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