
Subject: Splitting debs into two parts

Posted by [dolik.rce](#) on Wed, 07 Apr 2010 17:31:23 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello all,

I believe that there should be two debian packages. One architecture dependent, containing theide and the other should contain sources, as they are architecture independent.

I am aware that some of you might not like this idea, but there are two strong reasons to do so:

Size. Now, every build produces 14 packages (~23MB each) plus source package (~20MB). That makes almost 350MB every night. If we split the package into two, it will be about 170MB. I think that using the launchpad resources responsibly is kind of good etiquette...

Standards. Debian policy manual actually states, that large amounts of architecture independent files should be distributed in separate package. If we want to get into debian or ubuntu, we have to adhere to their rules as much as possible.

+ It supports the idea from index page: "You can use both, or you can use whichever you need."

Technical details: Packages would be connected through dependency, so for the end user nothing changes. The transition will be very smooth, just one update will pull two packages instead of one. I think best names would be "upp" for sources and "upp-ide" for theide.

What is your opinion on this?

Best regards,
Honza

Subject: Re: Splitting debs into two parts

Posted by [mdelfede](#) on Wed, 07 Apr 2010 18:14:01 GMT

[View Forum Message](#) <> [Reply to Message](#)

mhhhh... I don't see a package containing just executable very useful, as theide without uppsrc source tree has no purposes.

We could eventually make a sources-only package with makefile but then, also, without theide won't be so useful, either....

So, being true that you could spare the source overhead on every platform dependent executables, users would be forced to download and install both of them.

Ciao

Max

Subject: Re: Splitting debs into two parts

Posted by [nneilson](#) on Wed, 07 Apr 2010 19:01:37 GMT

[View Forum Message](#) <> [Reply to Message](#)

dolik.rce wrote on Wed, 07 April 2010 19:31 Standards. Debian policy manual actually states, that large amounts of architecture independent files should be distributed in separate package. If we want to get into debian or ubuntu, we have to adhere to their rules as much as possible.

"upp" for sources and "upp-ide" for theide.

Seems like a good step forward.

Since Ubuntu has such a large user base now maybe an installer could be implemented similar to the .exe for Windows, pre compiled/built.

Getting Upp into Ubuntu would increase the visibility/usage and maybe financial support.

Subject: Re: Splitting deps into two parts

Posted by [mr_ped](#) on Thu, 08 Apr 2010 15:59:48 GMT

[View Forum Message](#) <> [Reply to Message](#)

I agree we should split, but I think the proposal is not ideal.

But I can't improve it ... thinking about it... we are missing "lib" version.

libupp - binary .so + headers

libupp-src - uppsrc with core classes

upp - IDE + other binary stuff (should be able to work with libupp?) Maybe even precompiled best examples like Uword and such?

upp-src - ide/tools sources

upp-bazaar-src - bazaar sources

...blah blah...

this simply doesn't make sense, with the modular u++ build you must have sources, because you can't precompile library for all possible flags combinations.

So I'm posting it here as a warning, that this way doesn't look good either.

Maybe some upp-noarch with sources and other architecture independent stuff.... and upp[-archXX] for the rest would make most sense in the end.

Subject: Re: Splitting deps into two parts

Posted by [mr_ped](#) on Thu, 08 Apr 2010 16:01:25 GMT

[View Forum Message](#) <> [Reply to Message](#)

still in case of update you will in 99% cases download both packages anyway, so this will save just PPA space. (but worth of it)

Subject: Re: Splitting deps into two parts
Posted by [dolik.rce](#) on Thu, 08 Apr 2010 23:17:18 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,
Thanks for feedback. I agree with most of what you said.

Quote:mhhhh... I don't see a package containing just executable very useful, as theide without uppsrc source tree has no purposes.
This is not really true. Theide can be used (and I do it sometimes) to compile C/C++ code only, without any U++. You just need to add the sources in a package. The other way around is hard - I'm not sure if it is even possible to use the code with different IDE...

Quote:I agree we should split, but I think the proposal is not ideal.

But I can't improve it ... thinking about it... we are missing "lib" version.
Yes I know it is not ideal. I was hoping that someone would post some better ideas here

Quote:upp - IDE + other binary stuff (should be able to work with libupp?) Maybe even precompiled best examples like Uword and such?
I would say compiled examples should be in separate package. Technically it is easy to do. I could even imagine building reference, tutorial and examples nests. It would give as great quality control tool. When any of the demonstration code fails to build, we would know it next morning. There is nothing more discouraging than nonworking official examples...

I don't like the idea of having many packages as it makes it confusing for users. But maybe the code part could be separated into uppsrc+reference, bazaar and the examples+tutorial.

As most of you said, the only direct benefit is the reduced size. Even the rules in Debian policy manual that I mentioned in the first post are designed to save the repository space. But I think it won't hurt anybody, so there is no real reason not to split the deps. Also it might bring some small bonuses for advanced users. E.g. with separate sources and theide, you can use nightly build of theide while keeping stable release of sources (using "hold" in apt/aptitude/synaptic).

Also one additional idea: What about NOGTK builds? I would be the first to use that, as my ubuntu box has very small hdd and I don't have all the GTK stuff installed (I'm building my theide manually...). Probably there are some also other people who would appreciate that for whatever reasons.

Any more comments or ideas?

Best regards,
Honza

Subject: Re: Splitting deps into two parts
Posted by [mdelfede](#) on Fri, 09 Apr 2010 11:09:05 GMT
[View Forum Message](#) <> [Reply to Message](#)

dolik.rce wrote on Fri, 09 April 2010 01:17Hi,
Thanks for feedback. I agree with most of what you said.

Quote:mhhhh... I don't see a package containing just executable very useful, as theide without uppsrc source tree has no purposes.

This is not really true. Theide can be used (and I do it sometimes) to compile C/C++ code only, without any U++. You just need to add the sources in a package. The other way around is hard - I'm not sure if it is even possible to use the code with different IDE...

Well, I think Theide is an accessory part to the Ultimate library, not the way around. Installing Theide just to compile non-upp code, even if practicable, makes not much sense, imho.
Well, this is my opinion, of course. I'd not install theide alone just to compile c/c++ code, and I don't think many people would do it either, being theide not very user friendly at the beginning. To just compile c++ code I'd do it with codeblocks, kdevelop, anjuta, free M\$ stuffs...

Ciao

Max

Subject: Re: Splitting deps into two parts
Posted by [dolik.rce](#) on Fri, 09 Apr 2010 13:09:52 GMT
[View Forum Message](#) <> [Reply to Message](#)

mdelfede wrote on Fri, 09 April 2010 13:09Well, I think Theide is an accessory part to the Ultimate library, not the way around. Installing Theide just to compile non-upp code, even if practicable, makes not much sense, imho.

Well, this is my opinion, of course. I'd not install theide alone just to compile c/c++ code, and I don't think many people would do it either, being theide not very user friendly at the beginning. To just compile c++ code I'd do it with codeblocks, kdevelop, anjuta, free M\$ stuffs...

Ciao

Max

Yes, it is not very common to use theide alone. But that might change, if it will be in debian/ubuntu repository one day... Especially if we advertise a bit that it is possible to use it with other languages as well (java should work out of the box and many others can be done via scriptbuilds; also I remember someone claimed to use theide for PHP development).

Anyway, this is not the point of this discussion. The topic is how to best split the package to adhere to the packaging policies.

Honza

Subject: Re: Splitting deps into two parts
Posted by [nneilson](#) on Sun, 11 Apr 2010 22:02:50 GMT
[View Forum Message](#) <> [Reply to Message](#)

Quote:1. "... (java should work out of the box ...)"

2. Anyway, this is not the point of this discussion. The topic is how to best split the package to adhere to the packaging policies.

3. ... how to best split the package...

1. Interesting, are there any examples that show this?

2. Splitting theIDE from the source may work like many apps that are in the Ubuntu Synaptics. Selecting an item will automatically select the items it depends on, the user has the option of deselecting any of the additional items.

For a user that wants to try Upp then theIDE would be selected also.

An additional advantage would be if either the library or theIDE is up to date then only what needs to be updated will be downloaded.

One very important thing for a new user would be to have theIDE as an executable, 6.7 MB, rather than have them build it, that is just an extra hurdle that many may not even try or have problems with.

Looking at this from a different perspective, would it make sense to do away with the Windows installer and require a user to build theIDE. I don't think that would be a good idea.

Why make a Linux user build it??

3. Split theIDE from the source/library.

Send a proposal to whoever makes the decisions for what is to be included in Debian, whatever they accept or recommend seems to be the "ideal" way.

Subject: Re: Splitting deps into two parts
Posted by [dolik.rce](#) on Mon, 12 Apr 2010 06:21:10 GMT
[View Forum Message](#) <> [Reply to Message](#)

nneilson wrote on Mon, 12 April 2010 00:02Quote:1. "... (java should work out of the box ...)"

2. Anyway, this is not the point of this discussion. The topic is how to best split the package to adhere to the packaging policies.

3. ... how to best split the package...

1. Interesting, are there any examples that show this?
2. Splitting theIDE from the source may work like many apps that are in the Ubuntu Synaptics. Selecting an item will automatically select the items it depends on, the user has the option of deselecting any of the additional items.

For a user that wants to try Upp then theIDE would be selected also.

An additional advantage would be if either the library or theIDE is up to date then only what needs to be updated will be downloaded.

One very important thing for a new user would be to have theIDE as an executable, 6.7 MB, rather than have them build it, that is just an extra hurdle that many may not even try or have problems with.

Looking at this from a different perspective, would it make sense to do away with the Windows installer and require a user to build theIDE. I don't think that would be a good idea.

Why make a Linux user build it??

3. Split theIDE from the source/library.

Send a proposal to whoever makes the decisions for what is to be included in Debian, whatever they accept or recommend seems to be the "ideal" way.

1. There is JDK builder in Build methods. There is probably no documentation but you can have a look at `uppsrc/ide/Builders/JavaBuilder.cpp`.

2. Yes, that is what I mentioned in first post (in technical details). It is unlikely that only one of the packages would be updated, it would quite difficult technically. To provide binary instead of letting the user build theide is the whole idea of debian packages. But it has nothing in common with windows installer.

3. Source + application is the obvious choice, but there could be also finer-grained solutions (see [mr_peds](#) post above). Just for clarification: We are not in Debian distribution yet, we just publish packages in Debian format (actually only meant for Ubuntu right now).

Honza

Subject: Re: Splitting deps into two parts
Posted by [mr_ped](#) on Mon, 12 Apr 2010 08:15:33 GMT
[View Forum Message](#) <> [Reply to Message](#)

thinking about it more, maybe the best way is to keep it simple.
upp = architecture dependent files

upp-data = the rest (sources, resources, whatever else)

It means that when you want to install upp on fresh buntu, you will have to download full 20+MB or how much it is, then again that's just minor part of total download if there is not build-essentials+gcc installed yet.

It will hurt a bit during updates as well, but I don't think finer splitting will save that much updates, and it will make the platform less consistent (in case somebody runs into problem, we will have to figure out which packages did he install, and which ones did he omit, etc.).

So in the end I think it should be "take it all or nothing (download sources and build whatever you wish)", and two packages: binaries + data. This simplicity follows U++ spirit imho well.

Edit (more explanation): I think the "source" package doesn't make sense, the source is the very much base part of upp itself, so it doesn't make sense to download IDE binary without sources. (for extreme usecases it makes sense, but the packages should be targeted at "ordinary" users who want to create cross-platform C++ GUI application) ... that's why I'm suggesting "-data" package (although it will consist from 95% of C++ sources), to make it clear it's part of upp and the two can't exist without each other.

Subject: Re: Splitting deps into two parts

Posted by [dolik.rce](#) on Tue, 13 Apr 2010 05:17:48 GMT

[View Forum Message](#) <> [Reply to Message](#)

This might be a daring idea, but what about naming the platform dependent package (i.e. theide binary) just "theide"?

It might be good for PR... If someone searches for IDE (in synaptic, aptitude etc.), it would be one of the results. If it catches his attention, he will find out about U++ as well very soon.

The other package could then be just "upp" (which is fine, as the sources really are the body of U++) and depend on theide.

Quote:It will hurt a bit during updates as well, but I don't think finer splitting will save that much updates, and it will make the platform less consistent (in case somebody runs into problem, we will have to figure out which packages did he install, and which ones did he omit, etc.). I agree. Plus someone would have to write some ugly code to determine what to update

Honza

Subject: Re: Splitting deps into two parts

Posted by [nneilson](#) on Wed, 14 Apr 2010 02:20:27 GMT

[View Forum Message](#) <> [Reply to Message](#)

dolik.rce wrote on Tue, 13 April 2010 07:17 "theide" + "upp"

Quote: (in case somebody runs into problem, we will have to figure out which packages did he install, and which ones did he omit, etc.).

"theide" + "upp"

This seems like the way to go.

When a user updates both would be updated.

The only reason I can think of for someone not to update both is if they have a very slow internet connection.

"in case somebody runs into problem"

If someone has a problem and has removed a package they should replace it to see if the problem persists.

"theide" + "upp"

Good!!

Subject: Re: Splitting deps into two parts
Posted by [dolik.rce](#) on Wed, 14 Apr 2010 12:50:18 GMT
[View Forum Message](#) <> [Reply to Message](#)

At the end of this post is a proposal for the new packages control file. It should be readable even if you are not familiar with the details of the syntax. Just note that the first paragraph describes the dsc package (which is then built into the three packages listed below) and that `$(shlibs:depends)` is expanded during the build to list of libs linked to the binary.

The main idea is following:

Architecture independent package upp holds the U++ sources. It depends on theide, so it is always installed. The dependency can be also satisfied with theide-nogtk. Only one of those can be installed, but they can be switched whenever you want. If you install the other option, the user settings will persist. Theide can be installed without upp if someone wishes.

There is also one more option how to handle the theide vs. theide-nogtk: They could be installed both at the same times, if they used different names for the binary. Each would then have separate configuration. I don't think anyone needs two identical tools (only visually different) installed at the same time, so I prefer the first option listed above.

The description of theide packages was taken from the index page. I hope they make sense :-)

Any comments are welcome.

Here is the control file:

Quote:Source: upp

Section: devel

Priority: optional

Maintainer: Jan Dolinar <dolik.rce@seznam.cz>

Standards-Version: 3.8.4

Build-Depends: g++ (= 4.1) | g++ (>= 4.3), rsync, libx11-dev (>= 1.0.0), libxft-dev (>= 2.1.8), libpng12-dev (>= 1.2.8), libgtk2.0-dev (>= 2.8.17), libnotify-dev

Homepage: <http://www.ultimatepp.org>

Package: upp

Section: devel

Priority: optional

Architecture: all

Depends: theide, g++ (= 4.1) | g++ (>= 4.3), libx11-dev (>= 1.0.0), libxft-dev (>= 2.1.8), libpng12-dev (>= 1.2.8), libgtk2.0-dev (>= 2.8.17), libnotify-dev

Homepage: <http://www.ultimatepp.org>

Description: C++ library for cross-platform development

Ultimate++ is a radical and innovative GUI toolkit whose number one priority is programmer productivity. C++ is a great programming language but C++ programmers are sometimes hampered by the lack of effective libraries. U++ libraries enable genuine productivity gains with shorter development times and greatly reduced application source code size.

Package: theide

Section: devel

Priority: optional

Architecture: any

Depends: \${shlibs:Depends}

Homepage: <http://www.ultimatepp.org>

Description: IDE (not only) for Ultimate++ framework

TheIDE introduces modular concepts to C++ programming. It features BLITZ-build technology to speedup C++ rebuilds up to 4 times, Visual designers for U++ libraries, Topic++ system for documenting code and creating rich text resources for applications (like help and code documentation) and Assist++ - a powerful C++ code analyzer that provides features like code completion, navigation and transformation.

Package: theide-nogtk

Section: devel

Priority: optional

Architecture: any

Depends: \${shlibs:Depends}

Provides: theide

Replaces: theide

Conflicts: theide

Homepage: <http://www.ultimatepp.org>

Description: IDE (not only) for Ultimate++ framework

TheIDE introduces modular concepts to C++ programming. It features BLITZ-build technology to speedup C++ rebuilds up to 4 times, Visual designers for Ultimate++ libraries, Topic++ system for documenting code and creating rich text resources for applications (like help and code documentation) and Assist++ - a powerful C++ code analyzer that provides features like code completion, navigation and transformation.

This package contains TheIDE built with simpler look to reduce

package dependencies.

Subject: Re: Splitting deps into two parts
Posted by [nlneilson](#) on Wed, 14 Apr 2010 13:37:43 GMT
[View Forum Message](#) <> [Reply to Message](#)

This looks like three packages rather than two.

1. upp
2. theide
3. theide-nogtk

Wouldn't it be easier and simpler to have just two.

For those that are concerned about disk space the gtk dependencies could be removed with a simple script.

theide-nogtk would need to be built by the user but what percentage of users would need/want this? I am not familiar with this so I may be wrong.

Subject: Re: Splitting deps into two parts
Posted by [dolik.rce](#) on Wed, 14 Apr 2010 15:04:46 GMT
[View Forum Message](#) <> [Reply to Message](#)

nlneilson wrote on Wed, 14 April 2010 15:37This looks like three packages rather than two.

1. upp
2. theide
3. theide-nogtk

Wouldn't it be easier and simpler to have just two.

For those that are concerned about disk space the gtk dependencies could be removed with a simple script.

theide-nogtk would need to be built by the user but what percentage of users would need/want this? I am not familiar with this so I may be wrong.

Yes, it is tree packages. But only two of them are installed at any time. Note that usual user would install upp and theide would be installed as dependency. Advanced user has a choice to install nogtk version instead. Creating the nogtk version is just a matter of passing different flags to makefile. So no additional work for maintainer and no additional problems for end user.

You are right that anyone can compile his own theide, but it requires time and also knowledge how to do it. I have no idea how many people does use this. My guess is less than 5%... I know about one for sure - myself I like to have nice clean system without unnecessary GTK libs and I even prefer the Classic look. It is pretty tedious to compile theide on both of my computers every time some new feature was implemented. You might say I'm biased and you would be right

Honza

Subject: Re: Splitting deps into two parts
Posted by [dolik.rce](#) on Fri, 23 Apr 2010 16:32:31 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello all!

I finished the script changes, so now lpbld produces separate packages for theide and upp, as I proposed above. There is a test build of revision 2332 in my testing repository. It would be nice if someone could spare a few minutes of their time and test them before I upload the scripts on the infrastructure server. You can add the testing repository to your system by appending deb <http://ppa.launchpad.net/dolik-rce/uppnightly/ubuntu> YOUR_UBUNTU_VERSION_HERE main to your /etc/apt/sources.list.

Alternatively, you can have a look at the packages via the web UI.

Best regards,
Honza

Subject: Re: Splitting deps into two parts
Posted by [dolik.rce](#) on Fri, 30 Apr 2010 23:29:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi everyone!

All work is done and new version of scripts is on the server, hopefully working correctly. Launchpad is a bit lagging because of the release of Lucid Lynx, but the packages should be ready in couple of hours.

Please let me know, if you experience any difficulties when updating or when using them.

Best regards,
Honza

UPDATE: It seems that it will be in days rather than hours... The i386 queue is six days long now...

Subject: Re: Splitting deps into two parts
Posted by [Beankyu](#) on Thu, 17 Jun 2010 03:39:47 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thanks a lot for ur sharing.

I works well, really
