

# FAQ for Barf's dBox page

## Table of contents

1 Questions.....	2
1.1 1. Warum schreibst du nicht auf Deutsch? Du bist ja offensichtlich auf einem deutschen Server/(Why don't you write in German?).....	2
1.2 2. Why isn't this stuff in CVS?.....	2
1.3 3. What is a patch? What can I do with it?.....	2
1.4 4. What if the patching goes wrong?.....	2
1.5 5. Can you give me an idiot-proof step-by-step description of what I have to do to get one (or several) of your patches on my dBOX? .....	3
1.6 6. I don't know anything about this Unix/C crap. How much effort would it take to learn do all this? .....	3
1.7 7. Patching and compiling is sooo complicated, can't you provide a binary?.....	3
1.8 8. Ok, I have a binary (e.g. zapit). I use an image that I want to keep using. How can I put the new binary into my image? .....	3
1.9 9. Where does the name "Barf" come from?.....	3

## Questions

### 1. Warum schreibst du nicht auf Deutsch? Du bist ja offensichtlich auf einem deutschen Server/(Why don't you write in German?)

Englisch ist die Nummer 1. internationale Sprache der Wissenschaft und Technik, genau so wie Latein für die Medizin, Französisch für das Postwesen, oder Deutsch für die Theologie. Entweder schreibe ich auf Deutsch, und schliesse nicht Deutschsprechende aus, oder schreibe ich auf Englisch, und schliesse nicht Englischsprechende aus. Ich glaube das zweite ist besser. Die eigentliche Zielgruppe dieser Seite versteht Englisch, da bin ich mir ziemlich sicher.

Ausserdem gibt es recht wenige englischsprachige dBox-Sites. Und die, die es gibt, gehören fast ausschließlich zu "der dunkle Seite".

### 2. Why isn't this stuff in CVS?

It differs. Some of the stuff are, or will be checked in. In some cases the patch is not quite reliable, or there are other reasons for not committing it. This site makes it possible to publish patches that are not to be committed.

### 3. What is a patch? What can I do with it?

Short answer: If you really need to ask, then it is not for you. Long answer: A "diff" describes the difference between two files, one "original" and "fixed". It is generated by the program `diff`. It contains the changed lines, together with a few content lines around the changed lines. These content lines makes the patch and the patching procedure somewhat robust, for example if the original file changes in a different place. When the diff is distributed, in the intent of making it possible for people in the possession of the original file to generate the fixed file, the diff is called a *patch*. With e.g. a text editor, the original file and the patch, the fixed file can be constructed. There also exists a program which automates this task. It was written by Larry Wall possibly 20 years ago, and, surprisingly, carries the name `patch`. Typical use of the patch program is: `patch foobar.c < foobar.c-patch`. This is called "applying the patch" or "patching". This may go wrong, you must check the output of the program! See next Q:

### 4. What if the patching goes wrong?

You must not go on without thinking! Most likely, the "original file" has changed in a way that makes automatic applying of the patch impossible. If you have a reasonable understanding of the syntax and semantic of the files you are patching, you may be able to use your understanding and your brain to manually apply the patch in a way consistent with the original intention. Independently of this, you should inform the author of the patch, he/she may or may not be willing to update the patch (for you and for the rest of

the community!).

### **5. Can you give me an idiot-proof step-by-step description of what I have to do to get one (or several) of your patches on my dBOX?**

Short answer: No. Because there is no such thing. Slightly longer answer: Main steps are: Checking out the sources of the CVS, applying the patch(-es), configuring, compiling, possibly customizing, building an image, flashing it.

### **6. I don't know anything about this Unix/C crap. How much effort would it take to learn do all this?**

Consider it as getting into a new hobby. If you have no previous experience, it will probably take months before you have successfully built and flashed your first working image. Don't want to scare anyone away...

### **7. Patching and compiling is sooo complicated, can't you provide a binary?**

In some cases (like zapit) this makes sense. For programs that changes almost daily (like Neutrino) this would make no sense.

### **8. Ok, I have a binary (e.g. zapit). I use an image that I want to keep using. How can I put the new binary into my image?**

First of all: if your image is considerably different from the current CVS, it may not work at all, since the API and the communication between the programs tend to change fairly often. If your image has its root filesystem writable (so-called *jffs2-only-image*), you can just overwrite the old file (after making a backup :-)) with the new one. Unfortunately, this is seldomly the case :-). The images with non-writable root filesystem have their `/var` file system writable. It may be possible to put your binary there (say in `/var/bin`), and somehow tell the system to look for it there. If also this in not feasible, you have to extract the root file system partition, transfer it to a PC, unpack it, modify the unpacked file system, make a new file system, transfer it to the dBOX and flash that partition. How to do this is outside the scope of this FAQ.

### **9. Where does the name "Barf" come from?**

"Barf" is one of the main characters in one of my favorite movies of all time, [Spaceballs](#) by [Mel Brooks](#). In today's English, "to barf" means to vomit or puke (literally or symbolically). Googling for "barf" also gives a number of acronyms.