

# Managing and Delivering

## Your Content as Data

BY ALAN R. HOUSER, *Senior Member, Pittsburgh Chapter*

Imagine the ability to deliver the appropriate content to the appropriate person in the appropriate format at the appropriate time—with ease. This level of content customization may sound more like science fiction than a real-world solution. However, consider what some companies are already doing with their content:

A *manufacturer* of customized products employs a system that allows a foreman to swipe a barcode to begin a product's journey through the assembly line. The appropriate components, with the appropriate assembly instructions, are delivered automatically to each point in the assembly line.

A *major airline* automatically assembles the components of the pilot's "black bag"—the list of instructions for completing a flight itinerary, including specific information about each destination's airport. Among the critical pieces of information are takeoff and landing instructions based on the airport's geo-

graphic location and nearby structures. Instead of carrying a bag full of paper, pilots can view this information on a laptop computer.

A *semiconductor manufacturer* publishes device-specification information, traditionally printed in voluminous catalogs, on the company's Web site. Users can search for device information and filter results based on specific characteristics. After users identify specific devices that meet their requirements, they can print customized sets of data sheets for the devices in which they are interested.

These companies have changed the way they author and deliver content. No longer must content be presented as a linear series of information; instead, it can be managed and delivered as data objects—discrete, independent pieces of information that can be selected, manipulated, and presented to meet the needs of different audiences with different characteristics and different goals.

### Why Treat Documents as Data?

Many organizations have amassed large volumes of technical content and are trying to figure out how to manage that content in the face of today's complex information delivery needs. The value of an organization's information is directly related to the ability to efficiently create, manage, and deliver that information.

To assess whether your organization can improve its ability to manage and deliver your technical content, ask yourself how easily and effectively your organization can do the following:

- Share information among product marketing, product development, and technical publications teams.
- Identify and deliver relevant information quickly and in a format that is appropriate and useful to the user.
- Maintain information across product categories, easily identifying multiple places in which content must be changed.

- Publish information—including, but not limited to, hardcopy manuals, online help, Web content, and training materials—in forms that are appropriate to the end-user.
- Publish in forms that meet the U.S. government's Section 508 requirements for accessibility.

### When Does It Work?

Entering this new realm of information management requires careful planning and execution by corporate information technology and technical publishing departments. Data-oriented publishing tends to work particularly well for information that has the following characteristics:

- Highly modular—for example, procedural information that is appropriate in a very specific context.
- Can be (or must be) dynamically assembled and delivered.
- Output to multiple output formats.
- Targeted to audiences that can be classified into discrete categories.
- Passed or shared between organizations.

Although certainly not appropriate for all technical communication, data-oriented publishing provides the means to meet today's information delivery requirements.

### Assessing Your Requirements

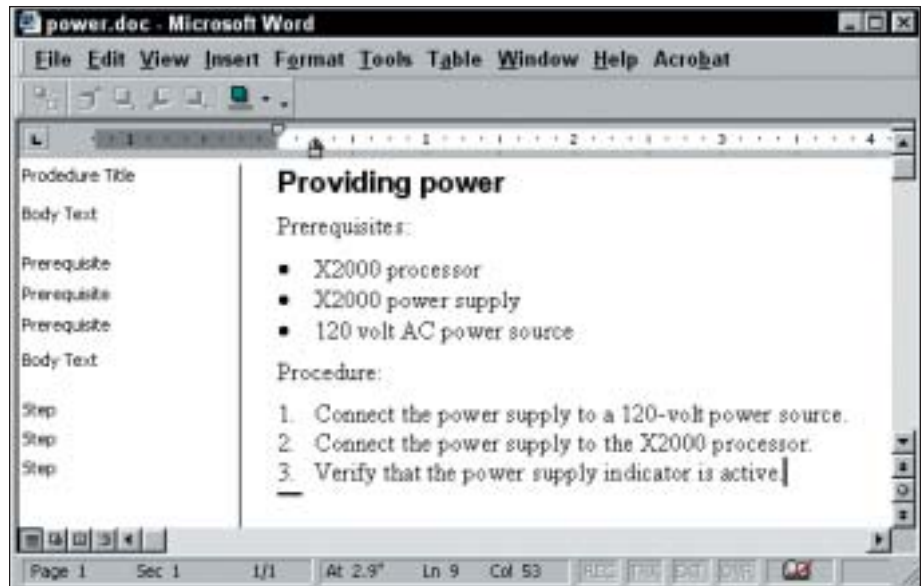
Adopting a technology for technology's sake is rarely successful. Adopting a technology as a solution to a business problem yields a much greater chance of success.

Begin with your business requirements. What would you like to do with your information that you are not currently doing? How would you like to deliver your information in ways that you currently cannot? The answers to these questions will drive the rest of the design and implementation process.

### Building an Information Model

Your business requirements will provide insight into an information model that will serve as a roadmap for delivering your content as data objects. Your information model will consist of your content and *metadata*.

Figure 1. Sample document created with template-based authoring.



This metadata, or data about data, provides a layer of information about your content. The metadata about each of your documents, as well as the documents themselves, will comprise the data objects you use to manage and publish your information.

There are several ways to add metadata to your information.

### Template-Based Authoring

Many writers think of template-based authoring as form-based authoring. This is not necessarily true. A template-based authoring tool allows writers to label information with paragraph and character styles. Adobe *FrameMaker* may be one of the more popular template-based authoring tools, but even Microsoft *Word* can be used this way.

Template-based authoring provides several benefits. Because formats are associated with paragraph and character styles, document formatting is largely automated. Authors can spend more time creating content and less time formatting that content. Also, paragraph and character styles can be used as labels for document components. An author might apply a "product name" style to product names in text. Procedure headings might use a "procedure heading" style. These paragraph and character styles provide an easy (albeit limited) way to associate metadata with your content. Figure 1

shows a simple example document with meaningful style names.

Success in template-based authoring requires several components:

- Designing templates that allow writers to label document components with semantically meaningful names.
- Training writers to use template formats consistently and correctly.
- Enforcing proper template formats (via an editorial review process, for example).
- Deploying publishing tools that maximize the benefit of template-based authoring.

Even though template-based authoring can provide an easy entry to creating information that you can manage like data objects, it has several limitations. Tagging information with paragraph and character styles can only take you so far. It is difficult to represent hierarchies of information and rich metadata with paragraph and character style tagging. Also, authors of information are largely dependent on proprietary authoring packages. Any manipulation and reuse of the information is limited to those proprietary tools.

### XML Authoring and Publishing

While template-based authoring provides a relatively low-cost entry into creating and maintaining reusable information, XML authoring and publishing

Figure 2. XML document as displayed by Microsoft Internet Explorer.

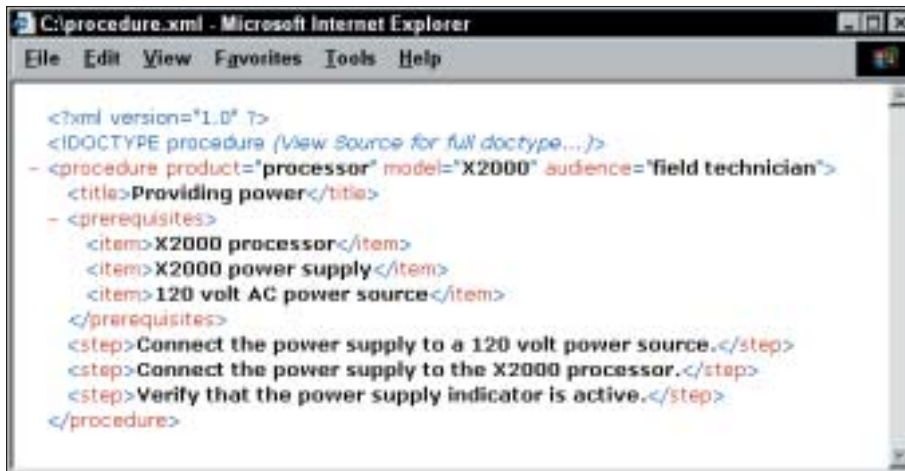
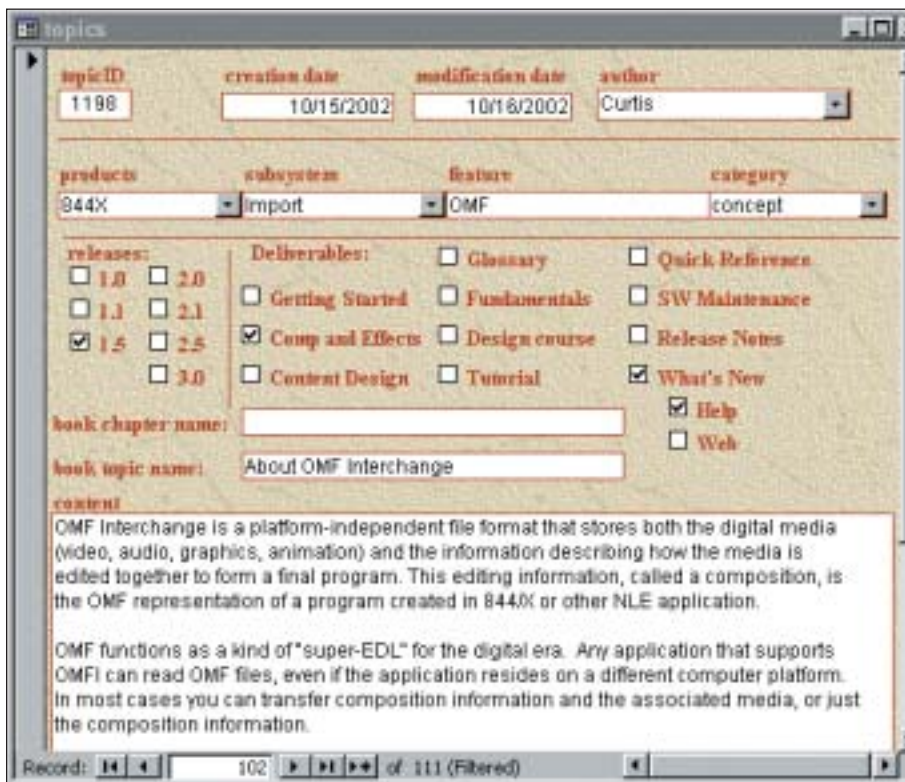


Figure 3. Database interface for collecting and storing metadata about topic information.



can radically transform an organization's ability to publish information. XML provides the capability to embed rich metadata within document content. This metadata can be used to select and manipulate XML content for publishing.

XML provides several ways to include metadata within a document. XML *elements* provide containers for document content. XML *attributes* provide a way to attach additional information to XML

elements. XML elements can be nested, which creates a document *structure*. For example, a "procedure" element might have an attribute called "product\_name," which refers to the name of the product or products for which the procedure is relevant. Elements that contain the text of the procedure, including such items as "prerequisites" and "steps," are likely to be nested within the parent "procedure" element. You can define your own element

names, attributes, and document structure to meet your organization's metadata requirements. Figure 2 shows element and attribute metadata in an XML document.

An advantage of XML authoring and publishing is that both XML and the tools for publishing XML documents are based on nonproprietary standards. For example, the XSLT programming language provides a way to select and manipulate XML content for creating customized documents. The XSLT language was developed by the World Wide Web Consortium (W3C), which also maintains the HTML language specification.

#### Database Publishing

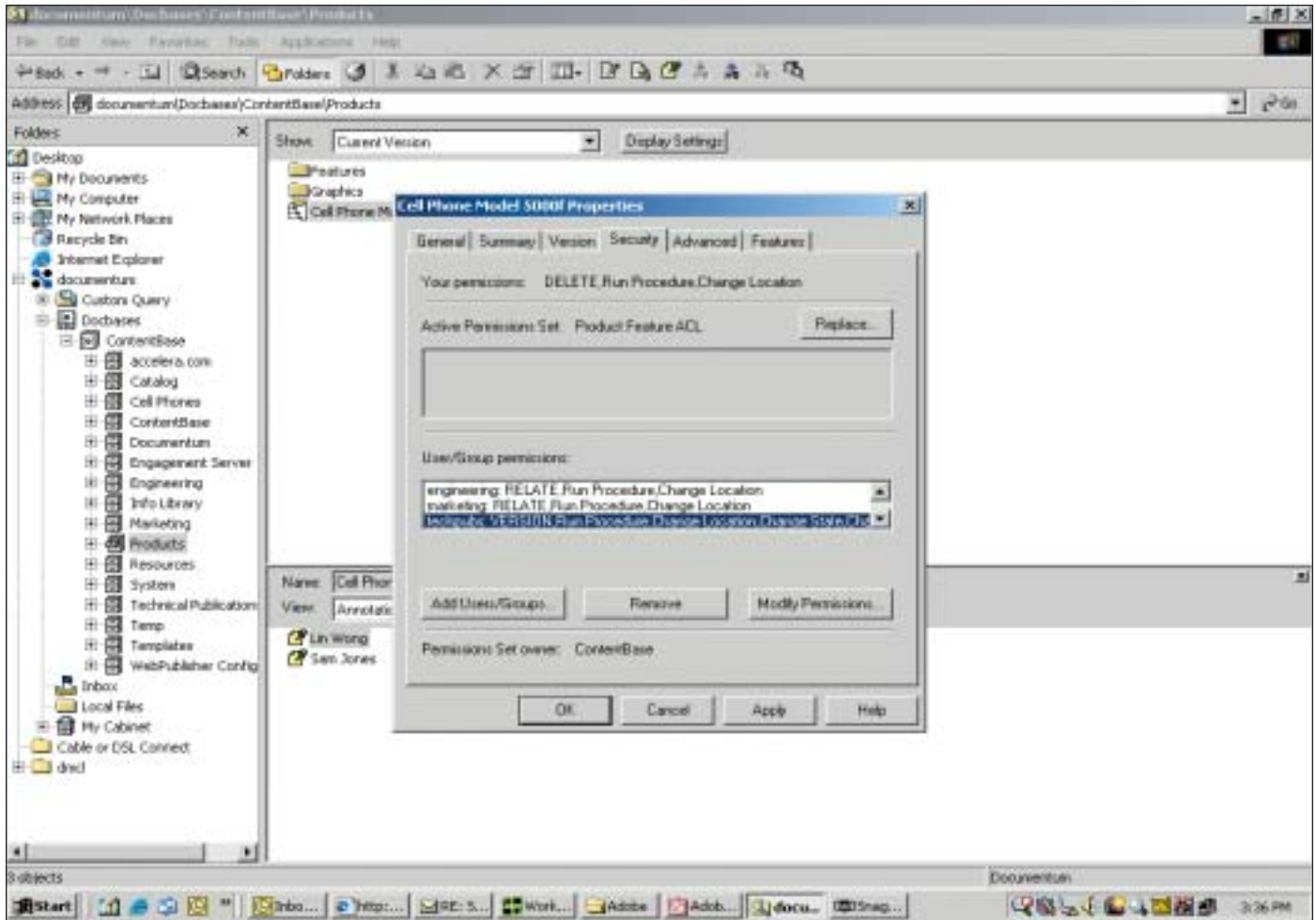
Some organizations maintain information in a database format. Maintaining content in a database offers several intriguing benefits. Databases provide the capability to associate rich metadata with document content through easy-to-use, forms-based interfaces. Also, database technology is older and more mature than XML technology. Most IT staffs can support databases, while not all can support XML (because they have not yet developed XML expertise). Figure 3 shows an example of a customized database interface for collecting and storing metadata about topic information.

These advantages are balanced by the disadvantages of database publishing. Database entry tools are typically forms-based, with text boxes for entering content. The database typically constrains how content must be written and entered. Conventional relational databases do not handle nested document structures well, as XML can. (However, major database manufacturers are now providing support for XML content.)

#### Content Management Solutions

Content management solutions appear to be growing in popularity as organizations attempt to manage large amounts of technical content. Content management solutions can be used in conjunction with either template-based authoring or XML authoring. They have a reputation for being expensive to purchase and complex to deploy, although prices are falling and new products are appearing in the marketplace. A content manage-

Figure 4. Interface of enterprise-level content management system.



ment solution typically provides the following functionality:

- Content management—maintaining multiple document versions, tracking changes to documents, recording information about document changes, attaching metadata to documents.
- Workflow management—automatic notification to appropriate persons when a document is created or modified, locking documents while an author is making changes, enforcing publishing workflows and schedules.
- Publishing management—assembly and publishing of appropriate content based on document metadata.

If you have ever asked yourself any of the following questions about a document, your needs could have been met by a content management system.

- Who wrote it?
- When did they write it?
- Why did they write it?

- When did they modify it?
- What did they modify?

Enterprise-level content management systems tend to be expensive and complex. However, a content management system may be a worthwhile investment for an organization that creates and manages large volumes of content. Figure 4 shows the interface of an enterprise-level content management system for managing document components and metadata.

As an alternative to full-blown content management systems, easier and less expensive solutions are available. Software development organizations will typically have a version management system in place for developers' source code. If you are saving document source files in XML, you can use the same version management software that your programmers use. Examples include Microsoft Visual SourceSafe and CVS (Concurrent Ver-

sioning System). These systems, in conjunction with XML content, can provide the source management functionality you need.

## Conclusion

Creating and delivering information as data objects requires reengineering many processes, including content authoring, source management, and publishing. However, for many applications, the effort is appropriate and necessary for meeting new business and publishing requirements. **i**

*Alan Houser is a principal partner of Group Wellesley, a Pittsburgh, PA-based company that specializes in information management, XML consulting and training, and technical writing. Alan is president of the Pittsburgh Chapter STC, and is co-author of XML Weekend Crash Course, published by John Wiley and Sons. You can reach him via email to [arh@groupwellesley.com](mailto:arh@groupwellesley.com).*