Subject: Archlinux AUR
Posted by Eremiell on Sun, 23 Apr 2017 01:25:37 GMT
View Forum Message <> Reply to Message

Hi!

I found U++ only recently (being many years C++ developer and tutor) and decided to give it a try. Sadly, reporting issues with AUR packages for nightly and git builds caused them to be orphaned. (No reaction from stable build yet.)

I'm now considering for longer time to start maintaining some AUR packages, as I dwell into PKGBUILDs nearly daily, so I could as well put this skill up for some good.

So my first question would be if that's OK with you?

If so, I'd like to ask about several things that could be done on U++ side, that would simplify the packaging a lot.

1) the builds as are now are using "uppbox/lpbuild2/mkfile". Could this be included into nightlies (and stables), so it doesn't have to be pulled from github, which causes potential version mismatch? Or could this probably be completely replaced by another build mechanism present? (I'm asking the later one simply because I don't know. If so, please enlighten me.)

2) could at least nightlies emit hash file(s), that could be grabbed for automatic checking? (Some sha256sums and sha512sums would be nice. Similar to say < h**ps://download-installer.cdn.mozilla.net/pub/firefox/night ly/latest-mozilla-central/firefox-55.0a1.en-US.linux-x86_64. checksums >, though a standalone file for each package and algo would be preferred.)

3) could at least nightlies emit some latest version file? Simply a text file, that contains nothing more that latest release number for quick and easy checking and also for automatic download and rebuild mechanism. (Similar to say <h**ps://www.hiawatha-webserver.org/latest>.)

4) I believe the previous two points would also help stable builds, though they are kinda far apart, so they're not that heavy to maintain.

Also the "#define DDUMP(x) @" in git is nasty, as it obviously emits errors, but they're sparse in a long build log that most people probably don't watch and while the build ends in error, the upp and umk packages still somehow "get done", only theide fails hard. But fixed that one with a simple sed. Just saying. Hope you don't plan to extend this strategy around.

Cheers,

Eremiell

P.S. Also I apparently "cannot use links until I have posted more than 1 message". Btw a dark theme and let's encrypt cert would do wonders.

Subject: Re: Archlinux AUR
Posted by mirek on Sun, 23 Apr 2017 17:37:46 GMT
View Forum Message <> Reply to Message

Eremiell wrote on Sun, 23 April 2017 03:25Hi!

I found U++ only recently (being many years C++ developer and tutor) and decided to give it a try. Sadly, reporting issues with AUR packages for nightly and git builds caused them to be orphaned. (No reaction from stable build yet.)

I'm now considering for longer time to start maintaining some AUR packages, as I dwell into PKGBUILDs nearly daily, so I could as well put this skill up for some good.

So my first question would be if that's OK with you?

Definitely. To dig into the issue a bit more, I believe that U++ developers responsibility should end at producing tarballs. Distro specific issues are currently being resolved by community and we provide links in Downloads pages just for convenience.

Quote:
If so, I'd like to ask about several things that could be done on U++ side, that would simplify the packaging a lot.

1) the builds as are now are using "uppbox/lpbuild2/mkfile". Could this be included into nightlies (and stables), so it doesn't have to be pulled from github, which causes potential version mismatch? Or could this probably be completely replaced by another build mechanism present? (I'm asking the later one simply because I don't know. If so, please enlighten me.)

2) could at least nightlies emit hash file(s), that could be grabbed for automatic checking? (Some sha256sums and sha512sums would be nice. Similar to say < h**ps://download-installer.cdn.mozilla.net/pub/firefox/nightly/latest-mozilla-central/firefox-55.0a1.en-US.linux-x86_64. checksums >, though a standalone file for each package and algo would be preferred.)

3) could at least nightlies emit some latest version file? Simply a text file, that contains nothing more that latest release number for quick and easy checking and also for automatic download and rebuild mechanism. (Similar to say <h**ps://www.hiawatha-webserver.org/latest>.)

All good points. Do not expect it to happend overnight, but I will put to this on high priority. If it is not done in next 14 days (and you are still around:), please poke me.

Quote:
Also the "#define DDUMP(x) @" in git is nasty, as it obviously emits errors, but they're sparse in a long build log that most people probably don't watch and while the build ends in error, the upp and umk packages still somehow "get done", only theide fails hard. But fixed that one with a simple sed. Just saying. Hope you don't plan to extend this strategy around.

DDUMP is doing exactly what it is supposed to do. If you have encountered that error, it more or less means this nightly is broken and you have to wait for next nightly.

Quote:
P.S. Also I apparently "cannot use links until I have posted more than 1 message".

Sorry. A couple of years back, we have spent a lot of time deleting spam. You will have to endure that. Just one more message to go

Mirek

---

Subject: Re: Archlinux AUR
Posted by Eremiell on Sun, 23 Apr 2017 20:03:54 GMT
View Forum Message <> Reply to Message

mirek wrote on Sun, 23 April 2017 19:37
Definitely. To dig into the issue a bit more, I believe that U++ developers responsibility should end at producing tarballs. Distro specific issues are currently being resolved by community and we provide links in Downloads pages just for convenience.

I mostly agree, though it's always good to know, you have some support, both theoretical as in "go on" and practical as in the few small changes I asked for. Mostly so that if some issues come up in the future (the build completely breaks, I need some minor meaningful adjustment like the checksums, or whatever) I can trust to have sensible ways of reporting and resolving the issue.

mirek wrote on Sun, 23 April 2017 19:37
All good points. Do not expect it to happend overnight, but I will put to this on high priority. If it is not done in next 14 days (and you are still around:), please poke me.

I will.

mirek wrote on Sun, 23 April 2017 19:37
DDUMP is doing exactly what it is supposed to do. If you have encountered that error, it more or less means this nightly is broken and you have to wait for next nightly.

Yes, I understand that. That was just a minor side note and an expression of a hopeful wish from packaging side of things that this (or similar) macro will remind contained in one place and not spread through the codebase. You should of course prioritize the development needs and ignore this wish completely should it be an obstacle to development. There's always more bash magic to fix stuff around for packaging.

I have an ugly deadline tomorrow, which I have to fulfill before going full crazy on this, but I have the core stuff working as is. Basically it downloads, compiles and packages alright for me, but I noticed some parts of the PKGBUILD are not in good shape, using potentially wrong, deprecated and misleading settings, so I want to polish it to conform the packaging rules first. (I may also be wrong with some of them. Let's say I need some time alone with it, the packaging guidelines and some tea.)

I'd like to find time to push the new builds tomorrow evening or during Tuesday. I'll let you know.

One more possible idea, which I expect to be more complicated and probably lower priority (but I'd love to be wrong):

I believe, the github repo is generated by some autonomous script and no one is actually mirroring that by hand, but if you were able to export some meaningful git tags for stable and nightly versions, that might be an interesting alternative for packaging too, especially for nightlies, which change often anyway. But tarballs, especially if you add the mkfile and checksums, are completely alright of course.

Cheers and seeya around,

Eremiell

---

Subject: Re: Archlinux AUR
Posted by dolik.rce on Mon, 24 Apr 2017 19:21:14 GMT
View Forum Message <> Reply to Message

Hi Eremiell!

Welcome to the U++ forum

Eremiell wrote on Sun, 23 April 2017 03:25I found U++ only recently (being many years C++ developer and tutor) and decided to give it a try. Sadly, reporting issues with AUR packages for nightly and git builds caused them to be orphaned. (No reaction from stable build yet.)

Maintainer of the AUR packages speaking here  I'm sorry I didn't reply right away when you flagged the package in AUR. Unfortunately, I'm rather busy lately and I must admit I didn't take proper care about the packages. Feel free to contact me directly if you're interested.

Eremiell wrote on Sun, 23 April 2017 03:25I'm now considering for longer time to start maintaining some AUR packages, as I dwell into PKGBUILDs nearly daily, so I could as well put this skill up for some good.

So my first question would be if that's OK with you?

I'd be more than happy to transfer those packages into good hands of another Arch user who will have more time to keep them up to date.

Eremiell wrote on Sun, 23 April 2017 03:25If so, I'd like to ask about several things that could be done on U++ side, that would simplify the packaging a lot.

1) the builds as are now are using "uppbox/lpbuild2/mkfile". Could this be included into nightlies (and stables), so it doesn't have to be pulled from github, which causes potential version mismatch? Or could this probably be completely replaced by another build mechanism present? (I'm asking the later one simply because I don't know. If so, please enlighten me.)
That file is a wonderful thing, but it is not really needed for the basic packaging. Less then a year ago, Amrein did a many fixes in the generated makefile that is included in the tarball when he was working on RPM packages. It should be fairly easy to use the regular makefile for Arch packages as well.

Points 2) and 3) were answered by Mirek already  And I agree with 4) as well.

Eremiell wrote on Sun, 23 April 2017 03:25Also the "#define DDUMP(x) @" in git is nasty, as it obviously emits errors, but they're sparse in a long build log that most people probably don't watch and while the build ends in error, the upp and umk packages still somehow "get done", only theide fails hard. But fixed that one with a simple sed. Just saying. Hope you don't plan to extend this strategy around.I think that it is actually pretty good idea. The only important part that is missing is that nightly builds should not be published if they don't pass the tests. Mirek do you think it would be possible to arrange that (if it's not already)?

Best regards,
Honza

---

Subject: Re: Archlinux AUR
Posted by Eremiell on Mon, 24 Apr 2017 23:05:06 GMT
View Forum Message <> Reply to Message

dolik.rce wrote on Mon, 24 April 2017 21:21
Maintainer of the AUR packages speaking here  I'm sorry I didn't reply right away when you flagged the package in AUR. Unfortunately, I'm rather busy lately and I must admit I didn't take proper care about the packages. Feel free to contact me directly if you're interested.

I acted mostly as upp-nightly and upp-git (and their umk and theide counterparts) became orphaned after me poking around. I still hoped for you to pop up. It's not my first round around flagging outdated packages, commenting on breakage and stuff not working and dropping wild PKGBUILD patches around. I got used to some maintainers acting in matter of minutes and other in days and weeks.

dolik.rce wrote on Mon, 24 April 2017 21:21
I'd be more than happy to transfer those packages into good hands of another Arch user who will have more time to keep them up to date.

It was more or less sheer pragmatism to volunteer, as I want these packages to work and I'm hacking around the PKGBUILDs already anyway, so why not share the work with others. I don't really wish to take the packages from you, if you're up to maintaining them further (talking mostly about stable now, the other two branches are unmaintained, orphaned and originated with fusion809 not you, but if you wish to take those under your control, I'd respect and support that).

On the other hand, if you feel you don't have time to maintain them, I could as well take care of all three branches, not much extra work.

I'm also ready to share the duty, so whoever notices the trouble first and has a slice of time fixes the fire.

As a side note, I'm designing the nightly and git channels as auto updating, grabbing new versions on fly without hand patching the PKGBULD in AUR unless something changes, similar to firefox-always-nightly AUR package or most of git packages around. The git package works so already after some fixing, as it previously read the AUR repo version and not the u++ repo one, so it never rebuilt. I'll fix nightly channel so once the checksums and version file are up, hand patched for now.

Still what I said before, I need to go though some parts carefully and slowly with packaging guidelines, as I fe. don't think, the package should be really flagged as arch: any, as it becomes platform dependent after build(), so it should probably list arches one by one.

dolik.rce wrote on Mon, 24 April 2017 21:21
That file is a wonderful thing, but it is not really needed for the basic packaging. Less then a year ago, Amrein did a many fixes in the generated makefile that is included in the tarball when he was working on RPM packages. It should be fairly easy to use the regular makefile for Arch packages as well.

I'll look into it, though I'd still prefer the file to just be packaged, as it "just works" and it's some 20kB decompressed.

The current package also sports two custom files, GCC.bm and theide.install, which seem not to come from repo. The second one seems to be quite simplified version of similar file in debian directory, the first one apparently comes into the compilation process and I guess is needed for the mkfile procedure. If using the regular makefile would drop need for this file, that would probably be nice. So yes, I'll look into it.

Cheers,

Eremiell

---

Subject: Re: Archlinux AUR
Posted by dolik.rce on Wed, 26 Apr 2017 19:41:42 GMT

Ok, here is my suggestion: If you're interested, take the upp-nightly and upp-git packages and bring them up to date with current AUR guidelines. Feel free to contact me via PM or email (or in person, I see you're from Prague too ) if you want to discuss how the building/packaging works. When those packages are in good shape, you can take over the stable branch and modernize it as well. For now, I have updated the stable to 2017.1, so there is no rush.

Quote:As a side note, I'm designing the nightly and git channels as auto updating, grabbing new versions on fly without hand patching the PKGBULD in AUR unless something changes, similar to firefox-always-nightly AUR package or most of git packages around.I used to have scripts to publish the nightly packages too. (Yes, I did that one too, but it got lost during migration from AUR3 to AUR4. Fusion809 created them afresh later and I totally forgot about it in my previous post.) Most of the time the scripts worked, but it always took me way too long to notice when something broke. So my tip for you would be to think about some kind of monitoring

Quote:The current package also sports two custom files, GCC.bm and theide.install, which seem not to come from repo. The second one seems to be quite simplified version of similar file in debian directory, the first one apparently comes into the compilation process and I guess is needed for the mkfile procedure.Theide.install can be dropped, since pacman hooks run 'update-desktop-database' automatically. The GCC.bm is not used during build. It is present in the package to be installed as a default build method for theide/umk. It would be actually much better if it could be generated automatically during build or even during installation (to make sure that the paths and settings are correct for the final user). Also, it would be nice to generate similar file for Clang. Amrein already did something like that for RPM packages, you can have a look in uppbox/Scripts - it also contains many other interesting things, e.g. the script which builds the tarball.

Honza

---

Subject: Re: Archlinux AUR
Posted by Eremiell on Wed, 26 Apr 2017 20:32:44 GMT


dolik.rce wrote on Wed, 26 April 2017 21:41Ok, here is my suggestion: If you're interested, take the upp-nightly and upp-git packages and bring them up to date with current AUR guidelines. Feel free to contact me via PM or email (or in person, I see you're from Prague too ) if you want to discuss how the building/packaging works. When those packages are in good shape, you can take over the stable branch and modernize it as well. For now, I have updated the stable to 2017.1, so there is no rush.

I'm onto it. Sadly still too much work to actually steal an evening for myself and some packaging. This week's terrible so far as work goes, which is luckily quite rarish lately.

dolik.rce wrote on Wed, 26 April 2017 21:41I used to have scripts to publish the nightly packages too. (Yes, I did that one too, but it got lost during migration from AUR3 to AUR4. Fusion809

created them afresh later and I totally forgot about it in my previous post.) Most of the time the scripts worked, but it always took me way too long to notice when something broke. So my tip for you would be to think about some kind of monitoring

I see. Sure, I plan to hand monitor it on some semi-regular basis. I just know, there are days, when I have too much other stuff to do and want to deliver the latest asap. It's also obviously much less work unless something changes and breaks stuff.

dolik.rce wrote on Wed, 26 April 2017 21:41Theide.install can be dropped, since pacman hooks run 'update-desktop-database' automatically. The GCC.bm is not used during build. It is present in the package to be installed as a default build method for theide/umk. It would be actually much better if it could be generated automatically during build or even during installation (to make sure that the paths and settings are correct for the final user). Also, it would be nice to generate similar file for Clang. Amrein already did something like that for RPM packages, you can have a look in uppbox/Scripts - it also contains many other interesting things, e.g. the script which builds the tarball.

Very well. I'll look into it asap, once I fix the deadline on fire.

I've seen some Czech sounding names around, but better to never presume, where people are from. If you wish to meet in person in Prague for some nice chat about U++, C++, Arch and stuff, I'm certainly open. I'm pretty much free on any day I don't have imminent deadlines, I don't teach and I don't have huge sleep deficit from the combination of the previous two.

---

Subject: Re: Archlinux AUR
Posted by Eremiell on Sun, 30 Apr 2017 11:15:57 GMT
View Forum Message <> Reply to Message

So I spent part of yesterday and most of today so far exploring contents of Scripts as well as stuff that comes with src rpm, and while I acknowledge the accomplishments reached, honestly a lot of it feels like somewhat dirty hacks. We all do dirty hacks at times to make stuff happen, but I'd probably prefer a somewhat cleaner way to package stuff. So at the moment, I'll probably prefer the mkfile to the rpm Makefile, which heavily calls bash scripts domake and doinstall to achieve pretty much anything.

One question pops up: did U++ ever consider using a meta build system like cmake or any of the many alternatives? It might clean up and ease the whole building process across platforms a lot. The current state feels in some ways as reinventing wheel. (Not that I haven't been there before.)

I'd really love to support clang as I use both gcc and clang most of the time, as they emit different warnings and catch up different troubles. I'll be looking more into that.

I originally planned to have the first PKGBUILDs yesterday (after not finding time for it on Tuesday), but while it compiles and packages for me without troubles (as it did week ago), I still feel there's more to polish. On top of that I'm leaving Prague in moments for a family event, so the

packages will probably come early next week once I get back and do some final cuts to achieve what I'd call minimal sanity level, rereading the packaging guidelines slowly once again (last time) and making sure, it's more or less clean at least on the packaging side. Then I'll release and dig deeper into the ways to build it better.

---