

WORDS



A Quarterly Bulletin for Technical Writers & Communicators

Volume 1 | Issue 4 | November 2009

The beginning

- There is no doubting the increasing buzz in our profession as more and more technical writers delve into this seemingly mysterious thing called *structured authoring*. Many think that this is something new. But it has, in fact, been around for more than 20 years—since the advent of Standard Generalised Markup Language (SGML). And not long after, FrameMaker—today’s tool of choice for many technical writers—became one of the first authoring tools to enable writers to tag text according to content rules (or at least that flavour of FrameMaker then called *FrameMaker+SGML*).

But the buzz is also coming from the fact that tagging text according to content—the structured authoring approach—is becoming increasingly flexible (with the rise of DITA and its specialisation capabilities) and more useful (with increasingly affordable content management systems). This issue of *Words*, and the next, looks at these new technologies, and from various perspectives.

- No longer do we doff our hat at what was once seriously, curiously and wrongly called *Standard English*. It was never standard by any means. And we now accept that no one English is better than any other. To think that the English of Oxfordshire might in some sense be better than, say, the English of Tasmania or of Maine is utterly preposterous.

Australian English has long been distinctive, with its unique vocabulary, colourful shortened forms and preference for economy over bombast. For many years, *Australian Style*—a free bulletin published by Macquarie University—has charted Australian English and chronicled its evolution. In this issue, Adam Smith, the current editor, reminds us of the bulletin’s purpose, as it moves its presence online.

- In the last issue of *Words*, we initiated a worldwide survey to gather the opinions of practising technical writers on what we should, or could, call ourselves: *technical authors*, *information designers*, or whatever. The survey generated quite some heat on internet discussion forums, suggesting that we do invest a lot in our name—which may account for the somewhat conservative results (reported on page 13 below).

Geoffrey Marnell

Editor [geoffrey@abelard.com.au]

Contents

Australian Style Online.....	1
A niche for DITA	3
Journal of Technical Writing and Communication.....	6
Is structured authoring a paradigm shift?	8
Creating and applying formats in FrameMaker	11
What should we be called? Survey results.....	13
Converting auto-numbers to text in MS Word	18
Miscellany	19
Mindstretchers	20
Volume index.....	20

Australian Style Online

Adam Smith

Australian Style, the national bulletin on Australian English, celebrated its first online edition in April 2009. The print publication of the newsletter had been produced biannually since 1992, supported by a variety of government departments, but funding cuts have meant that the online medium is the only way for *AS* to continue to be provided free. Paid subscription is an option we have always avoided, wanting to reach as broad an audience as possible—not simply those with a professional interest in language.

This democratic principle was expressed right from the start, in Volume 1.1, by Arthur Delbridge, founding editor of the *Macquarie Dictionary*: “After all, it is people who by their own practice in speaking and writing ultimately define the standards by which language as an institution may be judged”. *AS* has always had a strong connection with the *Macquarie Dictionary*, arising out of the national language forums, known as Style Councils, that were instigated by members of the dictionary’s editorial committee. The editor throughout its print existence was Pam Peters, a well-known authority on Australian English through works such as the *Cambridge Guide to Australian English Usage* (2007) and the *Cambridge Guide to English Usage* (2004). Ann Atkinson, senior editor at the *Macquarie Dictionary*, was executive editor until I took over that role in 2001.

The content of the newsletter has remained quite constant over the years. Articles on language issues from plain English to Arabic loanwords to the classification of shellfish, written by linguists, editors, journalists, or members of our word-loving audience with a special interest. Book reviews on dictionaries and other language-related volumes, reports on conferences such as Style Council and Australex, letters from readers. Regular features include SCOSE notes—a report from Irene Poinkin on the meetings of the ABC’s “Standing Committee on Spoken English”—linguistically playful cartoons by Judy Dunn, and David Astle’s popular and unique Rubicon puzzle.



A particularly important part of *AS* is the Feedback questionnaire. Responses from a good proportion of our c.8000 readers have allowed us to gauge current usage trends. This information has a practical function, with the data both serving the needs of the scholarly community—for example, in comparing the use of past tenses in Australia with other regions, in a recent study—and contributing towards advice given in the sixth edition of the *Australian Style Manual* (2002). The current online questionnaire was designed around questions of usage that need to be updated in the next edition of the *Style Manual*.

This emphasis on describing English, particularly Australian English, usage is very much at the core of *AS*’s take on language. Our role is not to give directives on correctness, in the manner of the Académie Française, but to report on the shifts and modulations in our language that reflect the changes in our society. The Centre for Language Sciences at Macquarie University has other tools to assist in this aim. Collections of written and spoken Australian

texts (corpora) from the 1980s and 1990s have been digitised so that they can be electronically searched¹. Data on grammatical changes and the forms and meanings of words can be analysed and compared with trends in other parts of the English-speaking world. Access to spoken data in particular allows us to observe innovations and variations that may then become accepted in standard written English.

With the passing of the print version of *AS*, some things have changed, though much remains comfortingly familiar. The regular features mentioned above all survive, alongside new sections such as a word column

focusing on Australian words, and a poet’s corner. An electronic archive has been included, which will make all the printed editions of *AS* available online for the first time, and reader feedback is encouraged so that a blog of issues of particular interest can be created.

The e-version of the newsletter can be accessed at http://www.ling.mq.edu.au/news/australian_style.htm, and if you would like to be added to our emailing list for notifications on updates to the website, you can contact me at adam.smith@mq.edu.au. The next issue of *AS* will appear online in December this year. We hope to continue the tradition of open and informed discussion on the English language in Australia far into the future.

Adam Smith

Adam Smith is the editor of *Australian Style*.

1. These corpora are also available for the use of researchers from universities other than Macquarie, although copyright restrictions don’t allow us to open them up to the public in general.

In the next issue

Due out on 1st February 2010

- Protecting yourself from MS Word
- Looking at DITA as just another tool
- Hacking resources to resize dialogs
- The lost art of paragraphing
- New book features in Adobe FrameMaker 9
- Master and sub-documents in MS Word
- Book reviews

The Words team

- Artwork: Christine Weaver
- Copy-editor: Marcia Bascombe
- Adviser: Mark Ward
- Cartoonists: Brian Fray
- Editor: Geoffrey Marnell
- Contact: words@abelard.com.au

© Individual contributors or Abelard Consulting Pty Ltd, 2009
[unless otherwise noted]

A niche for DITA

Dan Smith

This article describes the design principles of DITA, and the potential niche in the technical writing world that DITA can fill.

DITA is touted by some to be the next big thing in technical writing. It is described by others to be the next big thing to fall flat on its face. After all, the DITA structure of concept, reference and task is just a simplified version of the old information mapping system.

Back in the 1990s in the heady world of telecommunications, life seemed to be a series of technological next-big-things, debuting and dying to such an extent that a pattern emerged. The pattern was this: technology X would be touted as the next big thing. It was going to solve all communications and bandwidth problems. In the telecommunications community, excitement would mount to fever pitch, until cracks started to appear. Enthusiasm would wane as the cracks widened. At some point, everyone would jump off the bandwagon, and pronounce that technology X was in fact a dead duck, and the zeitgeist would focus its attention on the next big thing. Technology X would then be free to find its niche in the telecoms world, and would often prove to be a very useful, medium-sized thing. DITA could possibly follow this pattern.

A brief introduction to DITA

DITA is an XML standard, leveraging off the XML Document Type Descriptor (DTD) architecture to enforce structure on documents. As an XML standard, a DITA implementation can avail itself of all the functionality that XML offers, such as XML's sophisticated information categorisation and content delivery functionality. The possibilities that arise from using metadata to categorise information is a separate subject by itself.

As evolution organises life into a tree-like structure where new species evolve and branch off from existing species in response to changing environments, DITA organises information types into a similar tree-like structure. Corresponding to new species in the tree of life, new topic types, or specialisations, arise to meet the requirements that new information demands. At the root of the DITA tree is the Topic type, with a very broad structure remit. Up the tree are the Concept, Reference and Task types. These are branches, or specialisations, of the Topic type. They have more tightly defined

structures that reflect the type of information they are designed to contain.

To create new types of information, you can specialise one of the four primary types: Topic, Concept, Reference or Task. A new specialisation can only restrict the structure and content of the type on which it is based. The fact that you can only restrict existing types means that if a delivery or publishing system does not have access to a specialisation's details, it can simply navigate down the tree towards the Topic type until it finds the closest item it can use.

A key tenet of DITA is that a topic should contain a single item of information, and the information should make a degree of sense by itself. To provide structure, you link topics together using ditamaps. That is, in a ditamap, you arrange topics (and indeed other ditamaps) into a hierarchy (as shown in figures 1 and 2 on page 4).

Limiting topics to a single information item is important, as a key selling point of DITA is information re-use. Keeping

topics small and to the point maximises the possibility that they can be used across multiple publications. Re-use, however, is arguably more useful at the ditamap level, and it is good practice to keep ditamaps small and limited to a single subject grouping. You can then incorporate these ditamaps into larger ditamaps.

To enhance re-usability, DITA discourages the use of hyperlinks embedded within text. If a topic is re-used, there is no guarantee that the link target is a part of the publication, which would result in a broken link. In a topic, hyperlinks are included as a list of related links at the end of the topic. To define these related links list, you use DITA's *reltable* mechanism to set up the relationships between topics in a publication. A publication only displays the links to targets that are part of the publication. The process groups related links automatically into concepts, references and tasks.

DITA and CMS

To get anything like the best out of a DITA system, it must operate in conjunction with a Content Management System (CMS). A CMS provides functionality to maintain large volumes of information, and enables easy re-use of content. That is, with large volumes of information, you can easily find candidates for re-use, and in the documentation maintenance phase, you can easily find where an item of information is used throughout the system.

DITA (Darwin Information Typing Architecture): an XML-based environment for authoring and delivering information. It enables documents to be built from small chunks of information, called topics, each of which contains elements tagged according to the type of information they provide: concept, reference or task.

A CMS typically provides functionality that enables you to avoid using conditional text. Conditional text is even more error-prone than usual in a DITA system. The possibility for conditional text errors is magnified when re-using content. However, most CMS systems provide conditional text functionality if you want to risk it.

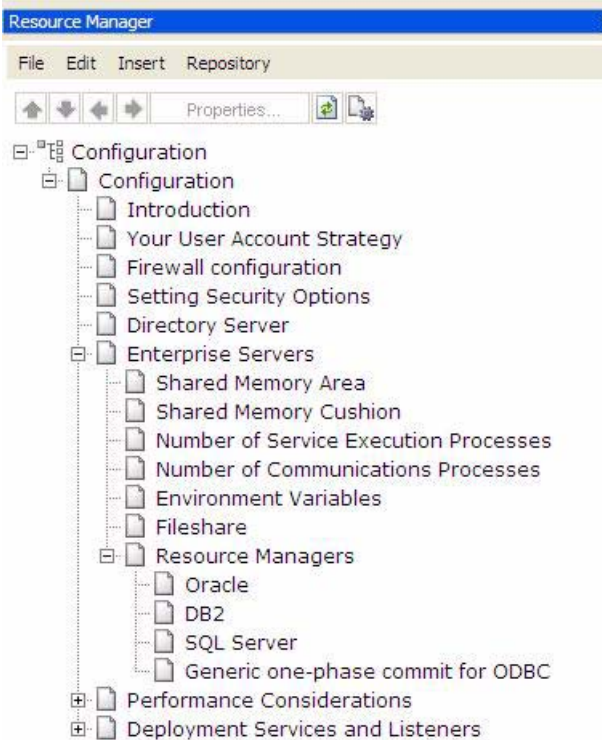


Figure 1: A DITA map (in XMetaL 5.1)

A CMS enables you to implement and maintain a complex XML metadata structure to categorise information. When properly designed, this structure serves as a sophisticated indexing system. It can deliver information-access options far removed from the simple search mechanisms that many web-based systems provide. For example, a DITA–CMS implementation has the potential to provide on-demand documentation tailored for a specific customer, or to a user’s role. This is an area that is just starting to evolve, and it is likely that, over time, methods of presenting information will arise that have yet to be imagined.

An organisation can maintain all its documentation, including marketing and training material, in DITA format in a CMS. This provides interesting opportunities for any area of an organisation to leverage off documentation from any other area, and also for ensuring that all written material originating from the organisation has a degree of consistency.

Where localisation is involved, a CMS enhances the use of translation memory technology, thus speeding up and simplifying the process, and reducing localisation costs. These savings alone can be sufficient to justify a DITA–CMS system. With a CMS, you can micro-manage a localisation process if you so desire, without having to entrust it to an agency.

Technologies are in the pipeline that will use variations on the translation memory mechanism as part of the documentation development process. Such technologies could provide functionality such

```

1  <?xml version="1.0" encoding="UTF-16" standalone="no"?>
2  <!DOCTYPE map PUBLIC "-//OASIS//DTD DITA Map//EN" "map.dtd">
3  <map id="BKCASKHTMLS002" title="Configuration" xml:lang="en">
4    <topicref format="dita" href="BKCACACONF" linking="none" locktitle="no" navtitle="Configuration"
5      scope="local" toc="yes" type="topic">
6      <topicref format="dita" href="BKCACACONFS002" navtitle="Introduction" scope="local"
7        type="topic"/>
8      <topicref format="dita" href="BKCACACONFU003" navtitle="Your User Account Strategy"
9        scope="local" type="topic"/>
10     <topicref format="dita" href="BKCACACONFS008" linking="none"
11       navtitle="Firewall configuration" scope="local" type="topic"/>
12     <topicref format="dita" href="GUID-16DC0BAA-3659-426A-92B4-095222A900A3"
13       navtitle="Setting Security Options" scope="local" type="topic"/>
14     <topicref format="dita" href="BKCACACONFS009" linking="none" navtitle="Directory Server"
15       scope="local" type="topic"/>
16     <topicref format="dita" href="BKCACACONFU015" linking="none" navtitle="Enterprise Servers"
17       scope="local" type="topic">
18       <topicref format="dita" href="BKCACACONFS011" navtitle="Shared Memory Area"
19         scope="local" type="topic"/>
20       <topicref format="dita" href="BKCACACONFS013" linking="none"
21         navtitle="Shared Memory Cushion" scope="local" type="topic"/>
22       <topicref format="dita" href="BKCACACONFU008" linking="none"
23         navtitle="Number of Service Execution Processes" scope="local" type="topic"/>
24       <topicref format="dita" href="GUID-DBC1B6A3-40A4-46E8-9BC8-1D7F27BE6FB2"
25         navtitle="Number of Communications Processes" scope="local" type="topic"/>
26       <topicref format="dita" href="BKCACACONFU004" linking="none"
27         navtitle="Environment Variables" scope="local" type="topic"/>
28       <topicref format="dita" href="BKCACACONFU017" linking="none" navtitle="Fileshare"
29         scope="local" type="topic"/>
30       <topicref format="dita" href="BKCACACONFU020" linking="none"
31         navtitle="Resource Managers" scope="local" type="topic">
32         <topicref format="dita" href="BKCACACONFS019" linking="none" navtitle="Oracle"

```

Figure 2: The XML behind the DITA map shown in figure 1

as suggesting existing topics for specific situations, and for alerting you if you have deviated from company styles and standards. In fact, the potential is there for a system to enable documentation development to be outsourced to writers with only marginal English skills, working for peanuts in Third World sweatshops.

DITA's niche

The niche that DITA systems occupy is in large organisations with correspondingly large volumes of user information delivered mainly over the web:

- The DITA enforcement of content structure is most useful with large systems where content is developed by many writers across many countries. DITA helps to deliver a degree of documentation consistency where writers' backgrounds, skill levels and even familiarity with the English language vary wildly.
- The ability to re-use content delivers most benefits to large organisations with many, related product lines.
- Although the DITA Open Toolkit is open source and freely available, a DITA-CMS system is

expensive to implement and maintain. At an organisation level, the expense can only be justified by tangible cost benefits, for example, by document development, maintenance and localisation savings.

- Implementing and maintaining a DITA system requires significant system resources: server and network hardware, XSL and scripting programmers, system administration and support resources.

Conclusion

The quality of output from a DITA documentation system will seldom compare favourably with that of small, dedicated publications lovingly handcrafted by quality technical writers using tools such as FrameMaker or RoboHelp. There will always be a place for such user documentation. DITA addresses a different area of the technical writing market entirely.

Dan Smith

Dan Smith is an expatriate Australian working in the UK. He has many years experience as a technical writer, working with large technology companies in Australia, Italy, France and the UK.

TRAINING COURSES

Technical, general & scientific writing, FrameMaker, Structured authoring

Abelard Consulting offers a number of courses and workshops designed to benefit technical writers, would-be technical writers, and those whose job requires them to be good communicators. The following courses are available:

- ▶ Technical Writing: An Introduction and Refresher [2 days]
- ▶ Scientific Writing: An Introduction and Refresher [2 days]
- ▶ Using Adobe FrameMaker: An Introduction and Refresher [2 days]
- ▶ Structured Authoring with Adobe FrameMaker [2 days]
- ▶ Business Writing: The Fundamentals of Clear Communication [2 days]

Where?

Scheduled classes are held in Melbourne, Sydney, Canberra, Perth, Brisbane and Adelaide. Customised on-site training is also available.

What?

To see the topics covered in a course, download a brochure from www.courses.abelard.com.au. We can also tailor a course to your specific requirements. Call 1800 601 116 for further information.

When?

Download a brochure or visit www.abelard.com.au/course_booking.html to see dates and locations.

By whom?

The FrameMaker and structured authoring courses are conducted by Mark Ward. Mark is a long-time user of FrameMaker and an expert in creating FrameMaker templates and structured applications.

The writing courses are designed and conducted by Dr Geoffrey Marnell. Geoffrey is the founder and principal consultant of Abelard Consulting. He also teaches Technical Writing and Editing in the English Department at the University of Melbourne and is accredited by IPed (Institute of Professional Editors). Geoffrey has more than 20 years experience as a technical writer, documentation consultant and educator.

Discounts?

Early-bird, concessional and student discounts apply. Further, on-site training is charged at a flat rate regardless of the number of participants.

ABELARD consulting Pty Ltd

Freephone: 1800 601 116

Telephone: +61 3 9596 3456

Email: courses@abelard.com.au

Web: courses.abelard.com.au

Journal of Technical Writing and Communication

In very few countries where technical writers are employed are there university courses for technical writers. Many technical writers enter the field through unrelated disciplines (and some just fall into it when they discover that what they have been doing for many years is in fact what technical writers do, even though their working rubric connoted nothing obviously related to technical writing (such as Policies and Procedures Writer). Thus there are many practising technical writers unaware of the fact that there is a flourishing academic field delving into the theories and methodologies of technical writing and communication.

One of the few publishing outlets for academics working in the field of technical writing and communication is *The Journal of Technical Writing and Communication*. The journal is compulsory reading for students in most technical writing courses worldwide and is equally informative for those already in the profession. The editor of *Words* asked the publisher for information about the journal. General information appears below, along with the contents of the current volume.

Aims & scope

The Journal of Technical Writing and Communication strives to meet the diverse communication needs of industry, management, government, and academia. For nearly forty years, this award-winning publication has served as a major professional and scholarly journal for practitioners and teachers of most functional forms of communication. Our purpose is to publish a thoroughly solid journal that performs as the needed bridge between academia and the world of practitioners.

Editor

Charles H. Sides
Chair, Department of Communication
Fitchburg State College, Massachusetts

Book Review Editor

Elizabeth Tebeaux
Texas A & M University

Executive Board

Paul V. Anderson
Master's Degree Program in Technical and Scientific
Communication, Miami University, Ohio

David N. Dobrin
B2B Analysts, Cambridge Massachusetts

Carel J. M. Jansen
Department of Business Communication, Nijmegen
University, The Netherlands

Frederick T. Kiley
Director, National Defense, University Press,
Washington, D.C.

John Kirkman
Communication Consultant, Ramsbury,
Marlborough, Wiltshire, England

Bernard J. McKenna
University of Queensland, Ipswich

Frederick C. Mish
Editorial Director, Merriam-Webster, Incorporated

Thomas E. Pearsall
Chair, Emeritus, Department of Rhetoric, University
of Minnesota

Janice C. Redish
President, Redish & Associates, Bethesda, Maryland

Stuart Selber
Department of English, Pennsylvania State
University

Editorial Board

Kirk St. Amant
Department of English, East Carolina University

Paul M. Dombrowski
Department of English, University of Central Florida

Brenton Farber
Clarkson University

Russell Hirst
University of Tennessee

Elizabeth Pass
Institute of Technical and Scientific Communication,
James Madison University

Jan H. Spyridakis
Human Centered Design & Engineering, University
of Washington, Seattle

Glen Thomas
Queensland University of Technology, Australia

Dorota Zielinska
Department of Linguistics, Jagiellonian University,
Krakow, Poland

Contents of current volume

Click a heading to read editorials, abstracts and reviews, or to purchase an article or subscribe.

Volume 39, Issue 1


- From the Editor's Desk, Charles H. Sides, Executive Editor
- The Public Presentation of a Hybrid Science: Scientific and Technical Communication in "Iraq's Weapons of Mass Destruction: The Assessment of the British Government" (2002), Keisha McKenzie
- *Ars Dictaminis* Perverted: The Personal Solicitation E-mail as a Genre, Derek G. Ross

- Frames in Reports and in Reporting: How Framing Affects Global Warming Information in the Intergovernmental Panel on Climate Change's "Summary for Policymakers" and in Documents Written about the Report, Bryan Tutt
- Language Problems to be Coped with in Web Localization, Pinfan Zhu
- Oral Communication and Technical Writing: A Reconsideration of Writing in a Multicultural Era, Sylvain K. Cibangu

Volume 39, Issue 2

- From the Editor's Desk, Charles H. Sides, Executive Editor
- Anti-Employer Blogging: An Overview of Legal and Ethical Issues, Mike Markel
- Presenting Consumer Technology with POP: A Rhetorical and Ethnographic Exploration of Point-of-Purchase Advertising, Geoffrey A. Cross
- Design for Effective Support of User Intentions in Information-Rich Interactions, Michael J. Albers

award-winning, authoritative international voice, publishing the latest research by recognized scholars from around the globe




Every article ever published in the JTWC is now available in clear, concise, comprehensive PDF format.

New Online-Only Delivery Options

- Enhanced subscription
(all articles from volume 1 through current volume)
- Current volume subscription
- Individual articles on pay-per-view basis

Click here for all the details and ordering information.

- Read all article abstracts free
- Download a complimentary sample issue

BAYWOOD PUBLISHING COMPANY, INC.
 <http://baywood.com>

- Composition Studies, Professional Writing and Empirical Research: A Skeptical View, Dana Lynn Driscoll
- The Banality of Rhetoric? Assessing Steven Katz's "The Ethic of Expediency" Against Current Scholarship on the Holocaust, Mark Ward, Sr.

Volume 39, Issue 3

- From the Editor's Desk, Charles H. Sides, Executive Editor
- Risk Communication, Space, and Findability in the Public Sphere: A Case Study of a Physical and Online Information Center, Ed Nagelhout, Julie Staggers, Denise Tillery
- Toward a Rhetoric of Locale: Localizing Mobile Messaging Technology into Everyday Life, Huatong Sun
- The Rhetorical Situations of Web Résumés, John B Killoran
- Examining Editor-Author Ethics: Real-World Scenarios from Interviews with Three Journal Editors, Nicole Amare and Alan Manning
- Introducing Heuristics of Cultural Dimensions into the Service-Level Technical Communication Classroom, Robert Schafer
- Writing an Introduction to the Introduction, James Hartley
- Book Reviews, Elizabeth Tebeaux, Review Editor, Jacqueline Palmer, Elizabeth Smith, Kirk St. Amant, Bernard McKenna

Volume 39, Issue 4

- From the Editor's Desk, Charles H. Sides, Executive Editor
- "Proof" in Pictures: Visual Evidence and Meaning Making in the Ivory-Billed Woodpecker Controversy, Wendy Winn
- Virtuous or Vicious?: Agency and Representation in Biotechnology's Virtuous Cycle, Naomi Sunderland
- Observational Data on Practical Experience and Conditions of Use of Written Instructions, Franck Ganier
- Coherence in Workplace Instant Messages, Jo Mackiewicz and Christopher Lam
- The Two-Semester Thesis Model: Emphasizing Research in Undergraduate Technical Communication Curricula, Julie Dyke Ford, Jennifer L. Bracken, and Gregory D. Wilson
- The Structure of Scientific Titles, Joseph E. Harmon and Alan G. Gross
- Creating Editorial Authority through Technological Innovation, Clinton R. Lanier

Does structured authoring constitute a paradigm shift?

Geoffrey Marnell

What do technical writers need to know in order to gain sustained employment in the technical writing profession? Contrary to the pronouncements of some technology evangelists, knowledge *of* XML (as distinct from knowledge *about* XML) is not necessary. It's true that much software these days is glued together with XML, and true too that some technical writers produce XML-tagged content. But most XML-tagged content is destined for the information technology (IT) domain. And while this domain is certainly an important source of revenue for technical writers, what some technology evangelists overlook is the fact that many technical writers work in non-IT domains: medical equipment, heavy machinery, mining and so on.

But isn't an authoring methodology separate from any specific domain, being a general approach that could be applied regardless of domain? In other words, isn't XML-based authoring suitable whether you are working in IT or in heavy machinery? Well, yes; but recall that the question is whether knowledge *of* XML (as distinct from knowledge *about* XML) is necessary. One can engage in XML-based authoring without needing to know the slightest thing about XML prologs, namespaces, IDREFs and the like. A parallel: do you really need to know the ins and outs of C++ in order to author a document in Microsoft Word or Adobe FrameMaker? Those who created the authoring infrastructure did, but not the day-to-day authors. It's much the same in technical writing. There are documentation technicians who set up the authoring infrastructure for XML-based

authoring—create schemas, DITA specialisations and the like—and there are those who use that infrastructure to author. The former need to know the nuts and bolts of XML; the latter do not.

So those coming new to technical writing need not worry too much if they know little of the nuts and bolts of XML. But what about the content-driven authoring *technique* that produces the XML (or XML-tagged content)? Isn't the move to content-driven authoring from format-driven authoring something new, implying a whole new approach to authoring? Isn't it a *paradigm shift*, something that writers old and new need to embrace?

Some technology evangelists certainly think so:

"The evolution of content creation from format-driven publishing to structured authoring is a paradigm-shifting ... transition."¹

First, what is a *paradigm shift*? The concept was introduced by Thomas Kuhn in *The Structure of Scientific Revolutions*.² Kuhn applied the concept to science. It comes about when a new discovery or new theory, in explaining something that had been considered anomalous by current theories, forces a radical or revolutionary change in scientific worldview. The shift from a geocentric to a heliocentric view of the universe is a paradigm shift. So too is the shift from creationism to evolution, and from classical mechanics to quantum mechanics.

The term has taken on a life of its own and is no longer applied only to scientific revolutions. Thus a move from a Keynesian view to a monetarist view is considered a paradigm shift in macroeconomics. Indeed, the term today seems to mean nothing more than a radical change in thinking about, or doing, something, regardless of domain.

So, is the move to structured authoring—the move to content-driven authoring as opposed to format-driven authoring—a paradigm shift? Does it constitute a radical change in the way technical writers write?

Firstly, does the question really matter? Isn't this merely quibbling over semantics? In one sense, yes; but it's more than just semantics when a term misleads, especially when it misleads those new to our profession or those wanting to break into it. To call a methodology *paradigm shifting* is to imply that the methodology it has supplanted, or is supplanting, is out-moded, erroneous, ineffective and the like: that is, it implies that the new methodology constitutes some intellectual advance on the old. More importantly, it implies that our profession has seen the light and moved across to structured authoring.



"THE RECIPE SAID TO SIMMER UNCOVERED!"

1. Sarah O'Keefe, "XML, Growing Up Fast", INTECOM, July/August 2008, p. 27.
2. University of Chicago Press, 1962.

Thus, for someone wanting to break into technical writing, they would need to master structured authoring.

But this is just not true. Very few technical writing projects adopt a structured authoring approach. For most projects, such an approach would be like taking a sledgehammer to a walnut. Large organisations producing multi-lingual deliverables along multiple delivery channels can gain advantages from a structured authoring approach. But for the common-or-garden projects most of us work on, there would be no gains from a structured authoring approach. So, despite the hoopla of some technology evangelists, those new to our profession, and those wanting to break into it, needn't feel that they must master structured authoring if they are to have any chance of advancing as a technical writer.

But give the new methodology time, some might say. This is all new, and it will take time for it to become as commonplace as today's predominantly format-driven authoring. But structured authoring is not a new methodology at all. It has been around since the 1980s, when Standard Generalised Markup Language (SGML) first appeared, more than a decade before the birth of its child, XML. (Indeed, XML is just a sub-set of SGML.) The hype and hoopla that greeted the arrival of SGML caused many technical writers to attempt to master SGML, and many bought *FrameMaker+SGML*, one of the first structured authoring tools on the market. But SGML has now largely faded away, and it is arguable whether modern approaches to structured authoring will overcome all of the difficulties—including wide-scale irrelevancy—that befell SGML. (SGML can still be found in some organisations, such as Caterpillar and the Department of Defence, but very few technical writing projects worldwide depend on SGML.)

But let's return to the issue of whether content-driven authoring constitutes a paradigm shift, a radical new way of authoring. The old paradigm is *format-driven* authoring, where the predominant concern, apparently, is the *appearance* of our text rather than the types of building blocks that make up the content. The new paradigm has us concentrating solely on content and ignoring formatting (or at least relegating it to the secondary task of applying some stylesheet or coded transform *post-drafting*).

But is this really a new approach to authoring? Does any writer, can any writer, really adopt a format-driven authoring methodology to the exclusion of a content-driven authoring methodology? Suppose, for example, that I am writing a scientific paper describing the results of some research. Do I really say to myself: first, I shall start with some heading 1 text, move on to a

heading 2 text, choose a smaller-than-normal body text format (maybe with some left and right indents) for the next paragraph, then add some more heading 2 text, and then some standard body text, and so on and so on. Of course not. We *naturally* think in terms of content: I start with a title, then I write the authors section. Next I add an abstract (introduced with its own heading), followed by the introduction, then the materials and methods section, followed by the results section, the discussion section, the acknowledgements, appendices and finally the list of references. The whole paper is written in content chunks, not format chunks. We will certainly format the content chunks (and the sub-chunks: headings, lists and so on). But we primarily think of the paper we are writing as composed of chunks of content or topics. Formatting is always secondary. And this is not new.

It is exactly the same with technical writing (and with any form of declarative writing). Formatting is always an after-thought. The fore-thought is the content types that will be the building blocks of my document. I don't say to myself as I am about to begin the steps in a procedure that I am choosing a list format. No, I say to myself that I am about to begin a procedure. This is a content type, not a format

type. I may format it in a particular way—as a numbered list—but that is not the primary consideration. The primary consideration is that I am about to set out the steps to describe how a specific goal can be achieved. When I write a warning or caution, I am not saying in my mind that I am about to enter some bold text with an accompanying danger symbol. No. I say I am about to enter a warning or a caution. This is a decision about content. When I am writing a trouble-shooting section, I don't say that I am choosing a particular format type. No, I am choosing a particular content type. And so on. (This is the logic behind *boilerplate templates*, the most common type of template: here are all the sections, that is, major content blocks; now plug in your text.)

So, in a fundamental sense we have *always* been engaged in structured authoring, despite the relatively recent appearance of structured authoring tools. Far from being a paradigm shift in how we author, structured authoring methodologies are actually doing little more than mirroring, at last, the way we have always authored. *It is the tools that have changed; it is not how we author that needs to change.* The structural components (or elements) that we see in DocBook, DITA and the like are just a reflection of the way we *naturally* chunk our writing, how we build a document from the blocks that are its necessary constituents. If it's always been that way—and it has—then authoring that way cannot be a paradigm shift.

Far from being a paradigm shift in how we write, structured authoring methodologies are actually doing little more than mirroring the way we have always written. It is the tools that have changed; it is not how we write that needs to change.

Strictly speaking, how we author in a structured authoring environment is a little different, but the difference is not such that a would-be or novice technical writer needs to be especially concerned. With the structured authoring we naturally adopt but with a non-structured authoring tool at our disposal (such as Microsoft Word), we are free to construct the structure as we please and free, too, to apply any format to any paragraph (and any character) that we type. With the structured authoring we naturally adopt but with a structured authoring tool at our disposal (such as Structured FrameMaker), we are free to apply whatever content type is appropriate wherever we have our cursor. This is determined by the content rules in the associated schema, document type definition (DTD) or element definition document (EDD). Formatting is another step: it is either applied via an associated style sheet or set out in the format rules specified in the associated EDD. But the principal difference is that with an unstructured authoring tool, you apply styles (aka formats) directly to the text you enter, whereas with structured authoring tools you directly apply content types (aka elements and their qualifying attributes) to the text you enter. Formatting is another step. With an unstructured tool, you select a style to apply to a chunk of text; with a structured tool you select a content type to apply to a chunk of text.

And what is so smart about this is that the structured authoring methodology exactly mirrors the way we author, and the way we have always authored. *This is no paradigm shift or quantum leap. Rather, the tools are catching up with us. It is not us who has to catch up with the tools.*

So don't be alarmed if you are new to technical writing and are confused by all the hype and hoopla about the need to learn XML and the need to embrace a new model of authoring. The model that many are apparently moving to is not new. It is a model that simply reflects the way everybody writes. If you have written anything at all—with quill, crayon, chalk or Microsoft Word—you have more than likely engaged in structured authoring: content first; format second. What is new is that modern authoring tools—from the advent of SGML onwards—can enforce a particular structure on a document. Those new to technical writing can learn how that can be done, if they wish. But not knowing how it can be done is no obstacle to advancing in the profession—especially given that so few technical writing projects insist on structured authoring.

Geoffrey Marnell

Journal



quarterly
52 full colour pages
£45 a year (£10 off for ASTC members)
contribute: journal.editor@istc.org.uk
archive (ISTC members only): www.istc.org.uk/Members_Area/communicator_archive.htm
subscribe: istc@istc.org.uk
advertise: felicity@tou-can.co.uk

Broaden your horizons with subscriptions to



periodicals

Institute of Scientific and Technical Communicators Newsletter



monthly
PDF (website or e-mail)
freely available at no charge
contribute: newsletter.editor@istc.org.uk
archive: www.istc.org.uk/Publications/Newsletter/newsletter_archive.htm

Creating and applying formats in FrameMaker

Mark Ward

Suppose that you have inherited a FrameMaker document and want to check if it has formatting integrity. You can see what styles have been set up by displaying the paragraph and character catalogs. But how do you know if those styles, *and only those styles*, have been used throughout the document? In other words, how do you know:

- if any paragraph or character style has been manually modified or overridden, and
- if there are paragraph or character styles in the document that are not in a catalog?
(Uncatalogued styles arise when text is imported or copied from another document and the text has a style name that is not in a catalog.)

The FrameMaker status bar shows, with an asterisk, if styles `Flow: A ¶: *Body` are modified or uncatalogued, but only for the paragraph or text that the cursor is currently in. Jumping from paragraph to paragraph would be a very slow way to find paragraph format overrides or uncatalogued paragraph styles. Finding *character* format overrides and uncatalogued styles in a similar way—this time jumping from word to word—would take even longer.

FrameMaker provides a feature that helps you quickly find format overrides and uncatalogued formats: *Create and Apply Formats*. This feature creates, and then applies, paragraph and character catalog tags for all paragraphs and character styles in the document:

- If a paragraph or character style is a format override—that is, a manual modification of a pre-existing catalogued style—a new style is created with the same name as the catalogued style but with a unique numerical suffix. For example, ***hyperlink** becomes **hyperlink1** (with the numerical suffix distinguishing it from the parent style **hyperlink**).
- If a paragraph or character style is not in a catalog, a new style is created with the same name. For example, ***steps** becomes **steps**.

Why use Create and Apply Formats?

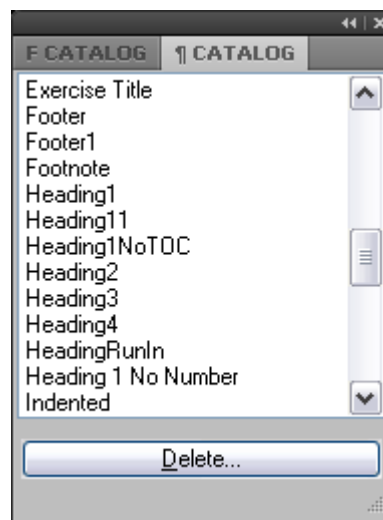
You could use this feature to:

- clean up a document you have inherited
- simplify a document that you want to use as a template
- find and fix manual formatting
- add the formatting imported with a file (for example, a Microsoft Word file or another FrameMaker file) to the catalogs of the current document

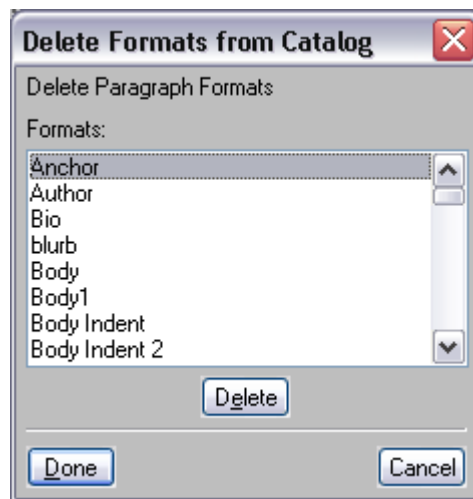
Step 1: Delete all formats from the paragraph and character catalogs (Optional)

This step is not strictly necessary. The reason I do it is so that when I run the Create and Apply Formats feature, the paragraph and character catalogs will only list the styles that are actually used in the document. This helps keep the catalogs uncluttered.

1. Display the paragraphs catalog.
(**Format > Paragraphs > Catalog**)



2. Click **Delete**.
The **Delete Formats from Catalog** dialog box appears.



3. Click **Delete** repeatedly until all the formats are deleted.
4. Click **Done** to close the **Delete Formats from Catalog** dialog box.
5. Display the character catalog.
(**Format > Characters > Catalog**)
6. Repeat steps 3 and 4 to delete all the character formats. (Note: you cannot delete the **Default ¶** font format.)

Step 2: Create and Apply Formats

1. Select **File > Utilities > Create and Apply Formats**.

A message appears warning that if you proceed, there is no **undo** option.

2. Click **Continue**.

FrameMaker creates a format tag for every paragraph and character format.

Note what has occurred:

- If you deleted all the paragraph and character formats (see step 1), the paragraph and character catalogs are repopulated with the original style names *but only for those styles that are actually used in the document*. Thus if you had a **Heading 4** style in the paragraph catalog previously, it will not appear in the paragraph catalog now if no paragraph has been styled **Heading 4**.
- If there were paragraph or character styles in use but they were not previously in the paragraph or

character catalog, there will now be an entry for them in a catalog. (These are the styles introduced when copying or importing text.)

- If a paragraph or character style is a format override—that is, a manual modification of a previously catalogued style—a new style is created with the same name as the catalogued style but with a unique numerical suffix. For example, ***hyperlink** becomes **hyperlink1** (with the numerical suffix distinguishing it from the parent style **hyperlink**).

Note that a numerical suffix is still added to a manually modified style even if the format it overrides was not previously catalogued in the current document. (It may have been a format override in a document or text inset imported into the current document.)

- All the formats created have been applied to the document, and no paragraph or character is a format override.

Step 3: Clean up

Now you can use FrameMaker's **Find/Change** feature to quickly search for the formats you don't want in your document. For example, you might find that you had two manually modified **hyperlink** character formats when only one hyperlink format is necessary. You will see this because you now have **hyperlink**, **hyperlink1** and **hyperlink2** in the character catalog. (The Create and Apply Formats feature has added a style for each manual override.) You can now search for instances of **hyperlink1** and change each one to **hyperlink**. Do the same for **hyperlink2**. (On the other hand, if a manually overridden style is needed, you might consider changing its name to one that more readily denotes its purpose, such as **hyperlink_in_footer**.)

Delete the unwanted styles and run the Create and Apply Formats feature again just to make sure the document formatting is now clean.



**MACQUARIE
DICTIONARY
ONLINE**

www.macquariedictionary.com.au

Subscribe to the complete Macquarie Dictionary online, updated annually with new words and definitions.

Also available online is the full Macquarie Thesaurus - that perfect word is just a click away.

Try it out now for FREE!

Macquarie Online is offering free extended trial access. Simply contact Macquarie Online to set up your 3 months free access. Quote code: 3mfTrialAC

Macquarie Online Support
phone: 1800 645 349
email: support@macquarieonline.com.au



Australia's national dictionary

Mark Ward

Mark Ward is an independent documentation consultant, adviser to Abelard Consulting, and delivers Abelard Consulting's *Adobe FrameMaker* and *Structured Authoring with Adobe FrameMaker* training courses.

What should technical writing be called?: Survey results

Geoffrey Marnell

In an article entitled “Technical writing: what’s in a name?” in the previous issue of *Words*, various reasons were considered for why the longstanding name for our profession—*technical writing*—might be considered inadequate. Over the years, many new names have been adopted by technical writers, but none has cemented itself as a generally preferred alternative name. (*Technical communication* has probably achieved greatest currency of all the alternatives to *technical writing*.)

The *Words* team created a web-based survey designed to gather the thoughts of practising technical writers worldwide on what their profession should be called. The survey also called for humorous titles that befit our profession.

First up, some of the humorous suggestions

bertram	geek-speak interpreter	scatanalyser
blarney interpreter	gobble-de-gook translator	semantic engineer
bull dispersion operative	gobble-de-gooker	sensemater
clarity consultant	ignorance obliterator	sentence constructionist
communication acrobat	information doctors	specification deboggler
communication carer	information distiller	stirologist
communication communicator	information tamer	techrator busters
could-have-been novelist	informationeer	texty
cubicle bard	jargon interpreter	word engineer
de-complicator	language hardhat	word monkey
demystifier	language legotist	word police
disambiguation engineer	manual labourers	wordsnotworth
doculogist	meaning sculptors	wordwrangler
documentationalist	message maestro	wordwright
geek-to-human translator	professional explainer	

The serious stuff: suggested titles

The survey attracted 165 responses and 25 titles for our profession. The titles, in order of popularity, are:

technical writer, 50 (30%)		content specialist, 1 (<1%)
technical communicator, 37 (22%)		content writer, 1 (<1%)
information designer, 24 (14.5%)		developmental editor, 1 (<1%)
technical author, 13 (8%)		document specialist, 1 (<1%)
information developer, 7 (4%)	And the winner is ...	information engineer, 1 (<1%)
documentation developer, 6 (3.5%)	... technical writer	information specialist, 1 (<1%)
documentation specialist, 4 (2.5%)		information technician, 1 (<1%)
instructional writer, 4 (2.5%)		information architect, 1 (<1%)
content developer, 3 (2%)		technical communication professional, 1 (<1%)
documenter, 2 (1%)		technical journalist, 1 (<1%)
communications specialist, 1 (<1%)		user assistance and language expert, 1 (<1%)
content delivery architect, 1 (<1%)		user support designer, 1 (<1%)
content provider, 1 (<1%)		

Suggested titles by country (and in order of preference)

Australia (45 responses)

technical writer, 20 (44.5%)
technical communicator, 10 (22%)
information designer, 5 (11%)
technical author, 3 (6.5%)
instructional writer, 2 (4.5%)
content developer, 1 (2%)
document specialist, 1 (2%)
information specialist, 1 (2%)
information developer, 1 (2%)
user assistance and language expert, 1 (2%)

Canada (6 responses)

instructional writer, 2 (33%)
technical writer, 2 (33%)
information designer, 1 (1.5%)
technical communicator, 1 (1.5%)

India (10 responses)

information designer, 3 (30%)
technical communicator, 3 (30%)
content developer, 1 (1%)
content specialist, 1 (1%)
information engineer, 1 (1%)
technical writer, 1 (1%)

New Zealand (35 responses)

technical communicator, 8 (23%)
technical writer, 7 (20%)
documentation developer, 6 (17%)
information designer, 6 (17%)
documentation specialist, 3 (8.5%)
technical author, 2 (6%)

content writer, 1 (3%)
documenter, 1 (3%)
information architect, 1 (3%)

United Kingdom (17 responses)

information designer, 4 (23.5%)
technical writer, 4 (23.5%)
technical communicator, 3 (17.5%)
technical author, 2 (12%)
communications specialist, 1 (6%)
documenter, 1 (6%)
information developer, 1 (6%)
information technician, 1 (6%)

United States (47 responses)

technical writer, 15 (32%)
technical communicator, 11 (23.5%)
technical author, 6 (13%)
information developer, 5 (11%)
information designer, 4 (8.5%)
content delivery architect, 1 (2%)
content developer, 1 (2%)
content provider, 1 (2%)
developmental editor, 1 (2%)
technical communication professional, 1 (2%)
user support designer, 1 (2%)

Other countries (5 responses)

documentation specialist, 1 (20%)
information designer, 1 (20%)
technical communicator, 1 (20%)
technical journalist, 1 (20%)
technical writer, 1 (20%)

Contractor preferences

Thirty per cent of respondents (50) are contractors. Their preferences are:

technical writer, 19 (38%)
technical author, 8 (16%)
technical communicator, 5 (10%)
instructional writer, 4 (8%)
information designer, 3 (6%)
content developer, 2 (4%)
documentation developer, 2 (4%)
[other, < 2], 7 (14%)

Employee preferences

Seventy per cent of respondents (115) are employees. Their preferences were quite different from those of contractors:

- technical communicator, 32 (28%)
- technical writer, 31 (27%)
- information designer, 21 (18%)
- information developer, 6 (5%)
- technical author, 5 (4%)
- documentation developer, 4 (3.5%)
- documentation specialist, 3 (2.5%)
- documenter, 2 (2%)
- [other, < 2], 11 (10%)

Preference by experience

0–4 years (32 responses)

- information designer, 10 (31%)
- technical writer, 8 (25%)
- technical communicator, 4 (12.5%)
- documentation developer, 2 (6%)
- [other, < 2], 8 (25%)

5–9 years (39 responses)

- technical communicator, 13 (33%)
- technical writer, 10 (25.5%)
- technical author, 4 (10%)
- information designer, 3 (7.5%)
- [other, < 2], 9 (23%)

10–14 years (29 responses)

- technical communicator, 10 (33.4%)
- technical writer, 7 (24%)
- documentation developer, 3 (10%)
- information designer, 2 (7%)

- information developer, 2 (7%)
- technical author, 2 (7%)
- [other, < 2], 3 (10%)

15–19 years (25 responses)

- technical writer, 12 (48%)
- information designer, 6 (24%)
- technical communicator, 4 (16%)
- [other, < 2], 3 (12%)

20+ years (40 responses)

- technical writer, 13 (32.5%)
- technical communicator, 6 (15%)
- technical author, 5 (12.5%)
- content developer, 3 (7.5%)
- information developer, 3 (7.5%)
- information designer, 2 (5%)
- instructional writer, 2 (5%)
- [other, < 2], 6 (15%)

What do most of us do?

The survey also asked what activities take up most of your time. This was an optional question, but with a purpose. It was designed to validate, or otherwise, a gathering belief that writing and editing are becoming minority activities in technical writing. The survey results suggest that this is not the case.

Of the 165 respondents overall, 58 chose not to answer this question. Of the remaining 107, 85 (or 79%) indicated that *writing* and/or *editing* are the activities that take up most of their working time. This leaves 22 (or 21%) who spend most of their working time doing things other than writing or

editing. (Some in this latter group stated that they were team leaders and led teams of folk engaged in writing and editing. Others said that they spend most of their time researching, which presumably means, in many cases, researching topics that they need to *write* about. Hence the proportion of those who are primarily involved in writing and editing is likely to be greater than what is suggested by the raw survey statistics.)

Conclusion: writing and editing still constitute the activities that take up most of the time of technical writers.

Writing and editing are the activities that take up most of the time of technical writers.

What else do we do?

researching
fact checking
planning
managing (or team-leading)
project and people management
instructional design
content design
teaching
illustrating (and graphic design)

interviewing
information design
content management
translation support
coaching or mentoring junior writers
negotiating
template design
photography
GUI design (and review of GUI designs)

A sprinkling of respondents' comments

"I think a name change is warranted because of the changing nature of the software industry. At one time, Technical Writer was the most appropriate name. However, as our skills are increasingly being applied to UI design and usability, and as printed manuals start to die out, I think it's time to let go of the 'writer' word. I believe we are experts in providing appropriate assistance. We do this using our knowledge of language and writing. It is important that we are considered by management whenever either of these (assistance or language) are in question."

"Please, please, please do not suggest made-up words like 'documenter' or, even worse, a term I heard floated at one company where I worked — 'documentationist'! This survey is rather silly if you ask me."

"[I prefer] *instructional writer* because it's slightly closer to reality for those of us who don't always write software manuals. Also, it might require slightly less explaining to acquaintances and neighbours. 'I'm a technical writer. I write and edit instruction manuals and web pages' just takes too long."

"I have found that 'technical writer' is the most easily understood label for our profession. Thanks for running this survey, it has stimulated a lot of debate on techwr-l and elsewhere."

"This is a hopeless but useful [sic] exercise."

"In spite of calling myself a communication consultant on my business card, I still think of myself as a tech writer and that's what I believe the profession should be called forever. Let's not try to confuse the world."

"Keep it simple with the name that has been around and is more or less understood. Changing it—even if more descriptive/accurate—will only create confusion and more explaining. When I am at a party I say 'I don't write software, I write *about* software' and people say 'Oh, you're a technical writer, and you do user manuals?' Yep. And

marketing copy and web site and other stuff. They get it."

"'Designing' info is a bit pretentious, isn't it, but often the design part of the job takes up as much, if not more, time than actually writing or editing it."

"Do *not* change the name. We deal with words and those words have to be written. We are writers. 'Technical' doesn't have to mean engineering; think of legal cases where someone gets off on a 'technicality'. When we talk about technical writing, we mean precision and accuracy. Poets can afford to be vague and leave the interpretation to the reader; technical writers cannot. The phrase just means that the words you use must be unambiguous, clear and concise. Therefore, whether you are explaining how to appeal against a parking ticket, describing how to change the brushes on an electric motor, or providing practical advice on achieving creative results with watercolours, what you are doing is technical writing. Learn to live with this."

"It surprises me that this debate crops up so often in a profession that supposedly rejects jargon."

"Technical author/writer' is what we're called, whatever the job entails. Companies understand this term. If you use anything else you're diluting the 'brand' and using jargon. Perhaps we should call teachers 'learning facilitators' or doctors 'wellness promoters'. Perhaps we should grasp more firmly a term that is already commonly used so that we can get more recognition as a profession."

"The problem with finding a term for our profession is that technical communication is such a broad and varied stripe. Not all tech comms is instructional, not all is help development or end-user documentation or even written documentation. That's why we can't agree on a term that both fits the profession and improves its image."

"I don't think that technical writing/comms etc is a discrete profession any more, and it's all a bit

People's Front of Judea versus Judean People's Front (to misquote *Life of Brian*). The issue for us is to get people to see that we are communicators, technical or otherwise."

"I've been called a Technical Author in England, plus I've been called an Information Developer, and now a Technical Writer. I've also been a Documentation Manager and a Technical Publications Manager. It doesn't really matter what we're called as long as we're employed and make a difference to the organisations that employ us through the value of our publications. When we stop adding value through our publications we won't get employed and we'll be called *unemployed*. Technical Writers/Technical Authors/Information Designers/Information Developers have resisted certification/registration that most professionals have. I've recently studied and qualified as a Network Administrator (CCNA) to help me write networking documentation in my current organisation, and have found the IT profession is now very heavily certified. The IT networking staff I work with are universally certified whether its Cisco CCNA or Microsoft MSCE."

"Many of the technical writers I've worked with over the years are neither *technical* nor *writers*, but prefer to work as desktop publishers. I think companies frequently see the two as the same, and they are frequently shocked when a true technical writer gets involved."

"Hopeless task ... you will never get consensus."

"Does the name matter? Surely a definition of what we *can* do is more important?"

"I am quite happy being a technical author, mainly because it amuses me to see the blank looks when asked what I do for a living!"

"It depends on what mood I am in and how many times I have been stonewalled by a developer or programmer, but the term 'Glorified Secretary' or 'Glorified Typist' sometimes makes me laugh and sometimes makes me want to shout at people that technical writers are not just typists, that research

and asking questions are more part of the job than the actual writing."

"I'm not in favor of the current push to call those of us in the profession *technical communicator* on the theory that technical writer doesn't describe everything we do. Well, my perspective is that technical writer is so entrenched in the marketplace that it's not going to go away and we're not going to be able to change it."

"The term *technical writer* is established in the industry. What you need is a robust glossary definition of the term. Why not ask for that?"

"Changing names at every subtle change of the main theme of our work is just plain silly. I am still having to explain *IT* and *ICT* to people. I tell them it is a silly name for *computing* and they are happy."

"I think a variety of names reflecting similar skills is probably the best. There are a variety of different jobs performed by people in our profession. My resume would probably not qualify me for the types of jobs performed by others who belong to the same professional organisations as I do. I also think this largely reflects the state of play in other professions: very few professions have names which match with universal functions."

"I like the existing, long-standing name and believe that it probably just needs to be marketed better. I like the idea that technical writing is a profession that belongs in many industries and that, like many professions, technical writers will probably do less writing and more management as their careers unfold. No to a name change!"

"If we want to create any kind of change in the recognition of our profession, then we need to have companies and HR groups understand and use our preferred titles. Or go the way of IBM, and have our responsibilities define us (e.g. their staff titles are *technical staff members*, not software engineers or tech writers). We can prefer to be called *Supreme Goddess of Words*, but that is meaningless unless the person reading our resume or bio understands that." ■

AUSTRALIAN STYLE

A NATIONAL BULLETIN ON ISSUES IN AUSTRALIAN STYLE AND ENGLISH IN AUSTRALIA

Announcing the online rebirth of the biannual newsletter published at Macquarie University, featuring:

- o articles on language usage and research
- o the Feedback questionnaire
- o the exclusive crossword Rubicon by David Astle
- o cartoons by Judy Dunn
- o book reviews, a new word column, and much more

Website: http://www.ling.mq.edu.au/news/australian_style.htm

Contact: adam.smith@mq.edu.au

Contributions Wanted

If you would like to contribute an article or review to **Words**, send an outline of what you would like to contribute to the editor at words@abelard.com.au.

Words is published four times a year: in February, May, August and November. Contributions must be on a topic of some relevance to technical communication.

Rough drafts will not be considered. You should only submit a draft if it is in a state that you consider is worthy of being published.

The editor reserves the right to make editorial changes, but will not publish your contribution unless you have agreed to any changes that have been made.

Copyright is retained by contributors.

Converting auto-numbers to text in MS Word

Bruce White

Microsoft Word's Achilles heel is its automatic numbering. While you are writing, automatic numbering is almost manageable. But when you are editing there can so easily come a time when you just want to say "I am not going to take this anymore!"

One solution is to convert all the automatically generated numbers to editable text before you start editing. You can do this with the following simple macro:

```
Sub RemoveNumbers()  
ActiveDocument.ConvertNumbersToText  
End Sub
```

The macro will "burn off" all the automatic numbering. The numbers don't go away: they are still there, but as editable text.

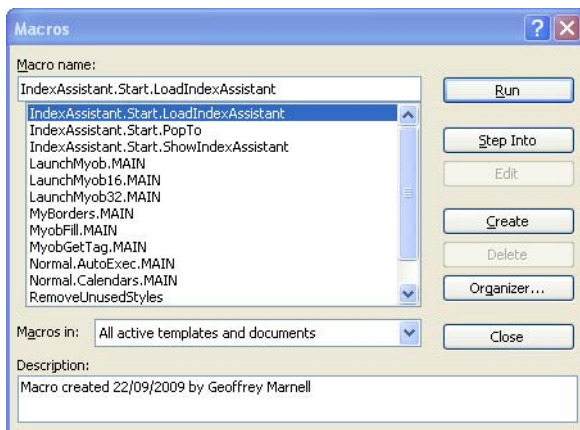
I have used this with Microsoft Word 2000 and 2003.

Step by step: create the macro

If you are not familiar with Microsoft Word macros, here is how you create the macro in question:

1. Select **Tools > Macro > Macros...**

The **Macros** window appears:



2. In the **Macro Name** field at the top of the window, overwrite the name currently shown with the name you want to give your new macro, say, `ConvertAutoNumbers`.

Note that there can be no spaces in the name.

3. Click **Create**.

The *Microsoft Visual Basic* application opens. Note the automatically inserted code in the **Normal - NewMacros (Code)** window:

```
Sub ConvertAutoNumbers()  
'  
' ConvertAutoNumbers Macro  
' Macro created 22/09/2009 by Bruce White  
'  
  
End Sub
```

4. Place the cursor at the start of the blank line above `End Sub` and type:
`ActiveDocument.ConvertNumbersToText`
Your code should now look like this:

```
Sub ConvertAutoNumbers()  
'  
' ConvertAutoNumbers Macro  
' Macro created 22/09/2009 by Bruce White  
'  
ActiveDocument.ConvertNumbersToText  
End Sub
```

5. Select **File > Close and Return to Microsoft Word**.

The macro can now be run whenever you want to convert automatically generated numbers to ordinary text.

Step by step: run the macro

1. Select **Tools > Macro > Macros...**

The **Macros** window appears, showing a list of all the macros available to you.

2. Click once on the name of the macro you want to run: `ConvertAutoNumbers`.
3. Click **Run**.

You'll notice that all the automatically generated numbers are still in place, but they are now editable text. They will not automatically change if you add or remove a numbered paragraph from a sequence of numbered paragraphs. Nor will they change in the crazy way they sometimes do when Microsoft Word is trying to be smarter than it really is.

Bruce White

For more information about creating and running macros, visit <http://www.winhelp.com.au>. Bruce is on Twitter at @winhelp3.

Miscellany

A strange fad indeed

It has never been conventional practice in English to place a single adjective or adverb in parentheses. We don't write, for example, *The sky was full of (pink) clouds* or *He (loudly) abused the umpire*. The parentheses serve no purpose at all.

However, this usage has started to appear, given momentum, perhaps, by its use by reputable writers:

"His approach ... is to situate us exactly in the middle of the (logarithmic) spectrum of magnitudes ranging from the astronomical to the sub-atomic."¹

Here are some more examples, with the meaning of each difficult to infer:

"To be the successful applicant you must have a PhD or equivalent qualification in (organic) chemistry ..."²

"concentrated in quartz–carbonate–(muscovite) filled fractures"³

Further, it is not conventional English to place a *string* of adjectives or adverbs in parentheses. We don't write, for example, *The sky was full of (fluffy pink) clouds*. And thus examples like the following also exhibit unusual and unnecessary punctuation:

"... just as the postwar social democratic model ran out of road after three (far more successful) decades in the mid 1970s."⁴

"The (historically mined) magnesite deposits of Lake Cargelligo ..."⁵

A related oddity is the placement of a single word between dashes:

"This is the definition of system switch, the same phenomenon that Flannery foretells in the realm of Gaia if we do not – quickly – step in."⁶

The author obviously wanted to emphasise the word *quickly*, but the conventional way of doing this in English is to use italics (or bold on screen):

This is the definition of system switch, the same phenomenon that Flannery foretells in the realm of Gaia if we do not *quickly* step in.

Today's conventions were yesterday's fads, so we might all be doing this in fifty or so years. But if your goal in writing is to get your message across with the least effort and distraction on the part of the reader,

fads are best eschewed, especially when they add no novel communicative richness to our writing.

How many languages are there?

According to recent research by the Summer Institute of Linguistics, there are 6909 living natural languages in the world.⁷ That sounds like a lot of languages, even when you consider that 473 are nearly extinct (that is, with only a few elderly speakers still living). However, in an earlier study by the Institute, conducted in 1999, 51 languages were found to have only one surviving speaker (and 28 of those were Australian indigenous languages). Further, 500 languages had fewer than 100 speakers, 1500 had fewer than a 1000 speakers and 5000 languages had fewer than 100 000 speakers. The Institute also concluded that 96% of the world's languages are spoken by just 4% of people.

So languages are dying out and, at the current rate, about half will be extinct by the end of this century (that is, there will be no native speakers of those languages still alive). *That equates to one language dying out approximately every two weeks.*⁸

Is this a bad thing or a good thing? The extinction of a language is tantamount to the extinction of a culture, and the less cultural diversity there is, the less rich are our lives (not only for those whose ancestral tongue has been lost).

And yet fewer languages also means that more people can more easily communicate with each other. At a time when global challenges impose a risk not merely to languages but to whole species—including *Homo sapiens*—perhaps greater global communication and understanding is the silver lining to the dark cloud of cultural loss.

■ [Nikki Ward's review of *Dying Words: Endangered Languages and What They Have to Tell Us* by Nicholas Evans will appear in the next issue of *Words*.]

2010 Technical Communication Summit

The Society for Technical Communication's 2010 technical communication summit will be held in Dallas Texas between 2 and 5 May. The venue is the Hyatt Regency Dallas at Reunion. Registrations open on 1 December. For further information, visit www.stc.org.

1. Richard Dawkins, *The Oxford Book of Modern Science Writing*, OUP, Oxford, 2008, p. 4.
2. Ad in *New Scientist*, 17 May 2008, p. 53.
3. PM Downes, "Petrology and ore microscopy of samples from the Calarie gold deposit", NSW Dept. of Primary Industries, Geological Survey Report G52008/0458, p. 4.
4. *The Sunday Age*, 28 December 2008, p. 13.
5. *Cargelligo 1:250 000 Geological Sheet: Explanatory Notes*, NSW Dept. of Primary Industries, 2005, p. 196.
6. Alanna Mitchell, *Quarterly Essay*, issue 32, 2008, p. 110.

7. See <http://www.sil.org/>. Viewed 12 September 2009. The languages are listed in the 2009 edition of the Institute's publication *Ethnologue*.
8. See David Crystal, *How language works*, Penguin, Camberwell, 2006, pp. 336–7. The 1999 Summer Institute of Linguistics's data, and the rate of language loss, is taken from Crystal's book.

Mindstretchers

Geoffrey Marnell

A winner must a loser make

On each of seven days, Allan, Bertram and Craig raced each other in the swimming pool. The record shows that Allan beat Bertram more times than Bertram beat Allan, and Bertram beat Craig more times than Craig beat Bertram. The record also shows that Allan came first, Bertram second and Craig third in three of the seven races, this being the most consistent result of all. Given just this information—and the fact that no races were drawn—what is the maximum number of times Craig could have beaten Allan during these races?

Dire divisional difficulties?

Write the numbers 1 to 20 on twenty slips of paper, one number to a slip, and place all the slips in a bowl. Randomly draw out the slips of paper and place them side by side with the numbers facing upwards. What is the probability that the 31-digit number so formed is divisible by three?

Solutions will appear in the next issue of *Words*.

Volume index

Volume 1, Issue 1: February 2009

- *Why good writing matters*, Geoffrey Marnell, 1
- *Getting metadata into a PDF*, Mark Ward, 3
- *Creating custom web styles with Dreamweaver CS4*, Geoffrey Marnell, 5
- *Quotation marks: over-used and under-done*, Geoffrey Marnell, 9
- *Localising documents with FrameMaker (Part 1)*, Geoffrey Marnell, 11

Volume 1, Issue 2: May 2009

- *How words get into the Macquarie dictionary*, Sue Butler, 1
- *Designing documents for maximum comprehension*, Geoffrey Marnell, 3
- *Localising documents with FrameMaker (Part 2)*, Mark Ward, 7
- *Entering uncommon characters*, Geoffrey Marnell, 10
- *Unnecessary precision: single-plural composites*, Geoffrey Marnell, 13

Solutions to the last puzzles

Puzzle 1

	2	5	7
×		3	6
1	5	4	2
7	7	1	
9	2	5	2

Puzzle 2

Half marks only if all you could find was that the second word in each set is an antonym of the third. But full marks if you found that, with the exception of the second set, the third word is an anagram of the first word and the second is an antonym of the third word.

Volume 1, Issue 3: August 2009

- *Contracting in the UK*, Dan Smith, 1
- *In defence of the passive voice*, Geoffrey Marnell, 3
- *Technical communication in the UK*, Marian Newell, 9
- *Technical writing: what's in a name?*, Geoffrey Marnell, 14
- *Survey: What should technical writers be called?*, 8
- *Superscripts and subscripts in HTML*, Geoffrey Marnell, 19

Volume 1, Issue 4: November 2009

- *Australian Style Online*, Adam Smith, 1
- *A niche for DITA*, Dan Smith, 3
- *Journal of Technical Writing and Communication*, Baywood Publishing, 6
- *Does structured authoring constitute a paradigm shift?*, Geoffrey Marnell, 8
- *Creating and applying formats in FrameMaker*, Mark Ward, 11
- *What should technical writing be called?: Survey results*, 13
- *Converting auto-numbers to text in MS Word*, Bruce White, 18



To subscribe, send an email to:
words@abelard.com.au