Ecma OpenXPS Whitepaper (TC46)

Ecma/TC46/2009/062
Ecma/GA/2009/087

Author: Ecma International Technical Committee 46 (TC46)
Date: April 2009
Version: 1.0

Introduction

Since mid-2007, a number of companies have been working as part of Ecma International Technical Committee 46 (TC46) on a standard for the OpenXPS paginated document format. OpenXPS has been designed and optimized to provide a bridge between physical paper and electronic paper, and to address the requirements for sophisticated rich-content document workflows between people, PCs and devices. The design selected has been able to achieve these goals in a format that is efficient and straightforward to implement, and that enables interoperability between a wide range of systems and solutions.

The OpenXPS Standard provided by Ecma TC46 formally defines the Open XML Paper Specification. This standard is written for developers who are building systems that process OpenXPS content.

A primary goal of the standardization effort is to ensure the interoperability of independently created software and hardware systems that produce or consume OpenXPS content. The standard defines the requirements that systems, producers and consumers, processing OpenXPS Documents must satisfy in order to achieve interoperability.

The OpenXPS Standard describes a paginated document format called the *OpenXPS Document*. The format requirements are an extension of the packaging requirements described in the Open Packaging Conventions (OPC) Standard. That Standard describes packaging and physical format conventions for the use of XML, Unicode, ZIP, and other technologies and specifications, to organize the content and resources that make up any document. They are an integral part of the OpenXPS Standard. In addition, OpenXPS builds on widely available technologies, such as ICC colour management, with the result that developers can utilize readily available hardened components in their implementations.

This paper, prepared by members of TC46, provides an overview of the OpenXPS format, background on the trends and design goals that influenced the development of the format, and additional details of the activities of the committee.

For feedback and questions please see the TC46 website at: http://www.ecma-international.org/memento/tc46.html

Contents

Rationale for an Electronic Paper format

Over the past several years a number of trends have emerged that impact how systems and devices work with printed documents. These trends are evident in the activities that users perform, requirements for procurement, and in products that have been developed to address them.

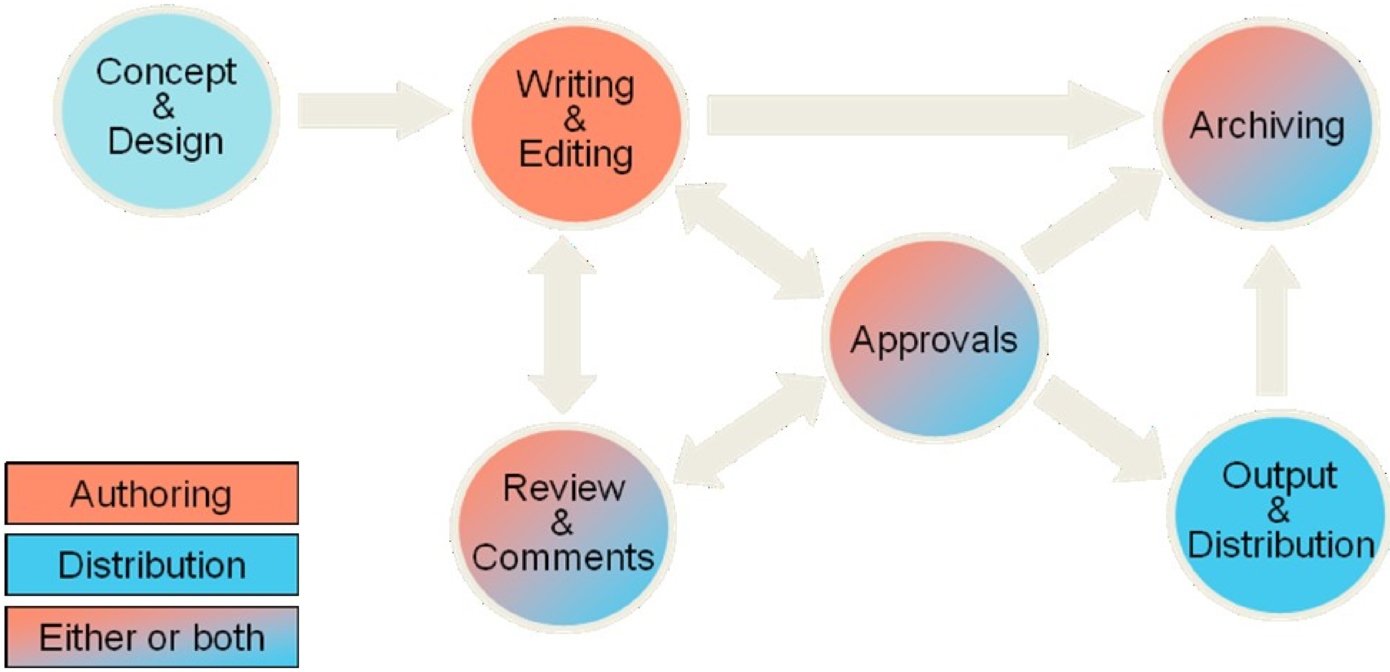

Figure 1: OpenXPS has been designed for the 'distribution' phases of the document lifecycle

The key trends that drove the development of the OpenXPS format are outlined below, followed by a description of the role of OpenXPS in providing a foundation for solutions that address these requirements.

- **Increasingly Complex Documents.** As design becomes an increasingly important part of all forms of communication, documents that are graphically richer, with more colour content, more advanced graphical content and imagery, have become increasingly common. More recently, applications have made the creation of rich documents easy for information workers (as well as design professionals) and frameworks that simplify the development of such applications are also emerging. Along with the advent of accessible applications, networked access to information components means that documents are often assembled from a variety of sources. Taken together, these factors are creating an explosion in the quantity and quality of rich documents, and in some cases are placing a strain on the systems in place to process content.

- **Document Workflows and IT Cost.** Document life-cycles are becoming increasingly complex, with capture, creation, editing, review, publication, printing and archiving in addition to all sorts of processing steps. With the increase in complex, intelligent, documents for creation, editing and review cycles, there's been a increasing need for formats that address the specialized needs of capture, print, archive and publication without compromising the formats used for the authoring stages of the cycle. In support of these workflows, customers are still looking for solutions to a number of problems:
  - First, PCs haven't eliminated paper, but they can be used to eliminate filing cabinets. Solutions are moving to electronic paper for filing and archiving, but these often don't integrate well with print parts

of the workflow, requiring explicit user interaction to flip between archiving/filing to electronic paper and printing to physical paper.
  - Secondly, within the average enterprise a significant portion of the overall IT budget is spent on printing and print-related costs. That represents a significant financial and environmental area for cost reduction given that assessments indicate that a large proportion is *unnecessary* print.
  - Third, corporations are looking for ways to manage unstructured data stored in documents across the company. There are a number of factors driving this, including competitive advantage, managing innovation, succession management, compliance, governance and risk management. Solutions in these areas often need to integrate across existing IT infrastructures for printing and document processing.

- **Intelligent Peripheral Devices.** After mice, keyboards and displays, printers have traditionally been the most connected peripherals in the PC ecosystem. As those devices get more sophisticated, with the emergence of multi-functional peripherals consisting of a scanner + printer + fax +…, customers are asking to integrate them more tightly into existing IT infrastructure including centralized user management and workflows that cross the boundaries from physical to electronic paper and then back again.

- **3D Content for Display, Print, and Exchange.** 3D content is becoming an increasingly important part of document workflows, both on the creation side where applications combine 3D data with conventional 2D documents, and on consumption where 3D printing is expanding the need for print formats that are capable of handling 3D.

OpenXPS has been developed to provide a foundation layer that enables solutions that address these trends.

Firstly: OpenXPS provides the ability for the rich content to get to and from intelligent devices – and, importantly, for that content to move at full fidelity, because as content becomes richer customers still expect devices to render it accurately. OpenXPS also enables content to be communicated efficiently, important both for performance and bandwidth consumption.

Secondly: OpenXPS enables devices to participate fully in document workflows and it connects document workflows to document peripherals more seamlessly than has previously been the case. For example, with OpenXPS devices can consume content streams like a conventional sequential page description language, but then access content randomly from a locally cached version of the same document.

Thirdly: OpenXPS supports electronic paper with rich content in document workflows – ensuring a high degree of reliability between authoring application and subsequent retrieval, often sometime later, from an archive.

When considering the features and benefits provided by OpenXPS and outlined in this paper, and when comparing the feature set with those provided by alternative formats, it’s useful to consider the key trends that OpenXPS enables solutions to address.

Features and Benefits

The feature set for OpenXPS has been carefully selected with a view to optimizing for the requirements of electronic paper, and only for the requirements of electronic paper.

Restricted Feature Set

Like all formats, the feature set of OpenXPS is limited. However, with OpenXPS there was an explicit design goal to eliminate two categories of features:

1. Features unnecessary for electronic paper
2. Features that, by their nature, provide an increased opportunity for malicious code

Restricting the feature set in this way provides OpenXPS with two very important benefits. Firstly, the format is relatively simple. This makes implementation straightforward and therefore increases interoperability. Secondly, the format is, relatively speaking, inherently secure. By this we mean two things: the format does not include features that may have side effects unexpected by a user (for example, the format does not contain programmatic elements and the reproduction of content is therefore static); and that by being straightforward to implement, the format enables implementations that are likely to be more secure to attacks from malicious code.

[Note that although this restrictive feature set is a benefit for OpenXPS, the presence of those same features may be considered a benefit in other formats that aim to address a much larger set of scenarios than OpenXPS.]

Stream and Random Access

The structure of OpenXPS enables implementations to access content efficiently using streamed and random access models, where, for example, content within an XPS Document can be accessed directly using the packaging model provided by OPC.

Streaming capabilities enable the format to be efficiently used for communication between systems, for example from PC to device, or from device to device, and enable consuming implementations to start to process content before receiving the whole document. In some cases a consuming implementation (a consumer) may be processing a document before a creating implementation (a producer) has completed generating the content, enabling simultaneous creation and consumption. The streaming model in OpenXPS is created by a combination of capabilities:

- Mark-up order within fixed page content defines the rendering ordering of elements
- The OPC container model enables flexible ordering of parts and separates logical structure of content from physical structure
- OpenXPS provides interleaving recommendations for generation of content for efficient consumption
- OPC provides a mechanism to separate parts into component pieces, which enables very granular interleaving of content
- OpenXPS Discard Control provides a mechanism for consuming implementations to efficiently manage resource caching strategies

This last capability is extremely powerful as it enables sophisticated physical streaming models that rely on interleaving of mark-up and resources without requiring consumers to deal with the additional complexity at the logical level.

OPC provides a comprehensive random access model that enables efficient processing of OpenXPS Documents when access to the entire package is possible. Random access capabilities are enabled by a combination of capabilities:

- OpenXPS content is referenced by URI conformant pack part names enabling efficient mapping between references embedded in part contents and the logical structure of the package
- OpenXPS implements the OPC relationships mechanism across the specification, ensuring that relationships between parts are well defined and easily discoverable. For example an implementation that is splitting pages can quickly identify all the resources used by specific pages without being required to analyze page content.

In addition to providing excellent support for random access and streamed models of content access, OpenXPS provides flexibility in how implementations further optimize between modes. For example, system implementations have flexibility over whether and when to optimize for low-resource consumption.

Ubiquitous XML

OpenXPS makes extensive use of XML, both in the container used to package contents and in the fixed document mark-up. Defining the page content using an XML-based language provides significant benefits for automated creation and processing of document content and also enables OpenXPS to easily integrate with existing document processing applications. For example, XSLT can be used to translate between OpenXPS fixed document mark-up and other XML-based representations.

Using XML as a foundation across OpenXPS provides benefits for implementations since there are existing robust XML parsing components available across platforms and with a wide variety of features and capabilities. This is a significant advantage as robust and secure low-level parsers are often a difficult area to implement correctly. XML also provides a standard mechanism to enforce basic validity of the document mark-up. This assists interoperability by providing a standard contract between consumers and producers; aids implementations by providing a mechanism to rapidly and automatically detect errors; and also provides an additional level of verification for deployed products that helps to prevent differences in the ecosystem from either gradually bifurcating the format over time, or of gradually increasing implementations costs as implementers are forced to support both specification compliance and cross-product compatibility and products diverge.

Existing Formats for Common Resources

OpenXPS makes use of many existing formats for common resources such as fonts, colour profiles and image formats. This provides OpenXPS with two advantages:

- By enabling the use existing ‘off-the-shelf’ components, implementation costs and time to market are reduced, assisting interoperability. In many cases this can also help product quality and security.
- By leveraging existing commonly used formats, OpenXPS can interoperate well with workflows that aggregate existing content into electronic paper documents and workflows that use existing components to analyze or process content in electronic paper. When combined with an XML-based mark-up for page content, this is a very powerful capability.

Resources Always Included

OpenXPS requires that resources are always included in the package. OpenXPS does not support external references to formats, or the reliance on an assumed or typical set of resources that consuming implementations may be expected to have. This provides two very important benefits. Firstly, users can rely on resources being available. For example, if retrieving from an archive after a long time users can be confident that the file contains all the required components necessary to accurately render it for an output. Secondly, including resources eliminates rendering issues caused by any differences in 'default' resources that are available to a consuming device, for example differences in font weights caused by using fonts supplied by different font foundries for the same typeface.

Static Fixed Content

OpenXPS only supports static and fixed content. This means that:

- The location of all content on every page is fixed. Elements are precisely sized and positioned and there are no layout elements. For example text is placed in individual runs, there is no capability to layout or reflow text content.
- Content is static. OpenXPS does not support animation or moving content, ensuring that pages always display consistently with how they would print.
- No variable content. OpenXPS does not support fields within the document page that respond to external events or other events such as the status of the document.
- No document automation. OpenXPS does not provide capabilities to automate document viewing that might otherwise result in different rendering or reading order depending on the consuming implementation.
- No programmatic interpretation or rendering that can result in differences in reproductions as a result of programming implementation differences that are outside of the purview of the OpenXPS specification.

Only Device Independent Colours

All colour information in OpenXPS is defined using device independent colour spaces, there is no concept of device dependent colour. This means that implementations always have an unambiguous description of the intended colour and are able to process correctly. There is another benefit in that users learn to rely on the colour provided by implementations when using the format since there's no get out clause to just 'do what you think is the right thing'.

Compatibility

OpenXPS has been developed with an easy progression from existing XPS implementations in mind. XPS and OpenXPS content can be easily distinguished, and XPS consuming implementations can be updated to support OpenXPS content with only minor modifications.

Technical Overview

This section provides a technical overview of the OpenXPS format. As an overview, it covers the main aspects of the format briefly. Readers are directed to the formal OpenXPS specification for an in depth description of the format.

Design Goals

OpenXPS was designed with a number of high level goals. These goals are summarized in the following bullets.

1. Provide a format optimized for use in the following roles:
    a. A **Page Description Language** for communicating document content to and from document peripherals such as printers and scanners.
    b. A **spool file format** for processing and temporary caching within an operating system when interoperating with document peripherals such as scanners and printers.
    c. An **electronic paper format** for publication, sharing, storage and long term archiving of document content in an electronic paper form that provides a facsimile of the physical paper representation.
2. Provide a format that can be trusted. OpenXPS was designed to be trustworthy on a number of levels
    a. Content stored in OpenXPS can be easily accessed. This means that the format is well documented, and also that the implementation overhead is minimal. In extreme cases, content can be access by expanding the OpenXPS file into a directory structure using a ZIP archiving tool.
    b. The OpenXPS format is designed to be a secure format, and to facilitate implementations that are also secure. Two significant trends make this a very important aspect of the design. Firstly, as operating systems and web browsers become increasingly hardened against malicious code attacks, the malicious perpetrators have moved attention to other ubiquitous components and attack vectors, document formats being just one high value target. Secondly, one of the emerging trends that OpenXPS addresses is intelligent document devices and increasing device-to-device interoperability. In such workflows, not only do devices become targets for malicious attacks, but there is frequently no user in the loop to verify that some triggered event is expected.
    c. OpenXPS is a static fixed format. Users can trust OpenXPS to communicate content just as it would have printed. Further, OpenXPS does not include variants that require users to differentiate between different intended use cases and adapt a mental model of what is expected from the format as a result.
    d. Support for digital signatures to verify document content and protect against repudiation.
3. Enable interoperability across a wide range of implementations
    a. OpenXPS is a format designed to be easy to work with, both for products provided by large corporations or small start-ups and for in-house development teams addressing specific point problems in an enterprise.
    b. OpenXPS bridges across physical and electronic paper and is designed for easy implementation in devices with a wide range of hardware capabilities.
    c. OpenXPS uses XML and existing formats to minimize cost and maximize quality of implementations and to maximize the reusability of existing content already encoded into popular formats.
4. Only electronic paper
    a. OpenXPS is designed to just address the needs of electronic paper and not to compromise the performance for that purpose by adding features required for other roles. Other formats do a good job of addressing broader scenarios with additional functionality and should be used wherever appropriate.

5. Support for emerging requirements for exchange of 3D data within 2D documents and for print formats capable of supporting both 2D and 3D printing of 3D content, to bridge between 2D and 3D output.

Technical Features

At the highest level, an OpenXPS document consists of XML-based mark-up that describes the graphical content of pages within a document together with the resources required to render that content description combined together as a single package. The package that combines these elements, or parts, together is defined by the Open Packaging Conventions (OPC) defined in ECMA-376 Edition 1 Part 2.

Open Packaging Conventions (OPC) defines a ZIP-based archive and an XML-based structure that serves to identify individual parts and their types, defines relationships between the parts, and provides generic features such as descriptive metadata, a digital signature foundation, and a low-level mechanism that enables fine-grained streaming of content.

Packaging

OpenXPS Documents are OPC compliant packages. OpenXPS documents can be processed by implementations that conform to the OPC Standard, for example accessing descriptive metadata. OPC also provides extensibility points for OpenXPS and is responsible for key foundation features, including the relationships model, low level interleaving using pieces of parts, and digital signatures.

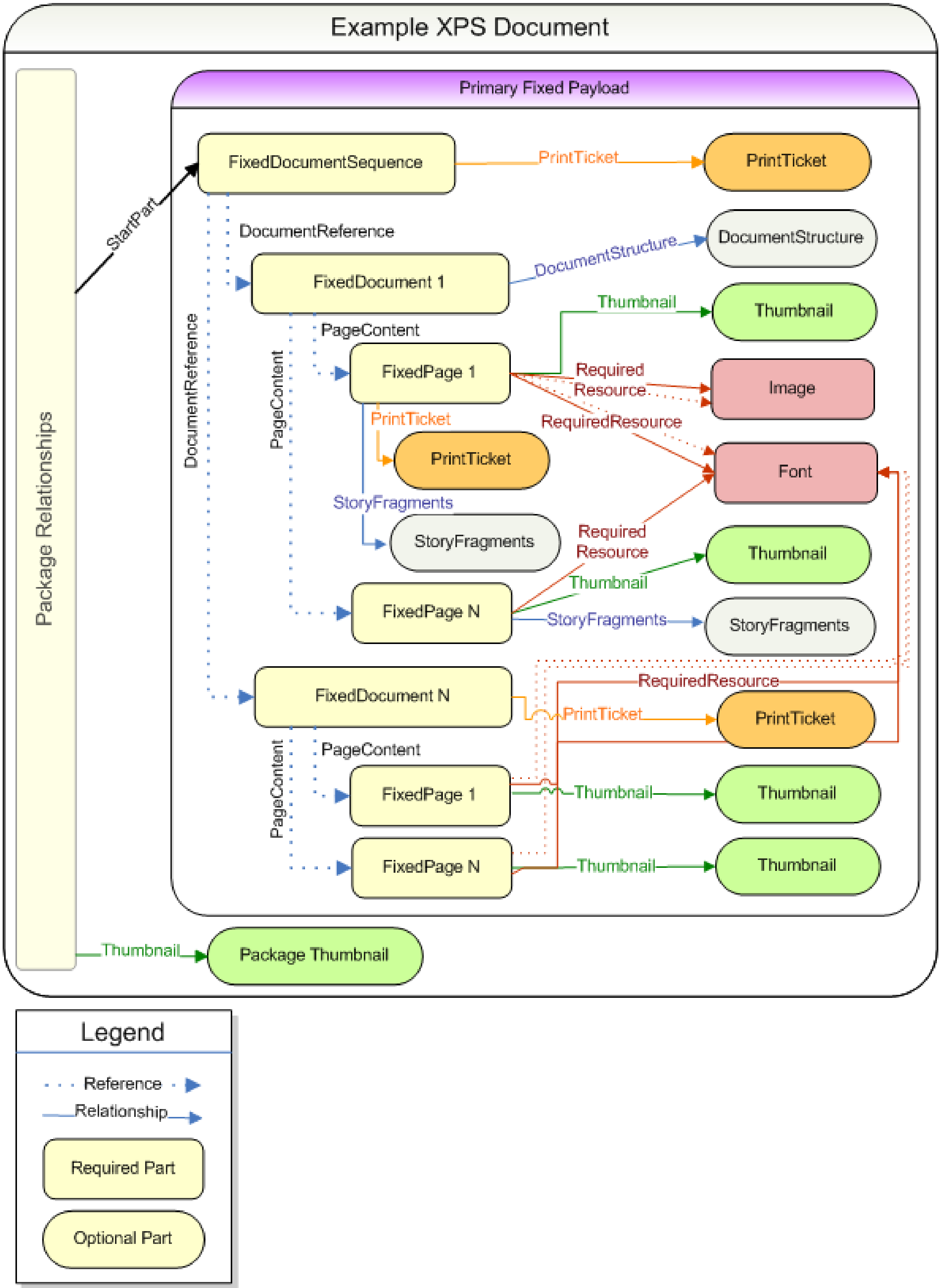

Figure 2: Internal OpenXPS Document structure builds on the Open Packaging Conventions Standard

Fixed Payload

OpenXPS Documents consist of Fixed Pages grouped together into a Fixed Document. There may be multiple Fixed Documents within an OpenXPS package, and an OpenXPS document may also include additional content (see Extensibility). OpenXPS documents are self-contained. That is they contain all the resources required to render the content of the document.

OpenXPS Document Mark-up

Page content in OpenXPS is defined using an XML-based mark-up known as OpenXPS Document Mark-up. OpenXPS Document Mark-up describes all graphical content on the page using the following constructs:

- **Glyphs element:** represents a fixed run of text
- **Path element:** represents a geometry that can be filled, or stroked, by a brush
- **Brush elements:** represents a brush that can be used to fill or stroke a path. OpenXPS provides brushes that can be defined for solid colours, gradient colours, images, and OpenXPS Document Mark-up.

Content can be grouped using a **Canvas element** and positioned on the page using a **Matrix Transform element.**

OpenXPS Document Mark-up is designed such that it provides a compromise between two design goals, firstly that the mark-up be parsimonious. That is that it should include a minimal set of primitives from which any graphical content can be constructed. Secondly, the mark-up should provide a compact representation of the content so that implementations can efficiently handle the mark-up even when uncompressed.

Resources

The fixed page mark-up that defines the page content references resources stored within the same package. Resources define content for fonts, images, colour profiles, print tickets and remote resource dictionaries.

Fonts

In OpenXPS text is specified in the form of a fixed run of glyphs that are defined by a referenced font resource. OpenXPS supports font resources that conform to the OpenType/Open Format Format standard using both TrueType and CFF outline forms. OpenXPS also defines mechanisms for font subsetting such that only the required glyphs are included in a font as well as mechanisms to support font foundry licensing flags that control use for embedding.

Images

Image content in OpenXPS is defined using the following file formats:

- JPEG compressed image data with JFIF and EXIF-based containers
- PNG compressed image data
- TIFF compressed and uncompressed image data
- JPEG XR compressed image data, including lossy and lossless compression methods

Support for JPEG, PNG and TIFF images meets the implementation requirements to leverage existing components and the scenario requirements to enable easy aggregation of existing content. Support for the new JPEG XR Standard enables high dynamic range, extended precision, high bit depth, and wide gamut image content to flow through OpenXPS workflows.

Remote Resource Dictionaries

Remote resource dictionaries facilitate sharing of common resources between pages within an OpenXPS document. Content that is repeated multiple times throughout a document can be defined in the remote dictionary and referenced for each occurrence.

Colour Representation

OpenXPS Document Mark-up enables colour information to be specified in a variety of colour space encodings. Further, OpenXPS requires that all colours be specified in device-independent coordinates to ensure that colour content is unambiguously defined. OpenXPS provides support for Standard sRGB and scRGB colour space encodings and, when used with ICC colour profiles, for other RGB, CMYK, named colour, and N-Channel colour encodings. OpenXPS uses standard ICC Colour Profiles to define colour content for all colour spaces except sRGB and scRGB.

OpenXPS provides the ability to specify colours with transparency and provides blending between elements with varying degrees of transparency, both at the object level and at the pixel level.

Print Ticket

OpenXPS provides comprehensive support for the specification of rich document content. Specification of how to process that content, for example what device-specific finishing options should be applied or how content processed with OCR should be handled is not covered in the OpenXPS Standard. However, to facilitate association of configuration settings defined in other standards, as well as within proprietary implementations, OpenXPS defines a Print Ticket mechanism that leverages the relationships mechanism to enable both the definition of processing instructions for OpenXPS jobs and an efficient document and page-level override mechanism for modifications to those instructions.

Extensibility

The OpenXPS format provides several extensibility points.

1. **OPC Parts:** OpenXPS defines the required parts for OpenXPS Documents. Additionally, a package containing OpenXPS content may include additional well-defined static parts. OpenXPS also enables hybrid formats where OpenXPS content may be included in what would otherwise be a proprietary file format such that users may access the standard OpenXPS content where a consumer for the proprietary version of the content is not available. This mechanism enables sharing of vendor-specific content without requiring an ecosystem of consuming applications to be first deployed.
2. **OPC Relationships:** The OPC relationship mechanism enables information to be associated with OpenXPS parts without modification of those parts. For example, a system that wanted to encode information about which pages within an OpenXPS document included elements that were not completely opaque could do so by defining a ‘NotOpaque’ part and relationships from that part targeting pages to which this new property applied.
3. **Mark-up compatibility:** OpenXPS supports Mark-up extensibility and compatibility defined byECMA-376 Edition 1. This enables the specification of new mark-up without breaking consumers that aren’t aware of the new constructs. It also provides a fallback mechanism to the future revisions of the specification can be backwards compatibility whilst still introducing new features.

3D Content

OpenXPS provides support for optional 3D content by defining a 3D brush element that supports X3D content. In mark-up, such 3D content is bracketed by an extensibility fallback block that defines an alternative 2D representation of the content so that consumers that are not 3D aware can continue to provide a complete rendering of the document.

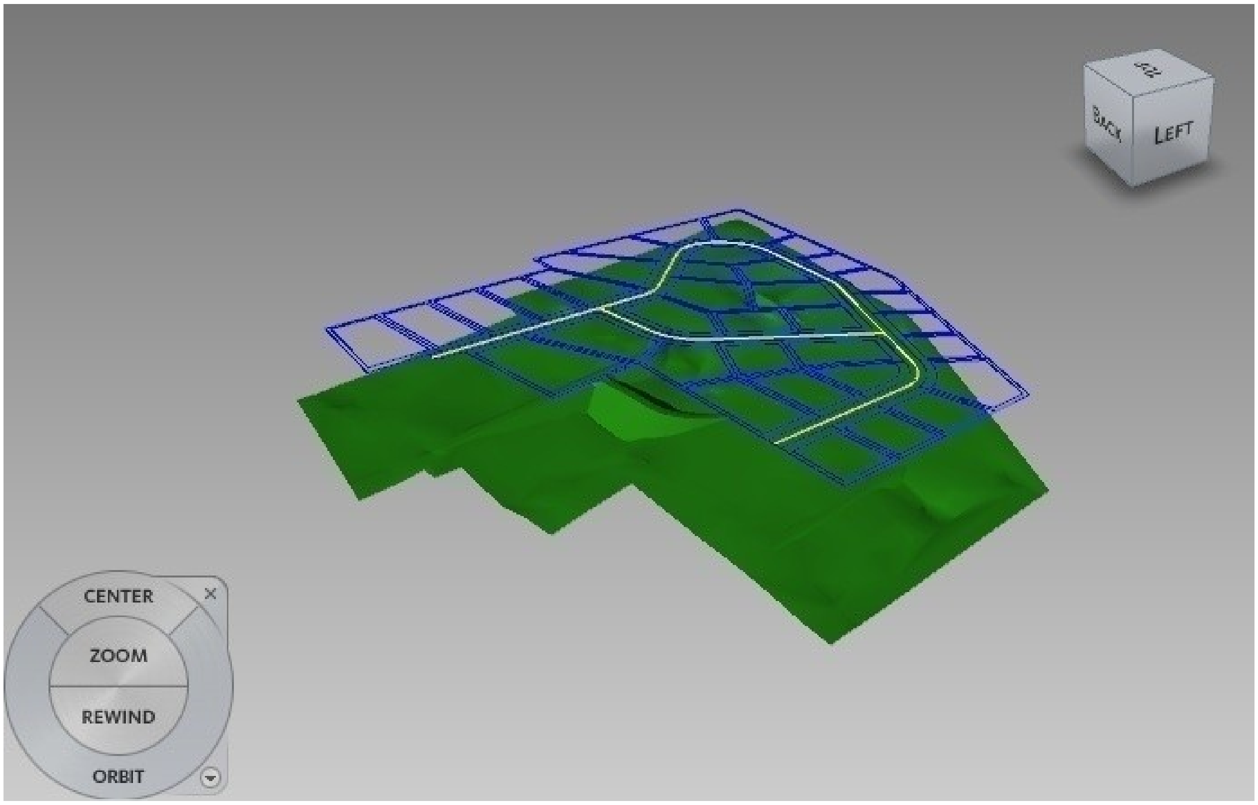

Figure 3: OpenXPS provides the ability to describe representations of 3D content.

Additional Features

OpenXPS also includes a number of features designed to enhance the experience of electronic paper. These include:

- Ability to specify internal and external hyperlinks associated with page content
- Document outline structure to specify table-of-contents-style document navigation
- Document structure to assist document accessibility, navigation and content extraction
- Digital signature capabilities

OpenXPS Differentiation

Many of the features and capabilities supported by OpenXPS are supported, individually, by other formats. For example there are formats that, like OpenXPS, use XML as the foundation for describing content and there are formats that provide a fixed layout representation of content on a page, but OpenXPS has both these characteristics. OpenXPS is therefore differentiated from other formats and standards by a *combination of features* that are not found, combined together, in other formats. The primary features that differentiate OpenXPS when combined are listed below:

- Fixed layout representation of page content
- All resources must be included in the package
- XML-based mark-up used to define page content, file structure, and metadata
- Feature set limited to only that required for efficient electronic paper
- Minimal implementation requirements
- Minimal threat footprint for malicious code
- Supports fine-grained streaming and random access

A brief history of OpenXPS

While developing what later became the Windows Vista operating system, Microsoft enhanced the Windows printing subsystem to support key emerging scenarios, including solutions for some of the trends outlined in the 'Rationale for an Electronic Paper format' section above. As part of this update, Microsoft decided to enhance the spool format used by Windows for printing with capabilities to support higher quality and more efficient printing. This format eventually became the XPS Document format.

Development of the XPS format was informed by extensive input from across the document and printing industry. This ensured that the format capabilities met the wider needs of the ecosystem, and that the format could be implemented independently on a variety of platforms and operating systems. During the development period several drafts of the specification were made public. Feedback was solicited from, and provided by, a wide range of stakeholders and interested participants. During this period an independent implementation of the format, to verify that the specification was complete and unambiguous and to demonstrate its suitability for implementation on multiple platforms, was also commissioned. The first version (version 1.0) of the format was published in October 2006 and remains available for download from http://www.microsoft.com/xps. To date, support for the format has been implemented in a wide range of products and solutions from a variety of vendors.

Following the publication of XPS version 1.0, it was determined that the best interests of the format, industry, and customers, would be served by industry taking ownership of the format and the future direction of any development. In June 2007, following an approach by a number of companies interested in the future of XPS, Ecma International formed Technical Committee 46 (TC46) to work on a formal standard for OpenXPS.

TC46 undertook to develop a standard that, while fundamentally compatible with existing implementations to maximize interoperability, provides additional clarity and industry compatibility to enable further usage of the format.

Date	**Event**
April 2005	Microsoft XPS draft specification released Comment and feedback invited
October 2006	Microsoft XPS 1.0 published.
June 2007	Ecma International announces formation of TC46 to standardize XPS.
July 2007	Ecma TC46 meeting in Cambridge UK. Hosted by Global Graphics.
September 2007	Ecma TC46 meeting in Tokyo Japan. Hosted by Fuji Xerox.
January 2008	Ecma TC46 meeting in Las Vegas USA. Hosted by Autodesk.
April 2008	Ecma TC46 meeting in Oxford UK. Hosted by Software Imaging.
July 2008	Ecma TC46 meeting in Redmond USA. Hosted by Microsoft.
September 2008	Ecma TC46 meeting in Yokohama Japan. Hosted by Canon.
December 2008	Ecma TC46 meeting in Geneve Switzerland. Hosted by Ecma International.
March 2009	Ecma TC46 meeting in Lexington USA. Hosted by Lexmark. TC46 approval of the Final XPS draft.
June 2009	Ecma General Assembly to vote on proposed OpenXPS Standard

Contributors to OpenXPS

A large number of experts from many companies and organisations worked on the Ecma OpenXPS Format, meeting physically and virtually through conference calls and email. In addition to companies and organizations that hosted face-to-face meetings (see table above), one way of visualizing the scope of participation is the following list of experts, and organisations represented, that attended the TC46 face-to-face meetings in Cambridge (UK), Tokyo (Japan), Las Vegas (USA), Oxford (UK), Redmond (USA), Yokohama (Japan), Genève (Switzerland), Lexington (USA)[1]:

Mr. N. Crews (Autodesk)
Mr. Kevin Tracy (Autodesk)
Mr. H. Mori (Brother Industries)
Mr. K. Fukaya (Brother Industries)
Mr. H. Kawamoto (Canon)
Mr. K. Dei (Canon)
Mr. S. Yamamura (Canon)
Mr. T. Oishi (Canon)
Ms. K. Matsuyama (Canon)
Mrs. D. Chiaramonte (Ecma International)
Mr. O. Elzinga (Ecma International)
Dr. I. Sebestyen (Ecma International)
Mr. R. Jaeschke (Editor)
Mr. M. Masui (Fujitsu)
Mr. T. Hashizume (Fuji Xerox)
Ms. Y. Fujiwara (Fuji Xerox)
Mr. M. Ohta (Fuji Xerox)
Mr. M. Bailey (Global Graphics)
Mrs. A. Mobbs (Global Graphics)
Mr. G. Godderidge (Hewlett Packard)
Mr. J. Papke (Hewlett Packard)
Mr. A. Garcia (Hewlett Packard)
Mr. T. Smith (Hewlett Packard)
Mr. T. Nohnishi (Konica Minolta)
Mr. F. Akiyama (Konica Minolta)
Mr. J Grams (Konica Minolta)
Mrs. A. McCarthy (Lexmark International)
Dr. A. Ford (Microsoft)
Mr. D. Merrill (Monotype Imaging)
Mr. C. Kush (Monotype Imaging)
Mr. D. Rÿbrock (Océ)
Mr. M. Dupré (Océ)
Mr. R. Porter (Pagemark)
Mr. M. Vrhel (Pagemark)
Mr. G. Free (Pagemark)
Mr. G. Karakousis (Panasonic)
Mr. K. Hirakawa (Panasonic)
Mr. Y. Aoki (Panasonic)
Mr. J. Mater (Quality Logic)
Mr. S. Kang (Quality Logic)

[1] List compiled from TC46 meeting minutes.

Mr. K. Suzuki (Ricoh)
Mr. G. Soord (Software Imaging)
Mr. J. Williams (Software Imaging)
Mr. P. Winwood (Software Imaging)
Mr. T. Bush (Software Imaging)
Mr. C. Bearchell (Software Imaging)
Mr. A. Saito (Toshiba)
Mr. A. Fujiwara (Toshiba)
Mr. T. Shimmyo (Toshiba)
Mr. C. Miyachi (Xerox)
Mr. D. d'Entrecastaux (Xerox)
Mr. M. Goldwater (Zoran)

OpenXPS Looking Forward

Following OpenXPS acceptance by the Ecma General Assembly as a formal standard, the priority for TC46 will be to encourage widespread adoption of the format and to facilitate a smooth transition of parts of the ecosystem that have already adopted the predecessor XPS format.

Beyond the initial OpenXPS standard, TC46 will remain engaged in the ongoing maintenance of the format, and will analyze any requirements that may lead to a future edition of the standard. In particular, the committee will assess issues raised during the development of the first edition and that were considered out of scope.