, Building 2030
Cambourne Business Park,
Cambourne, Cambridge
CB23 6DW UK
Tel: +44 (0)1954 283100
Fax: +44 (0)1954 283101

Global Graphics KK
Level 14, Hibiya Central Building
1-2-9 Nishi-Shimbashi, Minato-ku
Tokyo 105-0003. Japan
Tel: +81-3-5532-7340
Fax: +81-3-5532-7373

Copyright © Global Graphics Software Limited, 2007. All Rights Reserved

Global Graphics, *the smarter alternative,* the Harlequin Rip and Jaws PDF Creator are trademarks of Global Graphics Software Limited which may be registered in certain jurisdictions. Global Graphics is a trademark of Global Graphics S.A. which may be registered in certain jurisdictions. PostScript, Adobe, Adobe Acrobat, Distiller, InDesign, Illustrator and Photoshop are trademarks of Adobe Systems Inc which may be registered in certain jurisdictions. Microsoft and Windows Vista are trademarks of Microsoft Corporation which may be registered in certain jurisdictions. All other brand and product names are the registered trademarks or trademarks of their respective owners. Global Graphics makes no warranty and accepts no liability for any loss or damage arising from the use of information or particulars in this document.