## Tails - Feature #6220

Automated Debian package build infrastructure

08/07/2013 01:10 PM - intrigeri

<table>
<thead>
<tr>
<th>Status:</th>
<th>Confirmed</th>
<th>Start date:</th>
<th>08/07/2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
<td>Due date:</td>
<td></td>
</tr>
<tr>
<td>Assignee:</td>
<td>bertagaz</td>
<td>% Done:</td>
<td>0%</td>
</tr>
<tr>
<td>Category:</td>
<td>Continuous Integration</td>
<td>Estimated time:</td>
<td>0.00 hour</td>
</tr>
<tr>
<td>Target version:</td>
<td></td>
<td>Spent time:</td>
<td>0.00 hour</td>
</tr>
<tr>
<td>Feature Branch:</td>
<td></td>
<td>Starter:</td>
<td>No</td>
</tr>
<tr>
<td>Type of work:</td>
<td>Sysadmin</td>
<td>Affected tool:</td>
<td></td>
</tr>
<tr>
<td>Blueprint:</td>
<td><a href="https://tails.boum.org/blueprint/automated_builds_and_tests/Debian_packages/">https://tails.boum.org/blueprint/automated_builds_and_tests/Debian_packages/</a></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Description

It's been done before, no need to reinvent the wheel. See the blueprint for potentially useful tools.

Team: bertagaz

### Related issues:

<table>
<thead>
<tr>
<th>Issue</th>
<th>Status</th>
<th>Start date</th>
<th>Due date</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>Resolved</td>
<td>07/26/2013</td>
<td>02/28/2015</td>
</tr>
<tr>
<td>#2</td>
<td>Confirmed</td>
<td>08/26/2017</td>
<td></td>
</tr>
<tr>
<td>#3</td>
<td>Rejected</td>
<td>08/07/2013</td>
<td></td>
</tr>
<tr>
<td>#4</td>
<td>In Progress</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### History

#### #1 - 04/04/2014 12:03 AM - Dr_Whax

The Debile link is not working, this one is: [http://anonscm.debian.org/gitweb/?p=pkg-debile/debile.git](http://anonscm.debian.org/gitweb/?p=pkg-debile/debile.git)

#### #2 - 04/04/2014 09:35 AM - intrigeri

- Description updated

#### #3 - 04/04/2014 09:38 AM - intrigeri

- Description updated
- Blueprint set to https://tails.boum.org/blueprint/automated_builds_and_tests/#resources-debian-pkg

#### #4 - 08/05/2014 12:31 PM - intrigeri


#### #5 - 08/14/2014 05:15 AM - intrigeri

- Related to Feature #7036: Move custom software to main git repo added

#### #6 - 08/14/2014 05:16 AM - intrigeri

- Related to Feature #6090: Automated builds added

#### #7 - 11/03/2014 09:37 PM - sajolida

- Target version set to Sustainability_M1

#### #8 - 08/14/2015 11:48 AM - sajolida

- Description updated

05/04/2020
At this point I'm happy we did **not** set this up server-side, in a centralized manner, because I've entirely changed my mind wrt. how we should approach the problem at hand.

IMO we should build our custom packages as part of building an ISO (using good ol' sbuild/pbuilder or something that better isolates package builds from one another, such as [https://gitlab.com/uhoreg/whalebuilder](https://gitlab.com/uhoreg/whalebuilder) or one of the other options), and track their source code as Git submodules i.e. [#7036](https://gitlab.com/uhoreg/whalebuilder) for non-native packages, likely we want to track the branch that contains the Debian packaging).

This advantages I see with this approach, compared to what we currently do, and to extending our central infrastructure to build packages automatically:

1. We get more value out of reproducible ISO builds, by removing binary input and the required trust in the content of these packages, in the developer who built+uploaded them, and in our custom APT repo infrastructure.
2. Developers who don't have Git commit access can work on our custom software and build an ISO with their modified version in a straightforward way (no need to have a pbuilder setup nor to copy .deb's to config/chroot_local-packages, and the result of the work can be handled as any other Git pull request).
3. No need for all developers to set up and maintain a local pbuilder setup with N chroots.
4. We can get rid of our custom APT repository and, perhaps more importantly, of the complex & error-prone workflow that comes with it (e.g. for merging/resetting APT suites). Everything related to config/APT_overlays.d/ can go away, which simplifies our codebase and infrastructure.
5. More parameters that affect the resulting ISO are encoded in Git, which makes them easier to audit, log, and diff. The only remaining thing that's not encoded in Git is the **content** of the APT snapshots we use (we already encode in Git which set of APT snapshots we use).
6. We simplify what it means to have Git commit access: developers who have Git commit access **don't also** need upload rights to our custom APT repo. This simplifies sysadmin (no need to keep OpenPGP keys up-to-date) and makes it easier for developers to understand how the whole thing works.
7. The apt-cacher-ng cache can be shared between ISO build process and .deb packages build process.
8. We get finer-grained control over what exact state of the pbuilder/sbuild/whalebuilder chroot we use for building packages. E.g. we might want to use exactly the same APT snapshots that are used for building the rest of the ISO, which can matter a lot if we switch to Debian testing (#12615): building against current Debian testing can create packages that are not installable on the (older) snapshot of Debian testing we use in the SquashFS.
9. More of us have the skills and access needed to work, within our Git tree, on such a Debian packages build system, than to work on a centralized package build system hosted our infrastructure. This implies that we have more options wrt. who implements, reviews & maintains the new thing, instead of increasing our reliance on a tiny sysadmin team that would use a totally different stack (Puppet, Jenkins, jenkins-debian-glue, etc.).
10. Compared to what we do today, we have the option to enforce some quality checks on the packaging and upstream code (e.g. Lintian or autopkgtests).
11. No need to design & implement yet another mapping between Git branches, Jenkins jobs and APT suites.
12. No race conditions / async design skills required: we're guaranteed that the expected version of every custom package we want has been built when we need it.
#12 - 09/28/2017 12:14 PM - intrigeri
- Target version deleted (2018)

(as per updated roadmap)

#13 - 10/02/2017 10:53 AM - intrigeri
See also [https://tails.boum.org/blueprint/reproducible_builds/#custom-Debian-packages](https://tails.boum.org/blueprint/reproducible_builds/#custom-Debian-packages).

#14 - 10/03/2017 06:28 AM - intrigeri
- Related to Feature #14455: Reproducible Builds Stage 2 added

#15 - 08/19/2018 08:46 AM - Anonymous
Inspiration: [https://auto.debian.net/](https://auto.debian.net/)

#16 - 08/19/2018 08:48 AM - Anonymous
- Related to Feature #6219: Document our autobuild setup added

#17 - 09/19/2019 06:24 AM - intrigeri
- Related to deleted (Feature #7036: Move custom software to main git repo)

#18 - 09/19/2019 06:24 AM - intrigeri
- Blocked by Feature #7036: Move custom software to main git repo added

#19 - 09/19/2019 06:24 AM - intrigeri
It might be that once we have #7036, there's very few reasons left (if any) to bother doing what this very ticket is about.