PLEX

Plex  Conan

How to kill a dragon

Tobias Hieta

Dragon killer, cheese mover and likes to open cans that contains worms

PLEX

WTF is Plex?

PLEX

The leading media streaming software platform

PLEX

WTF is Conan?

PLEX

This is a story about love!

PLEX

Plex Media Server

PLEX

PLEX

60+ dependenc

Boost, zlib, freetype, ffmpeg, bzip, curl, expat, freeima

...

PLEX

JavaScript developer ->

PLEX

1269 dependencies

PLEX

27 targets

Linux (lot of different architectures), Android, iOS, macOS, Windows, FreeBSD

PLEX

Compilers …

PLEX

Standard libraries

PLEX

Many* build systems

*= I think the scientific unit is ‘a fuckton’.

CMake, Auto~~tools~~hell, SCons, Make, Visual Studio, ndk-make, meson, build2, waf and whatever that boost thing is

PLEX

They all suck

CMake probably sucks least.

PLEX

Compile

PLEX

plex-dependency-builder

PLEX

Rebuild the world

PLEX

scratchbox2

PLEX

Hidden changes

PLEX

Frustrated and angry

PLEX

(We need)

A new hope

PLEX

Individual packages

PLEX

Individual packages

Handle multiple build-systems

PLEX

Individual packages
Handle multiple build-systems

Cross-compile

PLEX

Individual packages
Handle multiple build-systems
Cross-compile

Manage deps and toolchain

PLEX

Individual packages
Handle multiple build-systems
Cross-compile
Manage deps and toolchain

Flexible

PLEX

Individual packages
Handle multiple build-systems
Cross-compile
Manage deps and toolchain
Flexible

Reproducible builds

PLEX

Buckaroo won

#shocking

PLEX

But...

PLEX

Conan really won

PLEX

Unified toolchain

PLEX

GCC ⚔ Clang

PLEX

Single binary
Multiple targets

PLEX

A single compiler

to rule them all.

PLEX

Cool logo

PLEX

16 targets
2 compilers

PLEX

Bumps on the road

PLEX

PlexConanTool

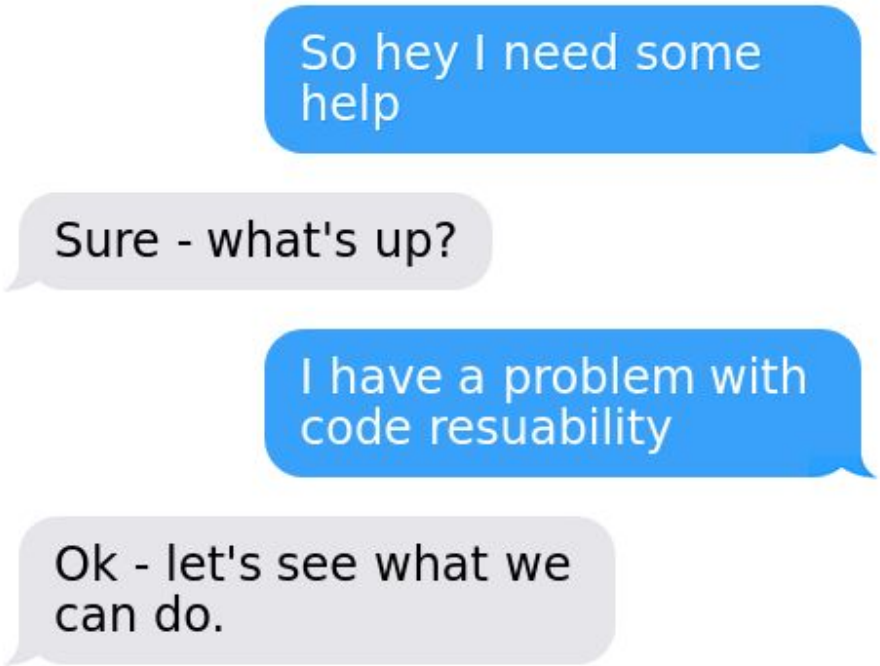
PLEX

```
def source(self):
  tools.download("http://www.libsdl.org/release/SDL2-2.0.tar.gz", "sdl.tar.gz")
  tools.unzip("sdl.tar.gz", ".")
  os.unlink("sdl.tar.gz")

def build(self):
  cmake = CMake(self.settings)
  cmakeflags = "-DSDL_Audio=OFF -DSDL_Video=OFF -DSDL_Render=OFF -DSDL_Power=OFF" \
               "-DSDL_CPUInfo=OFF -DSDL_Threads=OFF -DSDL_Loadso=OFF -DSDL_TImer=OFF -DDIRECTX=OFF -DSDL_SHARED=OFF"
  self.run("cmake %s -DCMAKE_INSTALL_PREFIX=%s -DSDL_Audio=OFF %s %s" % (cmakeflags,
      os.path.join(self.conanfile_directory, "install"),
      os.path.join(self.conanfile_directory, "SDL2-%s" % self.version),
    cmake.command_line))
  self.run("cmake --build . %s" % cmake.build_config)
  self.run("cmake --build . --target install")
```

Initial SDL file - lot of things in there.

PLEX


So hey I need some help
Sure - what's up?
I have a problem with code resuability
Ok - let's see what we can do.

PLEX

```
requires = "plexconantool/1.0"

def source(self):
  with tools.pythonpath(self):
    import plexconantool
    plexconantool.download(self, "...")

def build(self):
  with tools.pythonpath(self):
    import plexconantool
    cmakeflags = ["-DSDL_Audio=OFF", "-DSDL_Video=OFF", "-DSDL_Render=OFF", "-DSDL_Power=OFF"]
    plexconantool.cmake(self, cmakeflags)
    plexconantool.cmake_build(self)
```

Conan feature for code sharing - much better

PLEX

Profiles

PLEX

```
CC="clang" CFLAGS="-g -mmacosx-version-min=10.8"\
  conan test_package -s os=macos -s compiler=clang -s arch=x86_64
```

Everything was controlled with env variables.

PLEX


Hey - this whole thing with managing compiler flags and settings in the environment is pretty annoying.
Not that bad I think?
Well actually...

PLEX

```
[settings]
compiler=clang
compiler.version=5.0
compiler.libcxx=libc++
os=Macos
arch=x86_64
build_type=Release

[env]
CC=clang
CXX=ccache clang++
CFLAGS=-mmacosx-version-min=10.8 -g
```

A profile for macos

PLEX

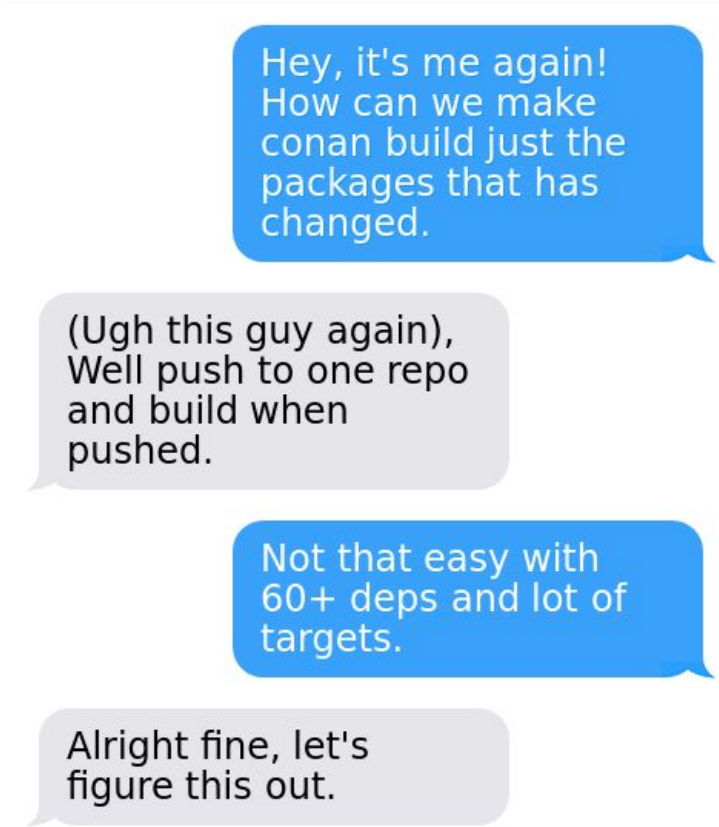
```
conan test_package -pr profiles/macos-x86_64
```

Much easier!

PLEX

Build what’s updated

PLEX


Hey, it's me again! How can we make conan build just the packages that has changed.
(Ugh this guy again), Well push to one repo and build when pushed.
Not that easy with 60+ deps and lot of targets.
Alright fine, let's figure this out.

PLEX

```
conan install -pr profile --build=outdated variants/standard
```

build=outdated was added

PLEX

```
argtable2/git-79dac1b-4@plex/stable: Package is up to date
breakpad/1.0-9d61e90-2@plex/stable: Package is up to date
bzip2/1.0.6-3@plex/stable: Package is up to date
cotire/1.8.0-391bf6b-0@plex/stable: Package is up to date
zlib/1.2.8-5@plex/stable: Package is up to date
argtable2/git-79dac1b-4@plex/stable: Already installed!
breakpad/1.0-9d61e90-2@plex/stable: Already installed!
bzip2/1.0.6-3@plex/stable: Already installed!
cotire/1.8.0-391bf6b-0@plex/stable: Already installed!
ffmpeg/3.3-8c220a2dd-0@plex/stable: Installing build requirements of: ffmpeg/3.3-8c220a2dd-0@plex/stable
```

Only processing changed packages

PLEX

Automation

PLEX


Heeey. So outdated is great and all - but how do we know what worked and not?
Seriously? We have other things to do here.
Pretty pleaaaase?
Hello? Please can we do this?
I'll owe you beer!
Ok fine

PLEX

```
conan install --build=outdated --json info.json -pr profile variants/standard
```

--json parameter was added

PLEX

Now we can parse this

PLEX

Reproducible builds

PLEX

zlib/1.2.8@plex/stable

PLEX

zlib/1.2.8-1@plex/stable

PLEX

```
class Bzip2Conan(PlexConanFile):
  name = "bzip2"
  plex_version = "1.0.6"
  plex_revision = 4
```

PLEX

PlexConanFile

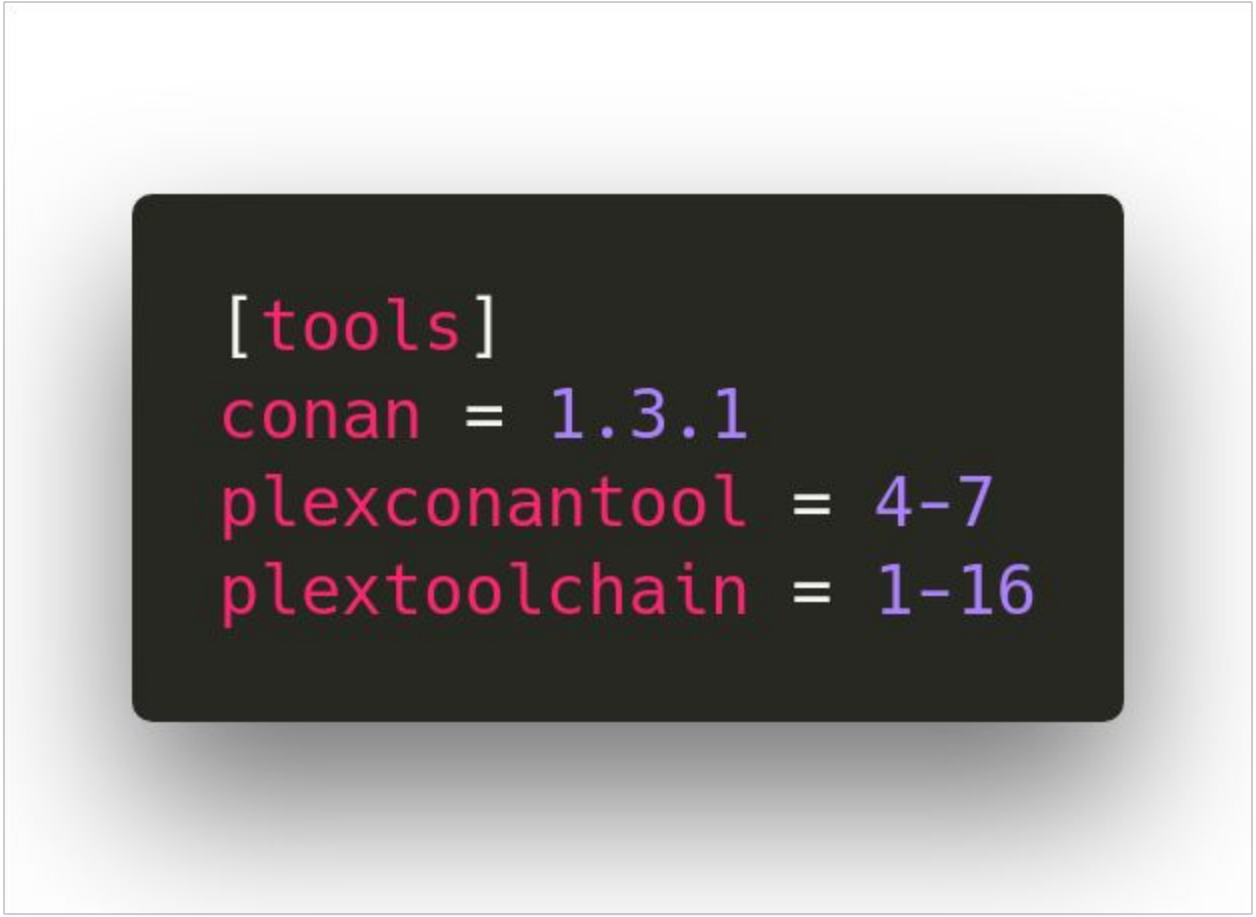
PLEX

Two phase setup

PLEX

conan_init.py

PLEX



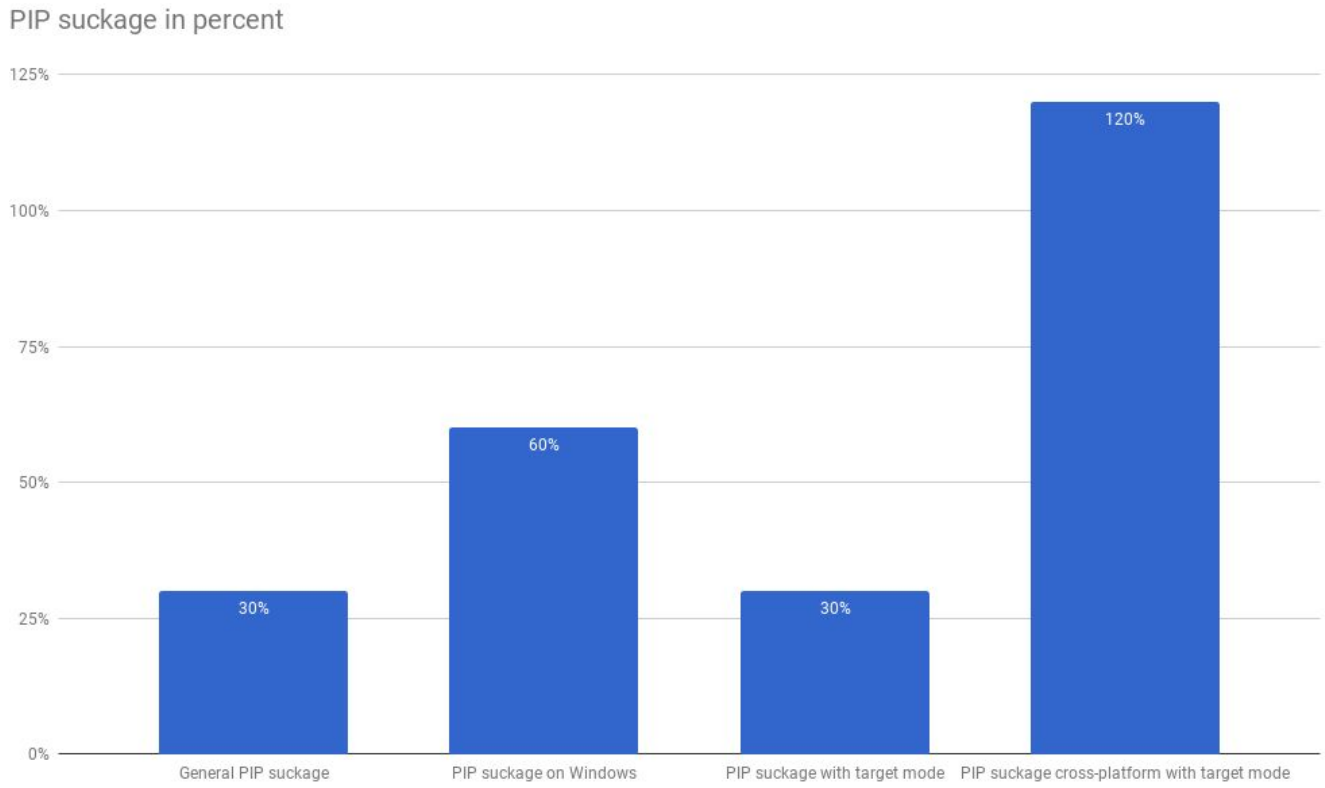
PLEX

Consistent environment

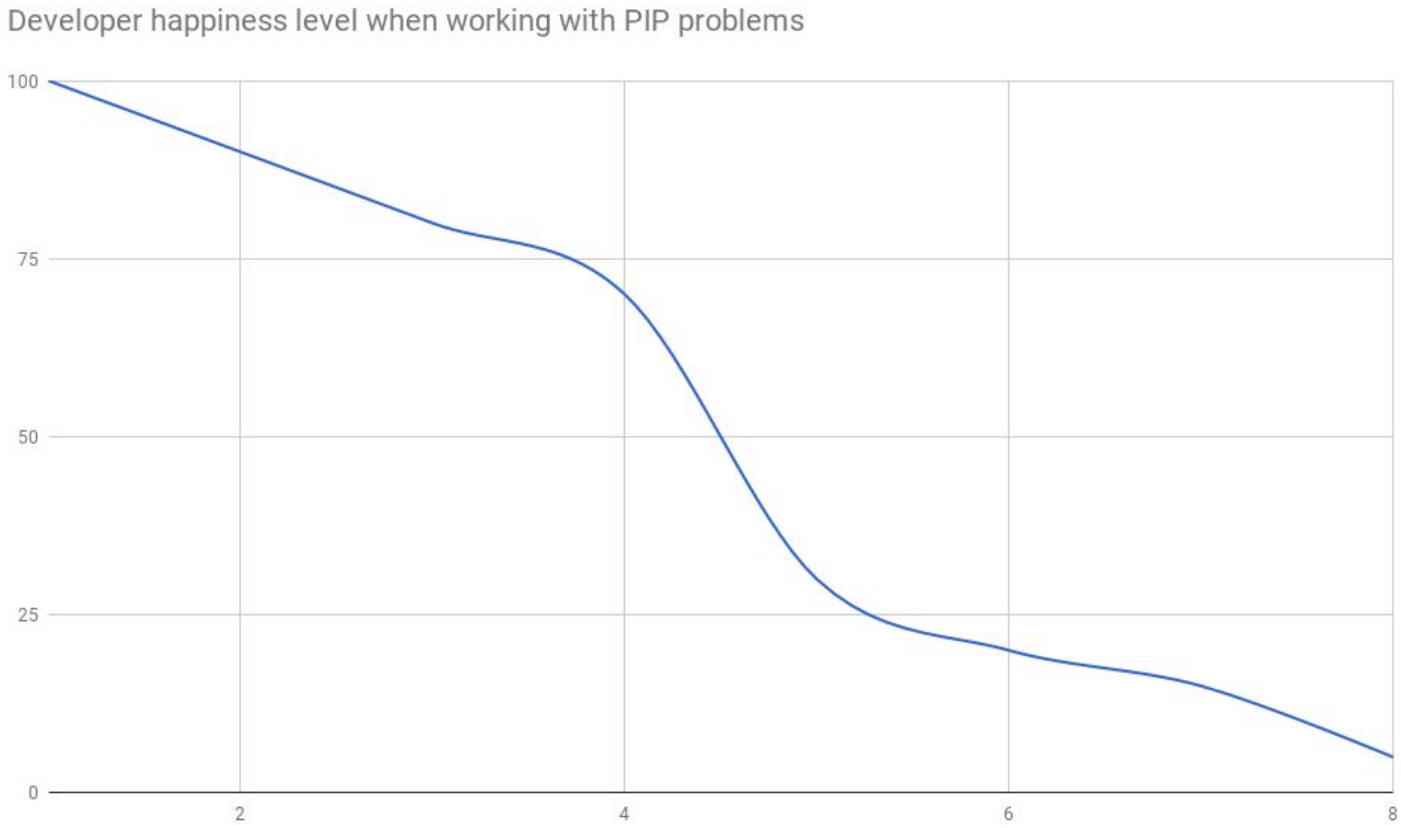
PLEX

Don’t try this at home

PLEX


PIP suckage in percent
125%
100%
75%
50%
25%
0%
30%
60%
30%
120%
General PIP suckage
PIP suckage on Windows
PIP suckage with target mode
PIP suckage cross-platform with target mode

PLEX


Developer happiness level when working with PIP problems
100
75
50
25
0
2
4
6
8

PLEX

Back on track

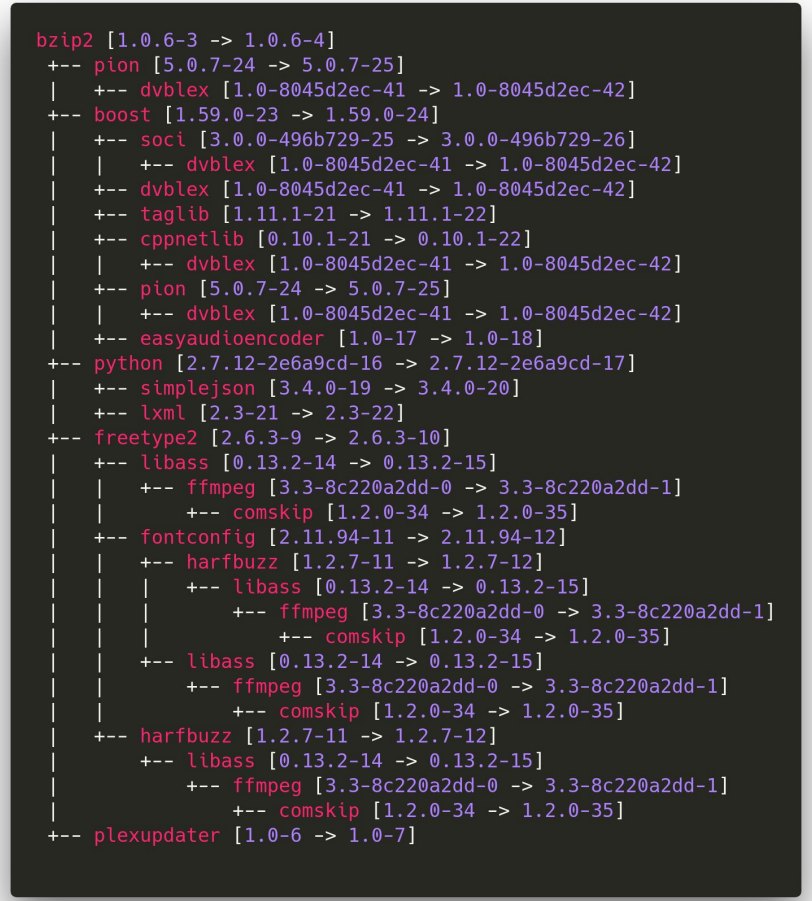
PLEX

Don’t trust manual revisions

PLEX

update_pkg_verison.py

PLEX


bzip2 [1.0.6-3 -> 1.0.6-4]
+-- pion [5.0.7-24 -> 5.0.7-25]
| +-- dvblex [1.0-8045d2ec-41 -> 1.0-8045d2ec-42]
+-- boost [1.59.0-23 -> 1.59.0-24]
| +-- soci [3.0.0-496b729-25 -> 3.0.0-496b729-26]
| | +-- dvblex [1.0-8045d2ec-41 -> 1.0-8045d2ec-42]
| +-- dvblex [1.0-8045d2ec-41 -> 1.0-8045d2ec-42]
| +-- taglib [1.11.1-21 -> 1.11.1-22]
| +-- cppnetlib [0.10.1-21 -> 0.10.1-22]
| | +-- dvblex [1.0-8045d2ec-41 -> 1.0-8045d2ec-42]
| +-- pion [5.0.7-24 -> 5.0.7-25]
| | +-- dvblex [1.0-8045d2ec-41 -> 1.0-8045d2ec-42]
| +-- easyaudioencoder [1.0-17 -> 1.0-18]
+-- python [2.7.12-2e6a9cd-16 -> 2.7.12-2e6a9cd-17]
| +-- simplejson [3.4.0-19 -> 3.4.0-20]
| +-- lxml [2.3-21 -> 2.3-22]
+-- freetype2 [2.6.3-9 -> 2.6.3-10]
| +-- libass [0.13.2-14 -> 0.13.2-15]
| | +-- ffmpeg [3.3-8c220a2dd-0 -> 3.3-8c220a2dd-1]
| | +-- comskip [1.2.0-34 -> 1.2.0-35]
| +-- fontconfig [2.11.94-11 -> 2.11.94-12]
| | +-- harfbuzz [1.2.7-11 -> 1.2.7-12]
| | | +-- libass [0.13.2-14 -> 0.13.2-15]
| | | +-- ffmpeg [3.3-8c220a2dd-0 -> 3.3-8c220a2dd-1]
| | | +-- comskip [1.2.0-34 -> 1.2.0-35]
| | +-- libass [0.13.2-14 -> 0.13.2-15]
| | +-- ffmpeg [3.3-8c220a2dd-0 -> 3.3-8c220a2dd-1]
| | +-- comskip [1.2.0-34 -> 1.2.0-35]
| +-- harfbuzz [1.2.7-11 -> 1.2.7-12]
| +-- libass [0.13.2-14 -> 0.13.2-15]
| +-- ffmpeg [3.3-8c220a2dd-0 -> 3.3-8c220a2dd-1]
| +-- comskip [1.2.0-34 -> 1.2.0-35]
+-- plexupdater [1.0-6 -> 1.0-7]

PLEX

Blizz

PLEX

Thanks!

@tobiashieta

PLEX

devstory

PLEX