Subject: [PROPOSAL] Changes to packaging scheme
Posted by dolik.rce on Sun, 29 May 2011 11:41:28 GMT
View Forum Message <> Reply to Message

Hi everyone

I would like to propose a couple of changes in the packaging scheme and hear your opinions. I will talk mainly about ubuntu packages and arch PKGBUILDs, which I maintain, but the ideas are applicable to any packaging system.

As of now, we produce three packages: two packages for theide (normal and nogtk version) and 'upp' for all the sources. I would like to split 'upp' into several smaller packages. The reason is simple, not all parts of U++ are changed equally often. For illustration, last changed revisions of the top level directories in svn trunk: for r in trunk/*; do echo "$r: `svn info $r | sed -n 's/Last Changed Rev: \(.*\)/\1/p;'`"; done;# bold directories are those that are usually shipped to end users
archive: 1997
bazaar: 3464
benchmarks: 958
examples: 3375
reference: 3453
tutorial: 1700
uppbox: 3467
uppdev: 3418
uppsrc: 3467
uppsrc2: 1175
upptst: 2925As you can see, some of them are seldom updated, so downloading all of them each time is quite an overkill. So I propose to split the current upp package into several smaller ones (upp-bazaar, upp-uppsrc, upp-reference, ...). To ensure a) fluent transition and b) easy handling for users, there would be still an 'upp' package, but it would carry no files, just dependency on all the upp-* packages (i.e. what's called meta-package in most package managers).

The advantages of the proposed changes:
- Faster updates, smaller download sizes
- Possible to explicitly install only packages you want (e.g. only uppsrc)
- Less work for the packaging server (we can build packages only for parts that changed).

The possible disadvantages:
- More packages
- Possibly confusing for user

I am not sure if the first one could affect anyone negatively, but I listed it as possible problem, just to be fair. As for the second, it should be IMHO easily solved by big bold warning (in each packages description and on download page) saying that the best choice is to install 'upp' and let package manager do the rest of the work.

I plan to do some upgrades in the uppbox/lpbuild packaging scripts in the near future, so if there is a general agreement with this proposal, I would incorporate it as well. So what are your opinions? Are there any serious drawbacks that I overlooked?

Best regards,
Honza

Subject: Re: [PROPOSAL] Changes to packaging scheme
Posted by koldo on Sun, 29 May 2011 13:17:05 GMT
View Forum Message <> Reply to Message

Hello Honza

IMHO I would not divide the packages. I am also confused with the nogtk version ...

Perhaps I would do a full file with all inside (TheIde, U++ and all packages), and a list of smaller files with packages only used by skilled U++ users.

Subject: Re: [PROPOSAL] Changes to packaging scheme
Posted by dolik.rce on Sun, 29 May 2011 15:19:35 GMT
View Forum Message <> Reply to Message

Hi Koldo,
Thanks for feedback.koldo wrote on Sun, 29 May 2011 15:17Perhaps I would do a full file with all inside (TheIde, U++ and all packages), and a list of smaller files with packages only used by skilled U++ users.Like "sudo apt-get install upp"? That works now (installing upp + theide) and would work with the proposed changes as well (just installing theide, upp-uppsrc, upp-bazaar, upp-tutorial, upp-examples and upp-reference). From the point of novice user nothing changes.

BTW: The splitting of packages to smaller self-consistent units is quite common (see for example libboost in ubuntu) and also encouraged in the debian packaging manual IIRC...

Honza

Subject: Re: [PROPOSAL] Changes to packaging scheme
Posted by koldo on Mon, 30 May 2011 06:26:10 GMT
View Forum Message <> Reply to Message

Hello Honza

If "sudo apt-get install upp" works, it is perfect. It is what some person beginning in U++ would expect.

All the additional packages are ok too.

Subject: Re: [PROPOSAL] Changes to packaging scheme
Posted by mr_ped on Thu, 02 Jun 2011 07:00:09 GMT
View Forum Message <> Reply to Message

I agree with suggested split.

Subject: Re: [PROPOSAL] Changes to packaging scheme
Posted by kohait00 on Tue, 12 Jul 2011 07:46:20 GMT
View Forum Message <> Reply to Message

though splitting can save some download time it adds to complicity.

see the nvidia graphiccard driver. 200MB building up all the drivers for all the cards ever produced. but one knows: 1 file, all there.

download speed / time is matter of convenience. but simplicity when setting up the building environment is a key feature.

i'd rather prefer to 'apt-get install upp' and have all there..
this is true for releases.. which is also a problem, because the sources have a version interdependancy..exchanging them at will without managing the dependncy properly opens door to broken releases. so why the hassle? leave'm together and you know it'l work..

if splitting, i'd rather go for:

upp Base package: uppsrc, rainbow
upp Goodies package: reference,tutorial,examples,bazaar

the others are mainly for upp developers. and they probably know how to set up the env, so this could be left out.

upp Dev package (not to be confused with the *-devel packages): uppdev, archive, benchmark, uppbox, uppsrc2, upptst

but i think there is still lacking a docu on how to set up a bleeding edge building environment for active development in and with upp, means setup svn checkout, managing upp.out and building methods, 3rd-party installations, additional compilers, TDMGCC, MINGW...custom buildsteps (for wichi there is IMHO 0 doc available, thus i haven't used it at all).

so maybe first update / reinforce on documentation...

Subject: Re: [PROPOSAL] Changes to packaging scheme
Posted by dolik.rce on Tue, 12 Jul 2011 11:11:32 GMT
View Forum Message <> Reply to Message

Hi kohait,

kohait00 wrote on Tue, 12 July 2011 09:46though splitting can save some download time it adds to complicity. It's not that complex E.g. for ubuntu/debian it is still single source package, just having few more sections.

kohait00 wrote on Tue, 12 July 2011 09:46download speed / time is matter of convenience. but simplicity when setting up the building environment is a key feature.

i'd rather prefer to 'apt-get install upp' and have all there..As I said above, this command will work exactly as before, resulting in complete installation of theide, uppsrc, all the examples etc.

kohait00 wrote on Tue, 12 July 2011 09:46this is true for releases.. which is also a problem, because the sources have a version interdependancy..exchanging them at will without managing the dependncy properly opens door to broken releases. so why the hassle? leave'm together and you know it'l work..I'm not sure if you are underestimating the packaging systems or my ability to specify the correct dependencies in the packages I have packaged enough software (we have debian servers at work) to be confident that I can make this work 100% correct.

kohait00 wrote on Tue, 12 July 2011 09:46if splitting, i'd rather go for:

upp Base package: uppsrc, rainbow
upp Goodies package: reference,tutorial,examples,bazaar

the others are mainly for upp developers. and they probably know how to set up the env, so this could be left out.

upp Dev package (not to be confused with the *-devel packages): uppdev, archive, benchmark, uppbox, uppsrc2, upptst

These are possible options as well. The choice is just a matter of taste. My taste (and also that of debian policy manual) is that the packages should be broken to the smallest possible logical units, which corresponds to a nest in this case.

kohait00 wrote on Tue, 12 July 2011 09:46but i think there is still lacking a docu on how to set up a bleeding edge building environment for active development in and with upp, means setup svn checkout, managing upp.out and building methods, 3rd-party installations, additional compilers, TDMGCC, MINGW...custom buildsteps (for wichi there is IMHO 0 doc available, thus i haven't used it at all).

so maybe first update / reinforce on documentation...
The documentation is not really related to this topic, but anyway: Most of it is documented, the problem is IMHO in the organization of the docs. E.g. custom build steps are explained in Configuring Packages and Assemblies, which I agree to be a bit unexpected, but easily found using google search field, which is conveniently located on each page of this site.

Best regards,
Honza

Subject: Re: [PROPOSAL] Changes to packaging scheme
Posted by kohait00 on Tue, 12 Jul 2011 11:23:45 GMT
View Forum Message <> Reply to Message

i'm not underestimating your skills,
you've just described it better this time.. i really thought having it sit in different packages..

in this case, IMHO go ahead. the question is, merely, is it worth your trouble, to save some folders with some text files..

Subject: Re: [PROPOSAL] Changes to packaging scheme
Posted by dolik.rce on Tue, 12 Jul 2011 11:46:25 GMT
View Forum Message <> Reply to Message

kohait00 wrote on Tue, 12 July 2011 13:23you've just described it better this time.. i really thought having it sit in different packages..
Just for clarity: There is two types (well, actually more, but that is not important now) of debian packages: source packages and binary packages. Source package .dsc is what packager sends to builder (lpbuild/distro). Binary package .deb is what end user usually installs on his machine. We would have one .dsc, which is built into several .debs.

The dependencies will than take care about installing all the packages when installing 'upp'.

Very similar holds also for Arch Linux and probably most of other packaging systems.

Honza