Chatting with ftpmasters

Andreas Tille

Debian

Kochi, 11. September 2023

Overview

1. Communication
2. Accept + RC-Bug
3. Existing packages passing NEW
4. Legal aspects
5. Hints

EtherPad: https://pad.dc23.debconf.org/p/31-chatting-with-ftpmasters

Did you bring some cake?

- Friendly chat between ftpmasters and developers
- Hopefully productive
- Communication between hard working people sometimes hard to schedule

Did you bring some cake?

- Friendly chat between ftpmasters and developers
- Hopefully productive
- Communication between hard working people sometimes hard to schedule

Did you bring some cake?

- Friendly chat between ftpmasters and developers
- Hopefully productive
- Communication between hard working people sometimes hard to schedule

Why cake?

- Since ftpmasters are doing a tough job[1] [2] . . .
- . . . where I'm happy that I do not need to do it
- Without ftpmasters we would not have a single package online
- So I usually bring cake
- Besides cake I end all my e-mails to ftpmaster with
  Thank you for your work as ftpmaster

[1] http://blog.alteholz.eu/

[2] Scott Kitterman: I didn't do anything beyond taking some time off work so I would have more time to pick up the FTP Team end of getting the new packages in.

Why cake?

- Since ftpmasters are doing a tough job[1] [2] . . .
- . . . where I'm happy that I do not need to do it
- Without ftpmasters we would not have a single package online
- So I usually bring cake
- Besides cake I end all my e-mails to ftpmaster with *Thank you for your work as ftpmaster*

[1] http://blog.alteholz.eu/

[2] Scott Kitterman: I didn't do anything beyond taking some time off work so I would have more time to pick up the FTP Team end of getting the new packages in.

Why cake?

- Since ftpmasters are doing a tough job[1] [2] . . .
- . . . where I'm happy that I do not need to do it
- Without ftpmasters we would not have a single package online
- So I usually bring cake
- Besides cake I end all my e-mails to ftpmaster with *Thank you for your work as ftpmaster*

[1] http://blog.alteholz.eu/

[2] Scott Kitterman: I didn't do anything beyond taking some time off work so I would have more time to pick up the FTP Team end of getting the new packages in.

Why cake?

- Since ftpmasters are doing a tough job[1] [2] . . .
- . . . where I'm happy that I do not need to do it
- Without ftpmasters we would not have a single package online
- So I usually bring cake
- Besides cake I end all my e-mails to ftpmaster with *Thank you for your work as ftpmaster*

[1] http://blog.alteholz.eu/

[2] Scott Kitterman: I didn't do anything beyond taking some time off work so I would have more time to pick up the FTP Team end of getting the new packages in.

Why cake?

- Since ftpmasters are doing a tough job[1] [2] . . .
- . . . where I'm happy that I do not need to do it
- Without ftpmasters we would not have a single package online
- So I usually bring cake
- Besides cake I end all my e-mails to ftpmaster with
  Thank you for your work as ftpmaster

[1] http://blog.alteholz.eu/

[2] Scott Kitterman: I didn't do anything beyond taking some time off work so I would have more time to pick up the FTP Team end of getting the new packages in.

Motivation for this BoF

- Debian is also known to be slow in getting something new quickly (Debian Med versus Bioconda)
- Community is observing us like in the *LWN Article: Debian reconsiders NEW review (Feb 11, 2022)*
- Can we do better do fulfill promise 4 of our social contract: *Our priorities are our users and free software*
- I'd like to start with better ways of communication

Motivation for this BoF

- Debian is also known to be slow in getting something new quickly (Debian Med versus Bioconda)
- Community is observing us like in the *LWN Article: Debian reconsiders NEW review (Feb 11, 2022)*
- Can we do better do fulfill promise 4 of our social contract: *Our priorities are our users and free software*
- I'd like to start with better ways of communication

Motivation for this BoF

- Debian is also known to be slow in getting something new quickly (Debian Med versus Bioconda)
- Community is observing us like in the *LWN Article: Debian reconsiders NEW review (Feb 11, 2022)*
- Can we do better do fulfill promise 4 of our social contract: *Our priorities are our users and free software*
- I'd like to start with better ways of communication

Motivation for this BoF

- Debian is also known to be slow in getting something new quickly (Debian Med versus Bioconda)
- Community is observing us like in the *LWN Article: Debian reconsiders NEW review (Feb 11, 2022)*
- Can we do better do fulfill promise 4 of our social contract: *Our priorities are our users and free software*
- I'd like to start with better ways of communication

Communication

- Communication with ftpmaster is rarely symmetric
- Threads on `debian-devel@l.d.o` rarely featuring ftpmaster
- Ping on IRC `#debian-ftp` sometimes works
- Replying to "`*.changes is NEW`"-mails frequently works
- Asymmetric "communication" via "`*.changes REJECTED`"-mails
- *Blog* from Thorsten Alteholz

Communication

- Communication with ftpmaster is rarely symmetric
- Threads on `debian-devel@l.d.o` rarely featuring ftpmaster
- Ping on IRC `#debian-ftp` sometimes works
- Replying to "`*.changes is NEW`"-mails frequently works
- Asymmetric "communication" via "`*.changes REJECTED`"-mails
- *Blog* from Thorsten Alteholz

Communication

- Communication with ftpmaster is rarely symmetric
- Threads on `debian-devel@l.d.o` rarely featuring ftpmaster
- Ping on IRC `#debian-ftp` sometimes works
- Replying to "`*.changes is NEW`"-mails frequently works
- Asymmetric "communication" via "`*.changes REJECTED`"-mails
- *Blog* from Thorsten Alteholz

Communication

- Communication with ftpmaster is rarely symmetric
- Threads on `debian-devel@l.d.o` rarely featuring ftpmaster
- Ping on IRC `#debian-ftp` sometimes works
- Replying to "`*.changes is NEW`"-mails frequently works
- Asymmetric "communication" via "`*.changes REJECTED`"-mails
- *Blog* from Thorsten Alteholz

Communication

- Communication with ftpmaster is rarely symmetric
- Threads on `debian-devel@l.d.o` rarely featuring ftpmaster
- Ping on IRC `#debian-ftp` sometimes works
- Replying to "`*.changes is NEW`"-mails frequently works
- Asymmetric "communication" via "`*.changes REJECTED`"-mails
- *Blog* from Thorsten Alteholz

Communication

- Communication with ftpmaster is rarely symmetric
- Threads on `debian-devel@l.d.o` rarely featuring ftpmaster
- Ping on IRC `#debian-ftp` sometimes works
- Replying to "`*.changes is NEW`"-mails frequently works
- Asymmetric "communication" via "`*.changes REJECTED`"-mails
- *Blog* from Thorsten Alteholz

Blog

When reading *this blog article* I found:

> *This month I accepted 420 and rejected 44 packages. The overall number of packages that got accepted was 422.*
>
> *I am sad to write the following lines, but unfortunately there are people who rather take advantage of others instead of doing a proper maintenance of their packages.*
>
> *So, in order to find time slots for as much packages in NEW as possible, I no longer write a debian/copyright for others. I know it is a boring task to collect the copyright information, but our policy still requires this. Of course nobody is perfect and certainly one or the other license or copyright holder can be overlooked. Luckily most of the contributors maintain their debian/copyright very thouroughly with a terrific result.*
>
> *On the other hand some contributors upload only some crap and demand that I exactly list what is missing. I am no longer willing to do this. I am going to stop processing after I found a few missing things and reject the package. When I see repeatedly uploads containing only improvements with things I pointed out, I will process this package only after all others from NEW are done.*

Reject mails

- Two types of reject mails
  1. OMG, I missed that, happy that ftpmaster found it
  2. Leaves the feeling that ftpmaster as well as you spent some time over some issue which is not really in the interest of our users
- Wild guessing: ftpmaster framework is binary `accept` / `reject`
- Leading to frustration on reject receiving end since an "answer" is just another upload taking days / weeks / months (?)
- Imagine some third option `accept + RC bug`

Reject mails

- Two types of reject mails
  1. OMG, I missed that, happy that ftpmaster found it
  2. Leaves the feeling that ftpmaster as well as you spent some time over some issue which is not really in the interest of our users
- Wild guessing: ftpmaster framework is binary `accept` / `reject`
- Leading to frustration on reject receiving end since an "answer" is just another upload taking days / weeks / months (?)
- Imagine some third option `accept + RC bug`

Reject mails

- Two types of reject mails
  1. OMG, I missed that, happy that ftpmaster found it
  2. Leaves the feeling that ftpmaster as well as you spent some time over some issue which is not really in the interest of our users
- Wild guessing: ftpmaster framework is binary `accept` / `reject`
- Leading to frustration on reject receiving end since an "answer" is just another upload taking days / weeks / months (?)
- Imagine some third option `accept + RC bug`

Reject mails

- Two types of reject mails
  1. OMG, I missed that, happy that ftpmaster found it
  2. Leaves the feeling that ftpmaster as well as you spent some time over some issue which is not really in the interest of our users
- Wild guessing: ftpmaster framework is binary `accept` / `reject`
- Leading to frustration on reject receiving end since an "answer" is just another upload taking days / weeks / months (?)
- Imagine some third option `accept + RC bug`

Reject mails

- Two types of reject mails
  1. OMG, I missed that, happy that ftpmaster found it
  2. Leaves the feeling that ftpmaster as well as you spent some time over some issue which is not really in the interest of our users
- Wild guessing: ftpmaster framework is binary `accept` / `reject`
- Leading to frustration on reject receiving end since an "answer" is just another upload taking days / weeks / months (?)
- Imagine some third option `accept + RC bug`

Reject mails

- Two types of reject mails
  1. OMG, I missed that, happy that ftpmaster found it
  2. Leaves the feeling that ftpmaster as well as you spent some time over some issue which is not really in the interest of our users
- Wild guessing: ftpmaster framework is binary `accept` / `reject`
- Leading to frustration on reject receiving end since an "answer" is just another upload taking days / weeks / months (?)
- Imagine some third option `accept + RC bug`

REJECT-FAQ

> *License III: You need to list all copyright information with all licenses in the copyright file itself. That one has to be the single point of information. Only files you find in /usr/share/common-licenses have a special exception here, everything else needs to be fully included!*

This was Revised Dec 2008. May be its time to revise it again?
Who should do the revision?

Accept + RC-Bug

Cases where applicable:

1. All code is DFSG free but `debian/copyright`
   - 1.1 Has not all free licenses of all files properly
   - 1.2 Specifies a wrong free license for one or more files
   - 1.3 Is lacking correctness in naming copyright holder
   - 1.4 Is missing copyright statements in a minority of files
2. License and copyrights are fine but
   - 2.1 Some files (for instance fonts) in unusual dirs
   - 2.2 Some missing documentation (README.source)
3. Package is in Debian but changed name(s) of binary package

Accept + RC-Bug

Cases where applicable:

1. All code is DFSG free but `debian/copyright`
   1.1 Has not all free licenses of all files properly
   1.2 Specifies a wrong free license for one or more files
   1.3 Is lacking correctness in naming copyright holder
   1.4 Is missing copyright statements in a minority of files
2. License and copyrights are fine but
   2.1 Some files (for instance fonts) in unusual dirs
   2.2 Some missing documentation (README.source)
3. Package is in Debian but changed name(s) of binary package

Accept + RC-Bug

Cases where applicable:

1. All code is DFSG free but `debian/copyright`
   - 1.1 Has not all free licenses of all files properly
   - 1.2 Specifies a wrong free license for one or more files
   - 1.3 Is lacking correctness in naming copyright holder
   - 1.4 Is missing copyright statements in a minority of files
2. License and copyrights are fine but
   - 2.1 Some files (for instance fonts) in unusual dirs
   - 2.2 Some missing documentation (README.source)
3. Package is in Debian but changed name(s) of binary package

Accept + RC-Bug

Cases where applicable:

1. All code is DFSG free but `debian/copyright`
   1.1 Has not all free licenses of all files properly
   1.2 Specifies a wrong free license for one or more files
   1.3 Is lacking correctness in naming copyright holder
   1.4 Is missing copyright statements in a minority of files
2. License and copyrights are fine but
   2.1 Some files (for instance fonts) in unusual dirs
   2.2 Some missing documentation (README.source)
3. Package is in Debian but changed name(s) of binary package

Accept + RC-Bug

Cases where applicable:

1. All code is DFSG free but `debian/copyright`
   1.1 Has not all free licenses of all files properly
   1.2 Specifies a wrong free license for one or more files
   1.3 Is lacking correctness in naming copyright holder
   1.4 Is missing copyright statements in a minority of files
2. License and copyrights are fine but
   2.1 Some files (for instance fonts) in unusual dirs
   2.2 Some missing documentation (README.source)
3. Package is in Debian but changed name(s) of binary package

Accept + RC-Bug

Cases where applicable:

1. All code is DFSG free but `debian/copyright`
   - 1.1 Has not all free licenses of all files properly
   - 1.2 Specifies a wrong free license for one or more files
   - 1.3 Is lacking correctness in naming copyright holder
   - 1.4 Is missing copyright statements in a minority of files
2. License and copyrights are fine but
   - 2.1 Some files (for instance fonts) in unusual dirs
   - 2.2 Some missing documentation (README.source)
3. Package is in Debian but changed name(s) of binary package

Accept + RC-Bug

Cases where applicable:

1. All code is DFSG free but `debian/copyright`
   1.1 Has not all free licenses of all files properly
   1.2 Specifies a wrong free license for one or more files
   1.3 Is lacking correctness in naming copyright holder
   1.4 Is missing copyright statements in a minority of files
2. License and copyrights are fine but
   2.1 Some files (for instance fonts) in unusual dirs
   2.2 Some missing documentation (README.source)
3. Package is in Debian but changed name(s) of binary package

Accept + RC-Bug

Cases where applicable:

1. All code is DFSG free but `debian/copyright`
   1.1 Has not all free licenses of all files properly
   1.2 Specifies a wrong free license for one or more files
   1.3 Is lacking correctness in naming copyright holder
   1.4 Is missing copyright statements in a minority of files
2. License and copyrights are fine but
   2.1 Some files (for instance fonts) in unusual dirs
   2.2 Some missing documentation (README.source)
3. Package is in Debian but changed name(s) of binary package

Accept + RC-Bug

Cases where applicable:

1. All code is DFSG free but `debian/copyright`
   1.1 Has not all free licenses of all files properly
   1.2 Specifies a wrong free license for one or more files
   1.3 Is lacking correctness in naming copyright holder
   1.4 Is missing copyright statements in a minority of files
2. License and copyrights are fine but
   2.1 Some files (for instance fonts) in unusual dirs
   2.2 Some missing documentation (README.source)
3. Package is in Debian but changed name(s) of binary package

Accept + RC-Bug

Cases where applicable:

1. All code is DFSG free but `debian/copyright`
   1.1 Has not all free licenses of all files properly
   1.2 Specifies a wrong free license for one or more files
   1.3 Is lacking correctness in naming copyright holder
   1.4 Is missing copyright statements in a minority of files
2. License and copyrights are fine but
   2.1 Some files (for instance fonts) in unusual dirs
   2.2 Some missing documentation (README.source)
3. Package is in Debian but changed name(s) of binary package

Examples will follow later

Advantages of Accept + RC-Bug

- Enables discussion in bug log for all sides
- Prevents that package will be officially released
- Documents issues of the package and enable asking for help
- No further rounds in new (= no further work of ftpmaster)
- Enables development against the package in question

? Am I missing any risk included in this procedure?

Advantages of Accept + RC-Bug

- Enables discussion in bug log for all sides
- Prevents that package will be officially released
- Documents issues of the package and enable asking for help
- No further rounds in new (= no further work of ftpmaster)
- Enables development against the package in question

? Am I missing any risk included in this procedure?

Advantages of Accept + RC-Bug

- Enables discussion in bug log for all sides
- Prevents that package will be officially released
- Documents issues of the package and enable asking for help
- No further rounds in new (= no further work of ftpmaster)
- Enables development against the package in question

? Am I missing any risk included in this procedure?

Advantages of Accept + RC-Bug

- Enables discussion in bug log for all sides
- Prevents that package will be officially released
- Documents issues of the package and enable asking for help
- No further rounds in new (= no further work of ftpmaster)
- Enables development against the package in question

? Am I missing any risk included in this procedure?

Advantages of Accept + RC-Bug

- Enables discussion in bug log for all sides
- Prevents that package will be officially released
- Documents issues of the package and enable asking for help
- No further rounds in new (= no further work of ftpmaster)
- Enables development against the package in question

? Am I missing any risk included in this procedure?

Advantages of Accept + RC-Bug

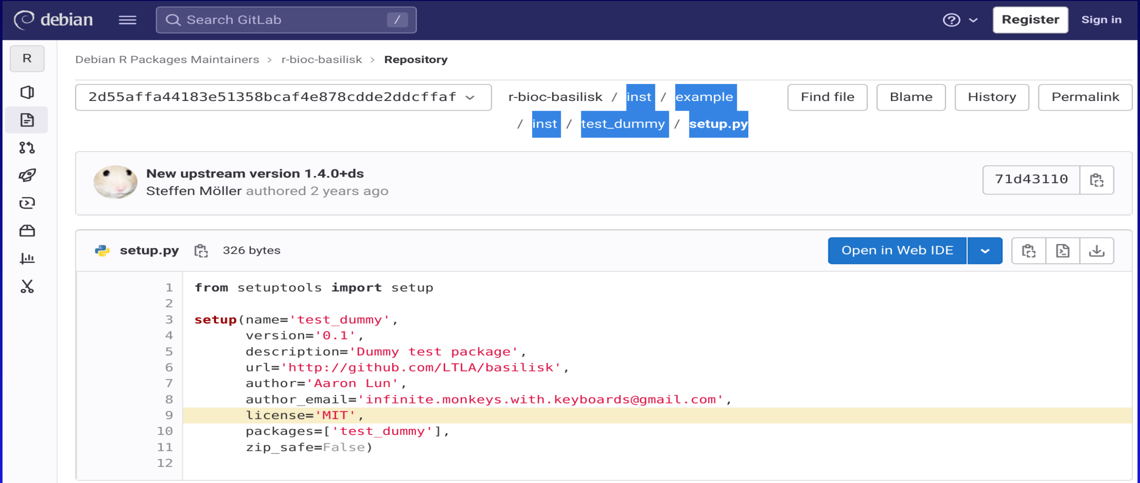
- Enables discussion in bug log for all sides
- Prevents that package will be officially released
- Documents issues of the package and enable asking for help
- No further rounds in new (= no further work of ftpmaster)
- Enables development against the package in question

? Am I missing any risk included in this procedure?

Examples for non-rejects: String or license?

r-bioc-basilisk: inst/example/inst/test_dummy/setup.py

Is this a string or some license statement?



Debian R Packages Maintainers > r-bioc-basilisk > **Repository**

2d55affa44183e51358bcaf4e878cdde2ddcffaf

r-bioc-basilisk / inst / example / inst / test_dummy / **setup.py**

New upstream version 1.4.0+ds
Steffen Möller authored 2 years ago

71d43110

setup.py 326 bytes

```
from setuptools import setup

setup(name='test_dummy',
      version='0.1',
      description='Dummy test package',
      url='http://github.com/LTLA/basilisk',
      author='Aaron Lun',
      author_email='infinite.monkeys.with.keyboards@gmail.com',
      license='MIT',
      packages=['test_dummy'],
      zip_safe=False)

```

Examples for non-rejects: String or license? (2)

- Ftpmaster *thinks* its a license statement
- Good example for asymmetric communication
- To act in the interest of our users to get the package quickly I would have simply added a paragraph to `d/copyright`
- . . . and by doing so I would have ignored what I think is correct
- I discussed this even with CTTE in *ITP bug log*
- Ftpmaster did not commented
- Finally upstream confirmed *It's just an example string*
- No "typical" example but shows how time of different people was burned over a non-issue

Examples for non-rejects: String or license? (2)

- Ftpmaster *thinks* its a license statement
- Good example for asymmetric communication
- To act in the interest of our users to get the package quickly I would have simply added a paragraph to `d/copyright`
- . . . and by doing so I would have ignored what I think is correct
- I discussed this even with CTTE in *ITP bug log*
- Ftpmaster did not commented
- Finally upstream confirmed *It's just an example string*
- No "typical" example but shows how time of different people was burned over a non-issue

Examples for non-rejects: String or license? (2)

- Ftpmaster *thinks* its a license statement
- Good example for asymmetric communication
- To act in the interest of our users to get the package quickly I would have simply added a paragraph to `d/copyright`
- . . . and by doing so I would have ignored what I think is correct
- I discussed this even with CTTE in *ITP bug log*
- Ftpmaster did not commented
- Finally upstream confirmed *It's just an example string*
- No "typical" example but shows how time of different people was burned over a non-issue

Examples for non-rejects: String or license? (2)

- Ftpmaster *thinks* its a license statement
- Good example for asymmetric communication
- To act in the interest of our users to get the package quickly I would have simply added a paragraph to `d/copyright`
- . . . and by doing so I would have ignored what I think is correct
- I discussed this even with CTTE in *ITP bug log*
- Ftpmaster did not commented
- Finally upstream confirmed *It's just an example string*
- No "typical" example but shows how time of different people was burned over a non-issue

Examples for non-rejects: String or license? (2)

- Ftpmaster *thinks* its a license statement
- Good example for asymmetric communication
- To act in the interest of our users to get the package quickly I would have simply added a paragraph to `d/copyright`
- . . . and by doing so I would have ignored what I think is correct
- I discussed this even with CTTE in *ITP bug log*
- Ftpmaster did not commented
- Finally upstream confirmed *It’s just an example string*
- No "typical" example but shows how time of different people was burned over a non-issue

Examples for non-rejects: String or license? (2)

- Ftpmaster *thinks* its a license statement
- Good example for asymmetric communication
- To act in the interest of our users to get the package quickly I would have simply added a paragraph to `d/copyright`
- . . . and by doing so I would have ignored what I think is correct
- I discussed this even with CTTE in *ITP bug log*
- Ftpmaster did not commented
- Finally upstream confirmed *It's just an example string*
- No "typical" example but shows how time of different people was burned over a non-issue

Examples for non-rejects: String or license? (2)

- Ftpmaster *thinks* its a license statement
- Good example for asymmetric communication
- To act in the interest of our users to get the package quickly I would have simply added a paragraph to `d/copyright`
- . . . and by doing so I would have ignored what I think is correct
- I discussed this even with CTTE in *ITP bug log*
- Ftpmaster did not commented
- Finally upstream confirmed *It's just an example string*
- No "typical" example but shows how time of different people was burned over a non-issue

Examples for non-rejects: String or license? (2)

- Ftpmaster *thinks* its a license statement
- Good example for asymmetric communication
- To act in the interest of our users to get the package quickly I would have simply added a paragraph to `d/copyright`
- . . . and by doing so I would have ignored what I think is correct
- I discussed this even with CTTE in *ITP bug log*
- Ftpmaster did not commented
- Finally upstream confirmed *It’s just an example string*
- No "typical" example but shows how time of different people was burned over a non-issue

Examples for non-rejects: GPL-2+ or GPL-3?

- *REJECT of* `r-cran-urlchecker`:

  > *please also mention the GPL-2 of urlchecker/inst/tools/urltools.R in your debian/copyright.*

- `inst/tools/urltools.R` actually has *"either version 2 of the License, or (at your option) any later version."*
- `d/copyright` had specified `GPL-3`
- Thus I *answered*:

  > *I've added an extra paragraph to d/copyright. However, IMHO there is no real point for a reject here since the "or later" statement of GPL-2+ should be fullfilled by GPL-3. I'd be very happy if you would save you and the Uploader some work in similar cases in future.*

- Thinking in terms of "Accept + RC bug" this would be at best "Accept + normal bug", IMHO

Item 1.2 in "Accept + RC-Bug" slide

Examples for non-rejects: GPL-2+ or GPL-3?

- *REJECT of* `r-cran-urlchecker`:

  > *please also mention the GPL-2 of urlchecker/inst/tools/urltools.R in your debian/copyright.*

- *`inst/tools/urltools.R`* actually has *"either version 2 of the License, or (at your option) any later version."*
- *`d/copyright`* had specified `GPL-3`
- Thus I *answered*:

  > *I've added an extra paragraph to d/copyright. However, IMHO there is no real point for a reject here since the "or later" statement of GPL-2+ should be fullfilled by GPL-3. I'd be very happy if you would save you and the Uploader some work in similar cases in future.*

- Thinking in terms of "Accept + RC bug" this would be at best "Accept + normal bug", IMHO

Item 1.2 in "Accept + RC-Bug" slide

Examples for non-rejects: GPL-2+ or GPL-3?

- *REJECT of* `r-cran-urlchecker`:

  > *please also mention the GPL-2 of urlchecker/inst/tools/urltools.R in your debian/copyright.*

- *`inst/tools/urltools.R`* actually has *"either version 2 of the License, or (at your option) any later version."*
- *`d/copyright`* had specified `GPL-3`
- Thus I *answered*:

  > *I've added an extra paragraph to d/copyright. However, IMHO there is no real point for a reject here since the "or later" statement of GPL-2+ should be fullfilled by GPL-3. I'd be very happy if you would save you and the Uploader some work in similar cases in future.*

- Thinking in terms of "Accept + RC bug" this would be at best "Accept + normal bug", IMHO

Item 1.2 in "Accept + RC-Bug" slide

Examples for non-rejects: GPL-2+ or GPL-3?

- *REJECT of* `r-cran-urlchecker`:

  > *please also mention the GPL-2 of urlchecker/inst/tools/urltools.R in your debian/copyright.*

- *`inst/tools/urltools.R`* actually has *"either version 2 of the License, or (at your option) any later version."*
- *`d/copyright`* had specified `GPL-3`
- Thus I *answered*:

  > *I've added an extra paragraph to d/copyright. However, IMHO there is no real point for a reject here since the "or later" statement of GPL-2+ should be fullfilled by GPL-3. I'd be very happy if you would save you and the Uploader some work in similar cases in future.*

- Thinking in terms of "Accept + RC bug" this would be at best "Accept + normal bug", IMHO

Item 1.2 in "Accept + RC-Bug" slide

Examples for non-rejects: GPL-2+ or GPL-3?

- *REJECT of* `r-cran-urlchecker`:

  > *please also mention the GPL-2 of urlchecker/inst/tools/urltools.R in your debian/copyright.*

- `inst/tools/urltools.R` actually has *"either version 2 of the License, or (at your option) any later version."*
- `d/copyright` had specified `GPL-3`
- Thus I *answered*:

  > *I've added an extra paragraph to d/copyright. However, IMHO there is no real point for a reject here since the "or later" statement of GPL-2+ should be fullfilled by GPL-3. I'd be very happy if you would save you and the Uploader some work in similar cases in future.*

- Thinking in terms of "Accept + RC bug" this would be at best "Accept + normal bug", IMHO

Item 1.2 in "Accept + RC-Bug" slide

Examples for non-rejects: GPL-2+ or GPL-3?

- *REJECT of* `r-cran-urlchecker`:

  > *please also mention the GPL-2 of urlchecker/inst/tools/urltools.R in your debian/copyright.*

- *`inst/tools/urltools.R`* actually has *"either version 2 of the License, or (at your option) any later version."*
- *`d/copyright`* had specified `GPL-3`
- Thus I *answered*:

  > *I've added an extra paragraph to d/copyright. However, IMHO there is no real point for a reject here since the "or later" statement of GPL-2+ should be fullfilled by GPL-3. I'd be very happy if you would save you and the Uploader some work in similar cases in future.*

- Thinking in terms of "Accept + RC bug" this would be at best "Accept + normal bug", IMHO

Item 1.2 in "Accept + RC-Bug" slide

Examples for non-rejects: GPL-2+ or GPL-3?

- *REJECT of* `r-cran-urlchecker`:

  > *please also mention the GPL-2 of urlchecker/inst/tools/urltools.R in your debian/copyright.*

- `inst/tools/urltools.R` actually has *"either version 2 of the License, or (at your option) any later version."*
- `d/copyright` had specified `GPL-3`
- Thus I *answered*:

  > *I've added an extra paragraph to d/copyright. However, IMHO there is no real point for a reject here since the "or later" statement of GPL-2+ should be fullfilled by GPL-3. I'd be very happy if you would save you and the Uploader some work in similar cases in future.*

- Thinking in terms of "Accept + RC bug" this would be at best "Accept + normal bug", IMHO

Item 1.2 in "Accept + RC-Bug" slide

Examples for non-rejects: Missing Public Domain

- `versioneer.py`
  - Several Python packages are containing the 2kB file *versioneer.py*
  - Its *license is CC0-1.0 (== public domain)* so authors explicitly do not claim copyright
  - Packages failing to mention this will be *rejected*
- Unlicense license
  - `r-cran-dials` was *REJECTED*:
    please mention the Unlicense license of some files

➜ We could save a lot of time if this would be "Accept + normal bug", IMHO

Item 1.1 in "Accept + RC-Bug" slide

Examples for non-rejects: Missing Public Domain

- `versioneer.py`
  - Several Python packages are containing the 2kB file *versioneer.py*
  - Its *license is CC0-1.0 (== public domain)* so authors explicitly do not claim copyright
  - Packages failing to mention this will be *rejected*
- Unlicense license
  - `r-cran-dials` was *REJECTED*:
    please mention the Unlicense license of some files

➜ We could save a lot of time if this would be "Accept + normal bug", IMHO

Item 1.1 in "Accept + RC-Bug" slide

Examples for non-rejects: Missing Public Domain

- `versioneer.py`
  - Several Python packages are containing the 2kB file *versioneer.py*
  - Its *license is CC0-1.0 (== public domain)* so authors explicitly do not claim copyright
  - Packages failing to mention this will be *rejected*
- Unlicense license
  - `r-cran-dials` was *REJECTED*:
    please mention the Unlicense license of some files

➜ We could save a lot of time if this would be "Accept + normal bug", IMHO

Item 1.1 in "Accept + RC-Bug" slide

Examples for non-rejects: Missing Public Domain

- `versioneer.py`
  - Several Python packages are containing the 2kB file *versioneer.py*
  - Its *license is CC0-1.0 (== public domain)* so authors explicitly do not claim copyright
  - Packages failing to mention this will be *rejected*
- Unlicense license
  - `r-cran-dials` was *REJECTED*:
    please mention the Unlicense license of some files

➜ We could save a lot of time if this would be "Accept + normal bug", IMHO

Item 1.1 in "Accept + RC-Bug" slide

Examples for non-rejects: Missing Public Domain

- *`versioneer.py`*
    - Several Python packages are containing the 2kB file *versioneer.py*
    - Its *license is CC0-1.0 (== public domain)* so authors explicitly do not claim copyright
    - Packages failing to mention this will be *rejected*
- Unlicense license
    - `r-cran-dials` was *REJECTED*:
      please mention the Unlicense license of some files

➜ We could save a lot of time if this would be "Accept + normal bug", IMHO

Item 1.1 in "Accept + RC-Bug" slide

Examples for non-rejects: Missing Public Domain

- *versioneer.py*
  - Several Python packages are containing the 2kB file *versioneer.py*
  - Its *license is CC0-1.0 (== public domain)* so authors explicitly do not claim copyright
  - Packages failing to mention this will be *rejected*
- Unlicense license
  - *r-cran-dials* was *REJECTED*:
    please mention the Unlicense license of some files

➜ We could save a lot of time if this would be "Accept + normal bug", IMHO

Item 1.1 in "Accept + RC-Bug" slide

Examples for non-rejects: Missing Public Domain

- `versioneer.py`
  - Several Python packages are containing the 2kB file *versioneer.py*
  - Its *license is CC0-1.0 (== public domain)* so authors explicitly do not claim copyright
  - Packages failing to mention this will be *rejected*
- Unlicense license
  - `r-cran-dials` was *REJECTED*:
    please mention the Unlicense license of some files

➜ We could save a lot of time if this would be "Accept + normal bug", IMHO

Item 1.1 in "Accept + RC-Bug" slide

Examples for non-rejects: Missing copyright holders

- `netpbm` was outdated, de facto non-maintained and buggy
- One bug needed change of binary name thus NEW processing
- Recieved REJECTs on *2022/05/12*, *2022/05/28* and *2022/08/04*
- Finally accepted at 2023/01/10
- Got help in October by Andreas Meltzer (spent a couple of hours for review)
- Fixing five bugs (#1007984 serious that required name change)
- Weak point of this upload-reject cycle: All those rejects after *lots* of copyright fixes lengthen the time we distribute the existing package with broken copyright

Item 1.3 / 1.4 in "Accept + RC-Bug" slide

Examples for non-rejects: Missing copyright holders

- `netpbm` was outdated, de facto non-maintained and buggy
- One bug needed change of binary name thus NEW processing
- Recieved REJECTs on *2022/05/12*, *2022/05/28* and *2022/08/04*
- Finally accepted at 2023/01/10
- Got help in October by Andreas Meltzer (spent a couple of hours for review)
- Fixing five bugs (#1007984 serious that required name change)
- Weak point of this upload-reject cycle: All those rejects after *lots* of copyright fixes lengthen the time we distribute the existing package with broken copyright

Item 1.3 / 1.4 in "Accept + RC-Bug" slide

Examples for non-rejects: Missing copyright holders

- `netpbm` was outdated, de facto non-maintained and buggy
- One bug needed change of binary name thus NEW processing
- Recieved REJECTs on *2022/05/12*, *2022/05/28* and *2022/08/04*
- Finally accepted at 2023/01/10
- Got help in October by Andreas Meltzer (spent a couple of hours for review)
- Fixing five bugs (#1007984 serious that required name change)
- Weak point of this upload-reject cycle: All those rejects after *lots* of copyright fixes lengthen the time we distribute the existing package with broken copyright

Item 1.3 / 1.4 in "Accept + RC-Bug" slide

Examples for non-rejects: Missing copyright holders

- `netpbm` was outdated, de facto non-maintained and buggy
- One bug needed change of binary name thus NEW processing
- Recieved REJECTs on *2022/05/12*, *2022/05/28* and *2022/08/04*
- Finally accepted at 2023/01/10
- Got help in October by Andreas Meltzer (spent a couple of hours for review)
- Fixing five bugs (#1007984 serious that required name change)
- Weak point of this upload-reject cycle: All those rejects after *lots* of copyright fixes lengthen the time we distribute the existing package with broken copyright

Item 1.3 / 1.4 in "Accept + RC-Bug" slide

Examples for non-rejects: Missing copyright holders

- `netpbm` was outdated, de facto non-maintained and buggy
- One bug needed change of binary name thus NEW processing
- Recieved REJECTs on *2022/05/12*, *2022/05/28* and *2022/08/04*
- Finally accepted at 2023/01/10
- Got help in October by Andreas Meltzer (spent a couple of hours for review)
- Fixing five bugs (#1007984 serious that required name change)
- Weak point of this upload-reject cycle: All those rejects after *lots* of copyright fixes lengthen the time we distribute the existing package with broken copyright

Item 1.3 / 1.4 in "Accept + RC-Bug" slide

Examples for non-rejects: Missing copyright holders

- `netpbm` was outdated, de facto non-maintained and buggy
- One bug needed change of binary name thus NEW processing
- Recieved REJECTs on *2022/05/12*, *2022/05/28* and *2022/08/04*
- Finally accepted at 2023/01/10
- Got help in October by Andreas Meltzer (spent a couple of hours for review)
- Fixing five bugs (#1007984 serious that required name change)
- Weak point of this upload-reject cycle: All those rejects after *lots* of copyright fixes lengthen the time we distribute the existing package with broken copyright

Item 1.3 / 1.4 in "Accept + RC-Bug" slide

Examples for non-rejects: Missing copyright holders

- `netpbm` was outdated, de facto non-maintained and buggy
- One bug needed change of binary name thus NEW processing
- Recieved REJECTs on *2022/05/12*, *2022/05/28* and *2022/08/04*
- Finally accepted at 2023/01/10
- Got help in October by Andreas Meltzer (spent a couple of hours for review)
- Fixing five bugs (#1007984 serious that required name change)
- Weak point of this upload-reject cycle: All those rejects after *lots* of copyright fixes lengthen the time we distribute the existing package with broken copyright

Item 1.3 / 1.4 in "Accept + RC-Bug" slide

Examples for non-rejects: Incorrect copyright holder

`r-cran-renv` ***REJECTED***:

according to renv/LICENSE, the only copyright holer is Posit Software, PBC

```
$ git diff -U1 c030e13^..c030e13
diff --git a/debian/copyright b/debian/copyright
index b2db422..96fce52 100644
--- a/debian/copyright
+++ b/debian/copyright
@@ -6,5 +6,3 @@ Source: https://cran.r-project.org/package=renv
 Files: *
-Copyright: 2000-2023 Kevin Ushey,
- Hadley Wickham,
- Posit Software, PBC
+Copyright: 2000-2023 Posit Software, PBC
 License: MIT
```

Item 1.3 "Accept + RC-Bug" slide

Examples for non-rejects: Files in wrong dir

- `r-cran-gfonts` became new dependency of other CRAN packages
- Recieved REJECTs *Fonts in wrong dir (2.1) 2023/06/08* and *Wrongly specified license (1.2) 2023/08/31*
- Package needed for several other packages meanwhile with RC bugs due to missing upgrade
- Accepted pretty quickly after explaining when responding to REJECT

Item 2.1 and 1.2 in "Accept + RC-Bug" slide

Examples for non-rejects: Files in wrong dir

- `r-cran-gfonts` became new dependency of other CRAN packages
- Recieved REJECTs *Fonts in wrong dir (2.1) 2023/06/08* and *Wrongly specified license (1.2) 2023/08/31*
- Package needed for several other packages meanwhile with RC bugs due to missing upgrade
- Accepted pretty quickly after explaining when responding to REJECT

Item 2.1 and 1.2 in "Accept + RC-Bug" slide

Examples for non-rejects: Files in wrong dir

- `r-cran-gfonts` became new dependency of other CRAN packages
- Recieved REJECTs *Fonts in wrong dir (2.1) 2023/06/08* and *Wrongly specified license (1.2) 2023/08/31*
- Package needed for several other packages meanwhile with RC bugs due to missing upgrade
- Accepted pretty quickly after explaining when responding to REJECT

Item 2.1 and 1.2 in "Accept + RC-Bug" slide

Examples for non-rejects: Files in wrong dir

- *`r-cran-gfonts`* became new dependency of other CRAN packages
- Recieved REJECTs *Fonts in wrong dir (2.1) 2023/06/08* and *Wrongly specified license (1.2) 2023/08/31*
- Package needed for several other packages meanwhile with RC bugs due to missing upgrade
- Accepted pretty quickly after explaining when responding to REJECT

Item 2.1 and 1.2 in "Accept + RC-Bug" slide

Example for delayed process

- Pangolin: some AI software used to investigate COVID-19
- Packaging "competitors" like Conda distribute it instantly
- *pango-learn*
  - First upload *2021/04/06*
  - REJECT *2021/11/06* OMG, unused non-free stuff in source
  - Re-upload *2021/11/08* unfortunately new upstream version
  - REJECT *2022/02/16* OMG, exclusion failed in new upstream
  - Re-upload *2022/02/16*
  - REJECT *2022/10/28* big data file in /usr/lib (Accept + RC 2.1?)
  - Re-upload *2022/10/28* moving files somewhere else
  - REJECT *2022/12/21* license GPL-3.0 only? (Accept + RC 1.1)
  - *answer* to my question related to license
  - Re-upload *2022/12/21*
  - Finally accepted *2022/12/24*
- *pangolin-data* (superseeded *pango-learn* meanwhile)
  - First upload *2022/12/24*
  - REJECT *2023/08/22* No source for data

Example for delayed process

- Pangolin: some AI software used to investigate COVID-19
- Packaging "competitors" like Conda distribute it instantly
- `pango-learn`
  - First upload *2021/04/06*
  - REJECT *2021/11/06* OMG, unused non-free stuff in source
  - Re-upload *2021/11/08* unfortunately new upstream version
  - REJECT *2022/02/16* OMG, exclusion failed in new upstream
  - Re-upload *2022/02/16*
  - REJECT *2022/10/28* big data file in /usr/lib (Accept + RC 2.1?)
  - Re-upload *2022/10/28* moving files somewhere else
  - REJECT *2022/12/21* license GPL-3.0 only? (Accept + RC 1.1)
  - *answer* to my question related to license
  - Re-upload *2022/12/21*
  - Finally accepted *2022/12/24*
- `pangolin-data` (superseeded `pango-learn` meanwhile)
  - First upload *2022/12/24*
  - REJECT *2023/08/22* No source for data

Example for delayed process

- Pangolin: some AI software used to investigate COVID-19
- Packaging "competitors" like Conda distribute it instantly
- *pango-learn*
  - First upload *2021/04/06*
  - REJECT *2021/11/06* OMG, unused non-free stuff in source
  - Re-upload *2021/11/08* unfortunately new upstream version
  - REJECT *2022/02/16* OMG, exclusion failed in new upstream
  - Re-upload *2022/02/16*
  - REJECT *2022/10/28* big data file in /usr/lib (Accept + RC 2.1?)
  - Re-upload *2022/10/28* moving files somewhere else
  - REJECT *2022/12/21* license GPL-3.0 only? (Accept + RC 1.1)
  - *answer* to my question related to license
  - Re-upload *2022/12/21*
  - Finally accepted *2022/12/24*
- *pangolin-data* (superseeded *pango-learn* meanwhile)
  - First upload *2022/12/24*
  - REJECT *2023/08/22* No source for data

Example for delayed process

- Pangolin: some AI software used to investigate COVID-19
- Packaging "competitors" like Conda distribute it instantly
- `pango-learn`
  - First upload *2021/04/06*
  - REJECT *2021/11/06* OMG, unused non-free stuff in source
  - Re-upload *2021/11/08* unfortunately new upstream version
  - REJECT *2022/02/16* OMG, exclusion failed in new upstream
  - Re-upload *2022/02/16*
  - REJECT *2022/10/28* big data file in /usr/lib (Accept + RC 2.1?)
  - Re-upload *2022/10/28* moving files somewhere else
  - REJECT *2022/12/21* license GPL-3.0 only? (Accept + RC 1.1)
  - *answer* to my question related to license
  - Re-upload *2022/12/21*
  - Finally accepted *2022/12/24*
- `pangolin-data` (superseeded `pango-learn` meanwhile)
  - First upload *2022/12/24*
  - REJECT *2023/08/22* No source for data

Example for delayed process

- Pangolin: some AI software used to investigate COVID-19
- Packaging "competitors" like Conda distribute it instantly
- *pango-learn*
  - First upload *2021/04/06*
  - REJECT *2021/11/06* OMG, unused non-free stuff in source
  - Re-upload *2021/11/08* unfortunately new upstream version
  - REJECT *2022/02/16* OMG, exclusion failed in new upstream
  - Re-upload *2022/02/16*
  - REJECT *2022/10/28* big data file in /usr/lib (Accept + RC 2.1?)
  - Re-upload *2022/10/28* moving files somewhere else
  - REJECT *2022/12/21* license GPL-3.0 only? (Accept + RC 1.1)
  - *answer* to my question related to license
  - Re-upload *2022/12/21*
  - Finally accepted *2022/12/24*
- *pangolin-data* (superseeded *pango-learn* meanwhile)
  - First upload *2022/12/24*
  - REJECT *2023/08/22* No source for data

Example for delayed process

- Pangolin: some AI software used to investigate COVID-19
- Packaging "competitors" like Conda distribute it instantly
- `pango-learn`
  - First upload *2021/04/06*
  - REJECT *2021/11/06* OMG, unused non-free stuff in source
  - Re-upload *2021/11/08* unfortunately new upstream version
  - REJECT *2022/02/16* OMG, exclusion failed in new upstream
  - Re-upload *2022/02/16*
  - REJECT *2022/10/28* big data file in /usr/lib (Accept + RC 2.1?)
  - Re-upload *2022/10/28* moving files somewhere else
  - REJECT *2022/12/21* license GPL-3.0 only? (Accept + RC 1.1)
  - *answer* to my question related to license
  - Re-upload *2022/12/21*
  - Finally accepted *2022/12/24*
- `pangolin-data` (superseeded `pango-learn` meanwhile)
  - First upload *2022/12/24*
  - REJECT *2023/08/22* No source for data

Example for delayed process

- Pangolin: some AI software used to investigate COVID-19
- Packaging "competitors" like Conda distribute it instantly
- `pango-learn`
  - First upload *2021/04/06*
  - REJECT *2021/11/06* OMG, unused non-free stuff in source
  - Re-upload *2021/11/08* unfortunately new upstream version
  - REJECT *2022/02/16* OMG, exclusion failed in new upstream
  - Re-upload *2022/02/16*
  - REJECT *2022/10/28* big data file in /usr/lib (Accept + RC 2.1?)
  - Re-upload *2022/10/28* moving files somewhere else
  - REJECT *2022/12/21* license GPL-3.0 only? (Accept + RC 1.1)
  - *answer* to my question related to license
  - Re-upload *2022/12/21*
  - Finally accepted *2022/12/24*
- `pangolin-data` (superseeded `pango-learn` meanwhile)
  - First upload *2022/12/24*
  - REJECT *2023/08/22* No source for data

Example for delayed process

- Pangolin: some AI software used to investigate COVID-19
- Packaging "competitors" like Conda distribute it instantly
- `pango-learn`
  - First upload *2021/04/06*
  - REJECT *2021/11/06* OMG, unused non-free stuff in source
  - Re-upload *2021/11/08* unfortunately new upstream version
  - REJECT *2022/02/16* OMG, exclusion failed in new upstream
  - Re-upload *2022/02/16*
  - REJECT *2022/10/28* big data file in /usr/lib (Accept + RC 2.1?)
  - Re-upload *2022/10/28* moving files somewhere else
  - REJECT *2022/12/21* license GPL-3.0 only? (Accept + RC 1.1)
  - *answer* to my question related to license
  - Re-upload *2022/12/21*
  - Finally accepted *2022/12/24*
- `pangolin-data` (superseeded `pango-learn` meanwhile)
  - First upload *2022/12/24*
  - REJECT *2023/08/22* No source for data

Example for delayed process

- Pangolin: some AI software used to investigate COVID-19
- Packaging "competitors" like Conda distribute it instantly
- `pango-learn`
  - First upload *2021/04/06*
  - REJECT *2021/11/06* OMG, unused non-free stuff in source
  - Re-upload *2021/11/08* unfortunately new upstream version
  - REJECT *2022/02/16* OMG, exclusion failed in new upstream
  - Re-upload *2022/02/16*
  - REJECT *2022/10/28* big data file in /usr/lib (Accept + RC 2.1?)
  - Re-upload *2022/10/28* moving files somewhere else
  - REJECT *2022/12/21* license GPL-3.0 only? (Accept + RC 1.1)
  - *answer* to my question related to license
  - Re-upload *2022/12/21*
  - Finally accepted *2022/12/24*
- `pangolin-data` (superseeded `pango-learn` meanwhile)
  - First upload *2022/12/24*
  - REJECT *2023/08/22* No source for data

Example for delayed process

- Pangolin: some AI software used to investigate COVID-19
- Packaging "competitors" like Conda distribute it instantly
- *pango-learn*
  - First upload *2021/04/06*
  - REJECT *2021/11/06* OMG, unused non-free stuff in source
  - Re-upload *2021/11/08* unfortunately new upstream version
  - REJECT *2022/02/16* OMG, exclusion failed in new upstream
  - Re-upload *2022/02/16*
  - REJECT *2022/10/28* big data file in /usr/lib (Accept + RC 2.1?)
  - Re-upload *2022/10/28* moving files somewhere else
  - REJECT *2022/12/21* license GPL-3.0 only? (Accept + RC 1.1)
  - *answer* to my question related to license
  - Re-upload *2022/12/21*
  - Finally accepted *2022/12/24*
- *pangolin-data* (superseeded *pango-learn* meanwhile)
  - First upload *2022/12/24*
  - REJECT *2023/08/22* No source for data

Example for delayed process

- Pangolin: some AI software used to investigate COVID-19
- Packaging "competitors" like Conda distribute it instantly
- *pango-learn*
  - First upload *2021/04/06*
  - REJECT *2021/11/06* OMG, unused non-free stuff in source
  - Re-upload *2021/11/08* unfortunately new upstream version
  - REJECT *2022/02/16* OMG, exclusion failed in new upstream
  - Re-upload *2022/02/16*
  - REJECT *2022/10/28* big data file in /usr/lib (Accept + RC 2.1?)
  - Re-upload *2022/10/28* moving files somewhere else
  - REJECT *2022/12/21* license GPL-3.0 only? (Accept + RC 1.1)
  - *answer* to my question related to license
  - Re-upload *2022/12/21*
  - Finally accepted *2022/12/24*
- *pangolin-data* (superseeded *pango-learn* meanwhile)
  - First upload *2022/12/24*
  - REJECT *2023/08/22* No source for data

Example for delayed process

- Pangolin: some AI software used to investigate COVID-19
- Packaging "competitors" like Conda distribute it instantly
- `pango-learn`
  - First upload *2021/04/06*
  - REJECT *2021/11/06* OMG, unused non-free stuff in source
  - Re-upload *2021/11/08* unfortunately new upstream version
  - REJECT *2022/02/16* OMG, exclusion failed in new upstream
  - Re-upload *2022/02/16*
  - REJECT *2022/10/28* big data file in /usr/lib (Accept + RC 2.1?)
  - Re-upload *2022/10/28* moving files somewhere else
  - REJECT *2022/12/21* license GPL-3.0 only? (Accept + RC 1.1)
  - *answer* to my question related to license
  - Re-upload *2022/12/21*
  - Finally accepted *2022/12/24*
- `pangolin-data` (superseeded `pango-learn` meanwhile)
  - First upload *2022/12/24*
  - REJECT *2023/08/22* No source for data

Example for delayed process

- Pangolin: some AI software used to investigate COVID-19
- Packaging "competitors" like Conda distribute it instantly
- *pango-learn*
  - First upload *2021/04/06*
  - REJECT *2021/11/06* OMG, unused non-free stuff in source
  - Re-upload *2021/11/08* unfortunately new upstream version
  - REJECT *2022/02/16* OMG, exclusion failed in new upstream
  - Re-upload *2022/02/16*
  - REJECT *2022/10/28* big data file in /usr/lib (Accept + RC 2.1?)
  - Re-upload *2022/10/28* moving files somewhere else
  - REJECT *2022/12/21* license GPL-3.0 only? (Accept + RC 1.1)
  - *answer* to my question related to license
  - Re-upload *2022/12/21*
  - Finally accepted *2022/12/24*
- *pangolin-data* (superseeded *pango-learn* meanwhile)
  - First upload *2022/12/24*
  - REJECT *2023/08/22* No source for data

Example for delayed process

- Pangolin: some AI software used to investigate COVID-19
- Packaging "competitors" like Conda distribute it instantly
- *pango-learn*
  - First upload *2021/04/06*
  - REJECT *2021/11/06* OMG, unused non-free stuff in source
  - Re-upload *2021/11/08* unfortunately new upstream version
  - REJECT *2022/02/16* OMG, exclusion failed in new upstream
  - Re-upload *2022/02/16*
  - REJECT *2022/10/28* big data file in /usr/lib (Accept + RC 2.1?)
  - Re-upload *2022/10/28* moving files somewhere else
  - REJECT *2022/12/21* license GPL-3.0 only? (Accept + RC 1.1)
  - *answer* to my question related to license
  - Re-upload *2022/12/21*
  - Finally accepted *2022/12/24*
- *pangolin-data* (superseeded *pango-learn* meanwhile)
  - First upload *2022/12/24*
  - REJECT *2023/08/22* No source for data

Example for delayed process

- Pangolin: some AI software used to investigate COVID-19
- Packaging "competitors" like Conda distribute it instantly
- *pango-learn*
  - First upload *2021/04/06*
  - REJECT *2021/11/06* OMG, unused non-free stuff in source
  - Re-upload *2021/11/08* unfortunately new upstream version
  - REJECT *2022/02/16* OMG, exclusion failed in new upstream
  - Re-upload *2022/02/16*
  - REJECT *2022/10/28* big data file in /usr/lib (Accept + RC 2.1?)
  - Re-upload *2022/10/28* moving files somewhere else
  - REJECT *2022/12/21* license GPL-3.0 only? (Accept + RC 1.1)
  - *answer* to my question related to license
  - Re-upload *2022/12/21*
  - Finally accepted *2022/12/24*
- *pangolin-data* (superseeded *pango-learn* meanwhile)
  - First upload *2022/12/24*
  - REJECT *2023/08/22* No source for data

Example for delayed process

- Pangolin: some AI software used to investigate COVID-19
- Packaging "competitors" like Conda distribute it instantly
- `pango-learn`
  - First upload *2021/04/06*
  - REJECT *2021/11/06* OMG, unused non-free stuff in source
  - Re-upload *2021/11/08* unfortunately new upstream version
  - REJECT *2022/02/16* OMG, exclusion failed in new upstream
  - Re-upload *2022/02/16*
  - REJECT *2022/10/28* big data file in /usr/lib (Accept + RC 2.1?)
  - Re-upload *2022/10/28* moving files somewhere else
  - REJECT *2022/12/21* license GPL-3.0 only? (Accept + RC 1.1)
  - *answer* to my question related to license
  - Re-upload *2022/12/21*
  - Finally accepted *2022/12/24*
- `pangolin-data` (superseeded `pango-learn` meanwhile)
  - First upload *2022/12/24*
  - REJECT *2023/08/22* No source for data

Example for delayed process

- Pangolin: some AI software used to investigate COVID-19
- Packaging "competitors" like Conda distribute it instantly
- *pango-learn*
  - First upload *2021/04/06*
  - REJECT *2021/11/06* OMG, unused non-free stuff in source
  - Re-upload *2021/11/08* unfortunately new upstream version
  - REJECT *2022/02/16* OMG, exclusion failed in new upstream
  - Re-upload *2022/02/16*
  - REJECT *2022/10/28* big data file in /usr/lib (Accept + RC 2.1?)
  - Re-upload *2022/10/28* moving files somewhere else
  - REJECT *2022/12/21* license GPL-3.0 only? (Accept + RC 1.1)
  - *answer* to my question related to license
  - Re-upload *2022/12/21*
  - Finally accepted *2022/12/24*
- *pangolin-data* (superseeded *pango-learn* meanwhile)
  - First upload *2022/12/24*
  - REJECT *2023/08/22* No source for data

Grow your ideas for Debian Project #11

Rethink or clarify the objectives of NEW queue review

- Discussing paradoxes about unequal handling of packages that need to re-enter NEW queue due to binary name change
- Rather arbitrary, and slows down development
- Links to suggested alternatives
- There is no visible sign that ftpmaster has noticed this page

Grow your ideas for Debian Project #11

Rethink or clarify the objectives of NEW queue review

- Discussing paradoxes about unequal handling of packages that need to re-enter NEW queue due to binary name change
- Rather arbitrary, and slows down development
- Links to suggested alternatives
- There is no visible sign that ftpmaster has noticed this page

Grow your ideas for Debian Project #11

Rethink or clarify the objectives of NEW queue review

- Discussing paradoxes about unequal handling of packages that need to re-enter NEW queue due to binary name change
- Rather arbitrary, and slows down development
- Links to suggested alternatives
- There is no visible sign that ftpmaster has noticed this page

Grow your ideas for Debian Project #11

Rethink or clarify the objectives of NEW queue review

- Discussing paradoxes about unequal handling of packages that need to re-enter NEW queue due to binary name change
- Rather arbitrary, and slows down development
- Links to suggested alternatives
- There is no visible sign that ftpmaster has noticed this page

Comments about existing packages in NEW

- Ftpmaster checks SONAME changes and other technical stuff
- Nice to have this extra pair of eyeballs
- Also *`d/copyright`* is re-checked
- What makes packages that need some new binary name distinct from any other package?
- If someone thinks extra copyright check is needed make a random pick into the pool and file bug reports

Comments about existing packages in NEW

- Ftpmaster checks SONAME changes and other technical stuff
- Nice to have this extra pair of eyeballs
- Also *`d/copyright`* is re-checked
- What makes packages that need some new binary name distinct from any other package?
- If someone thinks extra copyright check is needed make a random pick into the pool and file bug reports

Comments about existing packages in NEW

- Ftpmaster checks SONAME changes and other technical stuff
- Nice to have this extra pair of eyeballs
- Also *`d/copyright`* is re-checked
- What makes packages that need some new binary name distinct from any other package?
- If someone thinks extra copyright check is needed make a random pick into the pool and file bug reports

Comments about existing packages in NEW

- Ftpmaster checks SONAME changes and other technical stuff
- Nice to have this extra pair of eyeballs
- Also `d/copyright` is re-checked
- What makes packages that need some new binary name distinct from any other package?
- If someone thinks extra copyright check is needed make a random pick into the pool and file bug reports

Comments about existing packages in NEW

- Ftpmaster checks SONAME changes and other technical stuff
- Nice to have this extra pair of eyeballs
- Also `d/copyright` is re-checked
- What makes packages that need some new binary name distinct from any other package?
- If someone thinks extra copyright check is needed make a random pick into the pool and file bug reports

Grow your ideas for Debian Project #16

Formally consult legal advice on current NEW queue

- Idea similarly expressed by *Russ Allbery on debian-devel*
- In short: What does the project loose if we are less strict about packages in new
- We might remove packages (at least if never part of any release) from unstable easily
- There is no visible sign that ftpmaster has noticed this page

➜ We probably need some risk based assessment

Grow your ideas for Debian Project #16

Formally consult legal advice on current NEW queue

- Idea similarly expressed by *Russ Allbery on debian-devel*
- In short: What does the project loose if we are less strict about packages in new
- We might remove packages (at least if never part of any release) from unstable easily
- There is no visible sign that ftpmaster has noticed this page

➜ We probably need some risk based assessment

Grow your ideas for Debian Project #16

Formally consult legal advice on current NEW queue

- Idea similarly expressed by *Russ Allbery on debian-devel*
- In short: What does the project loose if we are less strict about packages in new
- We might remove packages (at least if never part of any release) from unstable easily
- There is no visible sign that ftpmaster has noticed this page

➜ We probably need some risk based assessment

Grow your ideas for Debian Project #16

Formally consult legal advice on current NEW queue

- Idea similarly expressed by *Russ Allbery on debian-devel*
- In short: What does the project loose if we are less strict about packages in new
- We might remove packages (at least if never part of any release) from unstable easily
- There is no visible sign that ftpmaster has noticed this page

➜ We probably need some risk based assessment

Grow your ideas for Debian Project #16

Formally consult legal advice on current NEW queue

- Idea similarly expressed by *Russ Allbery on debian-devel*
- In short: What does the project loose if we are less strict about packages in new
- We might remove packages (at least if never part of any release) from unstable easily
- There is no visible sign that ftpmaster has noticed this page

➜ We probably need some risk based assessment

Multiple roles of d/copyright

https://lists.debian.org/debian-devel/2022/02/msg00177.html

- copyright information is "social", not "technical"
- licensing and copyright-information bugs hard to find
- copyright file is doing several (perhaps conflicting) things
  - inform consumers of packages about restrictions
  - declare DFSG compliance
  - essentially write-only information about licenses
  - gives authors due credit (in order to be good citizens)
  - meet our self-imposed rules (lets talk about those!)

Ftpmaster member: some suggestions need policy change

➜ So just lets talk about this

Multiple roles of d/copyright

https://lists.debian.org/debian-devel/2022/02/msg00177.html

- copyright information is "social", not "technical"
- licensing and copyright-information bugs hard to find
- copyright file is doing several (perhaps conflicting) things
  - inform consumers of packages about restrictions
  - declare DFSG compliance
  - essentially write-only information about licenses
  - gives authors due credit (in order to be good citizens)
  - meet our self-imposed rules (lets talk about those!)

Ftpmaster member: some suggestions need policy change

➜ So just lets talk about this

Multiple roles of d/copyright

https://lists.debian.org/debian-devel/2022/02/msg00177.html

- copyright information is "social", not "technical"
- licensing and copyright-information bugs hard to find
- copyright file is doing several (perhaps conflicting) things
  - inform consumers of packages about restrictions
  - declare DFSG compliance
  - essentially write-only information about licenses
  - gives authors due credit (in order to be good citizens)
  - meet our self-imposed rules (lets talk about those!)

Ftpmaster member: some suggestions need policy change

➜ So just lets talk about this

Multiple roles of d/copyright

https://lists.debian.org/debian-devel/2022/02/msg00177.html

- copyright information is "social", not "technical"
- licensing and copyright-information bugs hard to find
- copyright file is doing several (perhaps conflicting) things
  - inform consumers of packages about restrictions
  - declare DFSG compliance
  - essentially write-only information about licenses
  - gives authors due credit (in order to be good citizens)
  - meet our self-imposed rules (lets talk about those!)

Ftpmaster member: some suggestions need policy change

➜ So just lets talk about this

Multiple roles of d/copyright

https://lists.debian.org/debian-devel/2022/02/msg00177.html

- copyright information is "social", not "technical"
- licensing and copyright-information bugs hard to find
- copyright file is doing several (perhaps conflicting) things
  - inform consumers of packages about restrictions
  - declare DFSG compliance
  - essentially write-only information about licenses
  - gives authors due credit (in order to be good citizens)
  - meet our self-imposed rules (lets talk about those!)

Ftpmaster member: some suggestions need policy change

➜ So just lets talk about this

Multiple roles of d/copyright

https://lists.debian.org/debian-devel/2022/02/msg00177.html

- copyright information is "social", not "technical"
- licensing and copyright-information bugs hard to find
- copyright file is doing several (perhaps conflicting) things
  - inform consumers of packages about restrictions
  - declare DFSG compliance
  - essentially write-only information about licenses
  - gives authors due credit (in order to be good citizens)
  - meet our self-imposed rules (lets talk about those!)

Ftpmaster member: some suggestions need policy change

➜ So just lets talk about this

Multiple roles of d/copyright

https://lists.debian.org/debian-devel/2022/02/msg00177.html

- copyright information is "social", not "technical"
- licensing and copyright-information bugs hard to find
- copyright file is doing several (perhaps conflicting) things
  - inform consumers of packages about restrictions
  - declare DFSG compliance
  - essentially write-only information about licenses
  - gives authors due credit (in order to be good citizens)
  - meet our self-imposed rules (lets talk about those!)

Ftpmaster member: some suggestions need policy change

➜ So just lets talk about this

Multiple roles of d/copyright

https://lists.debian.org/debian-devel/2022/02/msg00177.html

- copyright information is "social", not "technical"
- licensing and copyright-information bugs hard to find
- copyright file is doing several (perhaps conflicting) things
  - inform consumers of packages about restrictions
  - declare DFSG compliance
  - essentially write-only information about licenses
  - gives authors due credit (in order to be good citizens)
  - meet our self-imposed rules (lets talk about those!)

Ftpmaster member: some suggestions need policy change

➜ So just lets talk about this

Multiple roles of d/copyright

https://lists.debian.org/debian-devel/2022/02/msg00177.html

- copyright information is "social", not "technical"
- licensing and copyright-information bugs hard to find
- copyright file is doing several (perhaps conflicting) things
  - inform consumers of packages about restrictions
  - declare DFSG compliance
  - essentially write-only information about licenses
  - gives authors due credit (in order to be good citizens)
  - meet our self-imposed rules (lets talk about those!)

Ftpmaster member: some suggestions need policy change

➜ So just lets talk about this

Multiple roles of d/copyright

https://lists.debian.org/debian-devel/2022/02/msg00177.html

- copyright information is "social", not "technical"
- licensing and copyright-information bugs hard to find
- copyright file is doing several (perhaps conflicting) things
  - inform consumers of packages about restrictions
  - declare DFSG compliance
  - essentially write-only information about licenses
  - gives authors due credit (in order to be good citizens)
  - meet our self-imposed rules (lets talk about those!)

Ftpmaster member: some suggestions need policy change

➜ So just lets talk about this

Hints to deal with the current situation

- Bounce ftpmaster REJECT mail to ITP bug and keep the bug in CC in my answers
- Ftpmaster could CC ITP bug but not all packages in NEW have an according bug
- Having the reject issue in form of a `find|grep|sed`-ish command would be great

Hints to deal with the current situation

- Bounce ftpmaster REJECT mail to ITP bug and keep the bug in CC in my answers
- Ftpmaster could CC ITP bug but not all packages in NEW have an according bug
- Having the reject issue in form of a `find|grep|sed`-ish command would be great

Hints to deal with the current situation

- Bounce ftpmaster REJECT mail to ITP bug and keep the bug in CC in my answers
- Ftpmaster could CC ITP bug but not all packages in NEW have an according bug
- Having the reject issue in form of a `find|grep|sed`-ish command would be great

Slides will be on my talks page later

Andreas Tille <tille@debian.org>