



# You Mean I Have to Write About it, Too?

*Tamar E. Granor*

*Tomorrow's Solutions, LLC*

*Voice: 215-635-1958*

*Email: [tamar@tomorrowssolutionsllc.com](mailto:tamar@tomorrowssolutionsllc.com)*

*Web: [www.tomorrowssolutionsllc.com](http://www.tomorrowssolutionsllc.com)*

*Many software developers would prefer never to have to write in a natural language; they're much happier with programming languages. But the ability to communicate in writing is actually an essential skill for successful software development. Developers need to be able to write specifications and developer documentation, as well as to clearly ask questions of their customers and to answer customer questions. In some cases, developers are responsible for end-user documentation, too.*

*In this session, we'll see how to produce documents that say what you need to say in an accessible way, whether you're writing a user manual or an email. We'll look at all stages of the writing process, from organization to writing to editing to proofing. This session will be run as a workshop. Come prepared to write and edit. Please bring something on which to write and something to write with.*

## Introduction

I never planned to be a writer. In fact, by the time I started to think of myself as a writer (as opposed to a software developer who did a little writing on the side), I'd had seven books and hundreds of articles published.

Because I didn't plan to become a writer, I didn't take any courses specifically aimed at writers, and the courses I did have that emphasized writing definitely weren't focused on writing about technology. Most of the writing in those courses was analysis of one sort or another, whether literary or historical or something else.

My first real experience in writing in engineering and science came in graduate school, where I had to write a master's thesis and, later, a doctoral dissertation as well as a couple of related papers. After grad school, I moved into the software development world and started writing magazine articles and, eventually, books.

Looking back at those early writings, I can see that they're pretty solid. But it was five years editing FoxPro Advisor that really helped me understand and learn to articulate what works and what doesn't in writing about software and technology.

The goal of this paper is to share the key points with you, so you can write more effectively and with less dread and procrastination.

## Why do software developers have to write?

At first glance, it might seem that working as a software developer shouldn't require good writing skills, at least not in English or other natural languages. But most software developers are required to write coherently in a variety of ways. Let's start by looking at the kinds of writing that are part of a developer's job and who the various kinds of writing are aimed at.

### Email

For many developers, the most frequent kind of writing is email. We send email to report progress, to ask questions, to answer questions, to explain how things work, and to propose work. We email clients and prospective clients, co-workers, managers, colleagues, and users of the software we write.

### Comments

The other kind of writing we do every day is putting comments in code. While most developers probably don't think of commenting as writing, the ability to write clear,

concise comments makes one a better developer. Comments are written for our future selves and for those who will succeed us in maintaining and using the code we write.

### **Design documents and specifications**

Many developers have to write design documents and specifications for new applications or for enhancements to existing applications. These documents can be aimed at decision-makers and funders, as well as at other developers who will actually implement the systems. They need to describe the system clearly and completely.

### **Developer documentation**

It's not unusual for developers to be called on to document the code they create. Like comments, these documents are aimed at both our future selves and those who take over applications we write. For some kinds of software, such as APIs and libraries, they're aimed at other developers using or building on our work. Typically, this kind of documentation addresses things at a higher level than comments in the code, but it shares the need to be clear and concise. It also needs to be complete.

### **User documentation**

Some developers have to create documentation for the end-users of the applications they write. This can be in the form of complete manuals or of knowledge base articles addressing particular aspects of an application.

### **Technical support answers**

Some developers also serve as tech support for the code they write. While some support may be performed face-to-face or by phone, much tech support happens in writing, whether via email, chat, or in online forums.

### **Technical support questions**

Developers often need to ask for help, and there are lots of good places to find it online. But to get help, you need to be able to ask for it in a way that explains the problem clearly.

### **Books, articles, white papers, and blog posts**

Some developers (like me) write for a broader audience, hoping to spread knowledge about software development generally, particular languages and environments, or specific problems in software development. Most often, this writing is for other developers, though it's sometimes aimed at end-users or decision-makers.

### **Getting started**

The first step for any writing is to organize, whether that's spending a few moments in thought before cranking out an email or creating a detailed outline for a book. Consider the audience and the purpose of the piece of writing. Once you have those, identify your key point and create some kind of outline, even if it's only in your head. Finally, once you start

writing, you can help readers see the organization of the material by using an introduction-content-summary format.

## **Audience**

As the description of various kinds of writing in the previous section indicates, developers write for a variety of audiences from decision-makers to end-users to other software developers. Readers may be employers, employees, funders, co-workers, customers, or colleagues.

Audience matters because it indicates what you can assume the reader knows already and how formal or informal the tone should be.

## **Purpose**

Figure out why you're writing. Are you looking for information? Trying to explain something? Making a proposal? Trying to persuade? Sharing information?

Like the audience, the purpose of a document affects the style and formality of the writing. An email asking a colleague for information may be informal and breezy, while a design document is likely to be formal and serious.

## **Key point**

You've probably been to a talk where the speaker said something like "If you remember only one thing from this presentation, remember ..." That's the key point. Before you start writing, figure out what your key point is so you can aim your writing at making that point.

For some documents, especially things like manuals, there may be more than one key point. But even for a book, you should be able to tell someone why you're writing it in a sentence or two.

When there is more than one key point, you'll need a way to ensure that each point gets its due. Typically, a book is divided in chapters, with each addressing a key point. In an email, bullets or numbering let you highlight each point.

## **Outline**

As with the other aspects of organizing, how you outline is a function of what you're writing. For an email, an "outline" may be simply a mental list of the items you want to mention or a few words jotted down in the email before you actually start crafting sentences.

For example, if I need to send an email to a client about three issues we need to address, I might start with something like this:

New metadata

PDF issues

## Login problems

As I write the email, I'd flesh each of those points out, but having the list ensures that I don't forget any of them.

For a book or a manual, on the other hand, you're likely to start with a document listing the chapters and at least summarizing what each will cover. **Figure 1** shows part of the outline for my book "Taming Visual FoxPro's SQL." (As you can see, the final title hadn't been chosen yet.)

- + **SQL in VFP**
  - By Tamar E. Granor
- + **Chapter 1: Introduction**
  - *What is SQL?*
  - *Why use SQL?*
  - *Databases and Normalization*
- + **Chapter 2: Using VFP's SQL Data Definition Language**
  - *Creating databases*
  - *Creating tables and cursors*
  - *Changing table structures*
  - *Managing databases*
- + **Chapter 3: Basic data manipulation |**
  - *Adding data*
  - *Removing data*
  - *Changing data*
- + **Chapter 4: Retrieving data**
  - + *The components of a query*
    - **Simple queries**
    - **Destinations**
    - **Filtering data**
    - **Joining tables**
    - **Ordering results**
    - **Consolidating results**

**Figure 1.** For long documents, it's worth creating an actual written outline that shows the overall organization before you start writing.

For documents longer than an email, but shorter than a book, you can still develop a complete outline, but you may choose something less formal. For example, for conference papers and articles, I often start by jotting down things I want to cover as they occur to me and gradually organize them into a logical progression.

It's also important to realize that, in most cases, an outline isn't a commitment. It can and probably will change as you write. You'll think of things to add; you'll realize that there are better ways to organize the material; you'll realize that something you wanted to say doesn't really fit.

## Intro-Content-Summary form

I don't know who said it first, but there's an old rule about presentations that applies to writing as well: "tell them what you're going to tell them, tell them, and then tell them what you told them." The idea is that you start with a quick overview of what you're going to say, then you lay out the whole thing, and then you summarize what you said.

This is the basis for the "five-paragraph essay" technique that's widely used to teach writing, especially persuasive writing. The first paragraph provides the main thesis, and then in one sentence apiece, summarizes three arguments for it. Then, the next three paragraphs each lay out one of the arguments in detail. Finally, the last paragraph briefly reiterates the three arguments and draws the desired conclusion. (There are lots of issues with five-paragraph essays, but they're not a bad tool for introducing essay writing.)

More broadly, almost any piece of writing other than a list can be structured in this intro-content-summary form. In fact, for longer pieces of writing (such as non-fiction books, long-form journalism, or white papers like this one, the format is appropriate not just for the overall document, but for sections within the document. So a book may have an introductory chapter, then a series of chapters each exploring some aspect of the topic, and then a final chapter drawing some conclusions. Each chapter may have an introductory section, followed by a series of sections each covering some aspect of that chapter's focus, and close with a summary section. Even within one section of a chapter, you may find a few paragraphs to introduce the section's topic, then a number of paragraphs (or even subsections) to cover the topic, and then a few paragraphs to sum it up.

For shorter documents such as emails (or shorter pieces of longer documents), you may or may not want the summary part. For example, you can see in this paper that pretty much every section has the introduction and the content, but not all have a summary.

In emails that cover multiple topics, I typically open with a sentence or two to indicate that there's more than one subject, but depending how many there are and how long the email is, I may or may not summarize. When an email includes a number of action items, I will sometimes summarize those at the end, using a numbered or bulleted list.

For example, one long email that went to Southwest Fox/Southwest Xbase++ 2015 speakers started like this:

We have a number of items for speakers. There's a [TL;DR summary](#) at the end of this message.

And had this at the bottom:

[Here's the summary version:](#)

- 1) Check the [session schedule](#).
- 2) Book your flights and let us know.
- 3) We're paying for an extra room night.

- 4) Rehearse before the conference.
- 5) Speaker dinner 10/17. Tell [Tamar](#): Coming? Spouse/SO? Will drive?
- 6) Bring your own HDMI to VGA adapter.
- 7) Materials deadline is 9/21.

You can also see that this was informal writing. For a client or funder, “a TL;DR summary” (“TL;DR” stands for “too long, didn’t read”) would have been replaced with something like “a list of action items” and “Here’s the summary version” also would probably have something like “Here are the action items:.”

## Writing well

Once you know what you want to write and for whom, and you’ve figured out some organization for it, the next step is to actually write. A number of things determine whether what you write reads well.

While much of what makes writing good is universal, some of the things you may have learned in school don’t really apply to the kind of writing developers do at work.

This section addresses both good practices for writing in general and those items that are different about technical writing.

## Word choices

The words you choose in a document are important. Those choices set a tone for the document, and determine whether the document delivers a clear message or obfuscates its points.

## Technical terminology

So-called “words of art” are one big difference between creative writing and technical writing. (I’m using *technical writing* here in a broad sense to mean any kind of writing about a particular field. So this idea applies not just to technology, but to psychology, economics, physics, and even ancient history.) Teachers tend to emphasize the idea of using a wide vocabulary and finding many different ways to express the same idea. The idea is to make the result more interesting.

But in technical writing, it’s important instead to use the same terms for the same things every time. For example, if you refer to an option button in one paragraph and a radio button in the next, readers will try to figure out the difference between the two.

Choose the term you’ll use for each thing and use it every time. If there are multiple terms in common use for a thing or idea, clarify early which one you’ll use, making clear that that term implies all the variants.

If you’re dealing with a lot of terminology, create a document where you list the choices you’ve made. In fact, in team situations, having such a document can help avoid time wasted arguing over things where team members actually agree.

### General vocabulary

Aim to use the simplest, clearest words you can. On the whole, short words are better than longer synonyms. A document that's full of long words when shorter words are clearer reads like an attempt to show off.

Perhaps the most common example is saying "utilize" rather than "use." For example, in the previous section, I wrote "Choose the term you'll use for each thing and use it all the time." Note how stilted it sounds with "utilize," instead: "Choose the term you'll utilize for each thing and utilize it all the time."

The section at the end of this paper, "Appendix: Writing Tips" includes a number of specific recommendations on word choices.

### Active voice vs. passive voice

Most of us were taught in school to avoid the passive voice and always use the active voice. That is, we should write "Tim ate the banana" rather than "The banana was eaten by Tim." The passive voice is often seen as wishy-washy or vague.

As with many such rules, what's taught in school is often incomplete (or we forget the exceptions). Yes, the passive voice can be wishy-washy or vague, but there are places where it's the appropriate choice, and technical writing is more likely than other kinds of writing to include such cases.

Specifically, passive voice is the right choice when the actor is unknown or irrelevant. So, for example, this sentence makes a lot more sense than trying to convert to the active voice:

When the OK button is clicked, the query is sent to the server.

In this case, who's clicking the button? Who sends the query? Do we care?

On the other hand, avoid the passive tense when the actor is known and relevant. For example:

The `ACopy()` function copies data from one array to another.

is easier to understand than:

Data is copied from one array to another by the `ACopy()` function.

That example, though, shows that the choice sometimes isn't about the actor, but about the focus of the sentence. The example is drawn (slightly modified) from my book, *Hacker's Guide to Visual FoxPro*. It's the opening sentence of the section that discusses `ACopy()`, so clearly the function is the focus. In a different context, where the discussion is, say, about some piece of code that needs to copy data between arrays, the second version may be appropriate because the focus of the sentence is copying data and `ACopy()` is simply the implementation.

The passive voice can also be used as a way of hedging or avoiding blame (or credit). The classic example of this use is “Mistakes were made.” By whom? Why? Even if the sentence indicates who made the mistakes (“Mistakes were made by each member of the team.”), the passive formulation attempts to soften the blame. The active formulation (“Each member of the team made mistakes.”) takes responsibility. The purpose and overall tone of the document determine which is appropriate in a given setting.

## Tense

Be consistent about tense in your writing. If you’re writing about something that happened already, stick to the past tense.

If you’re writing about what’s going on now or how something works, stick to the present tense. It’s easy in writing about software to slip into the future tense:

When the Print button is clicked, the Print dialog will appear.

Present tense is better:

When the Print button is clicked, the Print dialog appears.

The only place for future tense in writing about software is when you’re writing about future plans:

After we fix this bug, we’ll determine priorities for the next release.

## Person

First person is “I” or “we.” Second person is “you.” Third person is anyone else.

Many of us were taught not to use the first person in technical writing. In fact, this is the reason the passive voice is considered appropriate for technical writing. Traditionally, a lab report or journal paper was expected to read:

Ten subjects with multiple sclerosis were identified.

rather than:

We identified ten subjects with multiple sclerosis.

These rules are more flexible today in a lot of fields, and in general, Computer Science seems less wedded than many other fields to these rules.

In the kind of writing software developers do, a combination of first person and second person is often appropriate.

In emails with clients or among team members, you’re likely to use first and second person to describe what you’ve done or plan to do and what you’re asking the recipient to do or to tell you.

Manuals and help files should avoid the first person, but may use second person to address the reader. For example, the Visual FoxPro Help file includes a walkthrough titled “Creating Applications with Visual FoxPro.” It opens with this paragraph:

You can use Visual FoxPro to create simple programs or more complex enterprise level applications, servers, and XML Web services. You can use tools to create programs or manually write them. When you compile Visual FoxPro programs, you can compile them as application (.app) files for running in a Visual FoxPro environment or as executable (.exe) or dynamic-link library (.dll) files that you can call from anywhere in a Microsoft Windows environment.

For magazine articles, white papers, books, and similar kinds of writing, using first and/or second person (as I’m doing in this paper) rather than third person makes the writing more informal. Decide whether that’s appropriate on a case-by-case basis.

Of course, many publishers (of both books and magazines) have a style guide that specifies what person to use, as well as other rules. (See “Style guides and style sheets” later in this paper.) When writing for them, use their specified style unless it results in language so clumsy as to obscure your meaning.

For design documents and formal proposals, third person is likely to be the most appropriate. Rather than writing about what you’re doing or what the reader will do, just describe the plan or the design or whatever, like this:

This product is a Stanifran for the upcoming Acme 2734. It will incorporate nearly all of the look and feel of the recently released Acme 1734 Stanifran while including new features specific to the new model.

## Gender

English is not a particularly gendered language. Most nouns are not masculine or feminine, and verb forms are the same whether they apply to a man or a woman. But there are still some gender-related traps, and standards in this area are evolving.

One area where English is gendered is third person pronouns. We have “he,” “him,” and “his” vs. “she,” “her,” and “hers.” There’s no standard genderless pronoun to refer to a person; it’s considered rude to use “it” for people. That causes a problem when you don’t know the gender of the person to whom you’re referring. One common solution is to say “he or she” as in:

When the next programmer takes over, he or she will have to revisit this code.

But many people find that construction clumsy and prefer to use “they”:

When the next programmer takes over, they will have to revisit this code.

While the use of “they” for this purpose actually dates back at least to Chaucer, until very recently, I would have automatically changed the second version to the first. But one

evolving standard on this front addresses how people themselves prefer to be referred to. An individual may use “they” (or some other non-standard pronoun) rather than “he” or “she.” When referring to a particular person, you should always use that person’s preferred pronouns, if you know them.

Gender is also an issue with respect to the assumptions one makes in writing. Not every software developer is male, nor is every secretary female. When the context doesn’t specifically call for a particular gender, technical writing should be gender-neutral. When you’re writing about unknown users of your software or unknown decision-makers or unknown software developers, don’t make any assumptions about whether they’re male or female. If avoiding gender entirely makes the writing too clumsy, consider switching between male and female choices within the document (though not within a sentence or paragraph, of course), or consciously making the choice that’s contrary to your first instinct.

Along the same lines, separate job names for men and women who do the same thing are on their way out. A person who acts is an actor and a person who waits tables is a waiter, whether male or female.

### **Remove clutter**

It’s easy to put a lot of extra words and phrases into your writing. It’s especially easy when you feel unsure of yourself.

Clutter can be a phrase like “It seems to me that.” The key to removing such clutter is noticing that the sentence says the same thing without that phrase.

A related form of clutter is having multiple words or phrases that express the same idea in a sentence. For example, this sentence appeared in the first draft of a publicity email for Southwest Fox 2007:

We are extremely excited by this as it offers everyone a superb opportunity for everyone to network, check out the most community vendors all in one place, and attend some fantastic sessions in between.

In particular, note the part that says “offers everyone a superb opportunity for everyone to network.” That’s clumsy and repetitive. The version we actually sent out changed it to “offers everyone a superb opportunity for us all to network,” eliminating the repetition of the word “everyone.” Here’s the version of that sentence that actually was sent to registered attendees:

We are extremely excited by this as it offers everyone a superb opportunity for us all to network and check out our community’s vendors in one place, as well as attend some fantastic sessions.

In fact, I think we can do better by eliminating the first use of “everyone” as well, since “us all” means the same thing. If I were doing it today, it would read:

We are extremely excited by this as it offers a superb opportunity for us all to network and check out our community's vendors, as well as attend some fantastic sessions.

Another form of clutter is using multiple words where one will do, such as “due to the fact that,” which is just a long way of saying “because.” <http://tinyurl.com/hv2aex9> has a great list of these phrases.

<http://tinyurl.com/zgzgi6m> offers a whole group of suggestions that, to some extent, boil down to removing clutter.

## Editing

Editing is not optional. Every document needs to be edited. How much editing it needs depends on the document. An email to a colleague likely needs just a quick read-through, while a user manual or a book needs multiple stages of editing.

The term “editing” covers multiple processes. For any kind of serious writing, they include revising, copy editing, and proofreading. For technical writing, you may also want a technical review step in between revising and copy editing.

## Revising

Revising means going back through your document (literally, with “fresh eyes”) and seeing where it needs work. When possible, it’s best to finish writing and then wait for days or even weeks before beginning revisions. (In his book “On Writing,” Stephen King says he puts a finished book in a drawer for about six weeks before he begins revisions.) The benefit of setting your work aside for a while is that doing so makes it more likely to feel new to you.

Of course, with technical writing, you rarely have the option of setting it aside for weeks. But for anything longer than an email (and, of course, for any angry email), give yourself a few hours or a few days before starting the revision process.

Revising is about the big picture. You’re asking whether the document is well-structured, whether chapters or sections should be moved around, and whether anything should be removed. While you’re likely to change some words and fix some mistakes during the process, your focus at this point should be on whether the document goes from point A to point B (or from point A to point Z, though points B, C, etc.) in a logical way.

One of the hardest things to do in revision is removing sentences, paragraphs or even larger parts of the document. After you’ve struggled to get the words onto paper, throwing them out is a challenge. But far more often than not, cutting makes the document stronger. Just as you should avoid word clutter when writing, in revision, look for sentences, paragraphs, sections and chapters that don’t serve the purpose of your document. (Because we’re talking about technical writing here, you may find that those pieces are a place to start a new document.)

## Technical Review

For some kinds of writing, you may be fortunate enough to have someone else to review technical content. Ideally, that's someone who's familiar with the topic you're writing about, who can catch mistakes or misunderstandings.

For a book-length work, you should insist on having someone else do this. Every book I've written has had a designated technical editor whose job was to read the manuscript for accuracy and test any related code. In addition to catching actual mistakes, the technical editor can point out what's missing.

For books I've written with others, the first round of technical review has always come from another author with the technical editor reviewing the document after I've incorporated the other author's comments.

For articles and white papers, you're unlikely to have an official technical editor. Over the years, when I've had an article I thought needed technical review, I've asked colleagues to do it, with the understanding that I'd return the favor.

If you don't have a technical editor or anyone you can ask, at least do a separate pass of your document with particular attention to technical issues.

## Copy Editing

Once the organization and content are right, it's time to focus on the writing. This is the stage where you fix the spelling and typos, get rid of [misplaced modifiers](#), make sure your sentences have subjects and predicates, and so forth. For longer works, this is also the stage where overall stylistic rules are applied to ensure that the document is consistent.

For a book-length work, you're likely to have someone else doing this. Ideally, it's someone with training in copy editing, who will create a style sheet (or apply an existing style sheet). For shorter documents and even some book-length ones, this is another task you'll have to do yourself.

### Style guides and style sheets

A *style guide* is a document that provides broad advice or writing style. Some are very well-known such as [The Chicago Manual of Style](#) and the [AP Stylebook](#). Because I started in the academic world, the one on my bookshelf is Turabian's [A Manual for Writers](#). A style guide covers a wide range of topics including punctuation, spelling, capitalization, citation styles and more.

Individual publishers may have a style guide as well. Sometimes, such a guide references a standard guide and then spells out the publisher's exceptions.

My book publisher, Hentzenwerke, had a style guide of about 20 pages. About half of it was a glossary, indicating the appropriate spelling of words like "uppercase" (no hyphen) and "object-oriented" (use a hyphen), and the appropriate capitalization of words like "Registry" (capitalize) or "intranet" (don't capitalize). The guide also explained how to use

the Hentzenwerke Word template to format the book. The goal was for all Hentzenwerke books to have a recognizable style.

Magazine and journal publishers and conferences also sometimes have a style guide, though more often, they simply give you a Word template and, if you're lucky, a guide to using it. For example, for Geek Gatherings conferences, we provide speakers with a Word template and a sample document that demonstrates the use of the various styles in the template. As with Hentzenwerke, the goal is that all the white papers for the conference have a common look.

A *style sheet*, on the other hand, addresses a particular document. It's created by the copy editor in the copy editing process. It tracks many of the same things as a style guide, but for the particular document being edited. It allows the copy editor to notice when you hyphenated a word in one place, but used it without a hyphen in another, and when you used the [Oxford comma](#) (a comma after the next to last item in a list) in one place, but not another. A professional copy editor creates a style sheet for each document. When you're editing your own work of more than a few pages, you may want to create a style sheet, as well.

### **How to copy edit**

Just as testing your own code thoroughly is hard because you know what it's supposed to do, copy editing your own work is hard because you know what it's supposed to say. But most of the time, you're the one who has to do so.

When I'm copy editing, whether it's my own work or someone else's, I'm looking for things that make me stumble as a reader. I'm generally a very good reader and can proceed quickly through most documents. Anything that slows me down likely needs to be reworked.

One way to see the problems better is to hear them. Read the document out loud, and you're likely to find things that need to be fixed. Another option is to print the document and edit on paper. The goal in both cases is to change the document enough that you're not just seeing what you think you wrote.

As for specific issues to look for, obviously, start with the items discussed in the "Writing well" section earlier in this paper.

These sites offer some additional tips:

<http://thewritelife.com/edit-your-copy/>

<http://blog.oxforddictionaries.com/2015/06/copyediting-your-own-work/>

### **Proofreading**

The final step in the editing process is proofreading. The term comes from the days before word processors and PDFs. Before sending a book or magazine to be mass-produced, sample pages called *proofs*, were printed for a final review. (I remember sitting on the sofa

of my office with a red pencil in hand marking up the paper proofs for the *Hacker's Guide to Visual FoxPro 3.0*.)

Because resetting type was expensive, the goal here was to find only serious problems. This was too late for revising or editing. But spelling and grammar mistakes could be fixed.

Today, of course, correcting errors is cheap. But the idea is still the same. Once you reach the proofreading stage, the basic document is set. Now you just want to make sure no errors have been introduced and no mistakes missed in the editing process.

Unlike technical editing or copy editing, this is a task for the author, not for a third party.

### **Which editing stages do you need?**

All that editing sounds like a lot of work, and it is. But while you need all four stages, how you actually perform them varies with the length of your document.

For an email or a web forum post, it's usually enough to finish writing and then make one or two passes to confirm that the message says what you meant it to say, says it clearly, and doesn't have obvious mistakes or omissions.

You may want to save an important or long email and review it after taking a break. If an email is sensitive in some way, it's wise to have someone else read it over before you send it to ensure that it can't be read as having a very different meaning than you intend. While we often write with an intended inflection, the reader can't hear that inflection. That's why emoticons were invented, but they're inappropriate in any but the most informal writing.

For documents from a few to a few dozen pages, plan on at least two editing passes. The first is to look at overall structure and content; the second, copy editing and proofreading. That doesn't mean you shouldn't fix errors you see during the first pass, just that your focus should be on the big picture.

For longer documents, it's worth taking the time to do it right and make at least three editing passes: structure, content and copy edit/proofreading.

Finally, no matter how long a document it is, if it's critical in some way, do more editing. You want no ambiguity in the instructions for wiping a hard disk or counting votes.

### **Write, write, write**

Like most skills, both writing and editing get better the more you do them. Today, we have more opportunities to practice writing than ever. Answer questions on a web forum, write a blog post, or update a wiki.

Reading is also a good way to improve your writing. As you read, try to notice when reading is easy and when it's not. (It's easier to notice when it's hard than when it's easy.)

When reading is easy, look at the words, sentences and paragraphs. Try to see how they guide you and keep you moving along.

When something is hard to read, try to figure out why. Watch for phrasing that makes you stumble. See if you can identify the problems and, if you're ambitious, try editing (at least in your head) to make it better.

The more you read, the more you write, the more you edit, the better you'll get.

## Resources

This list is not meant to be exhaustive. These are some of the books about writing that live on my bookshelves.

Zinsser, *On Writing Well*, Harper Perennial. This is a guide to non-fiction writing. My son had it as a text in a college writing seminar at the time I was teaching one elsewhere and strongly recommended it to me; he was right.

King, Stephen, *On Writing: A memoir of the craft*, Scribner. Part memoir, part guide to aspiring writers, this is the only King book I've read. I recommend it to anyone who wants to write.

Strunk, Jr., William & White, E. B., *The Elements of Style, Fourth Edition*, Allyn and Bacon. This slim volume is full of solid, practical advice on everything from punctuation to style, and includes a list of words and phrases commonly misused. (Incidentally, E. B. White was also the author of *Charlotte's Web* and *Stuart Little*.)

Turabian, Kate L., *A Manual for Writers of Term Papers, Theses, and Dissertations*, University of Chicago Press. My go-to for questions of format and style, especially regarding punctuation and citation.

Maimon, Elaine P. and Peritz, Janice H., *A Writer's Resource: A Handbook for Writing and Resource*, McGraw-Hill Education. This guide to writing aimed primarily at college students covers everything from types of papers to citation styles to editing, with lots of examples. I read it cover to cover when I was teaching a writing seminar; it taught me the correct terms for things I already knew how to do, and much more.

Alley, Michael, *The Craft of Scientific Writing, Third Edition*, Springer: This book addresses many of the topics in this paper, but at greater depth.

Alley, Michael, *The Craft of Editing*, Springer. This book is aimed at those who edit the works of others, but it has plenty to offer those editing their own work, too.

## Appendix: Writing Tips

This appendix lists a number of issues I've encountered in years of writing and editing. It is, of course, not exhaustive. Some of the examples below show incorrect usage. When that's the case, "(WRONG)" appears at the right margin.

## Word Choice

The possessive of "it" is "its." Think of it like "his" or "hers," neither of which have apostrophes.

-----

Use "for example" rather than "as an example" to introduce a normal example. Generally, "as an example" should be used with a word like "consider":

As an example of fans influencing TV programmers, consider Star Trek.

Non-human characters in Star Trek often have distinguishing physical characteristics. For example, the Vulcans have pointed ears.

-----

Use "whether" rather than "if" to indicate a two-pronged possibility:

The checkbox determines whether the message is sent.

-----

Avoid the word "utilize." There's no situation in which "utilize" is a better choice than "use."

-----

"Infer" and "imply" are not synonyms. The speaker or writer implies; the listener or reader infers.

-----

Don't make something conditional that isn't. People often say something like:

For those who don't know me, I'm John Smith. (WRONG)

Well, who he is for those who do know him? This happens because people take a shortcut. What that speaker was trying to say is:

For those who don't know me, let me introduce myself. I'm John Smith.

This comes up in writing, too, with phrasing like:

"If you've never used the Class Browser, it's a powerful tool. (WRONG)

What, it's not a powerful tool if you have used it? This one can be fixed by changing it to something like:

If you've never used the Class Browser, be aware that it's a powerful tool.

or just:

"The Class Browser is a powerful tool.

-----

Don't precede commands with "a" or "an." Either use "the" before it and "command" (or "function" or "property" or whatever) afterwards, or simply reference the command without an article. For example, use one of the following:

You can use the TRIM() function to eliminate blanks.

You can use TRIM() to eliminate blanks.

But you shouldn't say:

You can use a TRIM() to eliminate blanks. (WRONG)

Similarly, don't use "a" before "true" and "false."

-----

Don't follow "between" with "to"; use "between" with "and", or "from" with "to." For example, this version is wrong:

It takes between 5 to 8 days. (WRONG)

These versions are right:

It takes between 5 and 8 days.

It takes from 5 to 8 days.

Don't use "between" before a pair of values separated by a hyphen. The hyphen stands for the word "to."

It takes between 5-8 days. (WRONG)

No preposition is needed here.

It takes 5-8 days.

-----

The verb form of "recursion" is "recur." There is no English word "recurse."

-----

"Loosing" and "loose" do not mean that you don't know where something is. Those words are spelled "losing" and "lose." With two o's, the verb means to set something free.

-----

You don't "return data back to" something. You simply "return data to" whatever. In general, you don't "return back to" anything; you simply "return to" it.

-----

"As of yet" is bad form. Use "as yet" to indicate the current time. Use "as of <a particular time>" for other times. For example:

As yet, there's no solution.

As of last Tuesday, he hadn't finished.

-----

In general, "affect" is a verb and "effect" is a noun, though each does have a specialized use as the other part of speech. For example:

This drug affects the entire central nervous system.

The effect of the drug is to suppress negative thoughts.

As a noun, "affect" refers to feeling or emotion. For example:

The patient had a blank affect.

In addition, "affect" as a verb has the emphasis on the second syllable, while "affect" as a noun emphasizes the first syllable.

"Effect" as a verb is somewhat more common than "affect" as a noun. It means "to cause," and most often, precedes the word "change":

His primary goal was to effect change in the bureaucracy.

-----

Don't say "An example would be." Use "an example is," instead. "Would be" implies that it's true only under certain circumstances.

-----

"Advance" and "advancement" are not synonyms. "Advance" can be a noun, a verb or an adjective. As a noun, it means "the act of moving forward." "Advancement" is only a noun and it means "a forward step" or "an improvement."

Many people use "advancement" when "advance" is a better choice. Here are examples of correct uses for each of them:

The development of the steam engine was a tremendous advance in transportation.

Every piece of research contributes to the advancement of knowledge.

-----

"Advance" and "advanced" are also not synonyms. One buys advance tickets, not advanced tickets. Here, "advance" means "beforehand"; advanced tickets would be the ones you get after the beginner tickets.

### Grammar, Syntax and Punctuation

Understand the distinction between number and amount. "Number" refers to things that can be counted, like "number of people." "Amount" is for things that can be measured, but not counted, like "amount of sugar." While "more" applies to both number and amount, generally, it's better to use "fewer" for number and "less" for amount.

-----

Use "that" to introduce essential clauses and "which" (preceded by a comma) to introduce clauses that can be removed without losing the meaning of the sentence:

ASORT() is a function that sorts arrays.

ASORT(), which was enhanced in VFP 7, sorts arrays.

-----

Don't define terms by saying "x is where ..." or "x is when ...". Either use a direct definition "x is the ..." or find another approach, like "With x, ..."

-----

Don't say "the reason is because ..." Simply follow "is" with the reason.

-----

Use two commas, or none at all, when inserting a clause into a sentence. If the clause ends with a parenthesis, which is not uncommon (this one does), the comma should follow the parenthesis.

-----

Don't abuse the "self" words. These words ("myself," "yourself," etc., known as *reflexive pronouns*) belong in a sentence only if the corresponding pronoun has already appeared. For example, this is correct:

I'll take care of it myself.

But this is wrong:

He had a meeting with John and myself. (WRONG)

That should be:

He had a meeting with John and me.

-----

Use pronouns correctly. The most common mistake is to use "I" when it should be "me," especially when combined with another noun or pronoun. For example:

He brought the book for you and I. (WRONG)

That should be:

He brought the book for you and me.

The general rule for determining whether to use "I" or "me" in a combination is to consider which one is correct if you remove the other person. You'd never say "He brought the book for I."

-----

The word "being" should be a yellow flag. Most sentences that use "being" as the verb would be better rewritten without it. Often, a sentence with "being" is actually just a fragment. Here's an example:

Frangos and truffles aren't the same thing at all. The biggest difference being where  
can you can get them. (WRONG)

This should be rewritten as:

Frangos and truffles aren't the same thing at all. The biggest difference is where you  
can get them.

or:

The biggest difference between frangos and truffles is where you can get them.

or, if you really insist, change the period to a comma:

Frangos and truffles aren't the same thing at all, the biggest difference being where you  
can get them.

But that last version doesn't read as well as the other correct versions.

-----

Make sure you leave something for comparisons to compare to. Too often, people leave out the word "other" or "else" and fail to indicate what they're comparing to:

No one sells cars for this little. (WRONG)

Unless you're expressing the idea that the price can't be correct, it should be:

No one else sells cars for this little.

Or something like:

No one but me sells cars for this little.

### **Odds and ends**

Choose either italics or bold (not both) as your way of emphasizing, then use it uniformly. Don't use underlines at all; that's the typewriter method of specifying italics. Even with a uniform way of emphasizing, don't overdo it. Too much bold or too much italics makes a document hard to read, and defeats their purpose.

-----

When discussing software or other products with multiple versions, be version-independent wherever possible. When version is unimportant, don't mention it. When it does matter, use a phrase like "introduced in Word 2013" rather than "new in Word 2013." The former will still make sense when the current version is Word 2050, but the latter will be quickly outdated.