

## Re: CFO: Why ./configure?

**Source:** <http://linux.derkeiler.com/Newsgroups/comp.os.linux.development.apps/2003-09/0382.html>

---

**From:** VBDis (vbdis\_at\_aol.com)

**Date:** 09/15/03

Date: 15 Sep 2003 02:55:19 GMT

Im Artikel <bjsgrd\$soo\$2@mamenchi.zrz.TU-Berlin.DE>, Thomas Richter <thor@cleopatra.math.tu-berlin.de> schreibt:

>> *wouldn't it be easier to create a header memory.h ( as a symlink or a  
>> copy of bstring.h ) to gain the compatibility?*

>

>*No, I'm not the administrator of the system. I can't do that.*

Then ask your administrator to do that for you. Then the system wide solution can be used by every user, for every project. Your administrator also will be happy about procedures which are readily available for the system, so that he doesn't have to reinvent the wheel for every single problem.

>*Where's the X11 include I need?*

>*Where's the ncurses include I need? How or why does this call trigger*

>*makes in all the sub-directories? Where's the documentation created?*

>*Where's the archive packed?*

All these also are questions for an administrator. He should specify the appropriate places, once and forever. Then every build and installation procedure should consult the administrator of the system, i.e. read all required information from the appropriate files on the target machine.

>*It seems you haven't seen complex software projects yet. The makefiles*

>*here do more than just to build programs, and they do more than just to*

>*put a single executable into a specific disk location. They're not so easy*

>*to follow, and not five-liners.*

The currently used procedures are not easy to follow, because they boil down to exact and specific operations. Portability requires that all procedures are described in a more abstract way, which is understandable by the tools on every platform, and best also to the human reader, which may have to resolve remaining problems.

The autotools can be seen as a step towards an IDE, and the ideas behind these tools are not bad. But the implementation sucks :-(

An IDE should be part of a specific system, not part of every package. Such an IDE will provide all information about a software project, and will put all that information into the package. Then the IDE of the target system will use that information to build the package, in exactly the way which is appropriate for that target system. Then any assumptions or suspicions about the possible target platforms are obsolete, and must and should not become part of a distributed package.

DoDi