
Subject: Re: Windows drives vs POSIX mounts
Posted by [guido](#) on Fri, 28 Apr 2006 19:43:26 GMT
[View Forum Message](#) <> [Reply to Message](#)

luzr wrote on Fri, 28 April 2006 16:27

Well, I guess, "porting" is quite strong word here - right now I am writing this from ubuntu, while developing using TheIDE - no problems with fileselector

However, adding some common root pathes to the fileselector droplist looks like a good idea to me. But I guess it should be done more carefully - maybe depending just on names is not a good idea.

Concerning .dotfiles, yes, I guess, there definitely should be an option.

Mirek

Sorry, current file-selector doesn't support removable media on Linux. Even if the users knows about /media (which she shouldn't need to), FileSel still only show mounts as regular folders. If this miraculously isn't true for Ubuntu I'd like to know how so.

And the names below /media are *POSIX STANDARD* - I haven't picked them arbitrarily or because they happen to be true on my current favourite distro du jour. Obviously this isn't a 100% solution. But POSIX has no API for distinguishing (physical) devices. You can figure out, if a directory is a mount point in a portable way, but that's it. So you either need to add a libsysfs plugin for Linux and something else for FreeBSD and something else for MacOS X later on - or open the libhal/dbus can of worms, which for various reason you really don't want to. Ask, if you don't know what I'm meaning. libhal maybe ok in a couple of years, now it isn't for upp IMHO.

As the solution I presented doesn't break FileSel for anybody, while making it work better for modern standard compliant Linuxes, I suppose, I don't see why you would pass it off. View it as a stop-gap, if you will.

I understand that upp wasn't meant to be portable initially. Even though upp almost works on Linux now, the code still is strongly biased towards win32 in places. FileSel down to FileSystemInfo especially lacks many features on POSIX by design. I don't see how abstracting root and drive handling from the currently flat drive(letter) system can be avoided, if you are serious about POSIX implementations. On MacOS X /Volumes is a hidden directory, so users there might not even be able to access this directory from upp's file-selector, once ported. I'm not sure so on what level hiding is done - filesystem, libc, Carbon/Cocoa, Finder, so maybe /Volumes indeed would show up in upp apps - but Mac users are notorious for not tolerating such alien means.