

Maximising the value of QuarkXPress content

Written by

Gavin Drake

Marketing Manager, Easypress Technologies

gavin.drake@easypress.com

www.easypress.com

February 2002

Table of Contents

SUMMARY	3
Overview	3
Key Points	4
INTRODUCTION	5
The Issue	6
WHY MAXIMISE THE VALUE OF QUARKXPRESS CONTENT?	6
WHAT'S CHANGED?	7
HOW DO THE NEW DIGITAL MEDIA DIFFER FROM PRINT?	7
GETTING CONTENT OUT OF QUARKXPRESS.....	8
Content, Layout and Style Information.....	9
Content and Style Information.....	9
Content Only	9
The Solution.....	10
CREATING STRUCTURED CONTENT FROM UNSTRUCTURED CONTENT	10
Identification of Content	10
The Relationships between 'pieces' of content	12
XML AS A DATA FORMAT FOR STRUCTURED CONTENT	12
What is XML?.....	12
The Benefits of XML	13
GETTING CONTENT INTO XML.....	17
Copy and paste.....	17
Outsource.....	17
Custom Software Solutions	18
QuarkXPress-to-XML tool	18
Making conversion more efficient.....	19
QUARKXPRESS-TO-XML – THE PROCESS AND THE TOOLS	19
QuarkXPress to XML without structure.....	19
QuarkXPress to XML without a predefined DTD.....	20
QuarkXPress to XML with a predefined DTD.....	20
THE PRODUCTS.....	21
Avenue.quark	21
Atomik	21
QuarkXPress 5.0	22
InDesign 2.0	22
FIND OUT MORE	24
ABOUT EASYPRESS TECHNOLOGIES	25

Summary

Overview

This discussion document looks at how publishers can maximise the value of their content held in QuarkXPress.

The value of content held in QuarkXPress cannot be leveraged beyond print publishing while it is still in the proprietary QuarkXPress format. Content can be extracted from QuarkXPress in various ways but the most valuable solution is to end up with the content in a structured, granular and open format that can be repurposed in many different ways. Structure and granularity are two of the key factors that enable content value to be maximised. Creating structure from content that is held in QuarkXPress - in an unstructured format - is a difficult task, hence the lack of practical solutions for achieving it.

XML is an ideal format for holding structured content and therefore a key enabling technology for maximising the value of content held in QuarkXPress. The primary benefits of XML include flexibility, being an open standard, separation of content from structure and searchability. The challenge therefore becomes how to convert unstructured QuarkXPress content into structured, valid XML.

There are a variety of methods for converting QuarkXPress-based content into XML but the most cost-effective, timely and accurate is to use QuarkXPress-to-XML conversion XTensions software. Many XTensions software products are available for converting QuarkXPress content into XML, however they can produce wildly differing results, from simply extracting styling information through to creating fully structured and validated XML.

Avenue.quark, Atomik and QuarkXPress 5.0 are examples of products that can produce fully structured and validated XML with Atomik providing the most automated solution for this purpose.

XML will continue to increase in importance in relation to cross-media publishing, just as PDF has in relation to print publishing over the last 10 years. To maximise the value of content held in QuarkXPress, publishers quite simply need to get their content into XML. XML is without doubt the present and future of cross-media publishing.

Key Points

The Market

- In order to generate new revenues, publishers should now take their content and reuse it in the ever-widening array of available digital media.
- The publishing industry will need to move from print centric and application centric workflows to content centric workflows. This is likely to be one of the most significant challenges facing the publishing industry over the next 5 years.

XML

- The promise of XML is a universal data format, not owned by a single company, but embraced by all vendors and users to enable information to be stored and shared by anyone, anywhere.
- The benefits of XML include its flexibility, it being an open format, enabling the separation of content from presentation, storing of granular content, its searchability, enabling of communication and data exchange between applications and ease with which it can be manipulated.
- XML is currently the only export format from QuarkXPress that yields structured content
- The alternatives to XML include HTML, PDF and SVG however each of these has significant disadvantages as formats for maximising the value of content.

QuarkXPress

- In order to be able to reuse content in digital media without unnecessary manual intervention and processes, content needs to be held in a structured format. Almost all of the export formats from QuarkXPress produce unstructured content.
- There are three main categories of export from QuarkXPress; Content, Layout and Style Information; Content and Style Information; and content only.
- There are two key challenges to automating the process of converting unstructured content into structured content: identification of individual 'pieces' of content and correctly identifying the relationship between these 'pieces'. Without this, content cannot be put into a structured format.
- Granularity is of key importance if the value of content held in QuarkXPress is to be maximised.

Methods

- There are four main methods of getting QuarkXPress content into XML and these include: copy and paste; outsourcing; custom software solutions; and QuarkXPress-to-XML tools.
- The QuarkXPress-to-XML tools available fall into three main categories: QuarkXPress-to-XML without structure; QuarkXPress-to-XML without a predefined DTD; and QuarkXPress-to-XML with a predefined DTD. Of these, QuarkXPress-to-XML with a predefined DTD is considered the most cost-effective.
- There are several products on the market for extracting QuarkXPress content as XML but only a few are able to convert QuarkXPress content into XML using a pre-defined DTD.

Introduction

In the world of professional publishing, there is no other page layout tool that comes close to the market share and sheer popularity of QuarkXPress. Despite competition, particularly from Adobe - currently taking on QuarkXPress with InDesign - QuarkXPress shows no signs of seceding its throne in the short term.

The consequence of the dominant position of QuarkXPress over the last decade is the accumulation of literally billions of pages of content held in the QuarkXPress file format. Arguably a publisher's greatest asset is its content and that content is typically stored, electronically at least, in QuarkXPress.

The aim of this discussion document is to look at how this valuable content, stored in QuarkXPress, can be of maximum benefit to publishers. The intention is not to look at the commercial aspects of content reuse, of which there are many methods, but to look at the technical barriers to maximising the value of content held in QuarkXPress, an issue that is still perplexing IT, publishing, production and Internet managers at publishers throughout the world.

The Issue

Why maximise the value of QuarkXPress content?

The answer to this question is probably reasonably obvious. Putting the design and layout aside for one moment, content creation is a fixed cost for publishers. Therefore once you have created content, you should be able to reuse it as many times and in as many media as you wish, without incurring further content creation costs. It doesn't matter how many copies of a magazine, book or newspaper you print, how many digital media you use, or even how many people consume your product, the content creation costs do not increase. It is only the cost of distribution that changes (we count the paper and printing as distribution costs). Therefore, it stands to reason from a commercial perspective, that to maximise the content's value, you need to maximise its reuse.

To a greater or lesser extent publishers have recognised this for many years. For example, if you look at magazine publishers, licensing deals for overseas versions of magazines are commonplace. A foreign edition of a magazine is created by another publisher, using translated content from the original magazine, for which the originating publisher receives a licence fee. Book publishers have similar licensing arrangements.

The content is usually distributed to the licensee either just in printed form or as original QuarkXPress files. Up until the mid-nineties, this was probably as far as maximising the value of QuarkXPress content went. After all, what else could you do to maximise the value of content apart from sell as many copies of your publication as possible and licence it in as many countries as possible?

Although publishers have reused QuarkXPress content and maximised its value as far as possible, they certainly have not optimised how that content has been stored or managed.

What does that mean? If you relate it to financial savings, it's like storing your money under the mattress. Sure you can get to it, however if you have it stored in a bank, the same money is available from tens of thousands of cash machines around the world, can be seamlessly converted into any one of hundreds of currencies, invested in stocks and shares or bonds, used to make Internet purchases etc. Your money becomes flexible, fluid and accessible.

QuarkXPress is the mattress equivalent in the publishing world. When content is exclusively stored in QuarkXPress it is not flexible, fluid or accessible. When print was the only communication media for content none of this mattered. Now there is a world beyond the printed page.

What's Changed?

In the past ten years communication has been revolutionised, starting with CD-ROMs followed by the Internet – the first truly mass-market, global communication platform. This has facilitated digital media such as the Web, email, WAP phones, PDAs (Personal Digital Assistants) and more recently, Interactive Digital TV.

At first many publishers saw this as a threat to their businesses. The publishers seemed to view this in much the same way that rail companies viewed the advent of air travel. In publishing, “electronic only” content brands sprung up seemingly overnight and claimed that they would transcend traditional publishers. The dotcom bubble has now comprehensively burst, normal economic practices are in operation and thankfully most publishers now see the changing landscape of digital media as an opportunity, not a threat. It is an opportunity that they, as content providers, are best placed to benefit from.

That's what has changed but how does this relate to maximising the value of content? First and fundamentally, publishers have the opportunity to take their content and reuse it to generate revenues in an ever-widening array of digital media, i.e. maximising its value. The crucial thing about digital media is that it has a whole new set of requirements that are completely different to the requirements for content in print.

Ultimately what is required is a shift from print centric workflows to content centric workflows. This is likely to be one of the most significant challenges facing the publishing industry over the next 5 years.

How do the new digital media differ from print?

The reason that this change in workflow is inevitable is that the “new media” available to publishers have such fundamentally different requirements and characteristics from print. You can interact with the content, search for keywords or phrases, annotate, personalise the content to receive only what you are interested in, view content in many different formats all the way from in-depth news stories on the Web through to headlines and sound bites on a mobile phone. Publishers have the ability to distribute content almost instantaneously to multiple devices across the globe updating content in some cases on a minute-by-minute basis. As a publisher, you cannot provide any of this functionality while your content is in QuarkXPress.

We'll take a brief look now at a few examples of the media available to publishers and how their requirements vary from print.

CD-ROM

This is a medium that has in many cases been forgotten with the growth of the Internet but the CD-ROM was the first new medium that at least caused people to start thinking about alternative workflows. In many cases however, the CD-ROM became an extra, and usually expensive addition to the end of existing print workflows. The content was manually grabbed from the page layout tool, sometimes re-keyed and then the CD created.

Relative to print, the CD-ROM is very different from the printed page. You can store millions of pages of content on a single CD-ROM – no more need for a bookshelf full of the Encyclopaedia Britannia. Users can interact with CD-ROMs, content can be

hyperlinked for easy navigation, search facilities are available, the inclusion of video and sound i.e. multimedia.

The Web

Although the CD-ROM market caused companies to examine their workflows, the real kick was the advent of the World Wide Web (www). The Web shares many similarities with CD-ROM but is further differentiated from print in that the content can be distributed globally in seconds, updated instantly and limitless amounts of it can be made available. You can also gain instant feedback from users, interact in real time, deliver personalised content and link content to products that are available to purchase online.

Wireless devices (including mobile phones and PDAs)

Many wireless devices will likely merge into the Web category at some point in the future but for now continues to be a slightly different beast due to bandwidth restrictions, screen sizes and cost of transmission. Specifically wireless devices typically only receive a subset of the content that would go to a Website. There are still very few companies deriving revenues from publishing to mobile phones and so delivery needs to be low cost and automated. SMS alerts are an exception to this in some cases e.g. event-driven alerts.

InteractiveTV (iTV)

There are now over 8 million households in the UK with digital television and the UK is not alone in its take up. Television could well overtake computers as the most used device for viewing digital content and the primary viewing device for Internet content too. Like wireless devices, Interactive Television that is available to digital TV viewers tends to be affected by download times, and can take less content than would be put on a scrollable Web page for viewing on a computer. Close ties with e-commerce are likely to be another of the differentiating points for iTV.

Syndication

This is nothing new of course. As we've already mentioned, publishers have been licensing content to one another for many years. However this has previously been a very offline, manual process. Digital content is driving syndication, presenting publishers with huge potential revenue opportunities. The best syndication model for digital content is automatic, keeping costs to a minimum. To facilitate automated digital content syndication, content purchases usually require content to be in an industry standard format and commonly use the ICE (Internet Content Exchange) protocol.

As you can no doubt see from these few examples of digital media, the requirements are significant and the differences with print media obvious for all to see.

Getting content out of QuarkXPress

We've already predicted that one of the most significant challenges facing the publishing industry over the next five years will be the move from print-centric to content-centric workflows. The reason we've said the next five years and not immediately is that print continues to be the dominant revenue generator for almost all publishers and hence there is a reluctance to re-engineer the workflow to incorporate digital media. Additionally, there are still a lack of software tools available that deliver content-centric workflow without having significant impacts on existing print workflows.

With this in mind and assuming that the final version of the content will continue to end up in QuarkXPress for the immediate future, the challenge becomes how to take this

final version content in QuarkXPress and reuse it in the digital media described above. This is where we get to the crux of the issue facing publishers – getting content out of QuarkXPress.

In the age of space stations, genetic engineering and computers you can keep in your pocket, you would think that getting content out of QuarkXPress would be an easily solvable problem. QuarkXPress itself, like many software applications, uses its own proprietary file format. This presents the biggest hurdle in being able to reuse the content in other media. As you would expect from a mature software application though, there are many different ways of getting the actual content, if not the native file, into other formats. The appropriateness of each depends on the destination of the content. More detail on these export formats is provided later in this document. For clarity the various formats available have been divided between; content, layout and style; content and layout; and content only.

Content, Layout and Style Information

There are two main formats that fall into this category.

Postscript

The language of printing devices, developed by Adobe. Slowly being superseded by PDF.

PDF (Portable Document Format)

Based on Postscript. Has significant advantages over Postscript in that the file sizes are much smaller, it can be searched, hyperlinked and book marked, does not need the originating application that created the document and can be used on the Web.

Content and Style Information

There are all manner of formats available for receiving text content out of QuarkXPress with the styling information intact. The degree to which the styling information is maintained varies from format to format. QuarkXPress allows you to export text content in standard word processor formats including MS Word, WordPerfect and RTF (Rich Text Format). More recently HTML (style not layout) has been added to the standard set of text export formats. None of these formats will enable you to identically reproduce the same styling automatically in another application but they will get out a large amount of the formatting. Quark also has its own formatting language, called QuarkTags. This enables the export of all of the styling information with the text content. The only catch is that only Quark applications are able to perfectly reconstruct the QuarkTags back into the exact same styling again. This is simply because one of the unique things for any page layout tool is its H&J engine (Hyphenation and Justification), which determines how text is flowed on the page.

Content Only

This is the case that is potentially the most interesting for organisations wanting to reuse QuarkXPress content in other media. There are now two main options to choose from and the difference between them could not be starker. At one end of the spectrum you have ASCII text, which enables you to get all of the content out of QuarkXPress without any styling information. At the other end of the spectrum is XML – a format that cannot be achieved with QuarkXPress 4.1 on its own but can be using add-on products such as avenue.quark and Atomik. The main difference between XML and ASCII text is structure. In the next section we describe why this is so important.

The Solution

Creating structured content from unstructured content

In order to be able to reuse content in digital media without unnecessary manual intervention and processes, content needs to be held in a structured format. This is essential for its efficient reuse. Since QuarkXPress documents are typically formatted for human viewing rather than computer interpretation, extracting the data for use in systems that require structure is a difficult challenge. As we have discovered, almost all of the export formats from QuarkXPress produce unstructured content. How do you turn something that is basically unstructured in QuarkXPress into something structured?

It is worth saying now that this is not a trivial matter, highlighted by the fact that so few companies have got even close to achieving the “structure from non-structure solution”. There are two key challenges to automating the process of converting unstructured content into structured content: identification of individual ‘pieces’ of content and correctly identifying the relationship between these ‘pieces’. Without this, content cannot be put into a structured format.

Identification of Content

Making content easy to identify and read on a page is a function of good design. By laying out a page well and, where necessary, labelling content, as human beings we can look at a document and understand with almost 100% accuracy what each individual piece of content is. Although we do this almost instantaneously do not be fooled into thinking that this is a simple process. Human beings can identify the faces of people they know; yet it has taken years for computer systems to get even close to being able to do this. We are only now beginning to see computer systems that are able to do automated recognition of criminals on CCTV by looking at the mathematical compatibilities between the features on a person’s face.

If you think about a newspaper page, how do you know what the heading is, which text contains the table of contents, the temperatures for each city in the weather section, the body text for a book review etc.? The answer is that there are several things we can use to help us. For example we use the relative font sizes of the text, the position of the content on the page or even the actual content itself i.e. what it says.

Asking a computer to identify content in the same way as a human, would require an extremely high level of artificial intelligence (A.I.), combining visual information with pattern and grammar matching. This would be further complicated if it had to support multiple languages, industry vocabularies, tables, graphs and so on.

So it might sound as though humans have a distinct advantage in the process of identifying content. However computers have the edge in one department, which is the accuracy with which they can determine font types, font sizes, colours and other

typographical information compared with the naked eye. Could you tell the difference between 44pt Arial and 43pt Arial? Probably not.

Therefore one of the most obvious ways for a computer to identify individual pieces of textual content in a document is by using the text's typographical properties. If a piece of text is styled as Helvetica 44pt Bold, it is a heading. If a piece of text is Arial 10pt, it is body text and so on and so forth. This really plays to the strength of computers because they can very quickly and easily use simple rules about typographic properties to identify what a piece of text is supposed to be.

The issue with using an entirely typographical approach is how to handle a situation where two pieces of text use the same basic typographical information but represent different types of content. For example, if the author's name and the picture caption on the same page are using identical fonts, colours and sizes, how does the computer know which piece of content is which? Maybe it could look at the proximity of the caption text to the picture or the proximity of the author text to the headline but this is starting to get into A.I. again.

In QuarkXPress you actually have a potentially more powerful tool than typography with which to identify content correctly and hopefully avoid the type of ambiguity highlighted above. QuarkXPress style sheets enable users to style pieces of text in a document and in essence label the text. When a user changes a style sheet, every piece of text that has that style sheet applied to it, will be updated automatically with the style change. This makes document design much quicker but also means that content can be identified correctly. Microsoft Word uses a very similar concept. By using style sheets to identify content instead of just the typographical information, there is less room for ambiguity. Using the same example, the author text can be styled (labelled) as an author and the caption styled (labelled) as a caption.

So there's the answer then, use style sheets in QuarkXPress for all text content and then identification of content is easy!

If only life were that simple. There are downsides to this approach too. Let me start by saying designers are fantastic. Unfortunately, designers don't always use style sheets or if they do, they do not always use them consistently. The reasons for this vary between laziness, a reluctance to have any restrictions placed on their 'creativity' or to be fair to designers, more often than not because deadlines are tight. This last reason is a serious issue because when deadlines have to be met and time is short, the number one aim is to have the document ready for printing on time. In this case designers will make sure text looks correct but often don't have time to stick to the constraints of style sheets.

There is also a further situation where the military-like imposition of style sheets does not work which is in relation to archived documents. There are already literally billions of QuarkXPress documents containing valuable content that publishers would like to reuse. In this case, the aim is to reuse the content as cost-effectively as possible. If it were necessary to open every document and apply style sheets when they had not previously been used, the time and cost involved would make reusing archived content held in QuarkXPress economically unjustifiable.

What we can conclude then is that using either of these two methods in isolation has serious pitfalls that will compromise the integrity of the content being identified. However if the typographical identification method is used in combination with the style sheet identification method, then the pitfalls of both can be almost completely overcome.

The Relationships between 'pieces' of content

What constitutes "structured content" is not just a simple one-on-one identification of content on the page with descriptive labels. What starts to make the content useful is when the individual pieces can be related to each other. There is a fantastic example of this on an advert on the London Underground trains at the moment. The advert has text that reads something like this...

The vicar ate...

...my lovely green parrot...

...and the king ordered his execution...

...making for a delicious and filling Christmas dinner.

Obviously these four partial statements don't belong together but how did you know that? I expect because, although mildly amusing, they just don't seem to fit with our view of reality. Of course a computer would not know that. So how do you get a computer to form the correct associations between pieces of content on a page so that a title matches the body copy or a caption goes with an image?

There are actually several ways of enabling a computer to do this. The ideal solution is for a human to actually identify the relationships between the content for the computer, for example you could manually identify that the headline, the by-line, author, first paragraph, body text and photograph all constitute a single article in a newspaper. This is possibly the most accurate method however it is also the most labour intensive. Additionally it is open to human error and humans make more mistakes than computers. If you have to go to the trouble of manual identification of content, the process of reusing content from QuarkXPress through automation software is actually only fractionally faster than the manual copy and paste process.

Therefore we need another method of relating pieces of content together and in actual fact the facilitator of this is XML (eXtensible Markup Language). Before we explore just how automated extraction and relating of like content can take place using XML, we will examine XML itself in more detail along with the alternatives.

XML as a data format for structured content

What is XML?

XML stands for eXtensible Markup Language. The promise of XML is a universal data format, not owned by a single company, but embraced by all vendors and users to enable information to be stored and shared by anyone, anywhere. XML in and of itself is simply a file format. If you believed all you read in print (and online) you would be forgiven for thinking that XML is the answer to dynamic Web publishing, cross media publishing, content syndication, even world peace.

On its own XML solves none of the above but in relation to publishing, if your content is stored in XML, then you have a good starting point for achieving all of the above and more (possibly apart from world peace).

The Benefits of XML

There are many benefits to XML and these vary dependent on what you are going to use it for but here are just a few of the main ones.

Flexibility

XML is a markup language. This means that a set of tags is used to describe content. XML is more than just a markup language though. It is a markup language that can be used to create other markup languages that are based on XML. This is the real flexibility of XML.

Unlike HTML where the set of tags is fixed and defined by the WC3 committee, e.g. means bold and <I> means Italics, with XML anyone can define a set of tags for their XML. This has obvious benefits for publishers in that they can define a set of tags that reflect their content precisely. Publisher XYZ can create an XML-based markup language, XYZ XML, used just by them. Alternatively there are thousands of XML-based markup languages already in existence that can be downloaded from various Websites.

Open Format

One of the most obvious benefits of XML is that it is an open standard. No one company owns XML. You don't have to pay royalties to use it or need a special reader application to access it. In fact you can open XML in any text editor or word processor. It is both machine readable and human readable. This is a powerful combination. It also means that once content is held in XML it is 'future proof'. Now that is terminology that software companies have used to sell solutions for many years and there are many companies in the world that have been sold "future proof" solutions only to find the reality somewhat different. With XML this just is not the case. You can rest assured that no matter how the technology world changes, content held in XML will be accessible, useable and reusable for years to come.

Of course the reason we stress this point is that this is the whole problem that publishers are currently trying to overcome – how to reuse content held in a proprietary file format.

Separation of content from presentation - Create once, publish anywhere

I've included these two together because one facilitates the other. When you store your content in a proprietary file format like QuarkXPress, you store the content, i.e. the text and images, directly with the presentation (styling and layout) of that content but usually without the structure. The same applies to content stored in HTML. In both cases, all of the formatting and layout options are inextricably linked to the content. This makes content reuse extremely difficult because to reuse content in other media, you need to separate the content from the layout. When content and presentation are tied together either the whole lot has to move from media to media or significant manual processes are necessary.

XML is a format that can facilitate content being stored independently of its presentation. When it is time to deliver the XML content to its destination media, templates holding the presentation information can be applied to the content to deliver the same content to different media with a completely different look and feel in each media. The most commonly used media where this is true is the Web. By applying an XSL (eXtensible Stylesheet Language) to the XML, HTML can be created dynamically for delivery to a Web browser (See the XSL section later in this document).

Granularity

In order to maximise the value of content, publishers need to have their content in a very granular form, i.e., content that can be 'sliced and diced' to a fine level that ensures customers receive what they want and only what they want.

The way to achieve this is to enhance the content by adding rich metadata to the content. Metadata is a definition or description of data - or rather it is information that describes what the underlying content is about. If content is being stored in XML form, the metadata that describes the content can be contained in XML tags that are bundled with the piece of content. XML enables content to be described at a very granular level hence increasing its value.

Searchability

XML enables you to richly describe your content. This makes it infinitely easier to search content to find specific information. It also enables personalisation whereby only the content that is requested by or applicable to a user can be delivered. In relation to the Web, content stored in XML enables users to make very precise searches where they can search for content in a specific context.

Communication and data exchange between applications

Making different applications from different vendors (or sometimes even the same vendors) talk with one another or exchange content between them automatically can be an expensive and time consuming thing to achieve. System Integrators and consultancies have thrived on this type of work for many years. XML is facilitating communication between different applications even on different platforms in a way that has simply not been possible before. Most major software vendors are either supporting XML already or have certainly announced that they will in the near future. In a world where no one software developer can provide a complete solution and solve every issue, interoperability between applications is extremely important. Two common standards based on XML that facilitate content exchange and communication are ICE (Information & Content Exchange) and SOAP (Simple Object Access Protocol). Both of these, like XML, are open standards.

Data manipulation

Content stored in XML can very easily be converted from one XML structure to another. For example if you stored your content in XML structure 'A' but a partner company stored their content in XML structure 'B', conversion from structure 'A' to structure 'B' can be achieved quickly and easily. If you have your content stored in HTML or QuarkXPress documents, manual processes would probably be the only way that conversion of data structure would be possible (without the tools described below).

What are the alternatives to XML output and how does XML compare with them?

XML is currently the only export format from QuarkXPress that yields structured content, if you can indeed export it. All of the alternatives (with the exception of manual database connectivity extraction tools) result in unstructured content. These alternatives were described earlier in this document but we'll now look in greater detail at the more interesting formats, particularly in relation to cross-media publishing, as much confusion still exists amongst many QuarkXPress users as to what the differences are between these formats and XML. This is not to say that the other formats are not good formats, they are just not ideal as a storage format for maximising the value of your QuarkXPress content.

HTML (HyperText Markup Language)

This is of course the most common delivery format for the Web and therefore for many seems an obvious choice to go from QuarkXPress to HTML. This involves taking the QuarkXPress page and converting it as near as possible into HTML, ready for the Web.

Advantages

- Easy to get HTML out of QuarkXPress using third party XTensions software such as BeyondPress, WebXPress and now QuarkXPress 5.0.
- Content can be used directly on a Website without being converted into another format.

Disadvantages

- Can be time consuming and hence costly. The extracted HTML can require post export work in an HTML editor before it can be used online. Additionally, the QuarkXPress document may need to be redesigned to be appropriate for the Web or other digital media.
- Using flat HTML is the least efficient way to manage a large website. Publishers are increasingly moving towards database driven websites where content is stored independently of the final presentation.
- Difficult to repurpose - Although HTML is an open standard it is not a good data storage format and extremely difficult to repurpose - a key requirement for publishing to multiple digital media. This means that you have in essence simply traded one closed format (QuarkXPress) for another format that is equally difficult to repurpose.
- Not so usefully searchable - As HTML deals with formatting rather than structuring content, the content can be searched but only using free text searches or by searching the meta tags, which are only page specific. XML enables searching at a far more granular and hence more narrowly defined level, as long as the search tool takes advantage of the structure. An example of this might be searching for all of the televisions on a Website that cost less than £500 and support digital surround sound. This granularity of searching could mean the difference between getting five tightly defined search results and several thousand.
- Content has less value because it has almost no granularity (see section above on XML granularity).
- Difficult to reproduce QuarkXPress layouts on the Web. Although HTML has come a long way from HTML 1.0, it is still difficult to reproduce content in HTML that looks the same as it does in QuarkXPress. This is because QuarkXPress is a high-fidelity page layout tool that works by positioning boxes and Bezier lines anywhere on the page. There are special effects, like text on a path, which cannot be reproduced in HTML. You can create collages of images, or place boxes at any angle, create colour blends and more. These are just a few examples of layout and design effects that can be created in QuarkXPress but are not features of HTML.

A further complication is that Web pages can be continuous i.e. they don't have any set heights and widths. In contrast print pages in QuarkXPress always have constrained heights and widths. (this has been addressed in QuarkXPress 5.0 with the Web document feature). Another thing to consider is the physical size of the pages. For a magazine spread or a tabloid newspaper, the page size just wouldn't work on a computer screen.

All of this means that QuarkXPress pages can be reproduced in HTML but generally only if they are simple in layout and don't utilise QuarkXPress specific features. To get an accurate reproduction of more complex layouts, you will either have to render many of the QuarkXPress objects as images and suffer the

resulting slow download times or compromise on the layout. Of course the added disadvantage of converting complex items into images is that image files are not searchable.

PDF (Portable Document Format)

You often hear PDF and XML compared and contrasted as alternatives to one another. In reality XML and PDF are very different and have very different applications. They are both important and should work as complimentary formats not in competition. The important thing is to use them for what they are best at.

Advantages

- Relatively easy to create PDFs from QuarkXPress.
- The page is reproduced for the Web exactly as it looks in print so design consistency is maintained (provided font licensing issues are overcome)
- Can now be viewed on PDAs (Personal Digital assistants) such as Palm and Pocket PC devices.
- Relatively easy to protect content e.g. not allow printing or editing of the PDF or copying of the PDF content.

Disadvantages

- Not a fluid format i.e. it cannot easily adapt to the layout of the device it is being viewed on (unless using Acrobat 5.0 and text flow information is embedded in the PDF).
- Content is not stored in a granular format so it can only be reused in page or publication sized chunks (see section above on XML granularity).
- Print formatted documents are not always appropriate for viewing in a Web browser.
- Not easily searchable – i.e. you cannot identify the prices for product ABC because the PDF doesn't know what a product is or which text contains the price.
- Content is held in a proprietary, unstructured format.
- Files can be large and take longer to download than a text-based format
- Not human readable – PDF is only machine readable i.e. you need a proprietary viewer to be able to read the content – XML is both human and machine-readable.

SVG (Scalable Vector Graphics)

XML itself does not hold anything other than text. If you have images, video clips or other media, these are simply referenced in the XML as external files. SVG is an XML-based language for describing vector images and brings to the Web, the same high fidelity layout and printing capabilities that are available in print. It has the potential to have a revolutionary effect on the Web but only time will tell if it will take off as a format. Additionally Macromedia Flash and Shockwave already exist and have the market share.

Advantages

- Small file sizes without a loss of quality
- Content is in an open standard format i.e. not proprietary
- Content is selectable
- High quality printing possible directly from the Web browser

Disadvantages

- Same disadvantages to PDF with the exception of file size
- Additionally, there is currently no readily available export function from QuarkXPress into SVG. Quark has shown SVG output from QuarkXPress with new XTensions software they are developing. Availability for this software is not yet known.
- A plug-in is required to enable Web browsers to display SVG

Getting content into XML

So assuming that XML is currently the best format to store content in, facilitate its reuse in other media and hence maximise its value, what is the best way to get your content into XML?

Copy and paste

This is still probably the most popular way to repurpose QuarkXPress content into XML. It involves someone sitting in front of a copy of QuarkXPress, manually identifying content, the relationships between content and then copying and pasting the content into a database, content management application or template, ready for its reuse.

Advantages

- Conceptually very simple and has low technical requirements
- Does not require third party software

Disadvantages

- Extremely time-consuming
- Labour intensive
- Expensive
- Often carries with it unwanted formatting information such as line breaks
- Doesn't always support extended characters
- Doesn't enable image references to be easily transferred
- Prone to Human errors
- Difficult for untrained operators to reconstruct the content structure

Outsource

An alternative to copying and pasting yourself is to let someone else have the pain, i.e. outsource. There are many companies that specialise in data conversion. However as copying and pasting is so labour-intensive, another popular option is to use conversion companies located in countries with lower labour costs than your own e.g. India or China. You send them the QuarkXPress documents and you get back the content in whatever format you ask for, including XML. It has to be said that not all data conversion companies use the copy and paste method and some have sophisticated tools for this process or employ third party applications such as the XML tools described below.

Advantages

- Someone else has the pain

- In certain circumstances, it can be cost-effective

Disadvantages

- Can take a long time because a third party company is involved and especially when the work has to be sent abroad
- Loss of control over the timing, the process and the content itself
- Is typically expensive because ultimately the service provider has to make enough margin for it to be a viable business.
- Sometimes inaccurate - specifically if the conversion house uses inaccurate software tools for the conversion or performs the copy and paste function with low paid, unskilled workers, who probably have no interest or knowledge of what the content is.

Custom Software Solutions

In an attempt to solve this issue themselves, some publishers have developed bespoke software solutions to solve this problem. These include custom XTensions software and scripting tools such as AppleScript.

Advantages

- Cuts out the manual copying and pasting
- Gives publisher direct control over the process
- Can result in structured XML content

Disadvantages

- The usual disadvantages of developing custom software e.g. very expensive to develop, expensive to update, development risks of a company undertaking software development when it is not their core competency.
- Time has shown that these solutions rarely deliver accurate results and more often than not require much content manipulation and cleanup after the export. We have even seen publishers develop custom solutions but return to manual copying and pasting because of the lack of accuracy.
- These solutions tend to be “hand coded” for the specific task in hand and therefore very inflexible if the requirements change.

QuarkXPress-to-XML tool

This is a relatively new option for publishers. The first tools on the market for performing this task appeared a couple of years ago. A few far-sighted companies, including Quark themselves, saw the benefit of XML as a content storage format and hence as a facilitator for extracting content from QuarkXPress into XML.

This resulted in the development of the ‘Troika’ suite of tools by Quark, one of which is now known as `avenue.quark`, and the development of Atomik, by UK software developer Easypress Technologies. These were two of the first such solutions on the market, both of which we’ll cover in more detail later in this document along with further expansion of the whole subject of QuarkXPress-to-XML conversion.

For now, it is sufficient to say that these software solutions offer publishers a viable alternative to copying and pasting, outsourcing and custom solutions. Using the right QuarkXPress-to-XML tool can also bring publishers greater granularity of the extracted content and significant reductions in time and cost in reusing QuarkXPress content.

The reason I say, the right tool, is that there are several tools on the market that perform this QuarkXPress-to-XML function but not all of them are the same and as

you'll discover in the next section, QuarkXPress-to-XML can mean a lot of very different things and therefore bring varying degrees of success.

Making conversion more efficient

Before we look at the various types of QuarkXPress-to-XML tools available, let's look briefly at how the entire process can be made more efficient. We've already discussed the value of style sheets in QuarkXPress but this one is important enough to mention again. If you are only outputting to print, whether you use style sheets or not is really just a decision based on personal preference and the timesavings that they bring to your workflow. There is of course now another angle to this and that is the reuse of the QuarkXPress content. It might be difficult to convince designers of the value of utilising style sheets if it adds to their production and design times, but the commercial imperative for publishers to do so is significant. Most importantly it makes it possible to cost-effectively reuse content and leverage its value in other media with 100% accuracy.

The use of character-level style sheets can further increase the value of content by increasing its granularity. This is something you hear a lot about now but the value of content is related to its granularity. For example, if you have a travel brochure, you might want to label the pricing, the hotel rating and facilities as such. This would enable users on your Website to search for only holidays that meet their specific criteria e.g. just holidays in Canada that feature a 5-star hotel with an indoor swimming pool. Without the use of character style sheets, it would be extremely difficult to extract this specific information automatically. Another implication would be if a third party company wanted to licence only specific content rather than all of it.

Tables are another example of where the use of style sheets will have commercial advantages. Take the case of annual reports. The time taken to repurpose annual report financial tables into a structured format is significant, generally achieved through manual copying and pasting processes. This is because this data has to be 100% accurate. Unlike other types of content there is absolutely no room for error.

Ideally you want to have every figure in the table stored separately in the XML. If this is achieved, users of the data can for example query specific values within the financial information and compare it with other companies. By applying style sheets accurately, this can be achieved automatically.

QuarkXPress-to-XML – the process and the tools

QuarkXPress-to-XML on the surface of it sounds reasonably simple like QuarkXPress to HTML or QuarkXPress to text. How many ways can there be to do it? The answer is many.

QuarkXPress-to-XML XTensions fall into three main categories: those that actually extract styling information rather than structure; those that don't constrain the export to a predefined DTD; and those that structure the XML extraction based on the rules defined in a DTD (Document Type Definition). These might seem like minor differences but in actual fact the difference in the resulting XML is significant. We'll look at these three variations now.

QuarkXPress to XML without structure

This is not really QuarkXPress to XML at all, at least not in the context of XML as a format for holding structured content. Products that do this are simply extracting the

content using tags to describe the styling information. In many ways this is no different from using any of the standard text export formats in QuarkXPress such as QuarkTags. No structural information is extracted, nor in most cases is any image information.

QuarkXPress to XML without a predefined DTD

There are several XTensions solutions on the market that fall into this category. They generally rely entirely on the use of style sheets, which can present significant issues as we have already discussed. The resulting XML is simply the textual content of the page with XML tags surrounding each style sheet that was used in the QuarkXPress document. Some of the XTensions will even create a DTD for you that matches the XML extracted.

Disadvantages

- **The structure of the data** - With this method of extraction, the XML that is extracted has not been validated against your defined document structure. Therefore rather than specifying a DTD that describes how your publications are structured, what fields must be contained in the XML and a definition of each type of content, the content is extracted without any reference to your specific publication. The resulting XML is an entirely flat data structure. Therefore if you produced a magazine that contains sections and the sections contain articles, this hierarchical structure will not be reflected in the XML. This process can only work for very simple documents with few 'XML Elements' and where style sheets have been used religiously.
- **The quality of the XML** - With the 'DTDless' approach to extraction you have no idea what you are going to get out of the QuarkXPress document. There are no constraints on the data to be extracted. The old adage, GIGO (Garbage In Garbage Out) springs to mind. As there is no validation process against your DTD at the point of extraction, there is no way for the XTension to assess what fields need to be in the resulting XML. This means that the definition of a news story could contain 5 fields for one news story but 7 fields for another. They are both news stories but both have different data structures. The end result is that significant time must be spent turning the extracted XML into something that is usable. This post processing can take anything from hours to weeks depending on the complexity of the publication and quantity of content being extracted.

QuarkXPress to XML with a predefined DTD

The other method of extracting content from QuarkXPress as XML is by basing the extraction on a predefined DTD. There are several benefits to this approach, the most important of which is the accuracy of the extracted content and the immediate reusability of the XML. By basing the extraction on a DTD, the XTension is immediately able to constrain the extraction to fit an expected data structure and an expected set of fields. This vastly improves the accuracy of the extraction and makes the XML instantly reusable with very little or no post processing. If a hierarchical structure exists in the QuarkXPress document, this can be reflected in the XML.

Now is a good time to return to the earlier discussion on the issues of turning unstructured documents into structured documents. People often wonder how you can use styling information alone to turn an unstructured document into a structured one and with the exception of very simple documents, you can't. A computer is never going to be able to determine structure for you. This is the key to why basing the extraction from QuarkXPress on both styling information and a predefined document structure is the most successful way of automating the process and producing usable XML. The computer can use the DTD to tell it what it is expecting to find and then the styling information to indicate what it is looking at. The result - structured, predictable and immediately usable XML, with the minimum possible user-intervention.

The Products

There are several products on the market for extracting QuarkXPress content as XML. However some of them, as we have described above, either only extract styling information in XML format or extract structured content but not based on a pre-defined DTD. In both cases, the value of the XML is limited in relation to cross-media publishing. Hence the value of the QuarkXPress content is not immediately maximised. It is beyond the scope of this document to cover all of the available QuarkXPress to XML tools on the market so we have therefore stuck to two products that base their extraction on a pre-defined DTD that the user can select and control. The products we have chosen are avenue.quark – Quark's own solution to address this problem and Atomik – an automated solution for the extraction of QuarkXPress content into XML. Additionally we will mention how QuarkXPress 5.0 and InDesign 2.0 fit into this market.

Avenue.quark

First released in 2000, avenue.quark is a semi-automated solution for extracting QuarkXPress content as XML. It works by associating styling information with elements in a DTD. We say semi-automated because user intervention is required in each QuarkXPress document that you extract content from. The user must first identify the order in which content must be extracted from the page i.e. the box ordering, as well as associate content that is grouped together. Once this is done, text can be dragged and dropped onto the XML palette and the content is extracted from that particular group (or sequence as it is called in avenue.quark). Additionally, images must be manually dragged and dropped into the XML structure.

Avenue.quark is somewhat limited in the complexity of the DTDs that it supports. This makes it difficult to extract complex documents from QuarkXPress with hierarchical structures. It also makes it difficult to use a single DTD for many different types or layout.

On the plus side, the user can preview the XML in QuarkXPress, content can be posted directly through an IP address to 3rd party systems and there is an XML Import product available as beta software. Users can also use freely available XTensions software for avenue.quark to create ebooks in Microsoft Reader format (Windows only). Avenue.quark is best placed to handle simple publications where complex DTDs are not required and where the volume of content to be extracted is low.

More information: www.quark.com

Atomik

Atomik takes a different approach to avenue.quark in that its aim is to automate the extraction as much as possible and requires no pre-configuration of the QuarkXPress document in advance of the extraction. As with avenue.quark, Atomik works by associating styling information (whether that be style sheets or just typographics) with elements in a DTD. The point regarding style sheets is important, as some solutions require the user to adhere strictly to style sheets. In reality this doesn't always or often happen, so the flexibility of not requiring style sheets to be used throughout a document, or even at all, is very important. As we have already highlighted, there are benefits to more strict enforcement of style sheets though.

The support for and interpretation of complex DTDs, combined with advanced box ordering preferences, enable Atomik to extract the contents of entire QuarkXPress documents with no user intervention. With the click of a button, the entire document is

added, making the process extremely fast, efficient and cost effective. Unlike avenue.quark, Atomik also extracts images automatically. Tabular data can be extracted in XML table format based on the geometric positioning of the tabular content.

As well as AtomikXT - the XTensions software component, Atomik also includes Atomik Enhancer. Atomik Enhancer is an XSL processor that enables the user to transform the extracted XML into other XML structures. A simple example of this is adding metadata to the XML. More complex transformations might include turning the XML from one DTD structure into another or re-ordering the content. These are all processes that IT staff would typically have to perform on the extracted XML, often taking considerable time. Atomik Enhancer makes the whole process much quicker and pain free.

Atomik is ideal for medium to high volume publishers due to its automation and flexibility.

More information: www.easypress.com

QuarkXPress 5.0

The new version of the leading page layout tool includes HTML design and output features as well as XML import and export. We have already spoken about the inappropriateness of HTML as a format for holding structured content although HTML export will of course be useful for smaller scale users of QuarkXPress that are not concerned with cross-media publishing.

In relation to the XML functionality, this is provided for by the inclusion of avenue.quark 1.0, along with the XML import XTensions software. Therefore as far as XML functionality is concerned, QuarkXPress 5.0 does not take publishers beyond the current combination of QuarkXPress 4.1 and avenue.quark.

More information: www.quark.com

InDesign 2.0

Adobe has included both XML import and export functionality in the new version of InDesign. Users can create 'tags' in their InDesign documents, and then associate text and images with these tags. Users can define structure for their documents by manipulating these tags and their relationships within the InDesign structure palette. Additionally InDesign 2.0 includes support for attribute data.

The approach taken by Adobe for the XML export is a cross between avenue.quark and the 'DTDless' approach of some of the lesser QuarkXPress-to-XML tools. The result is a very manual XML creation process that involves the user tagging content in the InDesign document and creating a structure for it. A degree of automation can be achieved by mapping paragraph style sheets to tags, effectively 'auto-tagging' your document. As we have already mentioned above, this requires the stringent use of style sheets. You can also automate the creation of the tags by importing them from an XML file, although InDesign 2.0 just imports the tag names not the structure or attribute information that might be in the XML.

There are restrictions on the mapping of text to elements in that you have to tag text frames before you can tag any of the text within a frame. This seems an unusual approach in that this assumes that the text frame itself has meaning when in actual fact, it is usually the text that has meaning and the number of text frames used and their content depend entirely on the designer.

InDesign 2.0 does not support DTD or schema languages, which means InDesign will only create well-formed but not valid XML (see “QuarkXPress-to-XML – the process and the tools” for more details).

We should say that Adobe has implemented a good initial XML set in InDesign 2.0 for a product that previously had no XML support. They have taken some interesting approaches to both the XML Import and Export but our feeling is that without DTD support and validation, in terms of XML export, the product falls short of the necessary automation that high volume publishers require to make content reuse fast and cost-effective.

More information: www.adobe.com

Find out more

If you are new to XML, or even if you're not, there are a lot of issues to get your head around in relation to QuarkXPress-to-XML conversion. Therefore if you would like to find out more, have any questions after reading this positioning document or would like to give Atomik a test run, please let us know and we will be only to happy to help you. You can either call us on +44 (0)20 7704 0285, complete an online enquiry form on our Website (www.easypress.com) or fax back the below form to +44 (0)20 7704 6627. We look forward to hearing from you.

Enquiry Form

Please contact me to arrange an initial, no-obligation, consultancy meeting to discuss my QuarkXPress-to-XML requirements

Please use BLOCK CAPITALS

Which of the following best describes your company's main line of business (please tick ONE box only):

Name: _____	<input type="checkbox"/> Advertising Agency	<input type="checkbox"/> Publishing-Directories
Position: _____	<input type="checkbox"/> Automotive	<input type="checkbox"/> Publishing-Magazines
Company: _____	<input type="checkbox"/> Consulting	<input type="checkbox"/> Publishing-Newspapers
Address: _____	<input type="checkbox"/> Consumer Products	<input type="checkbox"/> Publishing-Other
Town: _____	<input type="checkbox"/> Financial Services	<input type="checkbox"/> Oil/Gas/Chemical
County: _____	<input type="checkbox"/> Food & Beverage	<input type="checkbox"/> Pharmaceutical
Postcode: _____	<input type="checkbox"/> Government	<input type="checkbox"/> Retail
Country: _____	<input type="checkbox"/> Health Care	<input type="checkbox"/> Telco
Tel: _____	<input type="checkbox"/> High Tech Manufacturing	<input type="checkbox"/> Transportation
Fax: _____	<input type="checkbox"/> Publishing-Books	<input type="checkbox"/> Travel
email: _____	<input type="checkbox"/> Publishing-Catalogues	<input type="checkbox"/> Utilities/Energy
	<input type="checkbox"/> Publishing-Contract	<input type="checkbox"/> Venture Capital
	<input type="checkbox"/> Publishing-Corporate	<input type="checkbox"/> Other

If Other, please state: _____

Faxback +44 (0)20 7704 6627

About Easypress Technologies

Easypress Technologies (www.easypress.com) develops easy-to-use Web-based tools that empower publishers to create, manage and publish content online efficiently and profitably.

Create Once - deliver anywhere

Until recently, delivering content to multiple media was never a serious issue. However, the proliferation of the Web, handheld devices, ebooks and a growing number of other new media channels has changed the publishing landscape forever. The Holy Grail of publishers is to create content once and deliver it to any number of media with minimal or no additional cost. This is still just an aspiration for most publishers.

By embracing open standards, particularly XML, Easypress Technologies is actively developing a range of solutions that will enable publishers to create once and deliver anywhere, making cross-media publishing a reality.

Efficiency and Profitability

Easypress Technologies is focussed on helping its customers achieve efficiency and profitability in their online publishing ventures. Not only does the company place the utmost importance on these two factors in its product development, our sales consultants regularly work with customers to formulate online revenue strategies and cost reduction opportunities that enable rapid return on investment.

Market Focus and Clients

Almost all the Easypress Technologies clients fall within the publishing sector, with many of the company's own staff also coming from a publishing background. Our blend of industry experience and intimate knowledge of Internet publishing technology, uniquely positions Easypress Technologies to serve publishers and meet their cross-media publishing needs.

The company has an extensive client list including FT Business, VNU, Roularta Media Group, Haymarket, Paragon Publishing and Daily Mail Group. Currently, over 60 publishing Websites are powered by Easypress and many leading publishers rely on Atomik to efficiently repurpose print content for multiple media.

Products

ATOMIK™

Atomik is a suite of XML tools that enable users to automatically turn QuarkXPress content into XML, the industry standard for holding digital content. By completely automating the process, Atomik makes repurposing content fast, reliable and efficient. Once the content is held in XML, it can be easily redeployed onto the Web, handheld devices, stored in a database, sent to mobile phones, or any other media.

EASYPRESS™

Easypress is an easy-to-use, Web-based content management system that empowers non-technical staff to create, manage and publish content online. It has been designed with publishers in mind and the comprehensive publishing-focussed feature set reflects this. Importantly, Easypress is ASP-based (Application Service Provider) which means there is no hardware or software to install. Publishers are therefore free to focus on content creation and revenue opportunities without having to worry about the technology or infrastructure.



EasyPress Technologies Ltd
International House, 59 Compton Road, Islington, London N1 2PB, UK

t: +44 (0)20 7704 0285
enquiries@easypress.com

f: +44 (0)20 7704 6627
www.easypress.com