DebConf20

How to use **apt-repos** to **inspect multiple apt repositories?**

2020-08-23
Christoph Lutz

About...

...the speaker = the author:

Christoph Lutz

2005 – 2011:
→ LiMux Office Team

2015 – 2019:
→ LiMux Developer

2020 – Nowadays:

→ still using apt-repos in the context of my new job in a commercial company

...the tool **apt-repos**:

written for Munich‘s LiMux Project

https://github.com/chrlutz/apt-repos

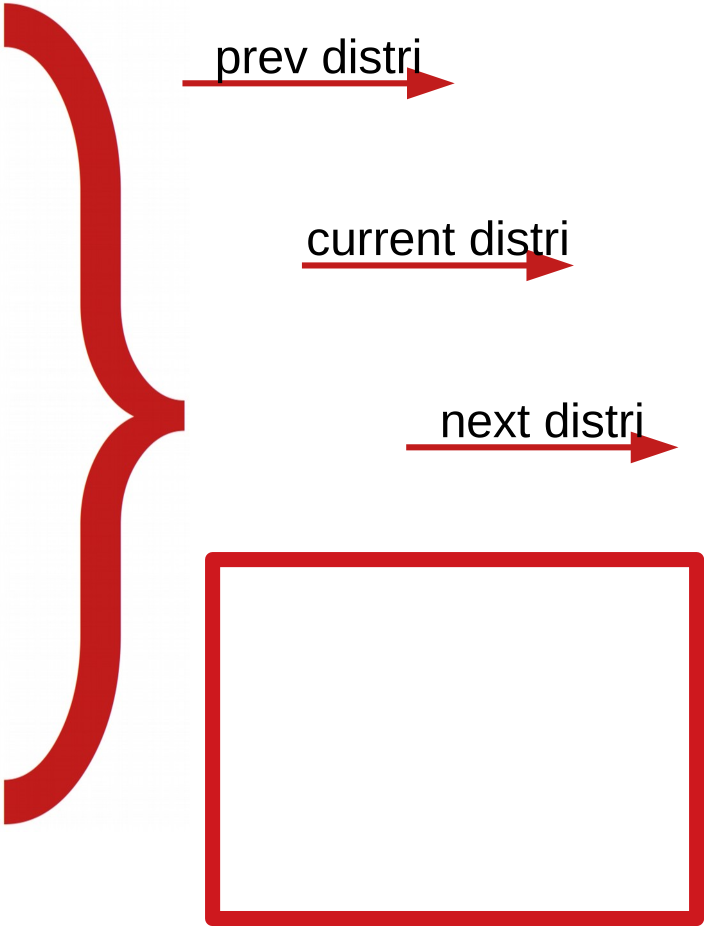
Useful for maintainers of debian based Linux distris to inspect apt repositories

LGPL v2.1

What you might want to inspect

as maintainer of a debian based Linux Distribution:

- Content of used **Upstream-Suites** (main-, security-, updates- and backports-Suites) and further **PPAs**
- **Business specific third party packages**
- **Custom, derived and adjusted** packages
- Packages & Versions in suites reflecting different **development stages** (eg. develop, testing, production)
- Different **mirrors** of the same things

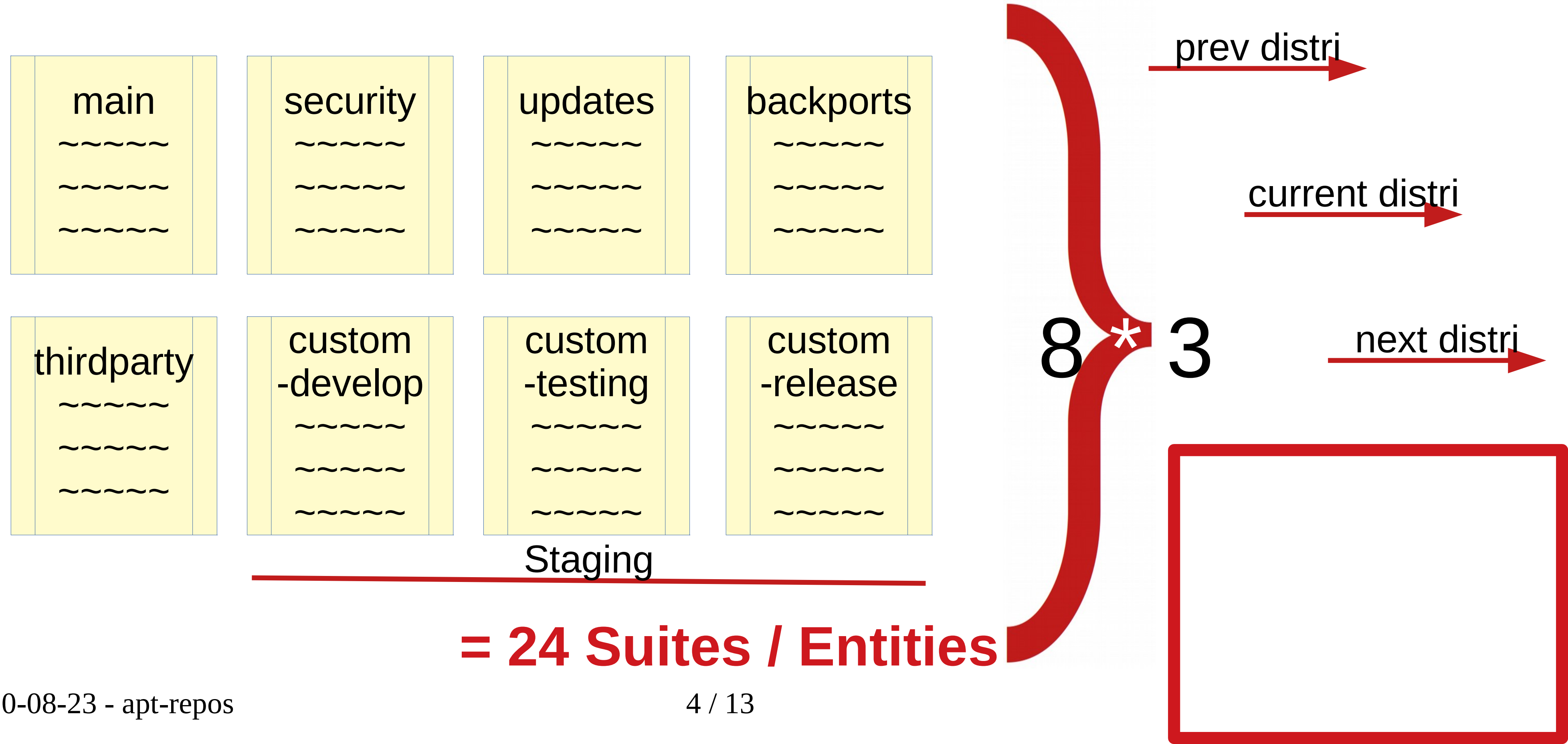


prev distri

current distri

next distri

What you might want to inspect



main ~~~~~~ ~~~~~~ ~~~~~~	security ~~~~~~ ~~~~~~ ~~~~~~	updates ~~~~~~ ~~~~~~ ~~~~~~	backports ~~~~~~ ~~~~~~ ~~~~~~
thirdparty ~~~~~~ ~~~~~~ ~~~~~~	custom -develop ~~~~~~ ~~~~~~ ~~~~~~	custom -testing ~~~~~~ ~~~~~~ ~~~~~~	custom -release ~~~~~~ ~~~~~~ ~~~~~~

Staging

8 * 3

prev distri

current distri

next distri

= 24 Suites / Entities

Questions you might want to answer

- Which **suites** do we have in our current distribution?
- **Where is** ‚git‘ and in which version?
- **From which** upsteam repository did we import ‚anacron‘?
- Which update packages do we provide (**list a repo/suite**)?
- Is my mirror of ‚buster-updates‘ **up to date (compare suites)**?
- Is there a **changelog** for my custom repository?

How to answer? Approach 1

Manually browse the APT-Repository Structure

http://deb.debian.org/debian

/dists/ → Packages.gz

/pool/

→ No efficient enough!

How to answer? Approach 2

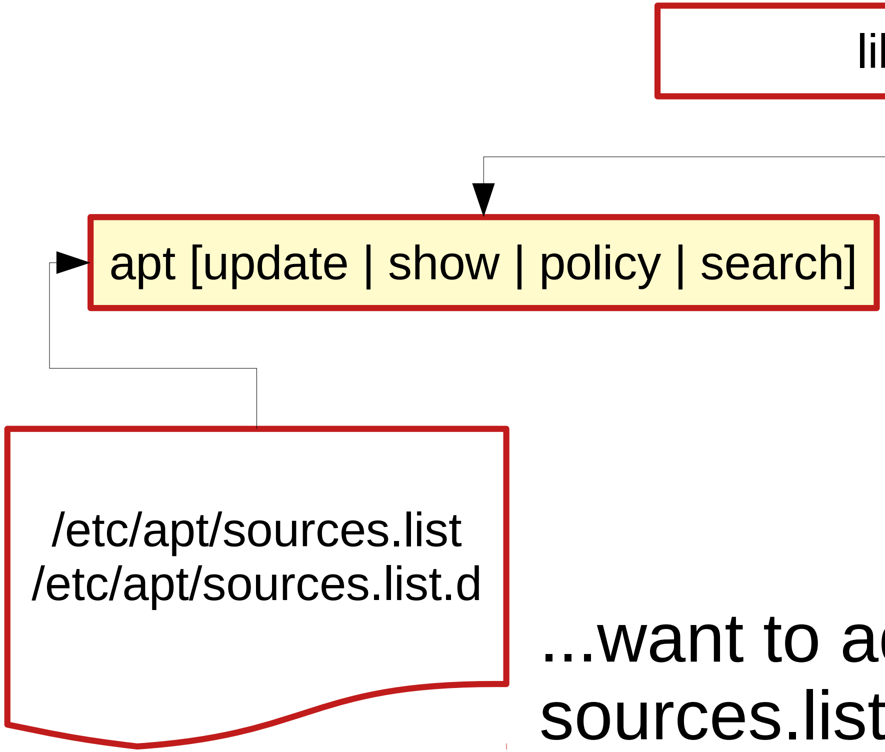
Use inspection provided by your repository management tool? e.g.

- **DAK:** dak ls …
- **Aptly:** aptly repo --with-packages list …
- **Reprepro:** reprepro list …
- **Debmirror?** → just mirroring, no inspection?

Issues:

- Typically needs to be run on the repository server
  --> Most repos are open! Want to inspect from anywhere!
- Different tools, syntax and switches for inspection…
- …**why can‘t we just use a tool to inspect the generic apt repository structure?**

How to answer? Approach 3



libapt-pkg(5)

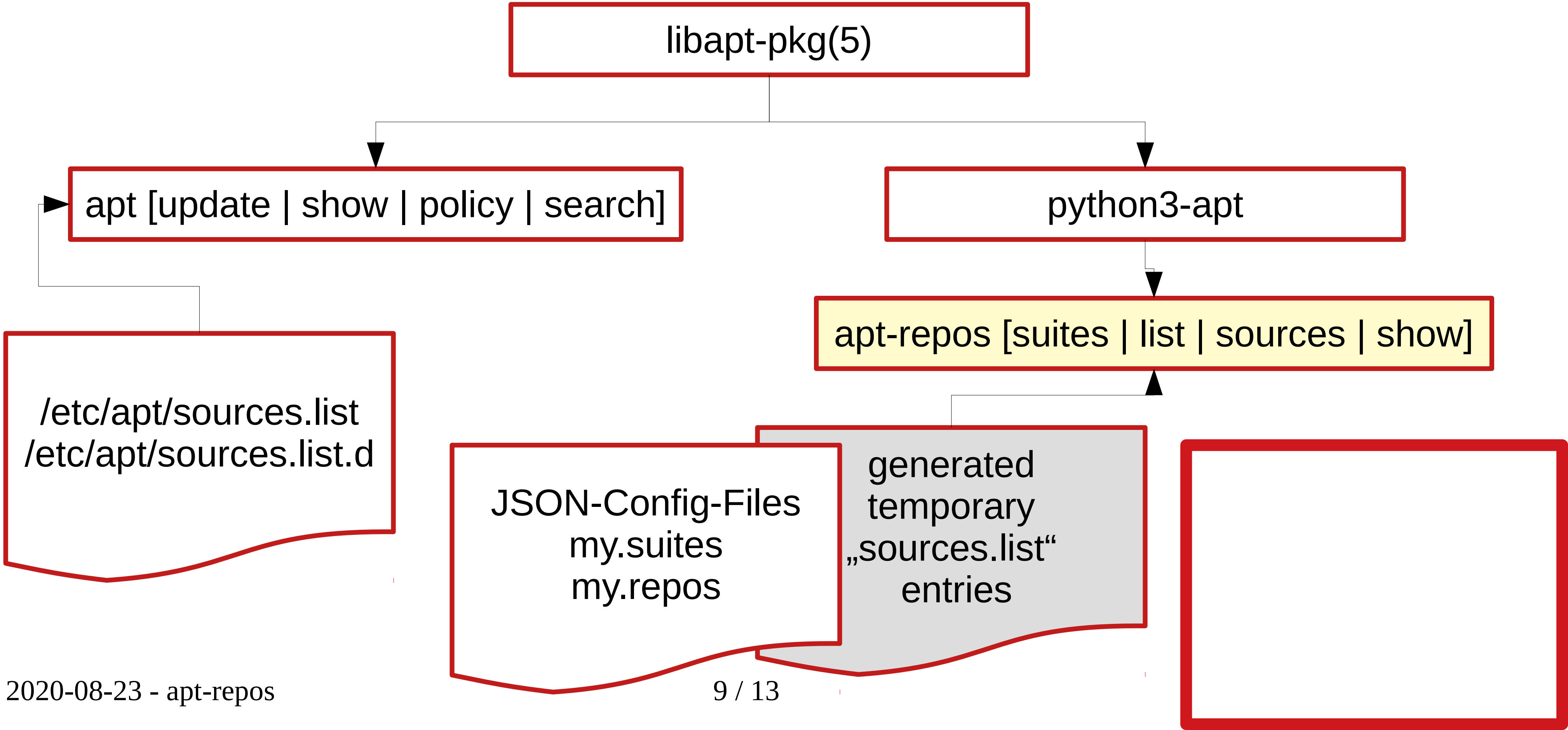
apt [update | show | policy | search]

/etc/apt/sources.list
/etc/apt/sources.list.d

APT can do that, but...

...want to add 24 suites to your sources.list and modify your local system?

How to answer? apt-repos!



libapt-pkg(5)

apt [update | show | policy | search]

python3-apt

apt-repos [suites | list | sources | show]

/etc/apt/sources.list
/etc/apt/sources.list.d

JSON-Config-Files
my.suites
my.repos

generated
temporary
„sources.list“
entries

DEMO

DEMO (1 / 2)

- Which **suites** do we have for our current distribution?

  → apt-repos suites

- **Where is** ‚git‘ and in which version?

  → apt-repos ls -s debian: git

- **From which** upsteam repository did we import ‚anacron‘?

  → apt-repos ls -s : anacron

- Which update pkgs do we provide (**list a suite**)?

  → apt-repos ls -s ubuntu:focal-updates -r .

DEMO (2 / 2)

- Is my mirror of ‚buster-updates‘ **up to date (compare suites)**?
  → apt-repos ls -s ref:suite,dest:suite -r . -di s [-dt meld]
- Is there a **changelog** for my custom repository?
  → apt-repos ls -s my:suite -r . --format=list >git-versioned.list
- What‘s the **source name** of my custom package?
  → apt-repos ls custom-package
- Which other **packages** derive **from that source**?
  → apt-repos ls src:custom-src

Thanks for listening

and have fun at the conference!