Asking Questions Online

Mark Bools

October 14, 2020

Last Modified: February 25, 2021

Abstract

Learn how to ask questions online that will get a response.

Questions asked online range from the trivial to highly complex. The more complex your question the more time you need to invest in asking it. This article addresses some issues surrounding online questions.

Before launching into my tips for asking questions that will get a response online a note for people responding to questions online.

Remember that the person asking may be working in a second language so be forgiving of poor grammar or word choice. Also, keep your answers clear using only commonly used words. While most writing advice is to keep jargon to a minimum, when answering a technical question it's actually most helpful to use appropriate jargon as this may well be more readily understood by non-native speakers.

Answer the question that is asked. I often see people respond to questions with, essentially, "don't use that tool, use this one". This is a response but not an answer.

Let's first look at how *not* to ask questions.

Don't ask a question without first:

1. RTFM (Read The Fucking Manual)—the people you are asking to answer your question will likely have a low tolerance for questions that can be easily answered by reading a `man` page or manual. Questions like "What does the X option do?" are likely to provoke answers linking to the `man` page or even a simple "RTFM".

2. Do a thorough search—it's both lazy and rude to ask people to donate their time to answering a question you could answer with a proper search (I include some tips on searching below).

Okay. So you've read the manual and done a proper online search but the answer is still unclear or unavailable.

The following are just tips for getting better and quicker responses. Not all tips apply to all questions. There is no such thing as a stupid question but there is such a thing as redundant or disrespectful questions (i.e. “I can’t be bothered to read the manual so I’m asking here”).

These tips are ordered to mirror the question type they apply to; trivial questions to complex.

Sometimes you won’t understand the answer.

Sometimes your question requires a highly technical response and you may not understand the answer. You might be able to get further clarification but sometimes you simple have to hit the books and learn more about the topic on hand.

For example, asking about complex netfilter firewall configuration without understanding how packet filtering works. Sure, you may get an answer but will you understand it enough to know it’s the right answer?

What sort of question do you have?

Your answer to this question will guide you to the correct place online to ask your question. It also determines how much preparation you need to do before asking your question.

If your question is simple one of preference (“which X do you prefer for Y”) then less technical, more general forums are preferred. There is little point, for example, going onto a Linux forum to ask “Which operating system is best?”, if you do it’s a fair bet people will assume your are trolling. On the other hand ask this question on r/AskReddit and you will get plenty of responses (whether they’re actually useful is another matter, but in this case this is more a function of the question than the forum asked).

Assuming you have a technical issue then is it a problem (e.g. an error in your system), a question of clarification (“I see I should do X but why?” or “I’ve done X but I’m not clear on how this works, can someone help me understand?”), or an ‘I don’t know’ (e.g. “I don’t know how to X?”).

Be clear. Be specific.

Clarity is your friend. Always.

The more specific your question the more useful the answer is likely to be. Asking non-specific questions opens the floodgate to very general answers and opinions.

More complex issues.

I you are having a specific problem then you have some work to do. There are a few approaches to consider. First, can you provide *all* the details surrounding the problem (your specific configuration files, for example)? This may not be possible because your setup is commercially sensitive or contains secrets you cannot reveal or it is so tangled with other material that to show the items specific to the problem would mean publishing a whole load of unrelated material. If you cannot publish the problem material directly then you need to put together a minimal example that causes the problem so that you can provide context to your problem (this applies even more when reporting a bug).

Depending upon your specific situation this context may be as simple as putting your minimal configuration into pastebin but often will require more effort *from you* (do not leave people with a lot of work to do reproducing your problem, if you do they're less likely to help). If you can, create a script to setup your problem scenario. For example, create a VM or container that reliably produces your problem and post the setup for this VM alongside your question. Your much more likely to provoke a response if people can simply run a script, see your problem, analyse it, and write a response.

This may all be overkill. Your problem may be a simple syntax or semantic error. You may get away with simply providing a pastebin containing your configuration or program.

Why pastebin (or similar)? Most forums have some form of rendering and this often makes it difficult to reliably cut and paste sample code (especially for file formats sensitive to spacing). Using a service that provides the raw text makes this problem go away. The less work you create for the people you're asking for help, the more likely they are to help.

I don't have time to do this! I need an answer NOW!

Well fuck you! Why would someone spend *their* time when you can't be bothered to spend your own?

Describe what you have already tried.

This may be as simple as "I tried X" or more complex (linking to a `git` repository with your failed attempts), but showing that you at least tried to solve the problem yourself motivates people to help and it reduces the number of unhelpful responses suggesting things you already tried.

Format your question.

Most forums allow some formatting. Use it! At the very least structure your question clearly so people can see:

- The question you have
- What you have already tried.
- Context

Searching

The number one skill for IT workers is the ability to search efficiently. Based on some of the lazy, redundant questions I see online people are either too lazy or too incompetent to search effectively. Here are some really basic search tips that will help you find help quickly (certainly quicker than asking for help online).

I always start by assuming that "someone *must* have come across this problem as well". Most of the time I'm right to assume this although I admit finding the answer can mean fiddling with the search for a bit to find that answer.

One pitfall of searching is when your search term is so general you get all sorts of crap in the response. My favourite example is the `python` tool `poetry`. I mean, almost any broadly worded search for 'poetry' is going to result in many non-python responses; try 'poetry help' or 'help using poetry'—you might get good responses if you regularly use Google to search python related material, but on search engine that does not track your every move it's a different story. The solution is to make all `poetry` searches into `python poetry` searches. I'm sure there are some poets producing poems about python, but you're much more likely to find the `poetry` tool.

Error messages are great search terms. If your problem is related to a specific error message, try entering that directly into the search engine. If that does not produce the answer you're looking for try putting the error message in quotes (`"`). If you do use quotes then remember to remove elements specific to your problem (e.g. filenames). You may also want to try prefixing your message with context. For example, if your error message is being output by the Python interpreter try prefixing your error message with `python`.

Try the next page! It's easy to assume that if your search does not turn up the correct answer on the first page of results then your search failed. Not true. Try looking at a few other pages of results.

Try more specialised search engines. It's tempting (I fall for this one myself) to assume DuckDuckGo will provide the answer straight away. Sometimes it's worth narrowing the search by choosing a more focused search engine. For example, why not try stackoverflow directly? This will automatically cut out a lot of non-relevant results. Returning to my early example, typing `poetry` into the stackoverflow search box returns lots of results relevant to the Python tool and none for rhyming couplets.

Use metasearch engines. I use my own SearX installation (try SearX if you don't want to set up your own). This searches multiple search engines simultaneously and allows you to choose which results to focus on. Metasearch engines can be a double edged sword though; they allow wide searched using a single search term but can produce a lot more (irrelevant) results. Try it out, see if it's helpful to you.

If a search produces too many irrelevant results, add more specific qualifiers to your search term. Remember, your search does not have to be in the form of a question, just keep adding terms to eliminate the crap from your results. If I enter `poetry` and get a load of guff about rhyming couplets I realise I can eliminate these by adding `python` to further limit my results.