

**Technical Communication UK**  
A triumphant launch for the UK's first national event



# Communicator

The Institute of Scientific and Technical Communicators  
Winter 2009

**Standardising  
documentation suites**

**Fostering good  
team dynamics**

**Reviewing version 5.3  
of Help & Manual**

**Communicating through  
hearing impairments**

# CMS integration for FrameMaker

You want to manage your DITA content in a CMS, but you are tied to FrameMaker. **Andy Lewis** and **Alex Masycheff** explain how.

Nearly any technical communicator realises the value that content reuse, flexible content customisation, multichannel publishing and consistency can add to the documentation process. When used properly, these techniques can significantly reduce costs for creating, maintaining, translating and delivering technical documentation, as well as increase overall customer satisfaction.

The methodology that helps technical communicators achieve these goals is often referred to as component content management. With this technique, a minimum unit of information is a component that represents a relatively small portion of content (for example, a single procedure or feature description). These components are stored in a central location, and can be arranged by authors into deliverables in the same way as Lego bricks can be used for building many different things.

Today the market is full of tools that facilitate and automate storage, usage and management of such components. These tools are often referred to as Component Content Management Systems (CCMS). Many of them work with content written in XML formats, such as DITA and DocBook.

While many CCMSs support integration with XML authoring tools (for example, Microsoft SharePoint integrates with XMetaL by JustSystems and SDL Contenta integrates with Epic by Arbortext), fewer CCMS platforms currently offer integration with Adobe FrameMaker. However, as the adoption rate of FrameMaker as a DITA authoring tool grows, more CCMS vendors are beginning to consider integration with FrameMaker. They have good reason to do so: a survey conducted by Scriptorium Publishing found that over 60% of DITA implementers are using FrameMaker as a DITA authoring tool, while over 62% are planning to use FrameMaker in the future (*The State of Structured Authoring in Technical Communication*, Scriptorium Publishing, [www.scriptorium.com](http://www.scriptorium.com)).

The first part of this article provides an overview of general features that any CCMS should support, describes some of the more important DITA-specific features to look for in a CCMS, and concludes with a short discussion of the advantages of integration with FrameMaker. The second part of this article is a detailed examination of the capabilities of a featured CCMS product: XDocs CCMS, developed by Bluestream ([www.bluestream.com](http://www.bluestream.com)).

## Features of a CCMS

### General CCMS capabilities

The goal of any CCMS should be to facilitate content reuse and content customisation, and to automate publishing to a certain degree. Essentially, you should be able to write a piece of content once, check it

into your CCMS repository and then use access permissions to allow authors to access the content and repurpose it as they want. Common CCMS features that support this working environment are:

- *Central repository.* The central location in which the CCMS stores all DITA topics, DITA map files and data such as graphics files and other material referenced by the XML content. The CCMS will support administrative operations on the repository for manipulation, upload and retrieval of content, and security features such as access control via user profile management.
- *Version control.* A CCMS stores previous versions of components to enable users to view any previous version and rollback to an older version, if necessary. CCMSs also usually help team members to work together and avoid situations in which changes made by one person are overwritten by someone else. This mechanism is known as *file locking*. Opening a file located in the repository for editing automatically locks that file so that other users cannot modify it until the active author has finished working and saved the file back to the repository.

Alternatively, files can be downloaded or opened just for reading, without being locked.

In addition, a CCMS may support *file caching*, whereby opening a file located in the repository automatically generates a copy of that file on the author's local computer. This enables the author to continue working on the file without having to be connected to the repository.

- *User management.* The CCMS enables you to define different types of user profile and to assign one of these profiles to each user of the system. Ordinarily, each user type has a default view of the CCMS interface and a default set of permissions (including user name and password, the type of operations that the user can perform on a document and the types of document that the user can access and modify). The CCMS stores user profiles in its own database. You can add, modify or delete user profiles as required.
- *Access permissions.* These ensure that users see and edit only the content that you want them to by controlling which user profiles have permission to access and modify specified documents. You can even grant permission for users to modify only specific parts of a document by setting access permissions for individual DITA topics or elements.

### DITA-specific features

For work with DITA XML files, you will require some additional capabilities from a CCMS. Basic DITA-related features include:

- *Search*, which enables users to search DITA topics and maps stored in the repository.

- *Link management*, which enables users to maintain the integrity of links between topics. For example, users can move or rename topics, maps or illustrations (or the folders where these are stored) without invalidating links to and from these resources.
- *Where used*, which enables users to find all DITA maps and topics that refer to a specified topic. If a topic has multiple versions, this feature can be used for each version individually.
- *Translation management*, which enables users to streamline the translation process and integrate with translation memory tools.
- *Integration with DITA Open Toolkit (DITA OT) and publishing engines*, which enables users to generate multiple deliverables in various formats, including PDF and online help, from a single source.
- *Publishing profiles*, which enable users who customise content (for example, for different products, platforms and audiences) to define possible variations once and then choose an appropriate profile when producing a specific deliverable.
- *Workflow management*, which enables users to create and manage workflow tasks and assignments.

#### *Advantages of FrameMaker integration*

Probably the biggest advantage of using FrameMaker as a DITA editor is the ability to produce high-quality PDF output. With FrameMaker, you get the DITA editor and a PDF publishing engine in one. You do not have to customise DITA OT stylesheets, which may require deep knowledge of XSL and entail high customisation and maintenance costs. You do, however, need to configure FrameMaker structured applications, but this is a far less problematic task. With FrameMaker integrated with a DITA-aware CCMS, you get most of the tools that you need for effective work with DITA.

When a CCMS integrates with FrameMaker, you should be able to perform tasks such as the following without leaving the FrameMaker environment:

- Searching and browsing a CCMS repository
- Creating new DITA topics and saving them to a CCMS repository
- Creating new DITA maps that refer to the topics stored in a CCMS repository
- Checking out DITA topics and maps from a repository for editing in FrameMaker
- Getting DITA topics and maps for viewing in read-only mode in FrameMaker
- Adding references to DITA topics and graphics stored in a CCMS repository
- Previewing content that you want to insert into a topic from a CCMS repository
- Using FrameMaker as a publishing system to generate page-oriented output.

#### **Feature implementation in the XDocs CCMS**

##### *Why integration?*

Usually, a CCMS consists of two main components:

- A *repository*, which is installed on a server and stores all content needed in your documentation.

- A *repository browser*, which can be either web-based or installed on every user's computer as a desktop application. A repository browser enables users to navigate the repository and perform various operations, such as creating folders, checking files in and out, managing versions, and previewing and publishing content.

In theory, you can use just these two tools to work with DITA content. For example, you may use a repository browser to check topics out of the repository and then open these topics using a DITA editor installed on a local computer. After you finish working with a topic, you use a repository browser again to check the topic into the repository. However, inconveniences may arise when you want to establish links between different pieces of content stored in the repository (for example, when you need to insert an image stored in the repository, add a cross-reference to another topic in the repository, use the DITA conref mechanism or create a DITA map that refers to topics in the repository).

Without having the ability to access the repository from a DITA editor, it would be like working with content stored on a remote drive, which is accessible from Windows Explorer but not from an authoring tool that can access only local drives. You would first have to copy all the content you need (including all images and topics to which you want to add cross-references) from the remote drive to your local computer — only then would you be able to add links. Similarly, to review a topic that refers to other topics or graphics, not only would you have to download that topic, but you would also need to copy all the referred files to the local computer. In addition, to keep links working, you would have to verify that the folder structure on your local computer replicates the folder structure in the repository. Otherwise, if the folder structure is different or folders are named differently, links will break after you upload DITA topics and maps to a repository.

##### *FrameMaker integration with XDocs*

XDocs is a DITA-aware CCMS developed by Bluestream Database Software Corporation. It provides most of the features that you might need when working with DITA, including storage of both DITA and non-DITA content in a central repository, search, link management, version control and workflow management.

To access the XDocs repository, you can use Local Explorer, which is a desktop tool installed on users' computers. (If you want only to view and search content, you can also use the web-based Content Management Portal, which can be a perfect tool for reviewers.) Local Explorer enables you to navigate the repository, search, check files in and out, preview and publish DITA content, manage versions and much more. You can even open topics and maps directly from Local Explorer by selecting an XML editor with which you want to edit the DITA content (Figure 1). The list of editors is completely customisable.

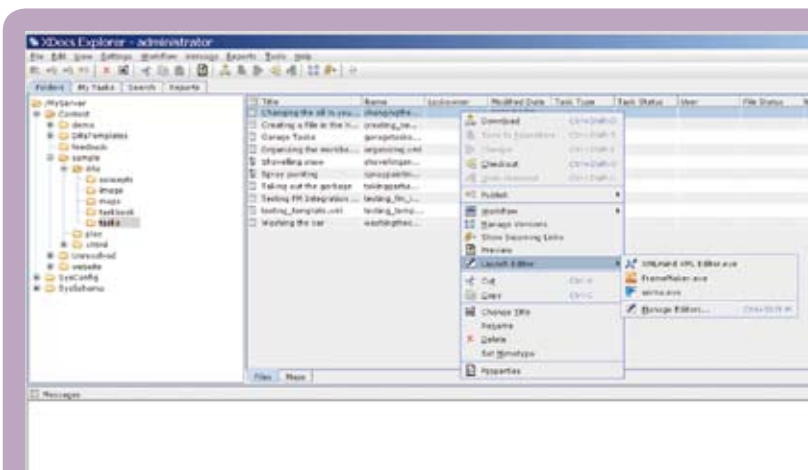


Figure 1. Selecting an XML editor

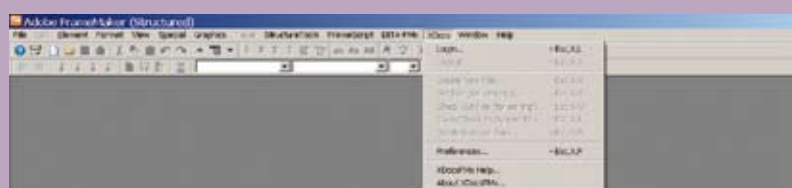


Figure 2. FrameMaker XDocs menu

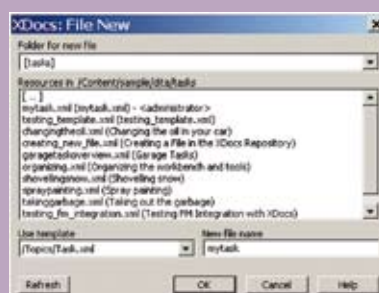


Figure 3. Creating a new topic

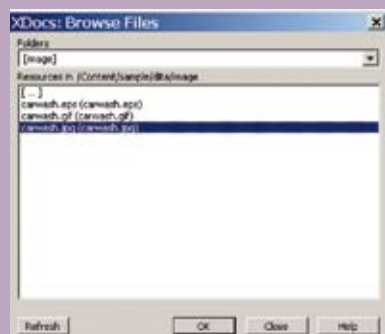


Figure 4. Retrieving an image from the repository

After installing this connector, you will see an additional command in the FrameMaker menu bar (Figure 2). From this menu, you can work with the repository, including creating new files, checking out existing files and getting files for viewing. For example, this is how the workflow looks when you need to create a new DITA topic, add an image from the repository and save the topic to the repository:

1. You start by logging into the XDocs repository. After you have logged in, the DITA-FMx menu for creating new files becomes disabled. This is done for your convenience: once logged in, you can create topics and maps in the repository only, which reduces the risk of accidentally working on files on a local computer.
2. When you select the **Create New File** option in the XDocs menu, the system prompts you to specify a repository folder in which the topic should be stored, the type of the topic (concept, task, reference, map or generic topic) and a file name (Figure 3).

After you provide this data, the new empty file opens. You may notice that new files have a few elements already inserted. For example, when you create a task, you get it with empty <title>, <shortdesc> and one <step>. This is defined in templates (which are just DITA files): you can add any empty elements there that you want to be inserted for all new files of that type.

3. Every time you add the <fig> or <image> element, instead of the regular **Open File** dialog box (which lets you choose a file from a local drive), XDocs displays its own **Browse Files** dialog box. This shows the contents of the repository and enables you to locate and insert an image from the repository rather than from the local computer (Figure 4). Similarly, when creating a DITA map that refers to topics stored in the repository, every time you insert a <topicref>, XDocs displays its own dialog box and prompts you to locate a topic in the repository.

What you do not currently have is the ability to search the repository from FrameMaker. As the number of individual topics grows, search capabilities become crucial. You cannot rely exclusively on reasonably named files and a logically organised folder structure. To focus on writing rather than searching, you may want to search by a topic title, text in a topic, attribute values and other parameters. The current version lets you search using Local Explorer and then open the file by selecting FrameMaker in the list of XML editors. You have to remember, though, that you need to check out the file if you want to lock it for editing by other users.

Another feature that is currently missing is the ability to preview a file before you insert it into a topic (although you can preview a link text). You may easily have hundreds of images and thousands of topics created by different people at different times: without

There are several options for using XML editors with XDocs:

- You can use the XMLMind editor provided with XDocs. XMLMind integrates tightly with the CCMS through the XMLMind virtual drive technology. This technology enables you to work with files stored in the XDocs repository as if you are working with files stored on disk.
- You can use the XMetaL editor, which can also be integrated with the XDocs repository.
- Recently, Bluestream added integration with FrameMaker. FrameMaker integrates with XDocs through the XDocs/FrameMaker connector developed by Leximation ([www.leximation.com](http://www.leximation.com)). The connector requires the DITA-FMx plug-in for FrameMaker, also developed by Leximation.

The installation process is very easy and took us just a few minutes. We especially liked the fact that making FrameMaker work with XDocs does not require any additional configuration and can be handled by any user who is skilled enough to click the **Next** button a few times. What is important is that XDocs integrates with all three versions of FrameMaker that support DITA: 7.2, 8 and 9.

a preview feature, the only way to check whether you are inserting the right file is to open it. Perhaps the lack of search and preview functionality is the weakest link of the integration with FrameMaker.

- When you close the file, the system asks you whether or not you want to check in the topic back to the repository immediately. You can select to check in the topic, or leave it checked out if you want to continue working with it later.

What we especially liked is how XDocs handles cross-references between topics and conref links. All you need to do is to open a source topic and a target topic from the repository, add an <xref> or conref, check in the topics back to the repository, and you are done! Whenever you open the topics from the repository, the links will always be valid.


Accessing repository content from FrameMaker when integrated with XDocs is like working with a remote drive: you create and open files directly in the repository without the need to copy anything to your local computer; XDocs will take care of all links.

After you have finished working on DITA content, a subject matter expert (SME) can review the document and leave comments. What is interesting is that SMEs do not have to install XDocs Local Explorer on their computers. Instead, they use the XDocs CM Portal, which is a web representation of the same XDocs

repository. SMEs can use the CM Portal to download a document for review or generate HTML or PDF output. Although SMEs can use XMLMind to edit content or leave comments (XDocs is provided with unlimited XMLMind licences so it won't be a problem for reviewers to install XMLMind), we believe reviewing will be much more compelling in XDocs when the ability to review and comment documents online is built into the product. Currently, web reviewing is an extra-cost item.

#### Summary

Overall, XDocs provides a fairly straightforward way to manage DITA content. If you are looking for a cost-effective, department-level solution that can be deployed within a few days, XDocs is a product that you definitely should consider. The product provides all fundamental and basic features that you need to work with DITA, though there are several features that are currently missing, some of which are mentioned in this article. Hopefully, the Bluestream development team will add them in upcoming releases.

You need to determine if the benefits you get provide sufficient value for you. The good news is that XDocs provides an application programming interface (API), and you can integrate this CCMS with other systems already in your organisation to provide additional tools for working with DITA. 

**Andy Lewis** has spent most of the 21st century creating technical content in FrameMaker for RADVISION.

E: andylewis0@gmail.com

W: www.linkedin.com/in/andylewis2003

**Alex Masycheff** is a senior XML project manager at WritePoint, a technical documentation company based in Israel. Alex develops and implements solutions that help documentation teams effectively create, customise and publish technical documentation. He also advises organisations on choosing the right techniques, technologies and tools, and teaches documentation teams to use content management concepts in practice.

E: alexm@writepoint.com  
W: www.writepoint.com

Translation provider since 1989

Certified Investor in People

ISO 9001:2000

## Translation specialists in your world

 **Lloyd international**  
T R A N S L A T I O N S

- Reflect the **quality** of your products across all languages
- **Reduce** your localization costs
- **Improve** your time to market
- **Succeed** globally

t: +44 (0)1829 730050

e: mail@lloyd.co.uk

w: www.lloyd.co.uk



Technical manuals

Software

Online help

Websites

Marketing materials