Tutorial: Functional Testing of Debian Packages

Antonio Terceiro
terceiro@debian.org
history
automated tests are great and we need more of it
automated tests don't replace all manual testing
Topics

- history
- specs
- tools
- functional tests by example
autodebtest (0.5.0) unstable; urgency=low

* Initial release [...].

-- Ian Jackson 〈iwj@...〉> Fri, 20 Jan 2006 ...
Title: autopkgtest - automatic as-installed package testing

Abstract:
Establish a **standard interface** to define and run "as-installed" **tests of packages**, i.e. the testing of packages in a context as close as possible to a Debian system where the **packages subject to testing are properly installed**.
( that is great, someone )
( should run that on the )
.o0( whole archive ... )
Debian Continuous Integration comes alive in January 2014

then $1\frac{1}{2}$ year, 2 GSOC projects, and lots of hacking time after ...
packages in ci.debian.net
~18.5% of the source packages in the archive is currently being tested up from ~1% in January 2014
<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Duration</th>
<th>Status</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.12</td>
<td>2015-08-02 17:07:55 UTC</td>
<td>0h 3m 37s</td>
<td>pass</td>
<td>debci log</td>
</tr>
<tr>
<td>0.12</td>
<td>2015-07-15 15:54:59 UTC</td>
<td>0h 3m 43s</td>
<td>pass</td>
<td>debci log</td>
</tr>
<tr>
<td>0.12</td>
<td>2015-07-05 05:49:28 UTC</td>
<td>0h 3m 24s</td>
<td>pass</td>
<td>debci log</td>
</tr>
<tr>
<td>0.12</td>
<td>2015-06-16 18:38:06 UTC</td>
<td>0h 3m 50s</td>
<td>pass</td>
<td>debci log</td>
</tr>
<tr>
<td>0.12</td>
<td>2015-06-13 14:34:00 UTC</td>
<td>0h 0m 17s</td>
<td>tmpfail</td>
<td>debci log</td>
</tr>
<tr>
<td>0.12</td>
<td>2015-05-28 00:29:55 UTC</td>
<td>0h 3m 55s</td>
<td>pass</td>
<td>debci log</td>
</tr>
<tr>
<td>0.12</td>
<td>2015-05-24 12:01:43 UTC</td>
<td>0h 0m 19s</td>
<td>tmpfail</td>
<td>debci log</td>
</tr>
<tr>
<td>0.12</td>
<td>2015-05-18 06:05:16 UTC</td>
<td>0h 3m 22s</td>
<td>pass</td>
<td>debci log</td>
</tr>
<tr>
<td>0.12</td>
<td>2015-05-11 08:07:12 UTC</td>
<td>0h 3m 28s</td>
<td>pass</td>
<td>debci log</td>
</tr>
<tr>
<td>0.12</td>
<td>2015-05-03 03:18:19 UTC</td>
<td>0h 3m 35s</td>
<td>pass</td>
<td>debci log</td>
</tr>
<tr>
<td>0.12</td>
<td>2015-04-30 21:21:25 UTC</td>
<td>0h 0m 52s</td>
<td>tmpfail</td>
<td>debci log</td>
</tr>
<tr>
<td>0.12</td>
<td>2015-04-26 22:18:07 UTC</td>
<td>0h 3m 23s</td>
<td>pass</td>
<td>debci log</td>
</tr>
<tr>
<td>0.12</td>
<td>2015-04-10 13:02:01 UTC</td>
<td>0h 3m 20s</td>
<td>pass</td>
<td>debci log</td>
</tr>
<tr>
<td>0.12</td>
<td>2015-04-07 02:20:41 UTC</td>
<td>0h 3m 22s</td>
<td>pass</td>
<td>debci log</td>
</tr>
<tr>
<td>0.12</td>
<td>2015-04-02 12:41:29 UTC</td>
<td>0h 3m 15s</td>
<td>pass</td>
<td>debci log</td>
</tr>
</tbody>
</table>
CI data is also in

- UDD/DMD
- DDPO
- package tracker(s)
Debian CI future

- almost ready to migrate to a distributed setup with lots of CPU power
- will be able to also test testing and more architectures
DEP-8: full text available from http://dep.debian.net/deps/dep8
debian/control
Source: foo
[...]
Testsuite: autopkgtest

debian/tests/control
Tests: foo bar baz
Tests: foo bar baz

foo, bar and baz must be executables in debian/tests/
Test-Command: foo --version \n        | grep 'foo version [0-9.]+'

no need to write one-line shell scripts
Tests: test-my-package

**Depends:** @, test-tool

*install extra dependencies*
Tests: test-my-package
Depends: @, test-tool

Tests: smoke-test

tests with different characteristics
Tests: upstream-tests

Depends: @, @builddeps@

need build dependencies
to run tests
Tests: my-tests

Restrictions: [...]

additional requirements for tests
Tests: break-the-world
Restrictions: breaks-testbed

testbed will not be reused for other tests
Tests: play-with-danger
Restrictions: needs-root

tests need to be run as root
(think twice before doing this)
Tests: complain-but-succeed
Restrictions: allow-stderr

does not fail if test program produces output on stderr
Tests: mess-with-services
Restrictions: isolation-container

skip if testbed is not at least as isolated from the host system as a container
Tests: mess-with-kernel
Restrictions: isolation-machine

skip if testbed is not at least as isolated from the host system as a VM
Tests: test-extra-features
Restrictions: needs-recommends

installs Recommends: into the testbed
tools
sadt(1) from devscripts

$ sadt

runs tests from source package at current directory, on the current system — somewhat limited
**adt-run(1)** from autopkgtest

Usage:

```
$ adt-run [adt-run options] \n--- [virt. args]
```

Yes, those are **THREE** dashes
$ adt-run ./ --- null

runs the tests from the source package at the current directory, on the current system ("null" virtualization)

IMO `adt-run` should default to this -- that's bug #795274
$ adt-run -u debci \
   /path/to/foo_1.2.3-1_amd64.changes \
   --- schroot debci-unstable-amd64

run tests from the source referenced by the changes file, using its binaries, under a user called debci, in a schroot(1) session based on the debci-unstable-amd64 chroot
$ adt-run -u debci
/path/to/foo_1.2.3-1_amd64.changes
--- lxc adt-sid-amd64

also supported: qemu, ssh
requested: pbuilder, docker
functional tests by example

(plus tips and tricks)
pinpoint

tip 1: shunit2(1)
redmine

tip 4: testing services
a random Ruby package
(let's try ruby-minitest)
tip 5: autodep8(1)
autodep8(1)

Testsuite: autopkgtest-pkg-ruby
Testsuite: autopkgtest-pkg-perl
Testsuite: autopkgtest-pkg-nodejs
Testsuite: autopkgtest-pkg-dkms
debci

tip 6: running "upstream" test suite
autodep8

tip 7: adding DEP-8 tests for package without them
vagrant

tip 8: really simple smoke tests
tip 9:

$ adt-run --shell
$ adt-run --shell-fail
Join the automated testing movement

- add tests to your packages
- add generic tests for your packaging team to autodep8
- help improving/maintaining ci.debian.net
thanks

• Ian Jackson — created autopkgtest
• Martin Pitt — maintains autopkgtest (and autopkgtest.ubuntu.com)
• Brandon Fairchild — debci UI work
• Lucas Kanbashiono — DEP-8 patches
contact

terceiro@debian.org

terceiro on OFTC

#debian-qa

#debian-devel