
Subject: Re: Windows drives vs POSIX mounts
Posted by [mirek](#) on Fri, 28 Apr 2006 14:27:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

guido wrote on Fri, 28 April 2006 09:36I tried to port FileSel to POSIX.
Unfortunately it makes assumptions which work for the MS-DOS drive letter system only.
In the POSIX world "drives" are mounted into /mnt and /media or /Volumes in the case of MacOS X.
Only the boot disk is mounted directly into the root (/).

I hacked FileSel to have it work better with modern Linux systems, conforming to POSIX FHS 2.3:

```
Image PosixGetDriveImage(String dir)
{
    if(dir.Find("cdrom") == 0 || dir.Find("cdrecorder") == 0)
        return CtrlImg::CdRom();
    if(dir.Find("floppy") == 0 || dir.Find("zip") == 0)
        return CtrlImg::Diskette();
    // if (??) return CtrlImg::Computer();

    return CtrlImg::Hd();
}

bool Load(FileList& list, const String& dir, const char *patterns, bool dirs,
           Callback3<bool, const String&, Image&> WhenIcon, FileSystemInfo& filesystem)
{
    if(dir.IsEmpty()) {
        Array<FileSystemInfo::FileInfo> root = filesystem.Find(Null);
        for(int i = 0; i < root.GetCount(); i++)
            list.Add(root[i].filename, GetDriveImage(root[i].root_style),
                    Arial(FNTSIZE).Bold(), SBlack, true, -1, Null, SCyan,
                    root[i].root_desc, Arial(FNTSIZE));
    }
    else {
        Array<FileSystemInfo::FileInfo> ffi =
            filesystem.Find(AppendFileName(dir, filesystem.IsWin32() ? ".*" : ""));
        if(ffi.IsEmpty())
            return false;
#ifdef PLATFORM_POSIX
        bool isdrive = dir == "/media" || dir == "/mnt";
#endif
        for(int t = 0; t < ffi.GetCount(); t++) {
            const FileSystemInfo::FileInfo& fi = ffi[t];
#ifdef PLATFORM_POSIX
            Image img = fi.is_directory ?
                (isdrive ? PosixGetDriveImage(fi.filename) : CtrlImg::Dir()) :
```

```

    CtrlImg::File();
#else
    Image img = fi.is_directory ? CtrlImg::Dir() : CtrlImg::File();
#endif
    WhenIcon(fi.is_directory, fi.filename, img);
    bool nd = dirs && !fi.is_directory;
    if(fi.filename != "." && fi.filename != ".." != 0 && fi.filename.Find('.') != 0 &&
        (fi.is_directory || PatternMatchMulti(patterns, fi.filename)))
        list.Add(fi.filename, img,
            fi.is_directory ? Arial(FNTSIZE).Bold() : Arial(FNTSIZE),
            nd ? SGray : SBlack, fi.is_directory, fi.is_directory ? -1 : (int)fi.length,
            Null, nd ? SGray : fi.is_directory ? SBlack : SLtBlue);
    }
}
return true;
}

```

(I made it hide dotfiles also - maybe open/save dialog needs a show-hidden toggle for POSIX)

Still, the drives DropList doesn't get populated.

But this code snipped as meant as demonstration only- it's really a hack. FileSel needs to be rewritten, allowing for the "conceptual root" of POSIX systems, no longer assuming all drives are to be found in "/" flat.

Sorry if code snipped comes out distorted!
How do I do that properly?

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
