

# Writing Becomes Industrial

BY NEIL PERLIN, *Associate Fellow*

**T**echnical writing today often varies according to each writer's preferences and styles, with little standardization and consistency. Writing aids like templates and style sheets are changing this by providing a more industrial approach to writing, and mainstream authoring tools make it easy to create and use these aids. However, many documentation groups have not yet made the strategic decision to include these writing aids as a part of their culture.

This article discusses the strategic importance of these writing aids and provides some notes about their creation, use, and maintenance. It describes how they make writing more efficient today and how they support the writing tools, technologies, and methodologies of tomorrow. The focus of this article is on online documentation—help systems and Web sites. Where I discuss hardcopy, I'm largely referring to *Word*.

## Overview

When I entered the technical communication field in 1979, IBM Selectric typewriters were cutting-edge authoring tools. Many writers, myself included, wrote with pencil and paper, then gave the pages to a word-processing group

for input. (STC's Andy Malcolm gave a speech in 1980 discussing whether technical writers should be *allowed* to write using computers. Andy reprised this speech for the Bleeding Edge at STC's 49th Annual Conference in 2002; it's one of the most ironic and funniest presentations I've ever heard.)

Our tools have come a long way since 1979, but the process of writing is still as individualistic as it was then. Some doc groups use templates, style sheets, style guides, and other aids to standardize their material; but, from my experience with clients in and outside high-tech, standardization is rare and rarely ingrained in the doc group's culture. Does this lack of standardization matter? Two answers:

No. The individualistic writing process is inefficient and inconsistent, but as long as your output is a finished end-product like a hardcopy manual or a help system, the drawbacks are acceptable.

Yes. Technical communication is changing in ways that make the drawbacks of individualism more critical:

- We're starting to go from creating one end-product to creating "content" as a raw material that we can process to create many end-products—that is, single sourcing. Web 2.0 is a good ex-

ample of this. Web 2.0 applications rely heavily on content supplied by the users themselves (think of wikis, MySpace, and YouTube). If you're going to be soliciting material for a Web 2.0 application, you have to decide whether the material is going to be presented as submitted or processed further to produce different outputs. Few companies single source beyond hardcopy and online help, but technologies like mobile devices, voice, and Web 2.0 are entering the business world. Single sourcing for these technologies, plus traditional hardcopy and online help, makes consistent structure and writing crucial if the single-sourcing tools are to work well.

- Technology keeps advancing, often in unexpected directions. But we don't know what technologies are coming and which will succeed, so we have to write on the assumption that today's material may have to be reused in unanticipated ways. This sounds like a future problem, but it's with us today. We see it when we try to re-use locally formatted *Word* documents online, for example. We got away with bad development practices in the past because we didn't anticipate having to further process the material. The old excuse was that we didn't expect major technical change. Now we know things will change, just not how. We can be ready for that unknown by creating material that's consistent in structure, style, and writing. Such material is flexible because it can be processed—converted—reliably and affordably.
- The "editor" position has disappeared in many companies, so there's no one to fix or even notice global stylistic, structural, or wording inconsistencies.
- Return on investment (ROI) and time-to-market requirements are getting tighter. There's less time to make mistakes and learn from them.
- The minimal syntax requirements of RTF and HTML let us develop bad coding and structural habits that XML doesn't tolerate.

What these changes, and others, have in common is the need for well-controlled content, sometimes called *structured content* or *structured writing*. What is structured writing? Here are three definitions:

- Writing with a structure based on heading styles in a style sheet.
- Writing with a structure based on templates (which may include a style sheet).
- Writing with a structure that's valid according to an XML DTD or schema.

Which definition is right? In a sense, it doesn't matter. What does matter is picking a definition that fits a particular doc group's culture. For example, a group that's just moving from hardcopy to online content will be so busy learning new concepts and technologies that requiring schema adherence as well is just asking too much.

So the simplest ways to create structured writing—through templates, style sheets, style guides, and other standards—are also the safest. Even these relatively simple approaches to structured writing will be difficult for doc groups whose culture, or whose larger company culture, doesn't support them. But it is a start.

Here are some high-level points about creating, using, and maintaining templates, style sheets, and style guides.

### Templates

A template reduces structural inconsistencies between similar types of content created by different authors or at different times. For example, if two authors have to write concept topics for a help system, they may use different wording and sequences for their headings. Even lone writers may do so if they forgot the wording or sequence they used in the previous topic. A template can reduce or eliminate this problem.

A template is a master document for a particular type of content, such as a concept topic or task. The template contains standard headings in a fixed sequence—the structure—plus common elements, such as a tip that must appear in all concept topics. When you create a topic using a template, the heading

Our tools have  
come a long  
way since 1979,  
but the process  
of writing is still  
as individualistic  
as it was then.

structure is predefined: some headings may already be filled in, and prompts indicate what to write. For example, a "Required Tools" section might include a prompt that says "[delete this prompt and type the list of tools]."

Creating topics based on a small number of templates takes many structural decisions out of the authors' hands. The result? All topics of each type will have the same structure—titles, subheads, and content—in the same sequence. They will be consistent.

To create *effective* templates, you have to evaluate the material you write; categorize it as a concept, task, etc.; define a standard structure for each category; and use that structure as the basis for each category's template. Every mainstream authoring tool offers template-creation features. If your tool doesn't, consider using a different one.

To create *maintainable* templates, do two major things:

- Build the template instructions into the template and write them to be usable by the newest or most novice author. For example, add prompts like "[type the tools list here and delete this prompt]."
- Make sure the template can stand on

its own. If you must add instructions other than prompts, for example, add them to the template as hidden text. Try not to create a separate instruction document for a template because that document will inevitably get lost, possibly making the template useless.

If you plan to go to XML, creating templates offers one more benefit. The work you put into identifying your documents' structure can serve as the groundwork for defining a DTD or schema.

### Style Sheets

A style sheet reduces formatting inconsistencies among similar content created by different authors or at different times. For example, if two authors have to write concept topics for a help system, they may use different formats for the headings. Even sole writers may do so if they forget which formatting they used in a previous topic. A style sheet can reduce or eliminate this problem.

Just as templates take structural decisions out of authors' hands, style sheets do so for the formatting decisions. You can also attach a style sheet to a template; authors using that template/style sheet automatically achieve both structural and formatting consistency.

To create effective style sheets, you have to evaluate your material; select formatting for headings, lists, tables, etc.; and define the styles. Every mainstream authoring tool offers stylesheet-creation features. If yours doesn't, consider using a different tool.

Style sheets must be managed carefully. Here are a few problems to watch out for:

- Competing style sheets from multiple groups. Ideally, there will be one company-wide style sheet, or one for each type of output you create.
- A tendency to accrue extra styles. You may define a main style sheet, only to find authors adding styles out of personal preference or because they can't tell which standard style to use for what text. Use clear and self-explanatory style names, weed out extra styles, and give authors a *process* for suggesting changes to the style sheet.

- Incompatible views on how to use style sheets in hardcopy vs. online output. I see many hardcopy style sheets with a hundred or more styles for very specific conditions, such as a “first paragraph in document” style that adds extra space between the title and the first paragraph. This level of detail is often unnecessary for online material. I find, for example, that a style sheet with a standard set of about twenty styles works for almost every help project.

The big problem with style sheets isn’t technical; it’s behavioral. How do we get authors to use styles when the formatting toolbar is so convenient? An obvious answer is to hide the formatting toolbar, but every authoring tool I’ve used puts the styles pull-down on the formatting toolbar, so hiding the toolbar hides the styles. Until new tools change this, authors need to develop the discipline to use the styles pull-down but ignore the rest of the formatting toolbar. I find that the best way to get authors to use style sheets is to stress the benefit to them—faster and simpler writing—rather than the benefit to me.

### Style Guides

A style guide reduces stylistic inconsistencies in content written by different authors or at different times. This is important for companies that grow by acquisition, with different doc groups having to compromise on their writing practices. A good style guide can ease these problems, but it needs certain characteristics, including:

- Required adherence. If there’s no enforcement of adherence and no penalty for lack of adherence, people will use the style guide inconsistently, if at all.
- A minimum amount of detail. One-hundred-plus-page style guides are masterpieces of detailed guidelines but often go unused because it takes too long to find anything. Assume that writers know how to write and provide guidelines only for specific issues.
- A defined owner and a defined process by which to change owners.

## How do we get authors to use styles when the formatting toolbar is so convenient?

Without that, the style guide will be forgotten as soon as the original owner changes jobs.

- A formal revision process. This allows the style guide to be updated to reflect changes in practice—but in a controlled way. Without this process, authors will just deviate from the style guide as they see fit.

### Other Standards

This is a broad area that may include anything from controlled vocabularies to metadata.

### Summary

Creating templates or style sheets seems hard if you’ve never done it before, but, mechanically, it’s easy. The hard part lies in designing these aids and ingraining them into your doc group’s culture. Also, be aware that the scope of these aids may extend beyond the doc group.

There’s also another reason to use these aids. Technical communication is entering one of its periodic upheavals. The first such period I saw was in 1980, when word processors replaced typewriters. In the mid-1990s, the spread of online help, followed in the late ’90s by

the spread of the Web, ushered in periods of instability. This new upheaval is tied to the need for flexibility, but there are some related requirements:

- The need to develop and enforce standards in general.
- The need for continuing education to keep up with new technology. Even if you never use Ajax or Ruby on Rails, you should know what they are to be able to gauge their importance to your company. (Plus, you’ll be better able to talk to the IT people.)
- The need for training in the use of our tools and their underlying technologies. Without that training, tools get used inefficiently at best.
- The need to position ourselves as developers and publishers of “content,” not just mere “technical writers,” and to back that positioning with a strategic view of content and the ability to discuss it in business terms.

All these changes are happening today, but in a hit-or-miss way. They need to become more common if we’re to make it into the next generation of technical communication. **1**

---

*Neil Perlin has twenty-eight years’ experience in technical communication, with twenty-two in training, consulting, and development for various types of online formats and tools such as WinHelp, HTML Help, CE Help, JavaHelp, RoboHelp, and some now forgotten. He is a columnist and frequent speaker for STC and other professional groups, a senior member of the Boston Chapter STC, the creator and manager of the Beyond the Bleeding Edge sessions at STC’s annual conference, and an Associate Fellow of STC.*

*Neil is a Madcap Certified Instructor for Flare, and a Macromedia Certified Instructor for RoboHelp and Captivate. He provides training, consulting, and development for online help and documentation, Flare, RoboHelp, Captivate, XML, and single sourcing through Hyper/Word Services of Tewksbury, Massachusetts. He can be reached at [nperlin@concentric.net](mailto:nperlin@concentric.net) or via [www.hyperword.com](http://www.hyperword.com).*

---

*The author thanks Deborah Sauer, principal, Deborah Sauer Consulting, for her comments.*