

# TECHNOTE: On the Elements of a Technote

By Tom Maremaa, Technote Editor  
**TM@applelink.apple.com**  
Apple Developer Technical Support (DTS)

There is no one formula or method for writing a good Technote — no way you can simply connect the dots or plug in the right values and expect that a finished product will automatically pop out. Each Note has its own unique flavor and character; no two are ever alike. This Technote about writing Technotes is, for example, uniquely recursive.

Technotes always have and always will start with one thing: *inspiration*. An idea comes to you that's important, that you want to share with other engineers or developers. And a Technote moves forward from there. Like anything else worth doing, you've got to sweat the details. Writing, inevitably, becomes rewriting; the conversion of idea to language, especially to a language that can be understood by more than an audience of one, presents its own problems. A good Technote will challenge you to be precise in your thinking and, in the course of writing, to validate the accuracy and usefulness of your original idea.

Knowing for whom you're writing — primarily an audience of your peers — is one of the keys to a good Technote. The other is working with a well-defined structure.

Each Technote, when it works, has a particular set of elements that, if you step back and put them in perspective, contribute to the Note's effectiveness. In this Technote, I've outlined some of those elements.

For more information about Technotes, check Apple's Technote Web site at <<http://www.info.apple.com/dev/technotes/Main.html>>.

## About the Elements

---

There are lots of schools of technical writing, as well as schools of thought, on how to shape and structure a work of technical information. I've evolved a few ideas of my own largely in the form of questions you may want to ask as you take a Note from start to finish.

### Technote Title

---

A good title or name captures the essence of the Note. It is important because it not only describes the content of the piece, but also provides a point of reference searching through a body of Notes. Your Note may live or die in terms of its usefulness simply because of its title.

### Metaphor

---

If you can come up with a good metaphor in your title and you can *sustain* that metaphor for effect throughout your Note, go for it. If your title just has throwaway humor or a glib one-liner, realize that your humor may not last long on the reader of your Note. A good metaphor, however, can live a long, long time, if it's done right.

### Intent

---

What is the Note trying to do? What is its purpose and/or reason for being? What particular engineering or developer issues are addressed or solved? Is it advancing or extending our knowledge of the Mac OS, illustrated a technique with code samples and snippets, solving a hardware-related problem, or telling us something not documented elsewhere? Intent is another way of keeping the focus of your Note.

## Motivation

---

What is your motivation in writing the Note? Motivation can drive you forward or backward, depending on what your intentions are. If your Note addresses important development or operating system issues, or is going to solve a particular engineering or design problem, that should provide you with compelling enough motivation to do the hard work necessary to complete the Note. Your Note will help others who've wrestled with similar problems and exhausted all their efforts to find answers.

## Scenarios

---

Providing a few scenarios that illustrate the design problem and how the Note attempts to solve it can go a long way toward making your Note valuable for another engineer or developer. These scenarios can be written narratively; they don't have to be written as numbered set of steps. You wouldn't tell a story to a friend using numbers, would you? If scenarios draw on real-world examples, if they show things as they really are, they can be even more effective in helping the reader of your Note understand abstract concepts and ideas.

## Applicability

---

What are the specific situations in which the techniques shown in the Note can be applied? What real-world examples can you draw from where the technique is applied? How can we recognize, or identify these situations?

## Enough Questions

---

At some point you've got to stop asking questions and just write. The best way to write is off the top of your head, spontaneously putting down your ideas in whatever way they come to you. Later, you can always revise your ideas, shape, edit, and recast them if necessary as you refine the elements and language of your Note.

## Using the Elements

---

If your Note focuses on a particular technique, you may want to work through the body of your Note systematically as follows:

### The Technique

---

Describe the specific technique and define it by its role and function.

### Collaboration

---

Specify how the technique collaborates — i.e., what it relies or depends on — to carry out its function or task.

### Consequences

---

How does the technique support its objectives? What are the tradeoffs and results of using this technique?

### Implementation

---

What pitfalls, hints or techniques do we need to know when implementing the technique? Are there language, platform or release-specific issues?

### Sample Code and Use

---

Show code fragments and snippets that illustrate how the technique is implemented and used.

### Known Uses

---

If this technique is used in systems, describe the scenarios here.

## Related Techniques

---

Describe any related techniques here at this point in your Note.

## X-References

---

Include cross-references to other sources and documentation.

# Writing for an Audience of Peers

---

Engineers and developers are a wily bunch, a breed apart from other scientists. They work under tremendous pressure, dealing with a mountain of technical and programming issues that the average person could not begin to grasp in a month of Sundays.

Writing for this audience is demanding; expectations are always high. Your Note must not only be technically accurate but also well-reasoned and, ultimately, useful to others who are trying to learn from you.

Think of this process as, simply, writing for an audience of your peers.

To write for an audience of your peers, you need to ask a lot of questions that may be in the minds of that audience, such as

- How do I do this?
- How do I *use* a certain technique?
- How do I use this specific system functionality?
- What do you recommend that I do in a specific situation? What are my choices?
- How can I *improvise* around using the technique?
- What do I need to *worry* about? For example, are there dependencies?
- What are all the *details* of a specific feature or resource?
- What are the limitations, restrictions, and outstanding issues connected with this technique?
- Why does this *behave* the way it behaves? What influences this situation?
- Is this a bug, limitation, or a shortcoming? What works today? What doesn't?

- Why was it designed this way? Do we *expect* it to change?
- What could I do to *change* the situation?
- What else do I need to know about the *background* and other issues related to this technique or design problem?

## Working with an Editor

---

Some engineers hate editors, others adore them. Whatever your preference, the truth is that in order to produce a quality Technote, you'll have to find a way to live with them.

The Editor is there as the first reader of your Note and will make sure you've got all your ducks in a row, that your Note doesn't veer off in some unknown direction, that your language is clear and lucid. The Editor is there to take you through the process of rewriting because most Notes need to go through a few drafts to work out all the kinks.

### Peer Review of a Technote

---

The Editor is also there to help you through the review process. That's when your Note is held up to the scrutiny of your peers for their comments and feedback. This can be a particularly harrowing experience and your Editor can act as a traffic cop to direct the information and feedback flow for your Note. Ultimately, however, you must be the arbiter of what goes in or comes out of your Note based on the comments and feedback of your peers.

#### **IMPORTANT**

In the True Spirit of Recursion, Return to Page 1 for a Recursive Discussion on the Elements of a Technote. ♦

### Acknowledgements

---

Special thanks to Kent Sandvik and Guillermo Ortiz