Finding an Easy Path to DITA
For Technical Publications

Abstract

New challenges are putting intense pressure on technical publishing departments. The Darwin Information Typing Architecture (DITA) is an emerging standard that can help organizations meet these challenges. However, DITA can be difficult to implement. It requires authors to create content in XML, and it requires integration with a content management system (CMS) if you want to take advantage of DITA's most powerful features. To complicate matters, there are many DITA editing tools and CMSs to choose from.

This white paper considers the nature of today's technical publications challenges, explains the benefits of DITA, walks through some of the roadblocks that prevent organizations from implementing DITA, and provides information about a DITA solution called Quark® Dynamic Publishing Solution (Quark DPS) for Technical Publications.

Quark

Executive Summary

Documentation groups must manage an immense amount of information, make intelligent use of that information, publish their documentation quickly and efficiently in multiple formats and languages, and deliver frequent changes and updates.

To compete in today's market, companies must create increasingly feature-rich products, customize those products for specific audiences and markets, and present new and improved incarnations of those products on a regular basis. All of this makes the task of product documentation increasingly difficult. Documentation groups must manage an immense amount of information, make intelligent use of that information, publish their documentation quickly and efficiently in multiple formats and languages, and deliver frequent changes and updates.

The Darwin Information Typing Architecture (DITA) is an XML-based standard that allows documentation groups to single-source[1] documentation for multiple products and audiences, to automatically publish that documentation in wide variety of media formats including web and print, and to efficiently maintain and update that documentation. However, to make the maximum use of DITA, you must have an authoring tool that hides DITA's complexity, and you must integrate that authoring tool with a content management system. Because DITA is such a promising solution to documentation problems, there are many DITA tools and CMS integrations on the market. It can be a challenge to select a solution, integrate that solution with existing systems, and get authors to actually use that system.

Quark DPS for Technical Publications addresses the challenges of DITA by providing a friendly, familiar authoring interface based on Microsoft® Word and SharePoint®.

Today's Technical Publication Challenges

Technical Publications departments face increasing challenges today:

- Competition and rapid technological advances have created pressure to release new and improved products on an increasingly frequent basis.
- As the technical capabilities of products evolve, the documentation for those products is becoming longer and more complex.
- To gain a competitive advantage in global sales and marketing, companies are localizing products for new markets, which creates additional documentation requirements.
- Customers demand that documentation be presented in media other than print, creating a need to produce the same content in multiple formats.
- Manufacturers are increasingly targeting specific demographic markets and sales channels with tailored editions of products and solutions. In documenting these tailored products and solutions, companies must choose between awkward, one-size-fits-all documentation and tailored documentation. While tailored documentation is clearly preferable, it can result in duplicated content, which makes it extremely difficult to maintain.

[1] "Single source" refers to the ability to use the same content in multiple publications or in multiple output formats such as print and web. This content comes from a single source, which is often managed in a content management system.

These challenges combine to present complex problems. For example, assume you are producing three different versions of a product, each of which has a slightly different feature set that is tailored for a particular market. Because these three versions of the product all share a large number of features, their documentation for these three versions is largely identical; if all three of the products have the X feature, their manuals all contain the documentation for the X feature.

Now, assume the customer requires that the manuals for these products be made available in PDF format, in HTML format, and as a Help file. With traditional publishing processes, three separate documentation groups must create three separate copies of each manual — one for PDF output, one in HTML, and for Help output — for a total of nine copies of the feature X documentation.

Now consider what happens when the X feature is improved or expanded. All nine versions of the documentation for feature X must be updated identically, possibly by people in separate groups. Not only does this take a lot of extra time, it also increases the risk of inconsistent or omitted updates.

Now, assume that a customer who is reading the HTML version of one of these manuals notices an error. Someone could easily fix that error in the HTML copy of the manual for that product without fixing any of the other copies. Even if there's a system in place to handle such fixes, it once again entails making those fixes to nine different copies of the feature X documentation.

For companies that sell in global markets, this problem is much worse. For example, a company that produces information in six different languages must deal with 54 copies of its documentation.

Publishing challenges have grown severe enough that they have outstripped many organizations' abilities to solve these problems through incremental process improvements.

Given the above, it's not surprising that the complexity of technical publications has reached a point where many organizations are struggling to produce documentation in time to meet aggressive product release schedules. As a result of this pressure, managers must expend significant resources to ensure that documentation meets the organization's quality goals. What's more, the challenges have grown severe enough that they have outstripped many organizations' abilities to solve these problems through incremental process improvements; clearly, a comprehensive change is necessary.

What is DITA?

DITA is an XML-based standard that provides a faster, easier way to publish technical documentation. DITA encompasses a number of critical best practices, including the following:

DITA uses the topic as the basic unit as the unit for both authoring and information assembly. Each topic, which DITA defines as "a specific subject covered in a specific way," is contained in a separate file, which makes it easy to share authoring and review tasks in a workgroup and to assemble those topics into publications.

DITA defines three basic topic types: concept, task, and reference. Each topic type is designed to cover a topic from a specific angle. For example, a product's theory of operation would be covered in a concept topic; each procedure for operating, maintaining, and servicing the product would be covered in a task topic; and the technical specifications would be covered in a reference topic.

DITA represents each publication with a DITA map. A DITA map is a file that lists the topic files that make up the finished publication and indicates the order and structure in which they should be assembled.

You can adapt the DITA schema to meet your own requirements using a mechanism called specialization. DITA specialization lets your adaptations remain compatible with off-the-shelf DITA applications. DITA specialization also lets you interchange content with others, even if their applications know nothing about your specializations and your applications know nothing about their specializations.

The DITA standard is maintained by the Organization for the Advancement of Structured Information Standards (OASIS). For more information, visit http://dita.xml.org.

How DITA Can Address These Challenges

The Darwin Information Typing Architecture (DITA) was designed by documentation experts to solve the problems inherent in technical publishing. DITA's advantages include the following:

- DITA supports automated publishing to multiple formats — including HTML, PDF, Help, and others —eliminating the need for manual document formatting. Rather than maintaining separate copies of a manual in different formats, you can generate automatically formatted documentation in a variety of media types, all from the same source content. When a change needs to be made, you only need to make that change once — to the master version — then you simply regenerate the content in all of the required formats.

- DITA divides documents into topics, allowing multiple authors to work on a document simultaneously. Rather than passing a document around between multiple writers so that they can contribute content serially, managers can assign different topics to different writers.

- DITA lets you use the same content in multiple documents without copying and pasting it. For example, the topic file that describes feature X can be included in the manuals for products A, B, and C by simply adding pointers to the topic file. Writers only have to maintain one copy of the topic, and all documents that use that topic can be updated automatically when the topic is revised.

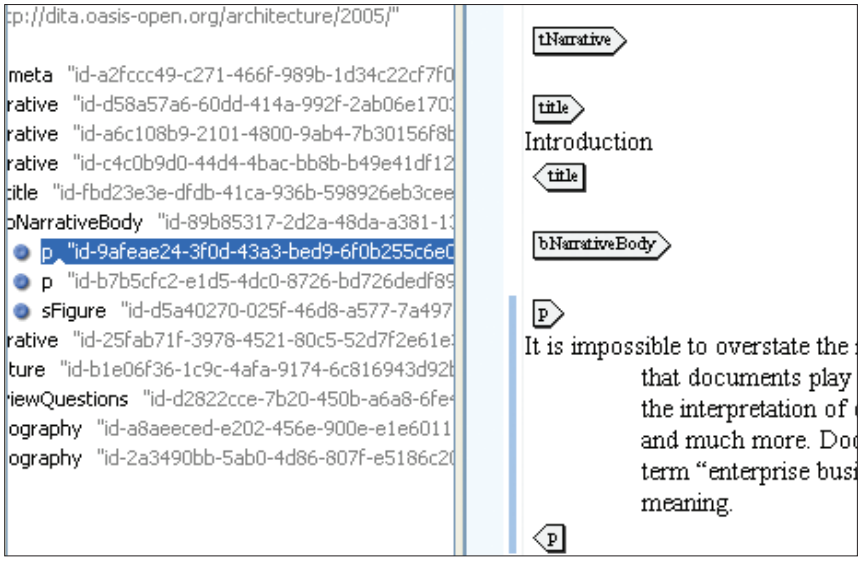
- DITA lets you use conditional text so you can produce variations of your documentation for different audiences' needs, while maintaining a single set of source files. For example, assume that the procedure for replacing a part is identical for products A, B, and C, but the part number is different for each product. DITA's conditional text feature lets you use product A's part number when you output the topic for product A, product B's part number when you output the topic for product B, and product C's part number when you output the topic for product C.

- DITA simplifies translation by allowing you to create all technical publication content in a single format, which allows you to make the most of translation memories.

Barriers to Using DITA

The road to DITA can be more difficult than one might expect.

Because DITA is a standard, you can work with DITA content using a variety of different tools. However, the DITA standard does not dictate how those tools should work, nor does it indicate how they should fit in with existing systems. And because DITA is rapidly gaining popularity, many vendors have brought DITA tools to the market — but not all of those tools have the same goals or capabilities. All of this means that the road to DITA can be more difficult than one might expect.[2]



Traditional XML editing tools do nothing to hide the complexity of XML, making them intimidating to many users.

Steep Learning Curve

Most writers are used to working with tools such as Microsoft Word, where a writer differentiates between different types of information using formatting. XML follows a significantly different model; rather than differentiating between different types of content through formatting, DITA XML requires writers to apply explicit semantic tags to different types of content. For example, where a Microsoft Word user might indicate a title by styling the text as 24-point bold, a DITA XML user must indicate a title by tagging it as a <title> element.

In addition, XML imposes restrictions on the way documents can be assembled — restrictions that users of word processes are not used to managing. There is nothing in a word processor to prevent a writer from, for example, creating a headline by applying 24-point bold to the first half of a paragraph and then inserting a return after the bold text. If a writer were to try this in an XML editor, however, either the editing tool would prevent it or the result would be an invalid document. And where most writers are used to being able to cut, paste, and move around text at will while working on a document, attempts to do the same in an XML editing tool can result in some very surprising results.

Most XML editing tools do nothing to hide the complexity of XML, and that makes them intimidating to many users. Even for users willing to adjust to such tools, the learning curve for XML authoring and editing can be long and expensive. And without guidance, a user who is not sure how to use the DITA schema can easily create XML that is technically valid but will result in confusing output when automatic formatting is applied at output.

[2] For additional examples of reasons why an organization may be reluctant to adopt the DITA standard, see "Drive Forward With Dynamic Publishing: DITA Helps Give Meaning To XML And Content Management" by Sheri McLeish with Kyle McNabb, Craig Le Clair, and Shelby Catino; Forrester Research, 2008.

Even if your technical publications experts are familiar with XML, other content contributors — such as technical specialists and reviewers – may not be. If such contributors cannot or will not work with an XML editing tool, they must submit their input in some other form. That means someone else must then convert that input to XML, which adds additional steps to the process and makes it difficult for those contributors to continue to participate in the process.

CMS Requirements

Because the DITA approach to documentation leads to far more individual components of information that must be managed, content management systems (CMSs) play an important part in a DITA solution.

DITA solves technical documentation problems through an implementation of "best practices" built around information types and component reuse. These practices are not new to XML, but DITA incorporates them formally. Because the DITA approach to documentation leads to far more individual components of information that must be managed, content management systems (CMSs) play an important part in a DITA solution.

While it is possible to use DITA without a CMS, such a system would come with a heavy load of manual tracking requirements. How do you keep track of which topics are used where? How do you notify stakeholders when content is updated? How do you make content available to multiple document architects? These are several of the areas where a CMS can be very useful. Because of this need, there are over a dozen CMSs on the market that are marketed as supporting DITA[3].

Even so, research shows that of the organizations that are currently using DITA, few have been able to select and implement a CMS.[4] There are several reasons for this:

- CMSs are expensive and require a good deal of time and effort to set up.
- Research on CMSs can be daunting. According to Content Management Professionals[5], there are almost 2000 products on the market that are billed as content management systems. That can mean a long and difficult selection process, even if the candidates are reduced to just the market leaders.
- Because a CMS can be such a large investment, it is important to gain buy-in from everyone who might end up using it.

Perhaps the main reason why organizations have not implemented DITA with a CMS is that these organizations discover that implementation may impose significant difficulties.[6] Just as DITA is new to technical publication departments, it is also new to content management systems. The more sophisticated CMSs offer tremendous flexibility in defining component models and may be able to accommodate DITA. However, several factors ultimately determine how much work is needed to support DITA completely: 1) the degree to which DITA's component model differs from CMS's native model; 2) the kind of development tools provided by the CMS; and 3) the flexibility of the other software that must also be integrated.

[3] See "DITA Tools from A to Z" by Bob Doyle, Intercom, April 2008.

[4] See Dr. David Dayton's discussion of an upcoming report, "Results of a Survey on the Usage and Impacts of Single-Sourcing and Content Management," as reported by Richard Hamilton (http://rlhamilton.wordpress.com/2008/06/10/stc-2008-day-four-june-4/).

[5] See http://www.cmprosold.org/resources/cmsml/index.html.

[6] See http://rlhamilton.wordpress.com/2008/06/10/stc-2008-day-four-june-4/.

In particular, organizations that select best-of-breed tools from various vendors often find that implementation and ongoing maintenance requires a large, upfront, and ongoing effort. This problem is more pronounced with DITA because it is a fairly new standard. Because DITA is still evolving, most vendors must quickly incorporate new features over the next few years to keep their DITA support current. When planning for a DITA project that integrates best-of-breed products, the question is not "will the interfaces change?" but "when?" Incorporating this level of change management is an added project overhead that organizations must consider in their project plans.

Analysis Paralysis

Given all of the factors listed above, many technical publications groups find themselves in a state of analysis paralysis. On one hand, DITA is clearly a game-changing approach to documentation with huge potential, and consequently there has been a rush to bring DITA solutions to the market. However, given the different needs of different organizations, none of these solutions is perfect. Some solutions are based on proprietary CMSs, requiring a significant up-front investment. Others may work with a CMS that an organization already has but that may not offer a rich-enough feature set or a simple-enough front end. And the expense required to get started with any of these solutions — in terms of software cost, integration costs, training, and so forth — can be daunting.

Consequently, many organizations shelve their plans to switch to DITA until they have time to sufficiently investigate all of the options, or until a clear leader emerges in the market. Meanwhile, they continue to struggle with outdated processes and miss out on opportunities for streamlining their publication process and achieving cost savings.

The Easy Path to DITA: Quark DPS for Technical Publications

Quark offers a simple solution that enables an organization to begin taking advantage of DITA with minimal disruption, and without requiring a huge investment in CMS technology and XML tool training. This solution is Quark Dynamic Publishing Solution (Quark DPS) for Technical Publications.

Quark DPS for Technical Publications is a standards-based solution designed from the ground up to help technical documentation groups quickly and efficiently implement a DITA solution.

Quark DPS for Technical Publications is a standards-based solution designed from the ground up to help technical documentation groups quickly and efficiently implement a DITA solution. It integrates directly with Microsoft Word, providing an XML editing environment where first-time users can become productive immediately. It also integrates with Microsoft SharePoint's content management capabilities, so it's ready to use out-of-the-box, allowing an organization to deploy quickly and avoid the months of integration effort that are often associated with other solutions.

With Quark DPS for Technical Publications, a technical publications department can:

- Meet growing documentation challenges with an international standard for technical publishing.
- Take advantage of the features of DITA without forcing writers to deal with the complexities of XML.
- Raise productivity, decrease time-to-market, and improve the quality of documentation — all at an attractive price point and with a low cost of ownership.

Quark DPS for Technical Publications consists of three components:

Quark XML Author: A plug-in for Microsoft Word 2003 and 2007 that lets authors and content contributors use Word to create XML.

Quark DITA Studio: A software application that allows document architects to build publications with DITA maps and to manage project assignments for writers and contributors.

Quark DITA-XML Adapter for SharePoint: An adapter that integrates Quark XML Author with Microsoft SharePoint.

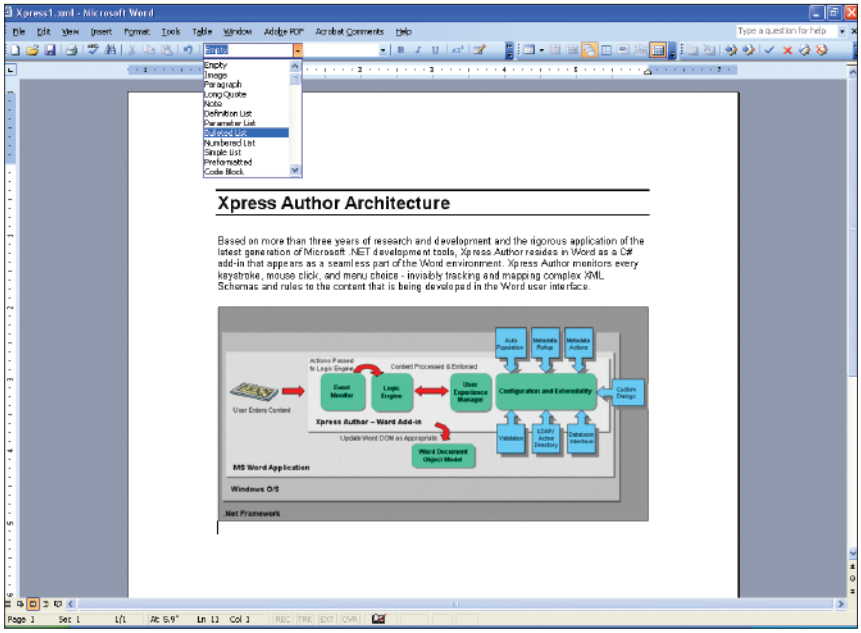
The following sections outline some of the specific advantages of Quark DPS for Technical Publications.

XML Editing in Microsoft Word

Quark XML Author minimizes the DITA learning curve by letting content contributors work in the familiar environment of Microsoft Word.

Quark XML Author minimizes the DITA learning curve by letting content contributors work in the familiar environment of Microsoft Word.

Quark XML Author presents DITA element types as visually distinct Microsoft Word styles, eliminating the need for reviewers to learn XML syntax. Rather than tagging content, content contributors can simply apply styles, the same way they always have.



Quark XML Author works in the background and minimally modifies the normal operation of Word to let users create XML documents. In this illustration, Quark XML Author uses the Word Style dropdown list to show all of the valid types of content that the author can insert at the current cursor location.

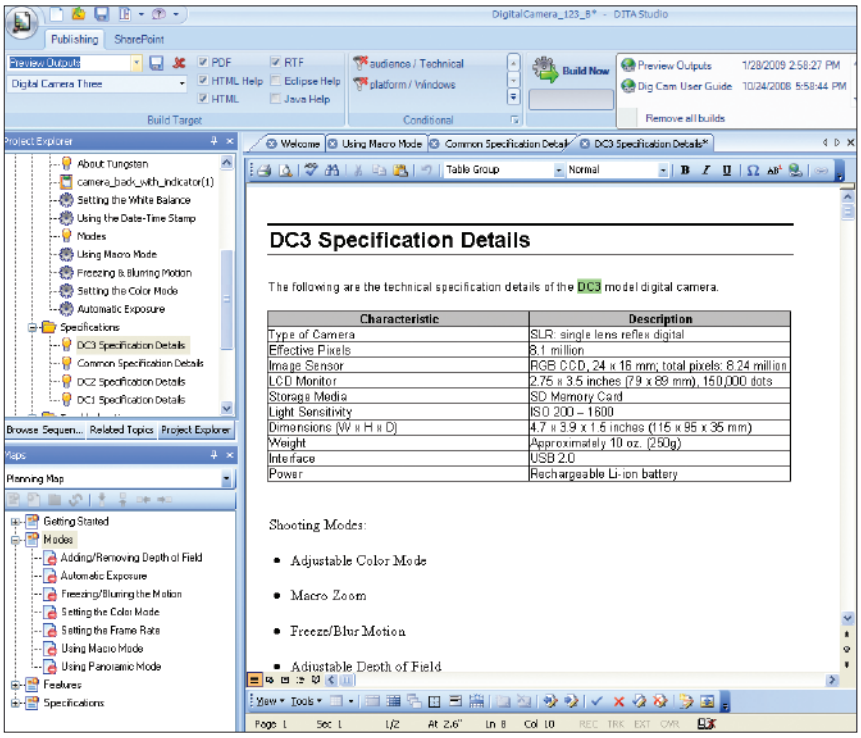
Quark XML Author also automatically limits users to valid content. When the text cursor is at a given point in a document, the only styles that are available to the content contributor are those that can validly be applied at that point. A content contributor can even right-click anywhere in a document and see a list of the types of elements that can be validly inserted at that point. This makes it impossible for content contributors to create invalid XML, and ensures that the output process will run smoothly.

XML editing in Microsoft Word means your content contributors can get up to speed faster. It also means you can bring content contributors into the authoring process without having to put them through extensive training.

Quark XML Author also can be configured to further limit authors to entering content that follows an organization's business rules. For example, if you don't want to include notes in your documentation — or if you want to make certain parts of a document mandatory even though they're optional according to DITA — you can automatically enforce those custom rules right along with the DITA rules.

XML editing in Microsoft Word means your content contributors can get up to speed faster. It also means you can bring content contributors into the authoring process without having to put them through extensive training. For example, rather than having technical writers interview developers, you can allow developers to contribute directly to the documentation.

Finally, Quark XML Author requires minimal IT support. Rather than having to maintain an entire third-party application, IT must merely install a Word plug-in that has been implemented with Microsoft technology.



Quark DITA Studio powers the publishing process by making it easy to build a "map" (which lists the topics that a publication includes) and then by automatically producing both print and digital versions of the publication.

Integrated DITA Project Management

Quark DPS for Technical Publications makes DITA project management easy with Quark DITA Studio.

Quark DITA Studio offers the following features:

- An intuitive project explorer that lets you easily access and reuse topics, maps, pictures, and other media in the SharePoint repository. This makes it easy for document architects to make the most of the organization's existing content and prevent needless re-work. Writers and other content contributors can collaborate on a single document simultaneously. Managers can also use the project explorer to monitor assigned topics and see whether or not they are checked out. SharePoint locking and versioning ensure that changes are always made in a controlled manner, and that you can always revert to earlier revisions of a topic.

- A drag-and-drop map editor that makes it easy for document architects to build documents as DITA maps. Combined with the project explorer, the map editor makes document development easier than ever before; to add an existing topic to a document, simply drag-and-drop the topic from the project explorer to the document's DITA map.

- A modern, configurable graphical user interface (GUI) that provides easy access to critical DITA features such as conditional processing. This GUI lets document architects produce different documents from a single-source content store without having to resort to technically challenging batch files and ant scripts. The GUI also lets users generate formatted drafts in a variety of formats with the click of a button.

- An integrated instance of XML Author, so that document architects can view and edit DITA content as well as maps and other resources.

- An API that allows the application to be customized to work with existing and legacy systems.

SharePoint Integration

Quark DPS for Technical Publications integrates with one of the most versatile enterprise content management systems on the market: Microsoft SharePoint.

Rather than relying on an expensive, proprietary content management system (CMS), Quark DPS for Technical Publications integrates with one of the most versatile enterprise content management systems on the market: Microsoft SharePoint. With more than eighty million licensees[7] in organizations today, SharePoint is an extremely popular and versatile tool for collaboration and content management. Even without the addition of a DITA tool, SharePoint offers numerous advantages over other CMSs, including the following:

- Web site management features

- File sharing and collaboration features

- Enterprise search and content management features

- Dashboard/portal construction and maintenance features

- A broad user base and support community

- Familiarity to IT personnel

- The Microsoft name and reputation

When paired with Quark XML Author, SharePoint provides DITA project management features that allow managers to assign and monitor the writing and review of technical publications.

[7] This statistic is reported in " Information Workplace Platform Vendors Light Up The World Of Work" by Erica Driver and Connie Moore with Kyle McNabb, Colin Teubner, Boris Evelson, and Jamie Barnett; Forrester Research, 2008. As reported by C.G. Lynch (http://www.cio.com/article/328613/Forrester_IBM_and_Microsoft_Likely_To_Dominate_Future_Information_Workplace_).

Future Proof

Quark DPS for Technical Publications is based on popular industry standards and widely used, well-established tools, and it doesn't lock you into a proprietary system. Your content is always in DITA format, so it's always compatible with other DITA solutions. That means you can take advantage of DITA's features now, even if your ideal DITA solution doesn't yet exist.

Install Quark DPS for Technical Publications and start using DITA today, with minimal sunk costs, and without locking yourself into a proprietary system. It's that easy.

About Quark

Revolutionizing Publishing. **Again™.**

Two decades ago, Quark drove the first revolution in publishing with QuarkXPress®, desktop publishing software that rapidly became the industry standard. Today, not only does QuarkXPress continue to innovate in the desktop publishing market — Quark is revolutionizing publishing again. With Quark Dynamic Publishing Solution, we're helping customers meet changing requirements and develop new revenue streams by extending the benefits of advanced technologies across the publishing process. Our dynamic publishing solution is setting a new standard in automated multi-channel publishing by combining the power of flexible layout and design with automated workflows and easy XML authoring for personalized communications across print, the Web, and electronic media.

To learn more about dynamic publishing or Quark DPS for Marketers, contact your dynamic publishing consultant or visit www.quark.com

North America
Quark Inc.
1800 Grant Street
Denver, CO 80203
solutions@quark.com
+1.800.676.4575
+1.303.894.8888

Europe
Quark Media House Sàrl
Rue Du Monruz 2
2000 Neuchâtel
Switzerland
solutionseu@quark.com
00800 1787 8275
+41 (32) 720 1414

Asia
Quark India Pvt Ltd.
A 45 Industrial Area
Phase VIII B Mohali
160059
Punjab, India
solutionsasia@quark.com
+91.172.2299.131

Japan
Quark Japan K.K.
West 248 bldg.
2-4-8 Ebisu-nishi
Shibuya-ku, Tokyo
150-0021 Japan
solutionsjapan@quark.com
+81.3.3476.9440

Quark