

Answers to Frequently Asked Questions (FAQs)

Every sentence I utter must be understood not as an affirmation, but as a question.

—Niels Bohr,
New York Times

When guests get stuck they most often turn to the FAQ, because the style seems friendlier than the average help system, and the genre promises answers to real questions from users, rather than a stonewalling corporate pile of documentation.

Most FAQs adopt a fairly conversational style. That style probably reflects the List Serv heritage, where a list moderator who got tired of answering the same questions over and over would collect all the answers in one file for downloading. Originally, the relationship was personal, or at least one-to-one, as the moderator answered one e-mail after another, and then simply pulled together all the best answers in one place to make the FAQ. The moderator usually had a real interest in helping out, and knew the niche audience very well, being part of it, so the answers often sounded like insider talk.

The basic structure of a FAQ suggests a conversation, where the visitor asks a question and you respond. That give-and-take shows the genre is a pure Internet play.

But, perhaps because of this conversational setting, FAQ sentences tend to ramble. Worse, some writers add clause after clause, and insert subordinate clauses within relative clauses, tossing in some *whens* and *ifs*. A guest who wants to understand may have to pull these multiple clauses apart, parsing meaning out of each piece, and then putting the pieces together to make sense. Tip: Use only one subordinate clause per sentence—and put that clause at the beginning or end of the sentence, because readers find extra clauses easier to absorb in those slots.

Alas, the passive voice also attracts a lot of FAQ writers. Turning an active verb into the passive (transforming “We confirm your order” into “Your order is now confirmed”) lets writers avoid saying who or what is doing the action, even though that kind of trick makes the sentence ambiguous, “Your entry is refused.” By whom? By what? If no one is responsible, how can a visitor get mad? Easy.

Try 'fessing up by using the active voice. “If you get this message, that means that our database can’t understand what you’ve typed. Please try again, using the tip next to the slot on the form.”

Break your paragraphs up. Too many FAQs try to compress the entire answer into a single paragraph, as if that made it a unit. But you’re on the Web. People come to the FAQ upset, anxious, angry, ashamed—not in a great mood for reading, and certainly not thoughtful enough to read 10 or 15 lines of prose extending from one side of the window to the other. Got a list? Reach for the bullets. Open that answer up even if you end up using half a dozen paragraphs.

Write the questions in the persona of the guest

Well-written questions—

- “How do I order?”
- “How do I find what I’m hoping to buy?”
- “How do I redeem a gift certificate?”

Notice the “I.” The writers are standing in for the guests, speaking for them, phrasing the questions as if they were really coming from the guests. Sure, these questions have been sanded, planed, and simplified way beyond what the average guest might ask. But consider the position from which they are asked: they come from the guest, not the site. Right on!

Not so hot questions:

- “What is the order process?”
- “What is Findoromatic Search?”
- “Gift certificate policy”

These questions are self-centered. They reflect the site’s concern with its own policies, concepts, and values. These questions do not sound like what a guest might ask. In fact, the last one is not even a question. Avoid “questions” that are really just labels for content: the point is to evoke a real conversation, even though it is virtual.

Should you start every question with the word *Question*? Sure, if you start every answer with *Answer*. But generally people understand the convention of Q&A, and grasp a boldfaced question without the label.

Students: “Aren’t these the same questions you asked on the exam last year?”

Albert Einstein: “Yes, but the answers are different this year.”

Key: Separate the answer from the question. Press Return after the question. Each paragraph has its own function, so it deserves its own location and format. The different look and place suggest taking turns in the conversation, with the users' questions bigger and bolder, as befits their importance, and your answers plainer, and perhaps even indented, to indicate your humble responses.

Create troubleshooting sections

Start with the symptom (not a question), as described by the user (not an engineer). "I downloaded the software, but it won't start." Not "Difficulty in launch."

Sum up the **problem** in terms that your guests would really use, so they can recognize their situation. When most people get in trouble, they describe their problem to themselves, and they use those words when they call or e-mail customer support. If you want people to recognize that you are offering a solution, use their words in the heading.

Then give your **diagnosis** of what may be going wrong, if you have any idea. Identify the diagnosis with a subhead, setting it off, so the anxious user can skip this section, and the curious can learn to perform this analysis.

Break out the **solution** with its own subhead. Usually, this section requires numbered steps if there is a concrete sequence of actions to perform. But sometimes you're offering a handful of options in a bulleted list. Only use numbers if the user must perform the steps in order.

If you face a really complicated problem, use a branching diagnostic. In response to the problem, ask a question and offer all possible answers as links. When the user clicks one, offer the next logical question, and the next, and so on, until you are pretty sure you know what the problem is. Then give the specific fix for that problem.

Branching diagnostics like this are wonderfully helpful. Look at the way Windows Help handles printing problems. The branching means the user is following his or her own path, unaware of all the other possibilities, able to maintain focus.

*Worrying is the most natural
and spontaneous of all human
functions.*

—Lewis Thomas,
The Medusa and the Snail

*The task is to be aware and
sensitive to your customers'
expectations and level of upset.*

—Ron Zemke and Tom Connella,
e-Service

To do all the talking, and not listen, is greedy.

—Democritus of Abdera

Give steps

Breaking your instructions out into numbered steps makes them more effective. People understand the sequence better, skip fewer steps, and succeed more often with numbers.

One meaningful action per step, though. Don't try to cram two or three actions into a single instruction, because people do the first, then the third, or get confused. Piggybacking makes your procedure look simple, because there are fewer steps. But the individual steps, clotted with more actions than a user can absorb at once, end up failing.

What's meaningful depends on the guest. For instance, beginners find command lines a tough way to interact with a program, and consider typing DIR a major action, and pressing Enter another big step. Of course, experienced DOS aficionados understand "Enter DIR" to mean, "Type these letters and hit the Enter key." Here's a case where personalization can help you differentiate your FAQs by skill level. The instructions for experts generally compress more actions into a single step, and skip most explanations.

Begin each instruction with a verb in the imperative, if possible. Don't say what the user can do, or may do, or might do. Tell them *what* to do. You may feel you are being rude, but bossing the user around is actually a courtesy because you save time, and make clear that the user really must do x, y, or z. No *maybes*, then.

What can you put before the verb? Motives and context. If you have to say why they should do the step, start with the purpose and then move to the action. "To confirm your changes, click Change." If you need to say exactly where to act, put that first. "At the bottom of the license click Submit." In this way, users know up front what the point is, and where to look. Knowing the goal and the locus of operations will help them grasp the instruction better. Also, this sequence follows psychology more closely: first, you have a goal, then you look around for an affordance to operate on, and then you act. (Evaluating whether you have succeeded comes later—in a following paragraph, if ever).

Don't pretend you are writing instructions and then decline into marketing talk. For instance, "3. Using Find Topics is a quick, easy, and powerful way to describe the particular product you may

be looking for.” Oh gee, that’s wonderful. But that text does not answer the basic question, “What do I do next?” Giving me an ad instead of a step just gets me mad.

Also, don’t adopt the system perspective. You are not writing specs, describing how the system is supposed to work. You are giving a user instructions. So you are not responding to the user’s needs if you write something like, “The system requires your billing and shipping information.” Switch your perspective. “Please enter your billing and shipping information.”

Separate the instructions from any explanatory material, putting the explanations in following paragraphs. Each step answers one question: “What should I do next?” The explanations tend to answer ancillary questions, such as, “What does that term mean? What’s the result?” So the explanations are different elements, with different responsibilities, and they deserve their own separate identity.

Breaking explanations away from the steps means the impatient user can just read the steps and do those, and never read your explanations. Nervous folks want their sweaty palms held tightly, and seem grateful for the separation, which makes clear “You do this,” and “Here’s more information if you want it.”

Gradually unfolding works better than jamming all the information into a single paragraph. Also, if you are working in an object-oriented environment, you want to identify and omit explanations when you send this stuff out to a mobile phone, or turn it into a quick reference.

Consider, too, the order of explanations, and try to follow the user’s likely sequence of follow-up questions. For instance, people usually ask what a particular term means in the instruction, before carrying out the instruction; so a definition of a concept would come pretty close after the instruction before any description of the result of the action.

Whenever people recognize that they are reading a series of instructions, they expect the structure to be strictly chronological. They expect everything—steps, explanations, pictures—to show up in exactly the order they experience carrying out the instructions. This approach seems simple, but if you really try to apply it to your answers, you’ll find that in many little ways you have jumped ahead

*The great end of life is not
knowledge but action.*

—Thomas Huxley,
Science and Culture

of yourself, or forgotten to mention some little action you take for granted, or you have actually scrambled the order of steps. Creating good instructions means beating on the sequence until you're certain that you have evoked the moment-to-moment experience.

Handle branching with bullets

Occasionally, you reach a decision point. The guests should do X if they want to print, or Y if they want to save. OK: the step should read something like this: “Decide what you want to do now.” Then drop in two bulleted items:

- If you want to print, do X.
- If you want to save the file, do Y.

In this way you bring the decision to the attention of the user, which is good, because many users don't realize they have to make a choice. A decision is a meaningful action; it deserves its own step.

The bullets separate out the options. Beginning each one with an *If* clause helps the user decide whether that particular option applies or not. If it doesn't, the user can skip to the next item. In this way, you avoid the peculiarly American problem, where people read an item such as “Do X if you want to print,” and press X right away, even though they do not want to print, because they act before they read the whole line. Starting with the IF clause helps make sure they make the right choice.

Go ahead, repeat yourself

Few guests actually read through the whole FAQ. If you think the same caution applies in half a dozen answers, include it in each one. You won't bore anyone but yourself, because only you actually read all of those answers.

For instance, if you have ten answers describing various settings the user can change, but none of these changes takes effect until they complete the current login session, then you better warn them of that little constraint after each answer, so they realize that, even if they have made the change as you suggested, nothing will happen until they log out and log back in. Little stupidities like this often get described only once, somewhere in a note where nobody may stumble on the information.

Cautions should be repeated—wherever the user could get in trouble. By the way, don't call a cautionary passage a *note*, or a *tip*—people often skip sections with those names as unimportant extras. You don't have to put a skull and crossbones next to your cautions or warnings, but you should make clear what they are with labels and special formatting. And don't wait until the end of the answer to mention, “Oh, by the way, back in step 3, if you happened to type the wrong name, the software probably froze up.” Better to put a caution in step 3, saying, “Please be careful to type your name right, so you don't confuse our rather simpleminded software.” Obviously, your engineers ought to stop ambushing users like this, but until they do, your job is to alert people before they get in bad trouble. If a misstep could result in lost or garbled data, put your caution ahead of the step. If the problems are easier to fix, put the caution right after the step.

Can steps be links?

Yes. If each step is actually described in a separate answer, because it involves several actions, you can tell people, “[Enter your credit card information](#),” making that instruction a link to the full details on entering the credit card dope.

Show pictures

As our audiences grow to approximate the general population, people who have trouble reading join the ranks of people who just resent being forced to read on-screen. Even though your work revolves around text, think visually. If possible, make up a storyboard for each important answer showing the key concept or diagramming the process.

Where you have a series of forms you want people to fill out, put a series of numbered instructions at the top of each page with an illustration, and highlight the form the user is on. In that way, as the user moves forward through the sequence, the visuals show progress, while explaining each major step.

Bring that kind of thinking back to the FAQ. Look at the Macintosh manuals, where each step starts with an image on the left, shows a giant number in the middle, and, finally,

over on the right, explains the picture with text. Pictures first, text second—that ought to be your mantra in FAQs.

Show pictures of icons. Show tools. Show screenshots.

Develop diagrams of complex concepts, process flows, and relationships. When the text inside the diagram acts as notes, you are getting through. When the art is so expressive that you do not need to add any text you've achieved the high level of Lego manuals.

Find out if you've answered the question

This ploy takes a lot of guts. At the end of each answer, ask whether you have provided the information the visitor needed.

And what if you failed? Give them an e-mail form, addressed directly to you, personally. You need to find out if your answers are working. And you owe it to the visitor to clear up the confusion.

Only really bold sites do this. But think of the impression they make. A guest reads an answer, and it may make sense, but the guest sees the offer. Wow!

Of course, as the writer, you have to answer a lot of e-mail if your prose is murky.

Keep growing the FAQ

Every time a guest asks a question—by e-mail, phone, chat, discussion board, or mail—you should consider adding the answer to the FAQ. If you believe the answer is already up on the FAQ, ask yourself: How come this particular guest couldn't find that? Or, if they read it, how come they couldn't understand it?

More importantly, recognize that you may not really have answered the question. This morning we had a little battle with Amazon's customer support, because once we had started an order using their famous patented, copyrighted, and trademarked One Click™ process, the damn thing wouldn't let us change the shipping method, even though we changed the settings repeatedly. So we complained. Their first e-mail told us how to change the settings for the One Click process, which, of course, we had already done. Well, maybe the customer service person just hadn't read our e-mail carefully; maybe we didn't make it clear that we had already tried this. We wrote back, asking why we weren't able to

Every man becomes, to a certain degree, what the people he generally converses with are.

—Lord Chesterfield, *Letters*

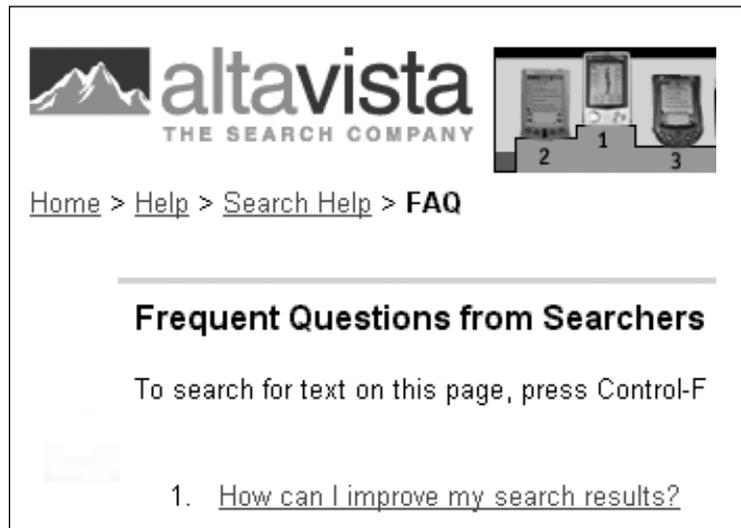
change the shipping method during the One Click process. They responded—nice to get a response, of course—with a full boilerplate FAQ about all the neat things you can change. Of course, the FAQ didn't recognize the problem we had encountered. Their writer, spotting a problem with changing the settings, just dumped those answers into the e-mail without really reading what we had written the second time. Net result: we were pissed.

Too often FAQ writers act as if the system is always working the way it is supposed to, or the way the engineers say it will. Try the task yourself and keep looking for areas where an ordinary person—like you—might get confused. Invent several scenarios and try each one out, using your instructions. You'll often find problems, variations, and unsuspected gotchas.

And when you feel your answers are really accurate, run a little usability test, the quick-and-dirty way, using half a dozen people to see if they can actually locate the answer and then succeed at acting on it. You don't need scientific proof or statistically meaningful measurements. You just need to watch people struggle with your prose. The experience of watching people not find your answer, not follow it, not understand it, and not succeed can be pretty sobering. Your fingers will begin to itch to rewrite.

See: Boggan, Farkas, and Welinske (1996), Duffy, Palmer, and Mehlenbacher (1992), Horton (1996), Price and Korman (1993), Price (2000), Weiss (1991), Zemke and Connellan (2001).

Case Study: FAQs at AltaVista



AltaVista, now billing itself as The Search Company, puts its marketing pitch inside the Help pages. For instance, in Help, when I choose the first topic, I am told that searching today is “far easier than ever before.” How come? You “don’t need to learn any commands.” OK! I’m for that. “Just enter words related to what you want and chances are good that useful results will go to the top of the list of matches.” I like the phrasing: the odds favor your getting decent hits.

The three-single-spaced-page topic contains thick paragraphs, wall to wall with 10-point Arial text—a daunting prospect, even on paper. The text is thoughtful, thorough, and unmarred with sub-heads, boldfacing, or tables (although these do show up on other Help pages). To escape, I go back to the Help menu and choose the Multimedia FAQ.

The questions are good. Most are simple, and some are even

phrased as if I were asking them: “How can I save an image?”

The answers appear in two to four short paragraphs. But even when the writers deal with a list of things, such as file formats, screen resolutions, or hardware requirements, the text remains resolutely running in paragraphs—almost no bulleted lists for options, no numbered steps for sequences of actions, no tables, just text.

Erudite, articulate text, making good points. Considerate text, recognizing my goals, and acknowledging problems I may face, even offering workarounds and remedies. But passive, *be-all, is-was* text, with almost no links to further resources.

I back out and choose Frequent Questions from Searchers. I guess I am a searcher. Here the writers believe in bulleted lists, and they boldface their sample search terms, so the page looks a lot brighter. “We” are talking to “You,” which makes clear who does what. When telling me what to do, the instructions often start with verbs in the imperative, which I appreciate, but then some of the steps devolve into several different actions, so I have to reread the text several times just to sort out what I do first, second, and third. And as soon as the writers undertake to explain how the search mechanism works, they get choked up, and start writing ponderous stuff like this: “A search engine is comprised of software that automatically performs many complex tasks.” Well, glory be, let me kneel down before the mainframe, and perform a salaam. In general, though, the writers keep their focus on the important players—we and you, our robots, and Webmasters. “Our robots will pick up these changes when they revisit a site or the Webmaster may resubmit the page.” There’s a little drama, and the writers describe the characters in action making the story pretty easy to follow.

And the writers also recognize that the FAQ may not be the final answer.

Did This Help Page Answer Your Question?

If not, try:

[Finding specific information](#)

[Search FAQ from Webmasters](#)

[FAQ for the AltaVista Network](#)

Or send us E-mail: search-support@altavista.com

When I go to the Frequently Asked Questions about the AltaVista Network I find a new style—numbered steps for actions and short single-paragraph answers for concepts. Crisp and clear, and easy to use.

So digging through the FAQs is like visiting an archeological site, where you can see different layers of ash and bones and pots left by the several generations of people camping there. The effect is not particularly confusing—in fact, I got the feeling that I was looking at ur-altavista material from its inception, tech comm stuff from its days as a Digital brand, and then shorter stuff from its newest incarnation, along with the marketing concepts in the surrounding Help. Various voices, then, are building up the site. But I think the results will vary, too. The most efficient answers—the ones I could put into action with the least thought—came in numbered steps. The least efficient answers were the all-text rambles about, of all things, multimedia (not a picture in sight). Perhaps the writers ought to get together at AltaVista, do some user testing, and agree on a common strategy, which I hope will focus on giving clear instructions.