

WORDS



A Quarterly Bulletin for Technical Writers & Communicators

Volume 1 | Issue 2 | May 2009

The beginning

- *Gerund grinder*: what a splendidly evocative term for a stickler: the sort of pedant who decries any change in language, arguing, wrongly, that change is necessarily diminution. These are the folk who scorn modern-day dictionaries for their descriptivism, for their respect for the common tongue. But not every word we hear or read makes it into a dictionary. As Sue Butler, publisher of the Macquarie Dictionary, points out in the first article in this issue, a word needs wide-ranging and settled currency before it gains admission. Alas, *gerund grinder* no longer cuts the mustard.
- *Single-sourcing* has been a buzz term in technical writing for the past five or so years, and some very high-tech solutions have been proposed to provide it. Many technical writers are unaware that Adobe FrameMaker—the long-time rival to Microsoft Word—has provided single-sourcing functionality for a very long time. Mark Ward explores single-sourcing in FrameMaker in his article on conditional text, a feature especially suited to localisation and translation.
- It is hard to escape the evidence of the modern writer's penchant for sans serif fonts. It's not just the web that's awash with such fonts—Arial, Helvetica, Verdana and the like. Many corporates have chosen a sans serif font for all corporate communication: web, email and printed. (Qantas has chosen Arial, a font of dubious quality.) University students are preferring sans serif; so are many arms of government. It seems that everyone is—with the exception of what you are reading now. Given this obsession with sans serif fonts, it seems only timely to revisit research into the link between comprehension and font choice. Colin Wheildon's research—which showed that serif fonts well and truly out-rank sans serif fonts in tests of comprehension—might have been limited to paper texts, but the fact that most online readers print out documents that call for intensive reading suggests that Wheildon's research is also relevant to the design of documents published online. This matter is considered in detail in this issue.

Geoffrey Marnell

Editor [geoffrey@abelard.com.au]

Contents

How words get into the Macquarie Dictionary.....	1
Designing documents for maximum comprehension ...	3
Localising documents with FrameMaker (Part 2).....	7
Entering uncommon characters	10
Client profile.....	13
Unnecessary precision: single-plural composites	13
Miscellany.....	15
Mindstretchers	16

How words get into the Macquarie Dictionary

A common question from the users of a dictionary to its editor is "How do words get into the dictionary?" Of all the countless words that are flung around the world each day, how is it that some are admitted to the select group encompassed by the dictionary headword list?

The test for inclusion is always currency, within the whole community or within a section of it. If enough people know the word and use it as a token of meaningful exchange, then it ought to be in the dictionary. Finding evidence of this comes down to finding instances of a word's use from a variety of different sources over a reasonable period of time—say five years. This second test ensures that a word has a permanent place in our English and isn't just a one day or one year wonder. We all know words which the media have picked up, tossed around like a brightly coloured ball and then abandoned.

For all the words that are in the dictionary there are in fact many more that could be there, particularly words from specialist activities. The jargon associated with computers could go on forever so the editor of a general dictionary has to decide how much of it the average writer or reader or computer user is likely to come across in their lives. Similarly the jargon of skateboarding is extensive and still developing, but while it has high currency among skateboard riders, how much of it should be included in a general dictionary? Again, general currency is the test. Who would have heard of the

term *moral hazard* until the governments of the world started discussing whether or not they should bail out the banks and what the terms of such a bailout should be. The discussion is faithfully reported in our newspapers and the term is included in the dictionary for those who might wish to get a better handle on it.

The dictionary's reach should also extend into the past to those words which once had currency but which no longer do so. The ghosts of these words still drift across contemporary life, particularly in our reading of earlier periods of literature. And where should the reader turn for an explanation if not to his or her dictionary.

How does a word come to the attention of the dictionary editors? There are lots of fish in this verbal sea but how do the fishers of words get them into the net?

There is a low-tech way and a high-tech way. The low-tech way is the way of Dr Johnson and of many dictionary editors in the past. The editor reads, and listens, and makes a record of anything interesting that they find. Words like those included in the latest update of the dictionary online have been gathered by watchful editors—words like *sexting*, *flashpacker*,

adventure running, *bromance*, *plastic soup*, *scene kid* and *helicopter parenting*. (For meanings go to www.macquariedictionary.com.au.)

The high-tech way is to get the computer to do the work of collecting and analysing texts. Data is plentiful these days and a spellchecker is the ultimate if rather crude means of identifying words not in the dictionary.

Finally we rely on contributors, these days mostly contributors from our website. A new word here, a meaning there, a definition which could be worded a little better. All offered in a spirit of helpfulness and prompted by a shared interest in Australian English and a love of language.

Susan Butler

Susan Butler is Publisher of Macquarie Dictionary.

For details in how to get a free three-month subscription to Macquarie Dictionary Online, see the notice below. You can find the dictionary at www.macquariedictionary.com.au.

To learn about some of the new words recently added to the Macquarie Dictionary, see "Word of the Year: toxic debt" on page 15.



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

commercial
translation
centre

For Effective
Translation/Localisation
Of Technical Documentation
In 80 Languages

Australia Wide Service Since 1988
Call 1800 655 224 (Australia Toll Free)
Email: mail@ctc.com.au

commercial translation centre

Designing documents for maximum comprehension

Between 1982 and 1990, editor and publisher Colin Wheildon conducted a number of experiments to determine if font, and the style of font, can significantly affect readers’ comprehension and recall. In some experiments, participants were asked a series of questions related to an article they had just read (thus testing their recall); in other experiments, they were simply asked to rank a number of styles according to the ease with which they can be read.

Font choice: serif or sans serif?

A *serif* is a short line added to the top and bottom of the strokes in traditional typefaces. A *sans serif* font is a font where the characters do not have serifs. In addition, serif characters mostly use strokes of varying widths while sans serif characters use strokes of the same widths. The difference is illustrated in figure 1.

Research conducted by the British Medical Council in 1926 suggested that serif fonts are easier to read than sans serif fonts. Publishing houses have for long set the body of their books in serif fonts. Wheildon set out to corroborate the results of the British Medical Council research, and to see if there was science behind the font decision of commercial publishers.¹

Wheildon’s research backs up the view that serif fonts are to be preferred for text that requires concentrated reading. He divided numerous groups of subjects into two subgroups. Half were asked to read articles with the body text set with a serif font and half were asked to read the same articles with the body text set in a sans serif font. The subjects were then given a series of comprehension tests (being 10 questions relating to the content of each article). The result Wheildon got is rather staggering: the number

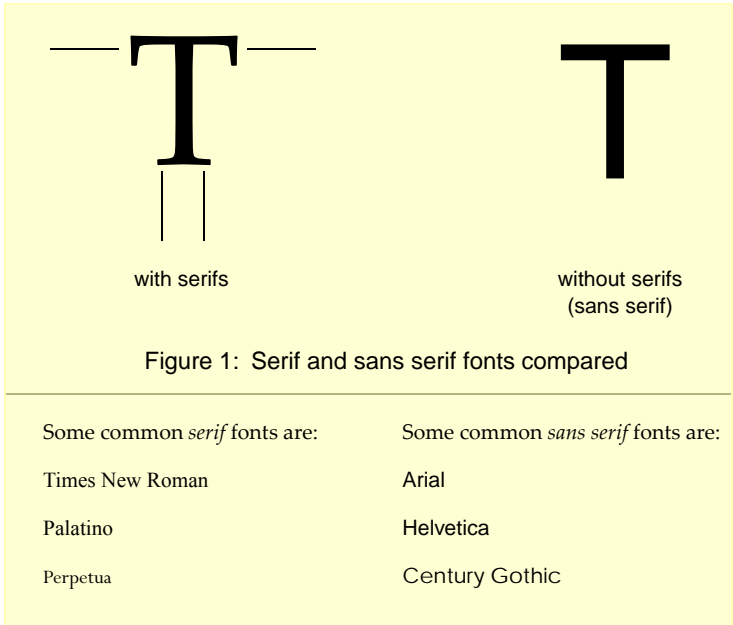
of subjects scoring 70% or higher in the tests was *more than five times higher* if they had read the articles set in a serif font (see table 1).

This was not an experiment to determine what fonts the subjects preferred; rather, it was a test of how well

the brain grasps and holds information presented in different typefaces. Many of Wheildon’s subjects in the sans serif sub-groups reported difficulty maintaining concentration after reading a dozen or so lines of text, and many reported that they needed to backtrack continually in order to maintain concentration.

Recommendation

Body text for material that is to be printed should be set in a serif font.



And for headings?

Wheildon presented his subjects with numerous styles for headings—serif, sans serif, sentence case, all upper case and so on—and each was asked if the styles were “easy to read”. The results:

Table 1: Comprehension scores after reading serif and sans serif body text²

	Percentage of correct answers given		
	70–100%	40–69%	0–39%
Serif font	67	19	14
Sans serif font	12	23	65

- serif sentence case and sans serif sentence case headings were declared easiest to read (by 92% and 90% of subjects respectively).

- serif sentence case and sans serif sentence case headings *set in italics* were declared easy to read by 86% of subjects.

- headings set all in upper case (serif or sans serif) were not as easy to read as headings set in sentence case (with a score of between 3% and 71%).

Recommendation Use either serif or sans serif fonts for headings, *but only in sentence case*.



1. C Wheildon, *Type & layout: Are you communicating or just making pretty shapes*, Worley, Mentone, 2nd edn, 2005.

2. op. cit., p. 47.

Capitals for body text?

As shown in table 2, Wheildon found that his subjects overwhelmingly did not find upper case text (that is capitals) easy to read, whether serif or sans serif.

Table 2: Ease of reading upper case text¹

	Easy to read	Not easy
Upper case, serif	7%	93%
Upper case, sans serif	7%	93%
Sentence case, serif	100%	0%
Sentence case, sans serif	22%	78%

Recommendation Do not set body text all in upper case.

To bold or not to bold?

Wheildon also presented his subjects with texts set solely in roman or solely in bold (with both texts being set in a serif font). The subjects were given comprehension tests. The results, given in table 3, strikingly show that bold should not be used for large slabs of text.

Table 3: Comprehension scores after reading body text set roman or bold

	Percentage of correct answers given ²		
	70–100%	40–69%	0–39%
Roman	70	19	11
Bold	30	20	50

Recommendation Use bold text sparingly.

To slant or not to slant?

Wheildon also presented his subjects with texts set solely in roman or solely in italics (with both texts being set in a serif font). The subjects were again given comprehension tests. The results, given in table 4, show that italicising text does not reduce readers' ability to comprehend it.

Table 4: Comprehension scores after reading body text set roman or italic

	Percentage of correct answers given ³		
	70–100%	40–69%	0–39%
Roman	67	19	14
Italic	65	19	16

Recommendation Although comprehension did not suffer as a result of recasting text in italics, Wheildon did note that his subjects found the all-italic setting

“unusual in such volume”. Thus it might be wise to use italics sparingly. (In print, italics are traditionally used to show emphasis.)

Ragged or justified?

Wheildon also tested whether paragraph alignments affect comprehension. Subjects were asked to read various texts, some set fully justified, some ragged right, and some ragged left. The results, given in table 5, show that fully justified text lent itself to the highest comprehension scores.

Table 5: Comprehension scores after reading body text set fully justified or ragged

	Percentage of correct answers given ⁴		
	70–100%	40–69%	0–39%
Fully justified	67	19	14
Ragged right	38	22	40
Ragged left	10	18	72

However, Wheildon doesn't mention whether his sample texts were created using typesetters or desktop word-processing tools. Typesetters are authoring systems of far greater sophistication than most of today's word-processing tools, and they justify text by adjusting the space between words *and* between characters. Microsoft Word, for example, creates justified text by adjusting only the space between words. The result is usually noticeably inconsistent word spacing (sometimes widely inconsistent) and often rivers of white space, both of which most readers find distracting.

These effects are reduced, but often not eliminated, by applying hyphenation. But hyphenation brings its own problem. There is a break in information absorption as the reader's eye moves from the head of the hyphenated word (at the end of the line) to the tail of the word (at the start of the next line). In some cases, readers correctly guess what the word is and there is minimal distraction; but in other cases, readers guess wrongly what the word is and then have to adjust their interpretation. This is an impediment to readability.

Recommendation Justified text is more compressed than, say, ragged right, and thus should be a consideration if space or page count is a design limitation you cannot avoid. Moreover, justified text is fine if you are composing your text on a professional typesetter (providing that the column width is not so small that ungainly spaces appear between words, as you often see in newspapers).

But if, as is more likely, you are using a commercial word processor (such as Microsoft Word) justified text usually results in uneven spacing between

1. op. cit., p. 100.

2. op. cit., p. 51.

3. *ibid.*

4. op. cit., p. 59.

words. This in itself is distracting, and even more so if spaces even partly overlap across a number of consecutive lines, creating distracting rivers of white space. Ragged right avoids these problems.

“... unjustified text is generally recommended because there is no completely satisfactory typesetting program for desktop production that automatically produces justified text with even word spacing.”¹

“Inconsistent amounts of white space, caused by spreading out of words to fill lines, have a disruptive effect.”²

For on-screen text

The recommendations given above apply to material that is to be printed. For material that is to be read primarily online, Wheildon suggests that many of these recommendations need to be reversed:

“people tend to use different reading modes for screen work, and screen definition is very poor compared with print definition. Indeed, the results [of the research] are almost turned on their head.”³

In other words, serif might be essential for text that is to be printed, but sans serif is best for text that will be read online.

Indeed, online designers do appear to prefer sans serif fonts, unlike book designers. But the issue of designer (or reader) preferences is distinct from the issue of the of communicative effectiveness, and some studies—not all—have shown that comprehension of material presented on screen is greater if *serif* fonts are used.⁴

Online or print?

But what fonts and styles are best suited for online reading is only part of the equation, *and may even be irrelevant for some types of material*. For in deciding on fonts and styles for online delivery, you need to consider whether the document you are writing is likely to be read online, or merely skimmed and then, perhaps, printed for reading offline.

Given the relatively poor resolution of computer screens—poor relative to the resolution provided by even a low-end laser printer—it is not surprising that many people prefer to read a printed version of what they encounter on screen:

If we are preparing documents designed for intensive reading, we should adopt styles that are best suited to offline reading, regardless of the intended medium.

Few topics that technical writers write about amount to one or two sentences. It is simply impossible to describe a complex concept, or explain a multistep process or procedure,

in a handful of sentences. So if the document you are writing fits into this category, you might consider adopting fonts and styles more suited to printed material than to on-screen material, *for offline is how most of your readers will read it*.

Recommendation Online delivery of information might be relatively inexpensive, and the documents more widely available and easier to find, but issues relating to comprehension and common reading practice suggest that:

if we are preparing documents designed for intensive reading, we should adopt styles that are best suited to offline reading, *regardless of the intended medium*.

Why do many people prefer to print out material delivered online? There are at least three reasons. First, reading from the screen is slower—by about 25%—than reading a print-out of the same material.⁵

Second, the online environment is overly distracting. A study reported in *Scientific American* in late 2008 found that:

“reading online may not be as rewarding—or effective—as the printed word. The reasons: the process involves so much physical manipulation of the computer that it interferes with our ability to focus on and appreciate what we’re reading;

5. C van de Velde & M von Grünau, “Tracking eye movements while reading: Printing press versus the cathode ray tube”, *Perception*, 2003, <http://www.perceptionweb.com/abstract.cgi?id=v031179>. Viewed 15 January 2008.

6. R Ackerman & M Goldsmith, “Learning Directly From Screen? Oh-No, I Must Print It!: Metacognitive Analysis of Digitally Presented Text Learning”, *Proceedings of the Chais conference on instructional technologies research*, 2008, <http://telem-pub.openu.ac.il>. Viewed 10 April 2008.

7. Jakob Nielsen’s “Why web users scan instead of read”. Available at <http://www.useit.com/alertbox/whyscanning.html>. Viewed 9 April 2008.

8. *ibid.* Results of between 20% and 30% have been reported in other studies.

1. *Style manual for authors, editors and printers*, 6th edn, John Wiley & Son, Canberra, 2002, p. 335.
2. J Kirkman, *Good style: Writing for science and technology*, 2nd edn, Routledge, Oxford, 2006, p. 107.
3. Wheildon, *op. cit.*, p. 211.
4. S Morrison & J Noyes, “A comparison of two computer fonts: Serif versus ornate sans serif”, *Usability News*, 2003, vol. 5, iss. 2. Available at http://www.surl.org/usabilitynews/52/UK_font.asp. Viewed 8 April 2009.

online text moves up and down the screen and lacks physical dimension, robbing us of a feeling of completeness ... The visual happenings on the screen and your physical interaction with the device is distracting ..."¹

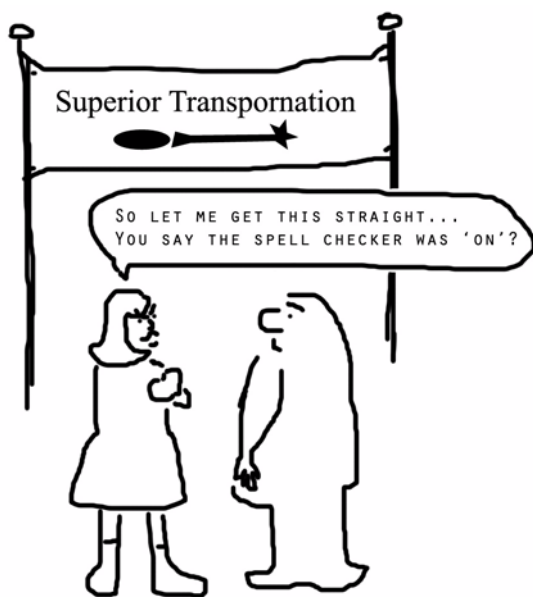
Third, and no doubt related to the previous point, comprehension of material is less when presented online:

"Most people, including our survey participants, believe that they learn less efficiently when reading from a computer screen than when reading from paper. We investigated the validity of this belief [and found that] on-screen learners performed worse than on-paper learners under self-regulated study."²

In fact, this particular study found that comprehension of online material was about 87% that of the comprehension of offline material.

Deciding on the right medium

If, as noted in the previous section, comprehension of online material can be 87% of the comprehension of the same material presented offline, could there be, on occasion, a moral obligation on technical writers to provide paper-based documentation?



© Matthew da Silva 2009

1. "Online v. print reading: which one makes us smarter?", *Scientific American*, 23 December 2008, <http://www.sciam.com>. Viewed 15 January 2008.
2. R Ackerman & M Goldsmith, op. cit.

Consider situations where the readers of our user guides could conceivably suffer injury or even death if they misunderstood what we had written. Some examples might be user guides describing how to use heavy machinery or medical equipment, or explaining what to do in an emergency at a nuclear power plant. Ensuring maximum comprehension in such circumstances is surely mandatory, in which case paper documentation would have to be a priority.

It is sometimes stated that *younger* readers prefer online reading to offline. As noted above, the empirical evidence doesn't appear to back this up, but let's suppose, for the sake of argument, that it is true. Suppose, further, that the majority of readers of the documentation we are writing are likely to be young folk. Should we give them the documentation only in the format they prefer? Given the greater likelihood of misunderstanding from online reading, perhaps a risk of injury or death might morally outweigh satisfying our readers' preferences.

Even in cases where misunderstanding is unlikely to lead to death or injury, there may be good reasons to override user preferences. Suppose, for example, that you manage a call centre. You have to decide whether the printed knowledgebase that your support staff now relies on should be converted to online and then decommissioned. You poll your staff and they all say that they prefer reading online (being of that generation that supposedly prefers online reading). But if comprehension of online instructions is markedly less than that of equivalent printed materials, the risk of customers getting poor advice from support staff is higher if support staff have only online references to rely on. Hence online delivery could lead to greater customer dissatisfaction, and possibly even more support calls (as disgruntled callers call back for further assistance). Online might be cheaper, easier to maintain and preferred by your staff—and yet possibly a poor business decision in the long run.

In a similar vein, what if our audience has a strong preference for sans serif fonts over serif fonts? Given the results of Colin Wheildon's research, we might again have an obligation to ignore those preferences, especially if there is a risk of injury or death should they not fully understand the instructional materials we are writing for them.

Thus in some cases, the *reader-is-king* philosophy that underpins much of what we do in technical writing may need qualification.

Geoffrey Marnell

► **Subscribe to Words**

Words is a free, quarterly bulletin for technical writers and communicators—indeed, for anyone with a passion for words and for what you can do with them. To subscribe, send an email to:

words@abelard.com.au

Localising documents with Adobe FrameMaker (Part 2)

In the last issue of *Words*, we described how to use one of FrameMaker's single-sourcing features, variables, to ease the task of localisation. In this issue we describe how *conditional text* can be used to localise documents.

With conditional text, you apply a user-defined condition tag to part of the document, and then, at production time, choose which tags to show and which to hide. This is an excellent way of ensuring that a particular audience gets only the material they need to see without you having to delete the material that is not relevant to them, for text tagged is hidden, not deleted. It can be shown again simply by changing its show-hide setting.

Localisation: making user documentation appear as if it was written by a member of the intended audience for that audience

The basics of conditional text

For this simple introduction, we will continue the example we used in the previous issue of *Words*—producing one user guide for an Australian audience and another for an American audience—and doing so from one document. This document refers to the fuel used to power a car: *petrol* in Australia and *gasoline* in America.

You can create condition tags called, say AU and US, and apply these tags to the appropriate terms: *petrol* or *gasoline* respectively. To prepare the Australian version of the user guide, you show all the text that has the AU condition tag applied to it and hide all the text that has the US condition tag applied. To prepare the American version, you do the opposite.

The following steps describe how to do this.

Create the condition tags

1. Select **Special > Conditional Text > Manage Condition**. FrameMaker displays the **Manage Condition Tag** dialog box.



The **Comment**

condition tag that is listed is built into all new FrameMaker documents by default.

2. Click **Add**. FrameMaker displays the **Add Condition Tag** dialog box.

3. In the **Tag Name** field, enter AU.

4. From the **Color** list, select a colour.

5. Click **Set**. FrameMaker closes the **Add**



Condition Tag dialog box and adds the new condition tag to the **Manage Condition Tag** dialog box.

6. Repeat from step 2 to create a condition tag called US.

7. In the **Manage Condition Tag** dialog box, click **Done**.

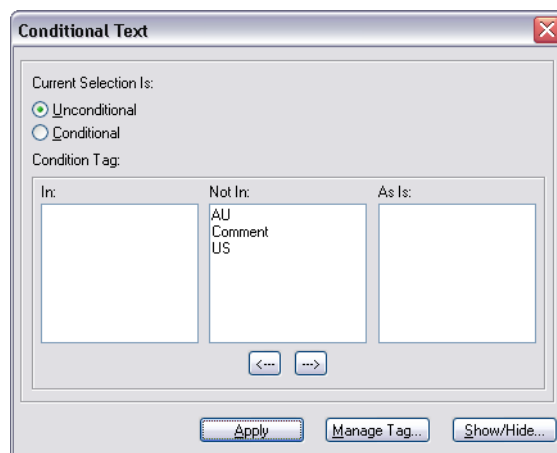
You can now apply the condition tags to your text.

Apply the condition tags

1. Select **Special > Conditional Text > Show Condition Indicator**. This is an optional step. It displays whatever text you apply a conditional tag to in the colour you assigned to that tag. This will help you identify conditional text.
2. Highlight some text that you want to apply a conditional tag to. In this example we highlight *Petrol*.

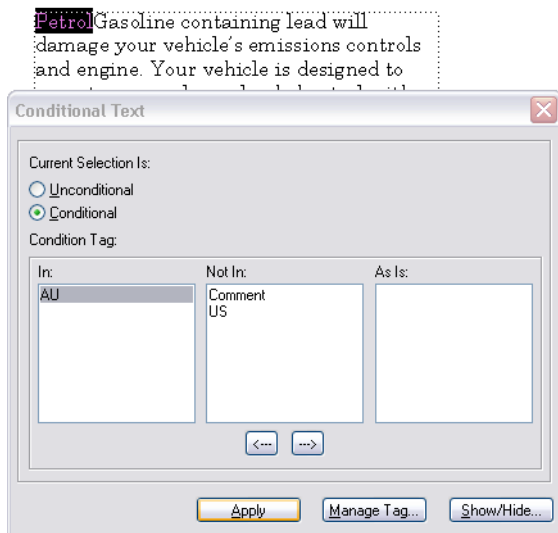
Petrol Gasoline containing lead will damage your vehicle's emissions controls and engine. Your vehicle is designed to operate on regular unleaded petrolgasoline with a research octane number (RON) of 91 or higher.¶

3. Select **Special > Conditional Text > Apply Conditional Text**. FrameMaker displays the **Conditional Text** dialog box.



4. Click the condition tag you want to apply to the selected text and then click <--. The tag is moved to the **In** list.





TIP: You can also show and hide graphics, tables and markers by applying condition tags to the corresponding anchors and markers.

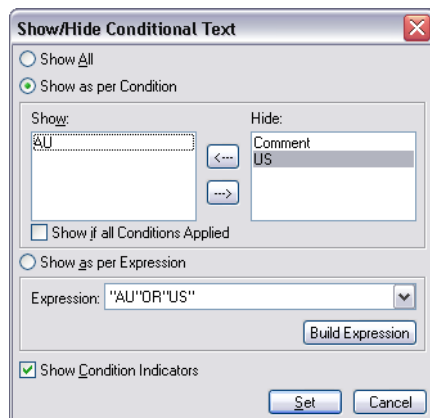
5. Make sure that the **Conditional** button is selected and then click **Apply**.
6. Apply the **US** condition tag to the term *gasoline* in the same way.

PetrolGasoline containing lead will damage your vehicle's emissions controls and engine. Your vehicle is designed to operate on regular unleaded petrolgasoline with a research octane number (RON) of 91 or higher.¶

When you have applied the appropriate condition tags throughout your document, you can produce a localised version of it.

Produce a localised document

1. If the **Conditional Text** dialog box is displayed, click **Show/Hide**. Alternatively, select **Special > Conditional Text > Show/Hide Conditional Text** from the main menu.
2. In the **Show/Hide Conditional Text** dialog box, select the condition tags and then click --> or <-- until the **Show** list contains only the **AU** tag.



3. Select **Show as per Condition** and then click **Set**. FrameMaker displays those parts of the document that:

- ◆ have the **AU** condition tag applied or
- ◆ do not have any condition tag applied

PetrolContaining lead will damage your vehicle's emissions controls and engine. Your vehicle is designed to operate on regular unleaded petrolwith a research octane number (RON) of 91 or higher.¶

4. Inspect the document, paying particular attention to the spaces between words. If necessary, change the applied conditions.
5. Display the **Show/Hide Conditional Text** dialog box, clear the **Show Condition Indicators** box and then click **Set**.

You can now generate the Australian document in the preferred medium. Repeat the procedure to generate the American document, this time selecting **US** as the conditional tag to show.

A more complex example

Suppose you want to translate the user guide into French, Italian, German and Spanish, and produce all these versions from one file set rather than creating a file set for each language.

One way to do this is to create a copy of each paragraph for each language, and mark each paragraph with a condition tag for that language.

The following graphic shows a segment of a document with **AU**, **French**, **Italian**, **German**, **Spanish** and **Approved** condition tags applied. We have included an **Approved** condition tag, as only translations that are marked as approved are to be published.

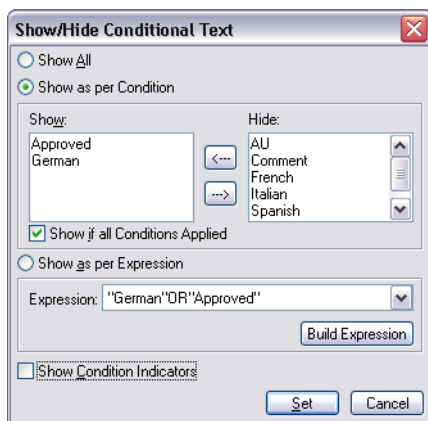
Petrol containing lead will damage your vehicle.
 Essence contenant du plomb vont endommager votre véhicule.
 Benzina contenente piombo per evitare di danneggiare il vostro veicolo.
 Benzin führen wird, die Schäden an Ihrem Fahrzeug.
 Gasolina que contiene plomo puede dañar el vehículo.

Note that text (including markers and anchors) can have more than one condition tag applied to it. In this example, the text for the German translation has also been tagged as approved, so the final version of the German is ready to be produced. To do so:

1. From the main menu, select **Special > Conditional Text > Show/Hide Conditional Text**.
2. In the **Show/Hide Conditional Text** dialog box, select the condition tags and then click --> or <-- until the **Show** list contains the **German** and **Approved** tags.



3. Select **Show as per Condition** and **Show if all Conditions Applied**, and clear the **Show Condition Indicators** box.



4. Click **Set**. FrameMaker displays just those parts of the document that:
 - ◆ have both the **German** and **Approved** condition tags applied or
 - ◆ do not have any condition tag applied

Berzin führen wird, die Schäden an Ihrem Fahrzeug.

Conditional expressions

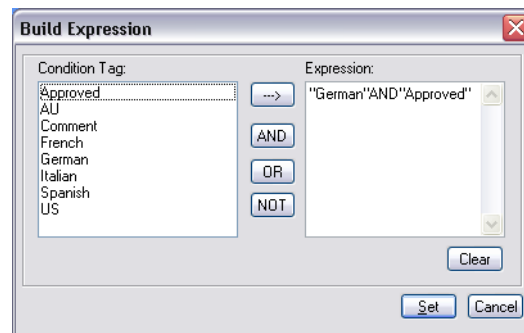
You can filter conditional text by joining together conditional tags with the Boolean logic operators AND, OR and NOT. The string of tags is called a *conditional expression*.

To obtain the same result as in the previous example but by using a conditional expression:

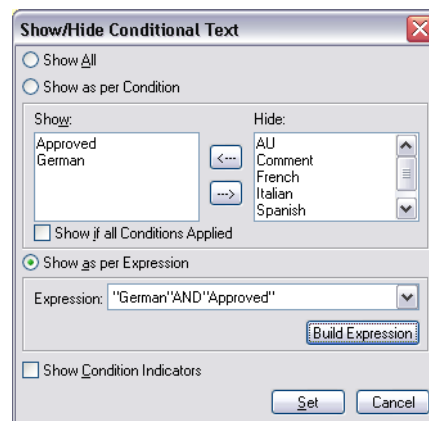
1. Select **Special > Conditional Text > Show/Hide Conditional Text**.
2. In the **Show/Hide Conditional Text** dialog box, click **Build Expression**.

The **Build Expression** dialog box appears.

3. If necessary, click **Clear** to remove a previous expression from the **Expression** pane.
4. In the **Condition Tag** pane, click **German** and then click **-->**.
5. Click **AND**.
6. In the **Condition Tag** pane, click **Approved** and then click **-->**.



7. Click **Set**. The **Build Expression** dialog box closes.
8. In the **Show/Hide Conditional Text** dialog box, make sure that **Show as per Expression** is selected and clear the **Show Condition Indicators** box.



9. Click **Set**. FrameMaker displays just those parts of the document that:
 - ◆ have both the **German** and **Approved** condition tags applied or
 - ◆ do not have any condition tag applied

Berzin führen wird, die Schäden an Ihrem Fahrzeug.

TIP: You can apply a condition tag to a book. To do so, select all files in the book and then apply the tag.

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.

FrameMaker courses

Abelard Consulting offers public and custom courses in Adobe FrameMaker.

In the next few months, the two-day course will be delivered in:

- Melbourne [May 18 & 19]
- Christchurch [June 8 & 9]
- Sydney [July 20 & 21]

Courses in structured authoring are also available.

For details of all Abelard courses, visit <http://www.courses.abelard.com.au> or call 1800 601 116.

Early-bird and concession discounts apply.



Entering uncommon characters

All languages and professions make use of symbols and signs. Some examples are the copyright symbol (©), punctuation marks (‘), mathematical operators (×) and currency symbols (€). Professional publishers use these standard symbols and signs, and many readers—possibly most—will be familiar with them and will find variations (or substitutions) distracting or baffling.

But substitutions are increasingly becoming common. For example, many writers are now using the asterisk as a multiplication sign (as in 3 * 5). The asterisk is widely used in computer programming to represent multiplication, but outside that domain, it is rarely used for that purpose. Other writers use a lower case x to represent multiplication (as in 3 x 5). The lower case x might look roughly like the true multiplication sign, but it is not the standard multiplication sign found in professional publications, namely, ×.

Likewise, many writers use a hyphen (-) for a minus sign, but the hyphen is not the standard minus sign, namely, −. The standard minus sign is longer than the hyphen and slightly higher above the baseline.

And one more example: inverted commas are increasingly used for the arcminute and arcsecond signs in expressions of latitude and longitude: 34° 20' 53" S. In professional publishing, primes are used for this purpose, not inverted commas: 34° 20' 53" S.

Why are such substitutions proliferating? It is probably because many writers do not know how to get, or can be bothered getting, the traditional symbol or sign from the standard computer keyboard.

It is true that some traditional symbols and signs could never be keyed from a PC or Macintosh keyboard, even indirectly. A typesetting machine, or metal type, was usually needed for texts that were especially typographically rich (although desktop font-creation software has been available for many years, ideal for creating the odd non-common character or two).

But with *Unicode* things are different. Unicode is a character specification standard that covers just about every character, sign and symbol—100 713 in total—in all the world's writing systems.

Every character, sign or symbol (hereafter called a *character* for simplicity) has a Unicode code (sometimes referred to as a *codepoint*). You might be familiar with the ANSI codes for some of the commonly used characters: 0150 for an en dash, 0176 for the degree sign, 0215 for the multiplication sign,

and so on. You can still enter these characters by specifying the ANSI codes: hold down the ALT key and enter the code on the numeric keypad.¹ However, the Unicode code for these characters (and for all characters that have an ANSI code) is different from its ANSI code. For example, the Unicode code for the en dash is 2013, for the degree sign 00B0, and for the multiplication sign 00D7.

The good news With current versions of the two main authoring tools on the market—Adobe FrameMaker and Microsoft Word—you can enter any character by specifying its Unicode code. The process is straightforward, and thus no longer is there an excuse for substituting the characters one finds in professional publishing with non-standard characters.

Unicode is a character specification standard that covers just about every character, sign and symbol—100 713 of them—in all the world's writing systems.

How do I find the code for a Unicode character?

Some often-used characters are listed in table 1 on page 12. Other methods of finding a character's Unicode code are given below.

Method 1: Internet search

Enter `+unicode +name` (where *name* is the name of the character) in an internet search engine. For example, entering `+unicode +infinity` into Google will give you the code for the infinity symbol (∞), namely, U + 221E. The part of the code you need to remember is 221E: the *U* prefix simply indicates that this is a Unicode code and that you need a special technique to enter this symbol as Unicode, explained on page 11.

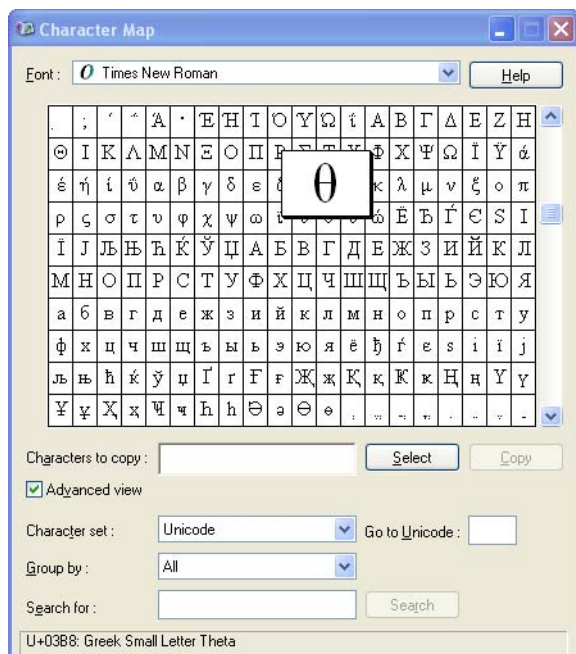
The drawback with this approach is that you need to know the name of the character. In many cases, the Unicode character name is the same as its common name—the multiplication sign is indeed called *multiplication sign*—but not in all cases. The thinnest space you can enter in illustration and layout applications is commonly known as a *hairline space*, but the name for it in Unicode is *hair space*.

Method 2: Windows Character Map

You can search for the character in the Windows character map. The illustration at the top of the next page shows the lower case Greek letter *theta* selected in the character map. The code for whatever

1. All keyboard instructions in this article are specific to a Microsoft Windows operating system. Similar functionality is likely to be available on systems running other desktop operating systems.

character is selected is shown at the bottom left of the window: 03B8 in this example.



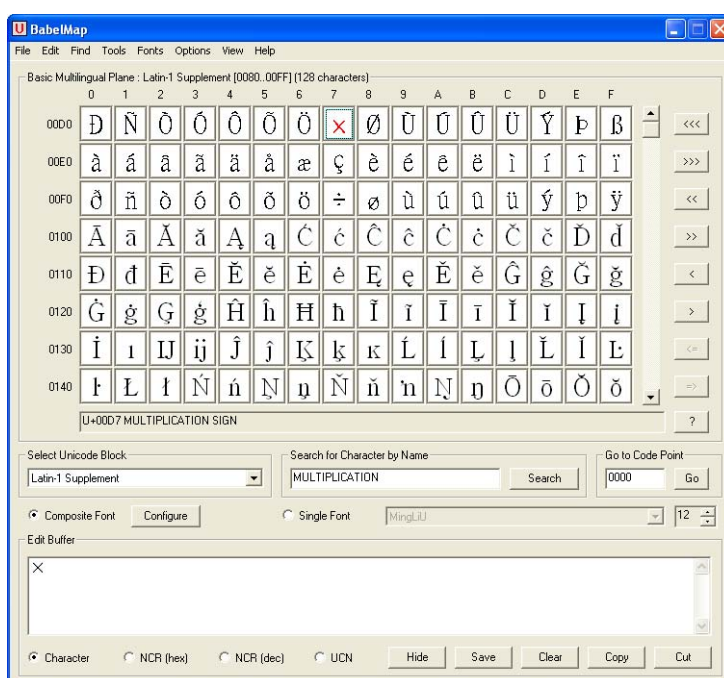
In Windows you display the character map by selecting **All Programs > Accessories > System Tools > Character Map**.

The drawback with this approach is that the Windows character map doesn't show all Unicode characters (showing less than 40 000 of the 100 713 characters). It might do for most of the characters you typically key, and then again it might not.

Method 3: Unicode map software

Free software is readily available that will show all Unicode characters and their codes. BabelMap is one such product (illustrated below).

You can scroll through the entire Unicode character set, or limit the characters displayed to particular blocks of characters. In the example at the right, part of the **Latin-1 Supplement** block is shown, and the multiplication sign, near the centre of the top row, has been selected. The code for the selected character (and the character's name) is shown below the displayed characters: in this example, the code is 00D7.



You can download BabelMap from <http://www.babelstone.co.uk/Software/BabelMap.html>.

Entering a Unicode character

Method 1: Copy and paste

You can select the character in the Windows character map or in BabelMap, copy it, and then paste it directly into your document. (When you select a character, it appears in the **Characters to copy** field (Windows character map) or the **Edit Buffer** field (BabelMap) from where it can be copied.)

If you are using Adobe FrameMaker, you may need to choose **Edit > Paste Special > Unicode Text** to ensure that the right character is pasted.

Method 2: Direct entry

Microsoft Word

Type the code for the character, hold down the ALT key, press the x key and then release the ALT key. For example, to enter the infinity character, enter 221E and then ALT + x.

Note that alphabetic characters in a Unicode code can be entered as upper case or lower case. Hence 221e and ALT + x also gives the infinity symbol.

Adobe FrameMaker

Hold down the ALT key, press + on the numeric keypad, enter numeric characters on the numeric keypad and alphabetic characters on the standard keypad, and then release the ALT key. (You may need to change a registry key for this to work. The key is HKEY_Current_User/Control Panel/Input Method and you need to set the EnableHexNumpad attribute to 1.)

Sample codes

Table 1 on page 12 shows the Unicode codes for a range of characters.

Note that not all fonts can display all Unicode characters. If an unexpected character appears (such as a question mark) select the character and apply another font. BabelMap can tell you which fonts installed on your system are capable of displaying the character.



Table 1: Unicode codes for some characters

Char.	Name	Code	Char.	Name	Code	Char.	Name	Code
–	minus	2212	...	ellipsis	2026	±	plus or minus	00B1
×	multiplication	00D7	†	dagger	2020	¶	pilcrow	00B6
°	degree	00B0	‡	double dagger	2021	¼	quarter	00BC
'	prime	2032	‰	per mille	2030	½	half	00BD
″	double prime	2033	√	square root	221A	¾	three quarters	00BE
©	copyright sign	00A9	∴	therefore	2234	é	lower case e with acute	00E9
®	registration	00AE	≠	not equal to	2260	è	lower case e with grave	00E8
™	trade mark	2122	≤	less than or equal to	2264	§	section	00A7
€	euro	20AC	≥	greater than or equal to	2265	²	squared	00B2
£	pound	00A3	≪	much less than	226A	³	cubed	00B3
¥	yen	00A5	≫	much greater than	226B	α	alpha	03B1
←	arrow left	2190	∞	proportional	221D	β	beta	03B2
↑	arrow up	2191	≈	approximately equal to	2245	γ	gamma	03B3
→	arrow right	2192	¬	not	00AC	σ	sigma	03C3
↓	arrow down	2193	‡	broken bar	00A6	μ	micro	00B5
⋮	vertical ellipsis	22EE	≠	not identical to	2262	≡	identical to	2261
Ω	ohm sign	2126	°C	degree Celcius	2103	∠	angle	2220

What about the web?

Not long into their careers, most HTML coders learn about, and appreciate, *named character entities*. These take the form `&entity_name;` and are particularly useful if you want to display a character that a web browser would otherwise interpret as an HTML character. For example, if you want to display angle brackets (< and >), you won't want to enter them directly from the keyboard, as the browser would interpret them as the beginning and end of an HTML tag. Instead, you enter these characters via their character entities: `<` and `>` respectively.

Other useful character entities are `&` and `"`. These produce the ampersand character (&) and a double quotation mark (") respectively.

Named character entities don't take you very far. There are just 252 of them, well short of the 100 713 Unicode characters. But all is not lost. Most modern web browsers will correctly interpret most of what are called *numeric character references*—and there is a numeric character reference for every character with a Unicode code.

A numeric character reference has the form:

```
&#xunicode_code;
```

The Unicode codes for web display are exactly the same codes you use to enter Unicode into Microsoft Word and Adobe FrameMaker documents (discussed earlier). For example, if you want a web browser to display the section symbol (§), you would enter the following in your HTML code:

```
&#x00A7;
```

Similarly, the proper minus sign (−) is entered as `−` and the euro symbol (€) is entered as `€`.

Not all browsers will render all Unicode characters, some particularly esoteric characters might require a change to the font-family specified, and you may need to declare a particular character encoding (for example, `<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />`) to get the characters you want. But numeric character references should work straight out of the box for most of the characters that most technical or science communicators generating web content will ever need.

Geoffrey Marnell

Thanks to Daryl Colquhoun for helpful comments on a draft of this article.

Client profile: Hydrix



Embedded Development Solutions

Hydrix is a leading product development company offering complete turnkey development and manufacture of products in the fields of:

- > Biomedical
- > Laboratory
- > Automotive
- > DSP and mathematical analysis
- > EFTPOS
- > Wireless
- > Industrial control

Hydrix has proven expertise in bringing product concepts to market quickly and cost effectively. Although Hydrix offers a complete product development service, many customers utilise a subset of Hydrix's offering to augment their own internal development team enabling them to reach the market quickly.

Companies with existing products are now turning to Hydrix to address issues causing excessive warranty claims. Design improvements can often be implemented quickly, making a significant, timely, improvement to their bottom line.

Please contact Mark Brydon on 0425 756 930 for a confidential, no-obligation, discussion about your technology needs.

www.hydrix.com

Unnecessary precision: single-plural composites

In everyday speech and writing, the singular includes the plural and the plural includes the singular. When we go to the airport with two bags to check in and are confronted with a row of check-in counters called *Bag Drop*, we don't ignore these counters and go looking for one that is called *Bags Drop*. The singular *Bag Drop* covers us regardless of how many bags we have. In other words, the singular includes the plural.

Similarly, if we have a single plastic water bottle to dispose of, we don't deliberately walk past a bin with the notice *Dispose of Recyclables Here*. The plural word *recyclables* doesn't exclude a single recyclable. In other words, the plural includes the singular.

Despite these joint implications, many writers feel that they need to cover both bases:

- ? Now replace the wheel(s).
- ? Now replace the wheel/s.
- ? Now replace the wheel or wheels.

If the material that preceded this instruction made it clear that only one wheel was to be removed, then clearly the singular alone is called for. But if the material indicated that *one or more* wheels could be removed—for example, if it included an earlier instruction like "Remove each wheel whose tyres need to be replaced"—then a plural instruction covers all bases. Just as a native English speaker won't ignore a *Dispose of Recyclables Here* bin if they only have one empty plastic bottle to dispose of, a mechanic is unlikely to ignore or be confused by "Now replace the wheels" if they had earlier removed only one wheel.

So there is no need for singular-plural composites. The plural alone covers both possibilities:

- ✓ Now replace the wheels.

Geoffrey Marnell

Style manual

for authors, editors and printers

The seventh edition of this authoritative guide to Australian style and usage will be published in late 2009 or early 2010.

The publisher will be holding focus groups with interested readers to uncover what information is needed in this edition.

If you want to be part of such a focus group, send an email to govstyle@finance.gov.au.

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

The Words team

Artwork: Christine Weaver

Copy-editor: Marcia Bascombe

Adviser: Mark Ward

Cartoonist: Matthew da Silva

Editor: Geoffrey Marnell

Contact: words@abelard.com.au

© Abelard Consulting Pty Ltd 2009

[unless otherwise noted]

Contributions welcome

Everyone is welcome to contribute to *Words*. You can send us a letter commenting on a previous article. Or send us an article for publication. All articles on language will be considered (and copyright is retained by the author).

Send contributions to: words@abelard.com.au

TRAINING COURSES

Technical, general & scientific writing, FrameMaker, Structured authoring

Abelard Consulting is offering 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]
- Punctuation: An Intensive Overview of Best Practice [4 hours]

Where?

Scheduled classes are held in Melbourne, Sydney, Canberra, Perth, Brisbane, Adelaide, Christchurch and Auckland. 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

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, documentation project manager 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

Miscellany

Word of the Year: *toxic debt*

The Macquarie Dictionary Word of the Year Committee has announced *toxic debt* as the 2008 Word of the Year. In making its decision, the committee—Dr Michael Spence (Vice-Chancellor of the University of Sydney), Professor Stephen Garton (Dean of the Faculty of Arts at the University of Sydney), Les Murray (renowned Australian poet) and Susan Butler (publisher of the Macquarie Dictionary)—considered the attention-grabbing events of 2008. And the event that overshadowed all others was the global financial crisis (GFC), the prime cause of which was unsustainable debt. The committee noted that “*toxic debt* ... as a lexical creation, [has] a visceral impact. It needed no explanation but said it all”.

Global financial crisis (and *GFC*) has also been added to the Macquarie Dictionary, along with *flashpacker* (a backpacker who travels in relative luxury), *bromance* (a non-sexual but intense friendship between two males), *textaholic* (someone who sends an excessive number of text messages), *guerilla gardener* (a person who plants gardens in areas controlled by councils or other organisations but neglected by them) and *lawfare* (the use of international law by a country to attack or criticise another country, especially a superior military power, on moral grounds).

The updated Macquarie Dictionary is available online at www.macquariedictionary.com.au. (See page 2 for details of a free three-month subscription.)

Technical writing courses

Abelard Consulting offers public and custom courses in technical writing, business writing, scientific writing and Adobe FrameMaker (structured and unstructured). We also offer half-day intensive courses on punctuation.

In the next few months, the two-day **technical writing course** will be delivered in:

- Adelaide [May 18 & 19]
- Brisbane [May 28 & 29]
- Melbourne [June 15 & 16]
- Sydney [June 18 & 19]
- Canberra [June 22 & 23]

For details of all Abelard courses, visit <http://www.courses.abelard.com.au> or call 1800 601 116.

Early-bird and concession discounts apply.



AODC 2009

The 12th annual Australasian Online Documentation and Content conference, AODC 2009, will be held at the Savoy Vibe Hotel in Melbourne from Wednesday 20th to Friday 22nd May. The conference targets technical writers, help developers, web authors and documentation developers.

Speakers include Penny Bradley, Emily Cotlier, Matthew Ellison, Gerry Gaffney, Dave Gash, David Lowe, Sarah Maddox, Gareth Oakes, Deborah Pickett, Allyn Radford and Tony Self.

To learn more about the conference, visit www.aodc.com.au

Most frequent words in written English

Ever wondered what are the most frequently written words in the English language? Here are the top ten.

- the
- of
- and
- to
- a
- in
- that
- it
- is
- was

Further, *the* appears twice as often as *of* and *of* twice as often as *and*. Thus *the* is the leader by a long way.

2009 accreditation exam for editors

The 2009 accreditation exam for editors is tentatively scheduled for Saturday 12 September 2009. The exam is run by Institute of Professional Editors (IPEd).

IPEd is the national professional society for editors. Its purpose is to advance the profession of editing “by planning and implementing national initiatives—an accreditation scheme, promotion and communications campaigns, training and mentoring—and other activities to support Australian societies of editors and their members, and editors in general” (from the Institute’s website).

Those who are accredited by IPEd are entitled to the postnominal “AE”.

To learn more about IPEd and to register an interest in sitting the 2009 exam, visit www.iped-editors.org.

