

Suggestion for Common Third Party Install system

Source: <http://linux.derkeiler.com/Mailing-Lists/Debian/2003-08/2857.html>

From: Edward Murrell (edward_at_murrell.co.nz)

Date: 08/16/03

To: debian-user@lists.debian.org

Date: Sun, 17 Aug 2003 00:48:39 +1200

Recently, I have been working over strategies for Linux on the desktop. In particular, I've been comparing usability of system administration tasks to Windows 2K/XP and Mac OS X, the main contenders on the desktop.

System administration tasks are done by almost everybody who owns a computer. It's not only things like applying security patch's, but includes mundane tasks like cleaning out old files, changing the dial-up number for your ISP, and installing and removing software.

The last point is a bit of contention under Linux. Sure, apt-get, urmpi, and the like make it easy to get and install the latest open source software from your favourite vendor, but they don't cover 'other' software. Third party, and closed source software is somewhat more difficult.

Third party applications are tricky. If a vendor wishes to support Linux, they currently have two options – to include packages for all the major distributions, or to roll their own install/uninstall system, often in the form of a script. If the vendor takes the first option, there are at least four major package formats to support (apt, rpm, emerge, and slackware tgz), with multiple sub-versions of those. If the vendor wishes to support more than just i386, or wishes to provide native support for say, Itanium and Opteron CPUs as well, the number of packages multiply even further.

The second option shifts the burden to the user. At first, it seems reasonable enough. A piece of software installs, with it's own script, probably into /usr/local/. A directory in /etc/, and files in the various users /home/ directories are created. Maybe the script also has the facility to clean up after itself. But where is this script kept? Perhaps on the install CD, or in the /etc/ directory, or with the program in /usr/local/application/, or was that /usr/local/application/bin/ ?

At first glance, this doesn't seem like an insurmountable problem. Most Linux or *nix software is open source, and files are kept nicely controlled by the package manager. Just how many pieces of closed source

Debian–User: Suggestion for Common Third Party Install system

software could there be on a Linux system anyway?

A lot. Personally, I already have the following installed; Quake 3 Arena, Team Arena, Real Player, Adobe Acrobat Reader, Unreal Tournament 2003, Enemy Territory, Return to Castle Wolfenstein. If Adobe ever releases Photoshop for Linux, I'll probably buy a copy of that as well. Add a couple of things like VMWare, maybe WineX, a video production package, and some random compiled from source packages, and it starts to get out of control. All these programs have their own way of doing things. I could drop everything into `/usr/local/`, delete it when I'm done, and hope there's not too much cruft left lying around the system.

The problem is that there will be cruft left lying around the system, and after a while, cruft builds up. Some of it possibly in horrible places where it's going to affect other things. You could argue that developers should release packages for your preferred distribution. You could argue that they should release the source code, and that I shouldn't install dodgy software that doesn't stick to `/usr/local/` and provide an uninstall script to remove the various files hanging around `/etc/` and my home directory. You could argue that end users should learn to use `/usr/local/`, `/opt/`, and install/removal scripts. You could argue a lot of things, but the world doesn't work like that. End users need a system that allows their file manager to handle the install and removal of third party applications without bothering them with the details.

One possibility, that I put forward here, is for a simple install and uninstall system, based around symlinks, and a common applications directory.

The application, to all outward appearances in most common file browsers, would appear as a file, but it's really a directory. Let us assume the application is called 'Foo'. Foo consists of binary files, supporting libraries, icons, configuration files usually found in `/etc/`, and so on. We can express most of these in terms of where they normally fit in the unix file system. A few, mainly icons, and programs shortcuts, vary from environment to environment.

The 'Applications' folder is accessible from the users Control panel as a special directory, allowing write access with `sudo` and the root password. The folder itself would most likely sit in somewhere like `/usr/local/Applications/`. Thus, our application 'Foo', would be in `/usr/local/Applications/Foo/`. It's file structure would look something like this:

```
/usr/local/Applications/Foo/  
/usr/local/Applications/Foo/icon.svg  
/usr/local/Applications/Foo/description.en.txt  
/usr/local/Applications/Foo/tree/linux/i686/usr/bin/foo  
/usr/local/Applications/Foo/tree/linux/i686/usr/lib/foo/libfoo.so  
/usr/local/Applications/Foo/install/etc/foo/foo.conf  
/usr/local/Applications/Foo/home/.foo/
```

Debian–User: Suggestion for Common Third Party Install system

To install, symlinks from are made from `/usr/bin/foo` and `/usr/lib/foo/` and `/usr/local/Applications/Foo/tree/linux/i686/usr/lib/foo/` to `/usr/local/Applications/Foo/tree/linux/i686/usr/bin/foo`, and to `/usr/local/Applications/Foo/tree/usr/lib/foo/`. A copy of `/usr/local/Applications/Foo/install/etc/foo/foo.conf` is made to `/etc/`, so that changes made to the configuration do not affect the original files of the program.

At the most basic level, `Foo/` needs to be copied from it's original source to `/usr/local/Applications/` and a script run to copy files in `Foo/install/` and make symlinks of the files in `Foo/tree/`. However, this simple design allows for a great flexibility. Multiple versions for multiple platforms can be held inside one directory, along with a complete listings of files that have been 'installed'. If required, you can make a copy of the various files to the normal `/usr/local/` and `/opt/` directories. If the environment flat out doesn't support the standard, you can create the symlinks yourself.

On a user–friendly distribution/file manager, attempting to delete the 'program' from the Applications directory would prompt the user if they wish to 'uninstall'. The computer would then remove the symlinks and any configuration files. It could also allow the uninstall program to ask if the user wishes to remove the system configuration files, originally copied from the the `Foo/install/` directory, and user configuration files, originally copied from the `Foo/home/` directory. This would be handy for say, games, where you want to remove the game from your system, but wish to keep the saved games and personal configuration options.

The file `Foo/icon.svg` can replace the standard folder icon, and `Foo/description.txt` can contain multiple language descriptions of the program, to be read by any program used to handle the installation and removal of applications. On more user–friendly systems, double clicking on the icon for the program on a CD would prompt the user if they wish to install the program.

There are other benefits to this method. Source installed packages, for example, `Mplayer`, instead of being installed directly into the file system, is instead copied into `/usr/local/Applications/mplayer/`, and installed in the above manner. `Mplayer` can then be copied easily to other systems, and removed easily when the version gets into your preferred package system. Automatic detection of CDs with these applications directories in them, possibly along with an autorun style file, would simplify installation even further. It is also possible that normal, non–root users could run these applications directly from their home directories, without needing to 'install' them. A plugin module for `apt` and `rpm` would allow existing free software managers such as `synaptic` and `Red Carpet` to also manage these applications.

Platforms which break with the standard filesystem, such as `GoboLinux`, can still work with this standard, and link files from the `tree/*/*/bin/`

Debian–User: Suggestion for Common Third Party Install system

directory to their preferred location.

I will emphasise again that this system is not meant to replace things like apt–get and rpm, but sit alongside them, and provide a common simple platform independent way of installing third party software.

While passing this idea around, I have come up against a few reoccurring 'complaints', some of which I have taken on and added to the above, and others which I'll try to address here to avoid repetition.

Q1) Apt/emerge/rpm already supports all the third party apps, either directly, or through meta packages.

A1) While currently, most packages are available in the above manner, as the popularity of free operating systems grows, more and more applications will be written for them. If Linux/BSD eventually has even a tenth of the number of Windows Applications on the shelves, package maintainers simply will not be able to keep up. It is also unlikely that brand new release purchasers of Half–Life 3 would want to wait for the Debian or Gentoo maintainers to release a package, simply so they can play the game.

Q2) I don't use closed source software, and neither should you!

A2) For better or worse, closed source software is very much here to stay. While you are certainly welcome to stick to free software, some of my work requires that I used software that is only available in closed source form.

Q3) Users / Vendors should just use apt/emerge/rpm.

A3) It is also unlikely that one platform and/or package manager will emerge as the 'one true manager' for quite some time. In the mean time, it would be nice if I could install and manage the third party software on my system with a minimum of fuss. While it's nice to get packages in my systems preferred format I'm not sure that it's going to be the same format in five years time. However, the third party software may still be loaded off the same CD in five years time.

Q4) Users should simply copy straight to the /usr/local/ directory.

A4) I should probably know how to replace, maintain, and repair all the major components in my car, but I don't, because my mechanic does. Likewise, the average user shouldn't need to know about something the computer is capable of taking care of.

Q5) You stole this from Apple!

A5) While the idea is very much inspired by Apple's bundles, you can't 'steal' an idea. And besides, they 'stole' it from NeXT OS. ;)

At this point, I'd mostly appreciate technical arguments about why it wouldn't work, or what could be done to improve it. Flames can be reserved for the final specification.

--

Debian–User: Suggestion for Common Third Party Install system

To UNSUBSCRIBE, email to debian-user-request@lists.debian.org
with a subject of "unsubscribe". Trouble? Contact listmaster@lists.debian.org