White paper

Print File Formats: A Comparative Analysis of EMF, OpenXPS and PDF for Enterprise Printing

A technical comparison, analysis and evaluation of the three most widely adopted print file formats employed by Universal Printer Drivers in enterprise environments.

August 2013

uniprint™

how the world prints

Contents

Executive Summary

Universal Printer Driver (UPD) technology can simplify printer management, increase security and reduce print output. UPDs make it possible to convert documents of any type, while a virtual printer can be installed on a server and made available for printing on the local network.

A wide range of software solutions are available, yet there is continued debate concerning the most appropriate UPD file formats. These formats provide the interface between the operating system and peripheral devices and translate commands between the two.

The three most widely adopted print file formats in UPDs are: Portable Document Format (PDF); Open XML Paper Specification (OpenXPS); and Enhanced Metafile (EMF).

Compression and bandwidth control are essential to overall printing efficiency, performance and productivity – especially in distributed and virtualized environments. Security of data can also be a key concern.

The purpose of this white paper is to provide a technical comparison and analysis of each of the three most widely adopted print file formats within UPD solutions, and how they influence printing performance in enterprise computing environments.

Ultimately, this white paper will demonstrate why PDF is the most appropriate print file format for UPD solutions designed for medium-to-large enterprises, and how the use of PDF can make enterprise printing more agile, efficient and secure.

It should be noted that although HP's Printer Control Language (PCL) has been adopted as a de facto standard by a number of printer manufacturers, it is proprietary to PCs and is not a universal transport or printer language. It is therefore not considered within the scope of this white paper.

1. Printing in the Enterprise Environment

Managing a multitude of printer makes and models has always been a challenge for enterprise systems administrators, and especially for larger organizations with multiple sites and cross-border operations. As users access printers from multiple networks and endpoints, it can be difficult for administrators to keep control of the printing environment, while maintenance and support can be resource-intensive.

1.1 Challenges

Printing becomes even more complex in virtualized and distributed environments such as those based on Citrix/Remote Desktop Services (RDS) or Microsoft's Terminal Server. This is because the application being printed from is running remotely, yet the printer is often situated locally (i.e. within the office or home). Compression and bandwidth control are therefore essential to overall printing efficiency, performance and productivity. Security of data also becomes a key concern.

Printing performance can be impacted greatly if large print jobs are being spooled on a remote host and sent down to the client. Spooled files were never intended to be transmitted across a wide area network (WAN) and major players in the virtualization market (Citrix, Microsoft and VMware) have added specific capabilities to their products to address the challenge of managing printers in virtualized and distributed environments.

To extend and enhance these capabilities, independent printing solutions based on Universal Printer Driver (UPD) technology have been developed by specialist providers such as UniPrint. These solutions add features such as the ability to print to network print servers without the use of drivers, secure printing and advanced performance and bandwidth controls.

1.2 Printer Workflows

The printing process varies according to whether a user is printing to a local printer or a network printer. Simply replacing a physical print server with a virtual one does not change the print queue setup and management process; thus, the following stages generally apply for all printer workflows:

- When the user presses the "print" button, the application creates a metafile known as "print data." This contains instructions for the printer such as characters, color, font and spacing.
- The operating system's print subsystem receives this metafile, determines whether the target printer is a local or a network printer and uses the printer driver to translate the print data into the printer's native language, at which point, the print data becomes a "print job."
- Once the print job is created, it is sent to a print spool – a temporary storage area usually on the hard disk or in memory – before being sent to the printer. Spooling allows multiple print jobs to be given to the printer at one time and gets its name from technology used in the 1960s, when print jobs were stored on large reels of magnetic tape.
- In the final printing phase, the printer receives the rendered print job from the print spooler and prints this file regardless of its format.

2. Print File Formats

There are multiple vendors offering print solutions based on Universal Printer Driver (UPD) technology. Designed for use in virtualized, server-based computing or conventional desktop environments, these solutions typically employ one of three print file formats:

- Portable Document Format (PDF) – a PostScript printing language (PostScript being a representation of a print job) invented for laser printers and championed by Adobe and Apple.
- Open XML Paper Specification (OpenXPS) – a relatively new standard designed by Microsoft to replace PDF.
- Enhanced Metafile (EMF) – the printing standard introduced by Microsoft as a protocol enabling communication between the printer driver and the application and/or Microsoft operating system.

The majority of UPDs use either EMF or OpenXPS. PDF however, is proven to deliver much better performance at the transport layer due to the fact that it uses compression. PDF also incorporates robust security mechanisms.

IT managers and systems administrators in particular, pay close attention to the print file format employed by a UPD because efficiency, performance and security are critical in virtualized and distributed environments where bandwidth and server capacity might be shared between thousands of users and network devices.

In the following sections, we will examine the evolution of each file format and salient features and functions.

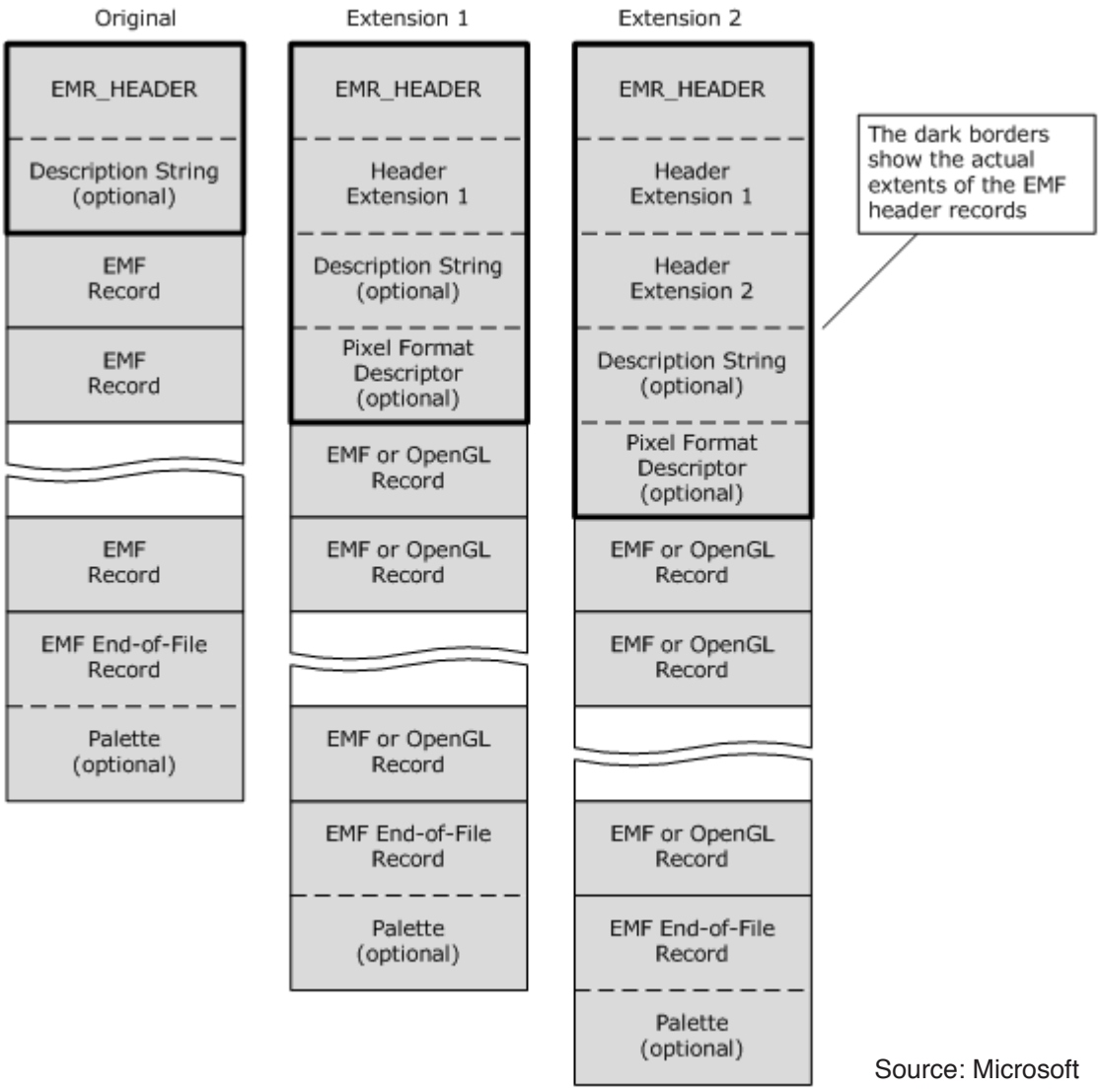
2.1 Enhanced Metafile (EMF)

EMF is the Windows format for rendering the print job. As the 32-bit version of the original Windows metafile (WMF) format, EMF was introduced in the early 1990s to solve deficiencies of the WMF format.

EMF files are intended to be portable between applications, with the EMF structure specifying a metafile format that can store a picture in device-independent form. The stored picture can be rendered by parsing and processing the metafile.

As summarized in *Figure 1*, an EMF metafile is a series of variable-length records. Called EMF records, these are objects that represent each element in the document and contain graphics drawing commands, object definitions and properties. Different records have different attributes, the values of which are defined by the data within the EMF file.

Figure 1: EMF File structure



Source: Microsoft

uniprint™
how the world prints

Notable records within the EMF file format are:

- EMR_EXTTEXTOUTW – draws a Unicode text string using the current font and text colors. This record has many attributes, most important of which are the StringBuffer and LogFont attributes. The StringBuffer stores the exact Unicode character values of the text with no compression. The LogFont is a stand-alone object that has its own range of attributes, and it describes the font being used.
- EMR_SETDIBITSTODEVICE – specifies a block transfer of pixels from specified scan-lines of a source bitmap to a destination rectangle and is used mainly to store images from documents. It has all the essential attributes, such as the size of the image or the coordinates, as well as the BitmapBuffer, which maps the coordinates of every pixel of the source image to the destination page.

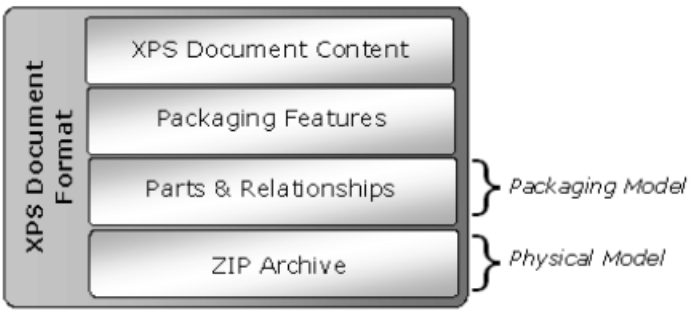
An EMF file does not compress the source image and paints it onto the page by mapping the coordinates of every pixel of the source file. In addition, where a single image is repeated, the EMF file is forced to draw the same image each time, whereas XPS and PDF can reference a single image multiple times *[see 'repeated image test case' in section 3]*. Coordinates can therefore take up a large amount of space in terms of storage when multiple instances of the same image exist within a single EMF file.

2.2 Open XML Paper Specification (OpenXPS)

OpenXPS is a document description language based on XML (Extensible Markup Language) and is similar in principle to Adobe's PS (PostScript). Developed by Microsoft as a replacement for EMF and standardized as an open standard document format in 2009, OpenXPS is designed to simplify the process by which digital documents are created, shared, printed, viewed and archived.

As depicted in *Figure 2*, the XPS Document format uses a ZIP archive for its physical model. Within the ZIP archive, the document is organized into different folders, each containing information regarding the document, such as relationships, resources, and content. Content files are stored in the document folder and are identified by the extension ".fpage." These files are written in XML and describe the content and properties of the pages.

Figure 2: XPS Document format



Source: Microsoft

The key elements within the OpenXPS file are Glyphs, which represent objects such as text, fonts and images. Like all XML documents, Glyphs have a variety of associated attributes that can be assigned to them. For example, a font style (FontUri), font size (FontRenderingEmSize), color (Fill), and the content of the text (UnicodeString).

It is worth noting that ASCII characters are not compressed and are stored exactly as they are represented in the Glyph element – i.e. as a Unicode string. However, OpenXPS is not able to distinguish between normal fonts and italicized fonts. It therefore generates two different files for a block of text written in the same font but using a mixture of normal and italic styles. Furthermore, XML is a wordy coding language as it employs a large volume of tags. This means file sizes are often larger than with a comparable PDF file.

When saving a file as an XPS file, images are stored within the resource folder as JPEGs – regardless of the original file format. One of the most important attributes is the image source attribute, which provides the path to the image stored in the resource folder, as this enables multiple images to be painted onto a page from a single source.

2.3 Portable Document Format (PDF)

PDF is a compact file format for creating, viewing and sharing documents. Developed by Adobe Systems and launched in 1993, the PDF specification now serves many applications and became an official ISO-standard (ISO 32000-1:2008) in 2008.

PDF preserves document formatting and enables file sharing. PDF printer drivers are based on the principle of virtual printing. These software products install on a system like regular printers, but have nothing to do with physical printers. When a document is sent to PDF printer drivers, they receive the data stream, convert it using the PDF specifications and save the resulting file on the computer's hard disk.

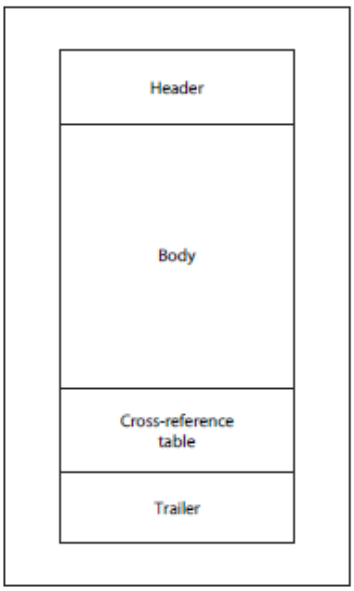
Among its other roles, PDF serves as a page description language: a language for describing the graphical appearance of pages with respect to an imaging model. An application program produces output via a two-stage process:

- The application generates a device-independent description of the desired output in the page description language
- A program controlling a specific output device interprets the description and renders it on that device

These two stages can be executed in different places and at different times. The page description language serves as an interchange standard for the compact, device-independent transmission and storage of printable or displayable documents.

Unlike other print file formats such as EMF and XPS, PDF does not use folders. It is a single file comprised of data [*see Figure 3]*, with images, fonts and blocks of text all represented as objects. Each object has attributes, data sets and values associated to it. Data and values are then organized into streams, which are compressed and encrypted and can only be read by the printer or the PDF reader. All of the data is compressed within a single file.

Figure 3: PDF format



Source: Adobe Systems

It is the content of the compressed streams that tells the printer exactly what to paint graphically onto the page. All layout and formatting decisions have already been specified by the application generating the content stream. As with XPS, font styles (e.g. normal, italic) are split into separate streams, while in instances where there are repeated images, the same stream from the image source can be repeated multiple times (therefore keeping the file size small).

3. Comparison and Analysis

The key performance indicators (KPIs) of a Universal Printer Driver (UPD) are efficiency, file size and how accurately the original document is reproduced by the print file format.

The following section provides a comparison, analysis and evaluation of the three most widely adopted UPD print file formats.

3.1 File Size and Efficiency

The most important aspect of a printer format is its efficiency. Efficiency is closely related to the format's file size, since the file size determines the amount of time taken for the data to be transmitted from the application to the server and from the server to the local printer.

In order to compare the file size for each format, a sample document comprising text only was generated in Microsoft Word. The document was then converted into each of the three formats for comparison and the file sizes recorded. This exercise was then repeated several times using various combinations of text, longer text and images. The results are shown in *Table 1*.

Table 1: Comparison of PDF, EMF and OpenXPS file sizes

	EMF	OpenXPS	PDF
Text only	407KB	461KB	134KB
Text and image	16.3MB	690KB	657KB
Varied	32.3MB	390KB	373KB
Repeated image	141MB	113KB	104KB
Long text	26.3MB	3.26MB	1.86MB

Source: UniPrint

In most cases, the file size of PDF and OpenXPS are similar, with PDF being slightly smaller. EMF is almost always exponentially larger except in the case of a text-only document conversion. This indicates that EMF uses a substantial amount of space to render the images in the document. When the document becomes larger or when the document is text based, the difference in file size becomes more apparent, especially in the case of long text and text only.

A specific result worth noting is the repeated image case. Even though there were five images in the original document, the size of the OpenXPS and PDF files generated remained comparatively small, whereas the EMF file size was extremely large. This is due to the fact OpenXPS and PDF only render the same image once, and then reference the same image in the document. Conversely, EMF renders the image every time it appears in the document, which generates a large amount of data.

3.2 Security

In a corporate environment, document security is a priority because confidential files may be compromised if they are not protected by a set of robust security features.

EMF

Currently, there is no way to reliably protect the content of an EMF file, other than setting the default permission restrictions in the file's properties.

Moreover, there have been vulnerability issues related to the EMF format. For example, the Microsoft Office Enhanced Metafile Processing Heap Overflow Vulnerability allows an unauthenticated, remote attacker to execute arbitrary code on a targeted system.

An unauthenticated remote attacker could exploit this vulnerability by convincing a user to open a malicious image file on an infected system. Subsequent patches have been released to resolve this issue.

OpenXPS

OpenXPS documents support digital signatures to provide greater document security. When applied to a document, these digital signatures can help ensure the identity of the author and the validity of the document content.

A digital signature enables users to verify that their XPS document hasn't been changed since it was digitally signed, and to verify the identity of the publisher of other XPS documents. This enables the user to establish without any doubt that the file originates from a trusted source before he/she opens it.

A user can also create permission restrictions for their XPS document so that no one can view, copy, print or digitally sign their document without their express permission. The user can set time restrictions or different levels of access on permissions and allow as many or as few permissions as they want.

PDF

PDF documents can be protected with the following password security based features:

- Password – to restrict users from opening the file for viewing, with a further password set to prevent any change to security settings (i.e. permissions, user or master password).
- Digital signature – can be used to authenticate and verify users.
- Permission settings – provide further controls that can limit copying, editing or printing. The restrictions on copying, editing or printing depend on the reader software to obey them.

PDF also supports a wide range of third-party software that can further enhance the safety of a user's documents. For example, one software solution allows the user to control the number of times the document can be viewed and printed, the length of time the document can viewed and even prevents the user from taking a screenshot of the document.

From time to time, new vulnerabilities are discovered in various versions of Adobe Reader, prompting the company to issue security fixes. Other PDF readers are also susceptible. A further aggravating factor is that a PDF reader can be configured to start automatically if a web page has an embedded PDF file, providing a vector for potential attack. Patches and fixes are released frequently to combat such issues.

3.3 Support and Compatibility

All three formats are supported by Windows, Mac OS and Linux. However, one notable difference is that only PDF has software compatible with mobile devices based on the iOS, Android and Windows Phone platforms.

EMF

Despite being the least recognized of the three printer file formats, EMF does have a small base of third-party software support. Notable examples include:

- ACDSee
- OpenOffice
- Adobe Illustrator CS6
- IrfanView

OpenXPS

OpenXPS is supported fairly widely by third-party software vendors but has a much smaller base when compared with that of PDF:

- MuPDF
- STDU Viewer
- XPS

PDF

Over the years, PDF has become the de facto standard for printable documents on the web. It is therefore supported by a huge range of third-party software and applications – from readers to web browsers. The most widely used third-party applications are:

- LibreOffice Foundation
- Foxit Reader
- Mozilla Firefox
- Google Chrome
- QPDF
- Solid Converter PDF
- PDFedit

4. Evaluation

When print data is converted into PDF, print jobs can be compressed by as much as 90%. Reducing bandwidth consumption and enabling faster overall print job transmission improves printing efficiency. Users can preview, print to any printer, email as an attachment and archive print documents on-the-fly. PDF incorporates robust security mechanisms and is therefore widely considered to be the safest of the three print file formats.

However, some industry commentators argue that EMF is the superior option because it does not involve any re-rendering of the file (to PDF), making it faster and resulting in a more accurate print job. Although it is true that the images within EMF files maintain their original quality because they are rendered for the printing device on the client and are not compressed, UniPrint's benchmarking process found the EMF file generated to be considerably larger than that of a PDF generated from the same document. Because fonts are not embedded, it is possible that the print job might not be delivered as desired.

EMF was never intended as a protocol to represent a page; rather, it's a set of instructions on how to print a job. Therefore, by design, EMF does not include compression, font embedding or objects printing. Furthermore, EMF is only available within the Windows platform and can only be used by Windows-based clients, meaning it is not suitable for use in mixed desktop environments (e.g. Apple Mac, Linux and PC).

By using XML, OpenXPS can describe the document in a resolution-independent, vector-based manner. This means that every part of the document (even individual letters and small images) is stored in a way that ensures it will not lose any quality when the image is enlarged or shrunk. OpenXPS also supports the Windows Color System management technology. As a consequence, there will be no loss of color quality and no color change when viewing XPS documents on different systems — a problem EMF to which can be vulnerable.

The main drawback to OpenXPS is that because it uses the complex XML coding language, it can consume a lot more CPU and bandwidth. Processing page content can be much slower than PDF. Also, it can take longer to transmit an OpenXPS file over the network because it is converted to EMF before it is transmitted.

Interacting with an XPS Document is also difficult because the format is primarily designed for document fidelity, data accuracy, file size and compression schemes. Thus, XPS Documents can neither be manipulated nor edited easily, and can only be viewed on computer systems that have an XPS Viewer installed.

5. Conclusion

Larger organizations tend to be more distributed and therefore lean toward a UPD solution that employs PDF. Not only is the file size small, the file can be generated quickly and is extremely flexible in terms of application. It is possible to convert any file to a PDF document as long as the computer has the necessary viewing and printing software.

With the rapid adoption of smartphones and tablet devices such as the iPad as business tools, it should also be considered that only PDF is widely supported on mobile devices. None among the current generation of non-Windows tablets come with built-in printer drivers, thus a virtualized printing solution that employs the PDF file format provides a flexible and future-proof option for those organizations with a virtualized and distributed environment that are looking to support more mobile knowledge workers.

PDF-based UPD simplifies printer management and enables anywhere, anytime, any device printing across different platforms. Furthermore, robust security is assured, as print jobs are converted into a compressed PDF, which can be encrypted. This enhances enterprise-wide document security, which is essential in highly regulated industries such as healthcare, financial services, government and legal services. Security will also be at a premium for any organization migrating to cloud computing solutions and applications.

As the cloud and software as a service (SaaS) models become more widely adopted, providing printing to clients that an organization cannot control, touch nor manage directly will present a fresh challenge for those managing their organization's printing requirements in complex virtualized and distributed environments. UPDs will therefore be essential in making printing more agile, efficient and secure.

Glossary

BMP	Bitmap
CPU	Central Processing Unit
EMF	Enhanced Metafile
JPEG	Joint Photographic Experts Group
OpenXPS	Open XML Paper Specification
PCL	Printer Control Language
PDF	Portable Document Format
PNG	Portable Network Graphics
PS	PostScript
RDS	Remote Desktop Services
SPL	Spool Parameter List
TIFF	Tagged Image File Format
UPD	Universal Printer Driver
WAN	Wide Area Network
XML	Extensible Markup Language

About UniPrint

UniPrint, a division of ACCEO Solutions Inc. (www.acceo.com), is the recognized leader in printing virtualization.

The newest solution in UniPrint's patented, award-winning product line, UniPrint Infinity is the industry's first truly secure enterprise-wide solution for any computing environment. Enabling anywhere, anytime, any device secure pull printing, UniPrint Infinity comes with a proven ROI through its statistics module, print document archiving functionality and Virtual Print Queue technology.

UniPrint Infinity replaces all manufacturer printer drivers with a single PDF generator to promote faster, more efficient printing, helping organizations to improve both productivity and return on investments. For additional information on UniPrint, please visit www.uniprint.net.

UniPrint Infinity, UniPrint Server, UniPrint Terminal Server Edition, UniPrint Gateway Edition, UniPrint Gateway Terminal Server Edition, UniPrint VDI Edition and UniPrint Host Edition are trademarks of UniPrint. All other trademarks and registered trademarks are properties of their respective owners.

For more Information or a free product trial, please visit www.uniprint.net or e-mail marketing@uniprint.net.

uniprint™

UniPrint
1 Eva Road, Suite 309
Toronto, Ontario
Canada M9C 4Z5
www.uniprint.net

Worldwide: (416) 503-9800
North America Toll Free: 1 (866) 488-3188
Europe, Middle East & Africa: (44) (0) 1344 206146
Fax: (416) 503-3113

©2013 UniPrint, a division of ACCEO Solutions inc.