

Packages - Packaging Request #2339

add TuxGuitar

2019-06-20 01:58 AM - AuroraBorealis

Status: open	% Done: 0%
Priority: wish	
Assignee:	
Category:	
Description Hello. It would be great to have TuxGuitar in Parabola's repo. AUR has it: https://aur.archlinux.org/packages/tuxguitar/	

History

#1 - 2019-06-20 02:09 AM - AuroraBorealis

It's an essential program for musicians that has no alternatives on GNU/Linux and other free OSes.

#2 - 2019-06-20 06:04 AM - bill-auger

- Priority changed from bug to wish

like many java programs, this does not build from source, but downloads new un-sourced files, often pre-compiled binaries, at build time - like many java packages, this one would probably be a lot of work to make fit for parabola

```
| ==> Starting build()...
| [INFO] Scanning for projects...
| [INFO] -----
| [INFO] Building tuxguitar-pom 1.5.2
| [INFO] -----
| Downloading: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-install-plugin/2.4/maven-
install-plugin-2.4.pom
| [INFO] -----
| [INFO] BUILD FAILURE
| [INFO] -----
```

#3 - 2019-07-19 12:11 AM - AuroraBorealis

Sadly, I will not use Parabola because of this single program.

#4 - 2019-07-19 09:47 AM - bill-auger

luckily, there is nothing preventing you from using this, or any program with parabola; and because an AUR PKGBUILD exists, it is very simple to install it - adding it to parabola repos would not make much difference, as long as the AUR maintainer updates it when that is needed

the easiest way for an AUR package to get into parabola is if it gets votes on AUR - when some AUR package accumulates enough votes, then arch will consider adding it to the main repos; and then parabola would get it automatically - this packages has 33 votes, which is significantly more than average - however, if the package does not build from source, and it is a large amount of work to make it build from source, then it is likely that arch would not take it either - furthermore, if that work were done, and it turns out that any of those dependencies are not freely licensed, then parabola would need to blacklist it; so like most java programs, this is probably already the ideal situation for any parabola users who want to use them

#5 - 2019-07-23 02:09 PM - AuroraBorealis

Thanks for your explanation, I think the dependencies under consideration are Apache Maven files that are all available in Apache's Source Repository, and are distributed under the free Apache License v.2.0.

#6 - 2019-07-23 06:40 PM - bill-auger

the license of the dependencies is not the entire concern though - each dependency would need to each be packaged separately - unfortunately with

java programs, that often numbers in many dozens - that is a large number of packages that would need to be maintained only to support one program

if most of those dependencies were widely used by other programs, then that would increase the value of maintaining them; but that could introduce a bigger problem because java developers are not thinking in terms of stable distros with standard libraries - different java programs would be likely to depend on different (possibly incompatible) versions of the same library; which could make some of them exponentially more difficult to support

that is the reason devs that use non-native languages (like java, golang, C#, javascript, ruby, python) most often package "the kitchen-sink" and require any user who wants to build from source to install every dependency to a writable location (or worse, to a system location) using the language-specific package manager, or they put everything in a much larger container package to overcome the fragility of this user-install method - those approaches are much simpler for the developers because it allows devs to pile on dependencies liberally, with little or no concern for efficiency, compatibility, or code simplicity; but imposes a large and fragile overhead onto the user and down-streams

that sort of development practice is the polar opposite of, and induces great friction with, what distros do on behalf of users in order to keep their system efficient and sane - its the very reason why distros exist in the first place - developers using non-native languages are predominantly users of proprietary operating systems that do not have proper package management, and who do not prioritize simplicity and efficiency which are the key tenets of "the UNIX philosophy" and the hallmarks of an arch distro

sure, these things could be packaged; but that there are relatively few java, golang, or C# programs that are widely used by users of nix distros; which makes the dependencies of each such program, more rarely used, and therefore less valuable to a distro or its users - python and haskell are the exceptions (and to a smaller degree, ruby and javascript), mainly because they are more widely used in the system tools of nix distros and they do not need to be compiled; which makes packaging and maintenance less work - as such, most of the commonly used dependencies for python and haskell have been packaged already

that is a long-winded explanation; but for anyone interested, it is why you do not see very many java, golang, or C# programs in any distro - it probably also explains why such an "essential program for musicians" is in the AUR and not in arch already - there is now a pro-audio team for arch, which added a large number of audio production programs last year - if this were to happen, it would be best discussed on the 'arch-proaudio' mailing list - i invite you to ask about the program on that mailing list - the lead dev of the 'arch-proaudio' team is very libre-minded; and we should work closely with them in regards to any audio production software

<https://lists.archlinux.org/listinfo/arch-proaudio>

#7 - 2019-07-24 08:14 AM - AuroraBorealis

Oh, you wrote a whole essay..

As far as I know, only 3 Maven plugins are needed for TuxGuitar, maven-install-plugin, maven-dependency-plugin, and maven-ant-plugin.

#8 - 2019-07-24 12:13 PM - freemor

bill-auger perhaps didn't give a clear enough example. and from your above comment it's clear that you haven't tried building it yourself. Because as soon as the build phase of makepkg starts it looks like this.

```
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building tuxguitar-pom 1.5.2_r1800
[INFO] -----
[INFO]
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ tuxguitar-pom ---
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ tuxguitar-pom ---
Downloading: https://repo.maven.apache.org/maven2/classworlds/classworlds/1.1-alpha-2/classworlds-1.1-alpha-2.
pom
Downloading: https://repo.maven.apache.org/maven2/org/apache/maven/maven-profile/2.0.6/maven-profile-2.0.6.pom
Downloading: https://repo.maven.apache.org/maven2/org/apache/maven/maven-artifact-manager/2.0.6/maven-artifact
-manager-2.0.6.pom
Downloading: https://repo.maven.apache.org/maven2/org/apache/maven/maven-repository-metadata/2.0.6/maven-repos
itory-metadata-2.0.6.pom
Downloading: https://repo.maven.apache.org/maven2/org/apache/maven/maven-artifact/2.0.6/maven-artifact-2.0.6.p
om
Downloading: https://repo.maven.apache.org/maven2/org/apache/maven/maven-plugin-registry/2.0.6/maven-plugin-re
gistry-2.0.6.pom
Downloading: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0.5/plexus-utils-3.0.5.po
m
Downloading: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus/3.1/plexus-3.1.pom
Downloading: https://repo.maven.apache.org/maven2/org/sonatype/spice/spice-parent/17/spice-parent-17.pom
Downloading: https://repo.maven.apache.org/maven2/org/sonatype/forge/forge-parent/10/forge-parent-10.pom
Downloading: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-digest/1.0/plexus-digest-1.0.pom
Downloading: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-components/1.1.7/plexus-component
s-1.1.7.pom
Downloading: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus/1.0.8/plexus-1.0.8.pom
Downloading: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-container-default/1.0-alpha-8/ple
xus-container-default-1.0-alpha-8.pom
Downloading: https://repo.maven.apache.org/maven2/org/apache/maven/maven-project/2.0.6/maven-project-2.0.6.jar
```

Downloading: <https://repo.maven.apache.org/maven2/org/apache/maven/maven-profile/2.0.6/maven-profile-2.0.6.jar>
Downloading: <https://repo.maven.apache.org/maven2/org/apache/maven/maven-plugin-registry/2.0.6/maven-plugin-registry-2.0.6.jar>
Downloading: <https://repo.maven.apache.org/maven2/org/apache/maven/maven-settings/2.0.6/maven-settings-2.0.6.jar>
Downloading: <https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-container-default/1.0-alpha-9-stable-1/plexus-container-default-1.0-alpha-9-stable-1.jar>
Downloading: <https://repo.maven.apache.org/maven2/junit/junit/3.8.1/junit-3.8.1.jar>
Downloading: <https://repo.maven.apache.org/maven2/classworlds/classworlds/1.1-alpha-2/classworlds-1.1-alpha-2.jar>
Downloading: <https://repo.maven.apache.org/maven2/org/apache/maven/maven-artifact-manager/2.0.6/maven-artifact-manager-2.0.6.jar>
Downloading: <https://repo.maven.apache.org/maven2/org/apache/maven/maven-repository-metadata/2.0.6/maven-repository-metadata-2.0.6.jar>
Downloading: <https://repo.maven.apache.org/maven2/org/apache/maven/maven-artifact/2.0.6/maven-artifact-2.0.6.jar>
Downloading: <https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0.5/plexus-utils-3.0.5.jar>
Downloading: <https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-digest/1.0/plexus-digest-1.0.jar>

I stopped it at that point as it was starting to compile the first bit.. But thats a bit more then 3
And there is a good chance it'd do another such pull to build other parts.

We, the devs whould have to track every one of those down. check the licensing package then, patch the build to use our packages rather then pull willy nilly from the Internet.
On top of that there'd be the version headaches he mentioned. Etc. Etc. This is just a bad fit from a maintenance/freedom standpoint.

#9 - 2019-07-24 03:53 PM - AuroraBorealis

Thanks for clarifying it, you're right, there's plenty of them, at least we know it's all Apache files. And no, I didn't build it myself, I took a quick glance at its build scripts, I'm not on a *nix system currently.

#10 - 2019-07-24 03:58 PM - freemor

An understandable mistake. One shouldn't have to anticipate that the process of compiling code would result in it downloading a bunch of stuff. But sadly in a world of Java/Go/Rust/Ruby/etc it is something you have to actively watch for.

#11 - 2019-07-24 05:47 PM - bill-auger

AuroraBorealis -

just to make it clear again, that using parabola would not prevent you from using this program - the very point of a free OS is that it does not prevent you from doing anything, including running proprietary software - the only implication is that you would need to build for yourself the packages that parabola does not carry - luckily, that is quite simple to do if someone in AUR has done the hard work already

#12 - 2019-07-24 05:53 PM - bill-auger

and BTW i did not mind writing an essay on this topic - it comes up often enough that i have myself, explained this in much the same way several times - spelling it out into such detail in a published form, is an investment for the future - the next time this comes up, that verbose comment may become the seed of a wiki article, so that i, and other parabola devs, wont need to explain it again in full

#13 - 2019-07-25 09:27 AM - AuroraBorealis

it seems that you guys have a great community here.. I didn't expect to see such comprehensive input and readiness to assist on my humble request page.

#14 - 2019-12-08 03:42 AM - AuroraBorealis

- *File depositphotos_9066007-stock-photo-confused-emoticon.jpg added*

AuroraBorealis wrote:

It's an essential program for musicians that has no alternatives on GNU/Linux and other free OSes.

Ah, recently I've figured out that TuxGuitar in its present state is crap, and that MuseScore is a much better program for guitar tabs, so it's not really urgent. I'm not going to use TuxGuitar anyway. And it's definitely not an essential program, it's just s*it, as it turns out.

#15 - 2019-12-08 07:50 PM - bill-auger

in a case like this, urgency is no longer an issue at all, unless this program has some useful functionality that no other program in the parabola repos can accomplish

the main consideration for adding a new package is the desirability/cost-of-maintenance ratio - we have seen that the cost-of-maintenance factor is quite high for this one; so its desirability factor would also need to be very high to make it valuable - if musescore does the same job as tuxguitar, that

decreases the desirability factor of tuxguitar significantly

Files

depositphotos_9066007-stock-photo-confused-emoticon.jpg	56 KB	2019-12-08	AuroraBorealis
---	-------	------------	----------------