Maintaining 8000, erm, 9000 Packages

Large Scale Package QA in the PostgreSQL Ecosystem

Christoph Berg
myon@debian.org
christoph.berg@credativ.de

21st August 2015

PostgreSQL in Debian

Simple plan:

- There is Debian
- There is PostgreSQL
- Package it
- Done!

PostgreSQL in Debian

Reality is more complex

- There is Debian sid, jessie, wheezy, squeeze
- There is PostgreSQL 9.4, 9.3, 9.2, 9.1, 9.0
- PostgreSQL major releases have incompatible on-disk format
- Upgrading needs both versions installed in parallel
  (or lots of disk space and a complicated plan)

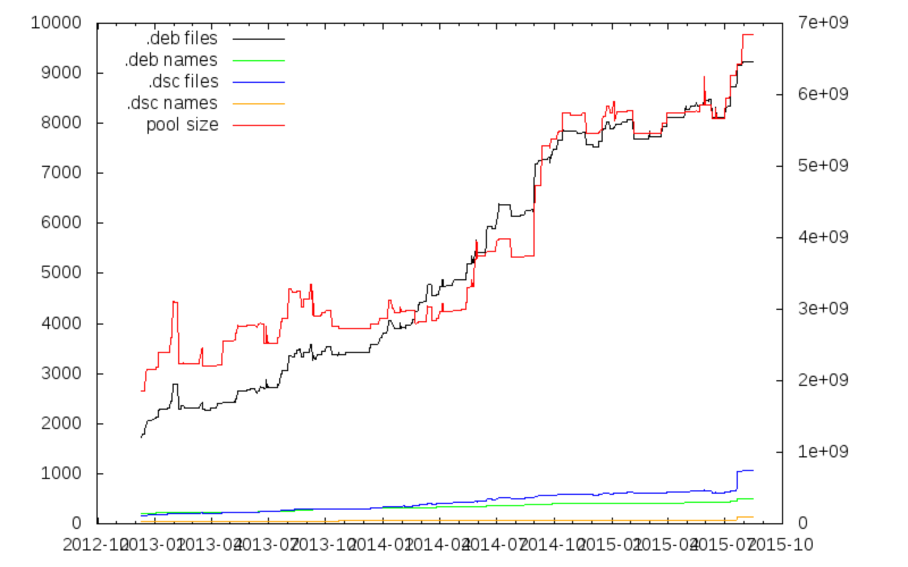
postgresql-common

- Solution: make packages for 9.4 9.3 . . . co-installable
- postgresql-common takes care of creating database clusters in the correct locations
- Still, Debian jessie has 9.4 only (wheezy has 9.1, squeeze had 8.4)
  - Users want to try new PostgreSQL version
  - Users want to upgrade OS without upgrading PostgreSQL

apt.postgresql.org

- Apt repository hosting packages for all PostgreSQL major releases built for all Debian (and Ubuntu) releases
- Somewhat a superset of what backports would do
- 6 PostgreSQL releases:
  8.4 9.0 9.1 9.2 9.3 9.4 (9.5 9.6)
- 7 Debian and Ubuntu releases
  squeeze wheezy jessie sid precise trusty utopic
- 2 architectures
  amd64 i386
- **6×7×2 = 84 targets**
- About 130 source packages
- Some are built “only” per distribution/architecture (14 targets)
- Extension modules depend on the PostgreSQL version as well (84 targets)

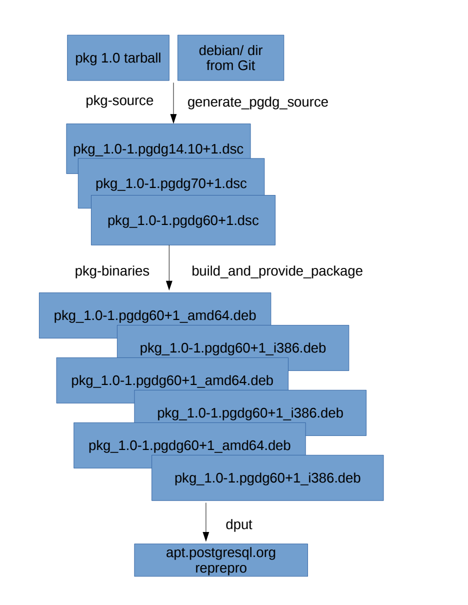
Statistics



.deb files
.deb names
.dsc files
.dsc names
pool size
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
7e+09
6e+09
5e+09
4e+09
3e+09
2e+09
1e+09
0
2012-10
2013-01
2013-04
2013-07
2013-10
2014-01
2014-04
2014-07
2014-10
2015-01
2015-04
2015-07
2015-10

Packages and Tests

Simple packages



pkg 1.0 tarball

debian/ dir from Git

pkg-source

generate_pgdg_source

pkg_1.0-1.pgdg14.10+1.dsc

pkg_1.0-1.pgdg70+1.dsc

pkg_1.0-1.pgdg60+1.dsc

pkg-binaries

build_and_provide_package

pkg_1.0-1.pgdg60+1_amd64.deb

pkg_1.0-1.pgdg60+1_i386.deb

pkg_1.0-1.pgdg60+1_amd64.deb

pkg_1.0-1.pgdg60+1_i386.deb

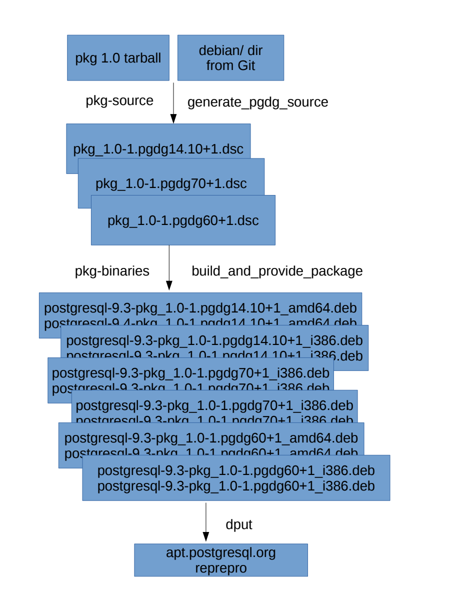
pkg_1.0-1.pgdg60+1_amd64.deb

pkg_1.0-1.pgdg60+1_i386.deb

dput

apt.postgresql.org reprepro

PostgreSQL extension packages



pkg 1.0 tarball

debian/ dir
from Git

pkg-source

generate_pgdg_source

pkg_1.0-1.pgdg14.10+1.dsc

pkg_1.0-1.pgdg70+1.dsc

pkg_1.0-1.pgdg60+1.dsc

pkg-binaries

build_and_provide_package

postgresql-9.3-pkg_1.0-1.pgdg14.10+1_amd64.deb
postgresql-9.4-pkg_1.0-1.pgdg14.10+1_amd64.deb

postgresql-9.3-pkg_1.0-1.pgdg14.10+1_i386.deb
postgresql-9.3-pkg_1.0-1.pgdg14.10+1_i386.deb

postgresql-9.3-pkg_1.0-1.pgdg70+1_i386.deb
postgresql-9.3-pkg_1.0-1.pgdg70+1_i386.deb

postgresql-9.3-pkg_1.0-1.pgdg70+1_i386.deb
postgresql-9.3-pkg_1.0-1.pgdg70+1_i386.deb

postgresql-9.3-pkg_1.0-1.pgdg60+1_amd64.deb
postgresql-9.3-pkg_1.0-1.pgdg60+1_amd64.deb

postgresql-9.3-pkg_1.0-1.pgdg60+1_i386.deb
postgresql-9.3-pkg_1.0-1.pgdg60+1_i386.deb

dput

apt.postgresql.org
reprepro

Quality Assurance

- No one is going to test 9000 packages
- We need testsuites
- PostgreSQL has regression tests
- Many extensions have regression tests
- *postgresql-common* has a testsuite on top of the PostgreSQL packages

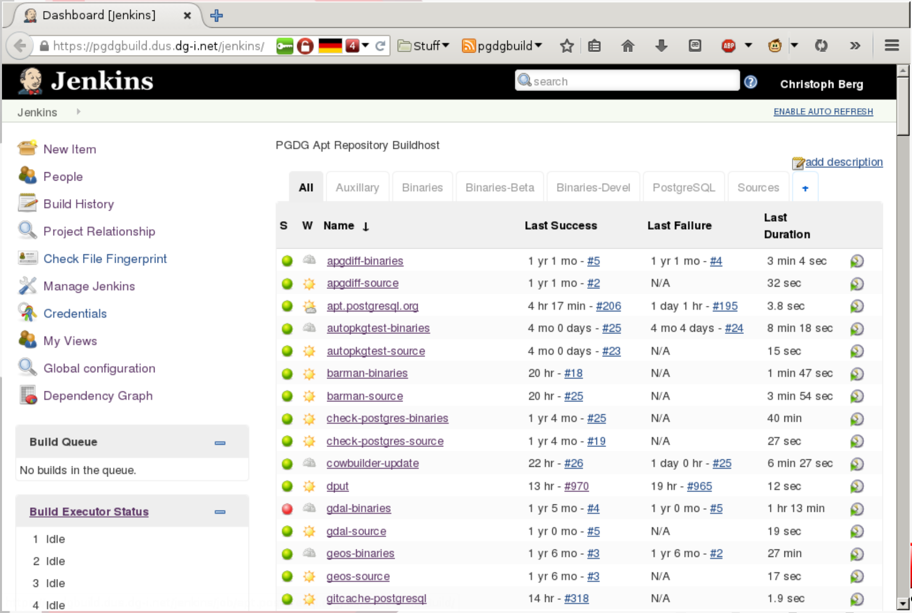
Quality Assurance

- No one is going to test 9000 testsuites
- We need automation

Jenkins

- Continuous Integration Server
- Takes your code and compiles it all the time
- More generally a framework for running scripts
- Matrix jobs with “configurations” (distribution, architecture)
- “source” jobs take a tarball and a VCS with a debian/ directory to build source packages per distribution
- “binaries” jobs take source packages and build (many) binary packages
- Thanks to Mika Prokop and his jenkins-debian-glue!

Jenkins



PGDG Apt Repository Buildhost

S	W	Name ↓	Last Success	Last Failure	Last Duration
●	☁	apgdiff-binaries	1 yr 1 mo - #5	1 yr 1 mo - #4	3 min 4 sec
●	☀	apgdiff-source	1 yr 1 mo - #2	N/A	32 sec
●	⛅	apt.postgresql.org	4 hr 17 min - #206	1 day 1 hr - #195	3.8 sec
●	☁	autopkgtest-binaries	4 mo 0 days - #25	4 mo 4 days - #24	8 min 18 sec
●	☀	autopkgtest-source	4 mo 0 days - #23	N/A	15 sec
●	☀	barman-binaries	20 hr - #18	N/A	1 min 47 sec
●	☀	barman-source	20 hr - #25	N/A	3 min 54 sec
●	☀	check-postgres-binaries	1 yr 4 mo - #25	N/A	40 min
●	☀	check-postgres-source	1 yr 4 mo - #19	N/A	27 sec
●	☁	cowbuilder-update	22 hr - #26	1 day 0 hr - #25	6 min 27 sec
●	☀	dput	13 hr - #970	19 hr - #965	12 sec
●	☁	gdal-binaries	1 yr 5 mo - #4	1 yr 0 mo - #5	1 hr 13 min
●	☀	gdal-source	1 yr 0 mo - #5	N/A	19 sec
●	☁	geos-binaries	1 yr 6 mo - #3	1 yr 6 mo - #2	27 min
●	☀	geos-source	1 yr 6 mo - #3	N/A	17 sec
●	☀	gitcache-postgresql	14 hr - #318	N/A	1.9 sec

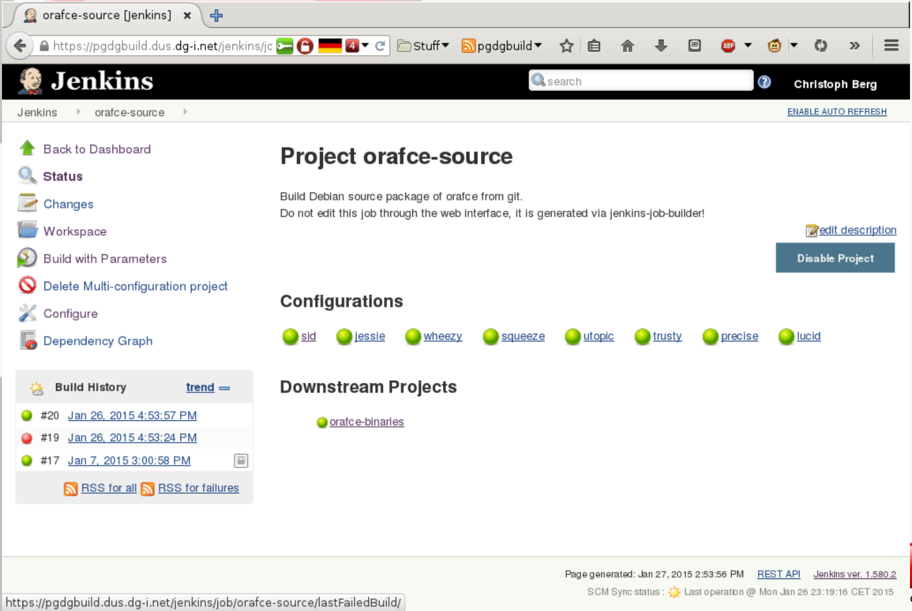
Build Queue

No builds in the queue.

Build Executor Status

1 Idle
2 Idle
3 Idle
4 Idle

orafce Source



- Back to Dashboard
- **Status**
- Changes
- Workspace
- Build with Parameters
- Delete Multi-configuration project
- Configure
- Dependency Graph

Build History trend

●	#20	Jan 26, 2015 4:53:57 PM
●	#19	Jan 26, 2015 4:53:24 PM
●	#17	Jan 7, 2015 3:00:58 PM

RSS for all RSS for failures

Project orafce-source

Build Debian source package of orafce from git.
Do not edit this job through the web interface, it is generated via jenkins-job-builder!

edit description

Disable Project

Configurations

● sid ● jessie ● wheezy ● squeeze ● utopic ● trusty ● precise ● lucid

Downstream Projects

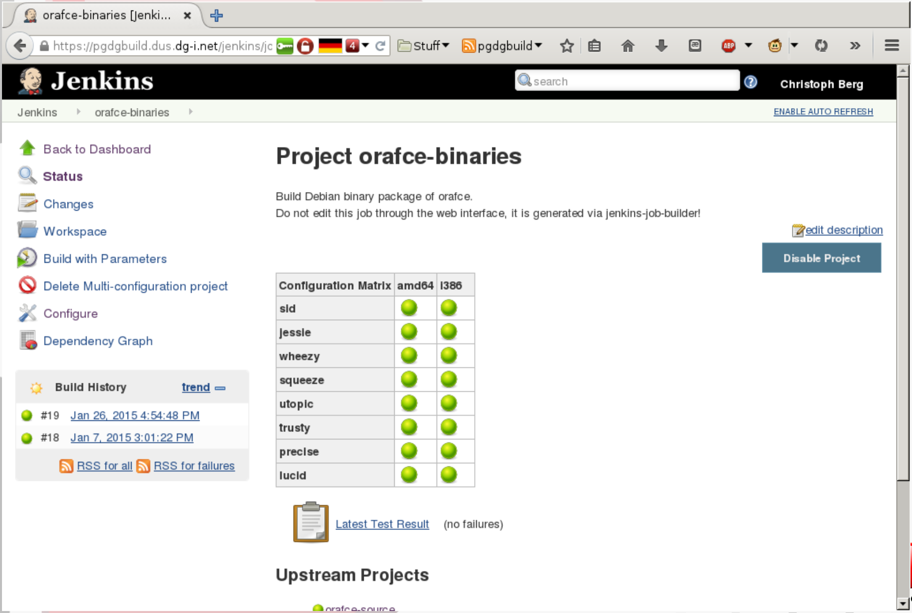
● orafce-binaries

Page generated: Jan 27, 2015 2:53:56 PM REST API Jenkins ver. 1.580.2

SCM Sync status : Last operation @ Mon Jan 26 23:19:16 CET 2015

https://pgdgbuild.dus.dg-i.net/jenkins/job/orafce-source/lastFailedBuild/

orafce Binaries



Back to Dashboard

Status

Changes

Workspace

Build with Parameters

Delete Multi-configuration project

Configure

Dependency Graph

Build History trend

●	#19	Jan 26, 2015 4:54:48 PM
●	#18	Jan 7, 2015 3:01:22 PM

RSS for all RSS for failures

Project orafce-binaries

Build Debian binary package of orafce.
Do not edit this job through the web interface, it is generated via jenkins-job-builder!

edit description

Disable Project

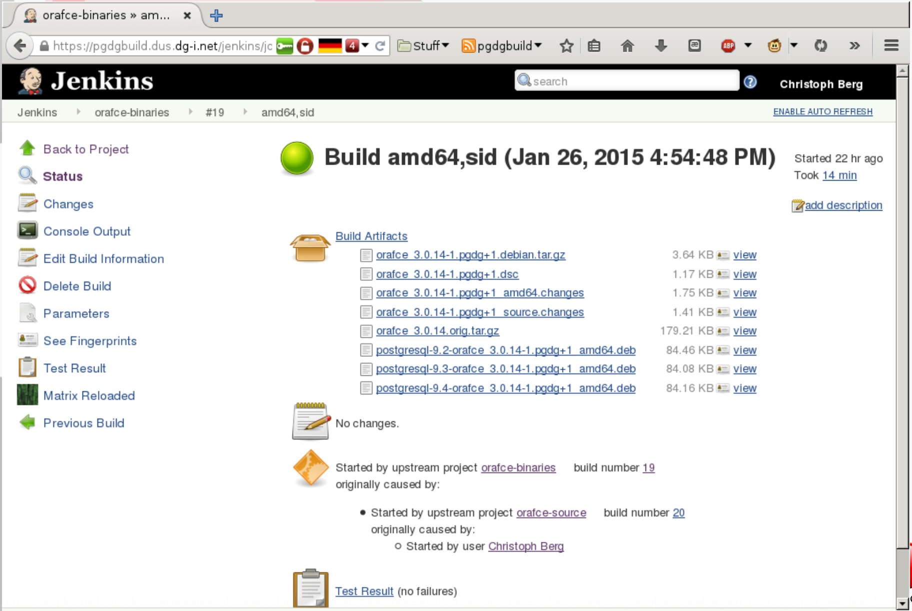
Configuration Matrix	amd64	i386
sid	●	●
jessie	●	●
wheezy	●	●
squeeze	●	●
utopic	●	●
trusty	●	●
precise	●	●
lucid	●	●

Latest Test Result (no failures)

Upstream Projects

● orafce-source

orafce Buildresult



Jenkins ▸ orafce-binaries ▸ #19 ▸ amd64,sid

- Back to Project
- **Status**
- Changes
- Console Output
- Edit Build Information
- Delete Build
- Parameters
- See Fingerprints
- Test Result
- Matrix Reloaded
- Previous Build

Build amd64,sid (Jan 26, 2015 4:54:48 PM)

Started 22 hr ago
Took 14 min

add description

Build Artifacts

orafce_3.0.14-1.pgdg+1.debian.tar.gz	3.64 KB	view
orafce_3.0.14-1.pgdg+1.dsc	1.17 KB	view
orafce_3.0.14-1.pgdg+1_amd64.changes	1.75 KB	view
orafce_3.0.14-1.pgdg+1_source.changes	1.41 KB	view
orafce_3.0.14.orig.tar.gz	179.21 KB	view
postgresql-9.2-orafce_3.0.14-1.pgdg+1_amd64.deb	84.46 KB	view
postgresql-9.3-orafce_3.0.14-1.pgdg+1_amd64.deb	84.08 KB	view
postgresql-9.4-orafce_3.0.14-1.pgdg+1_amd64.deb	84.16 KB	view

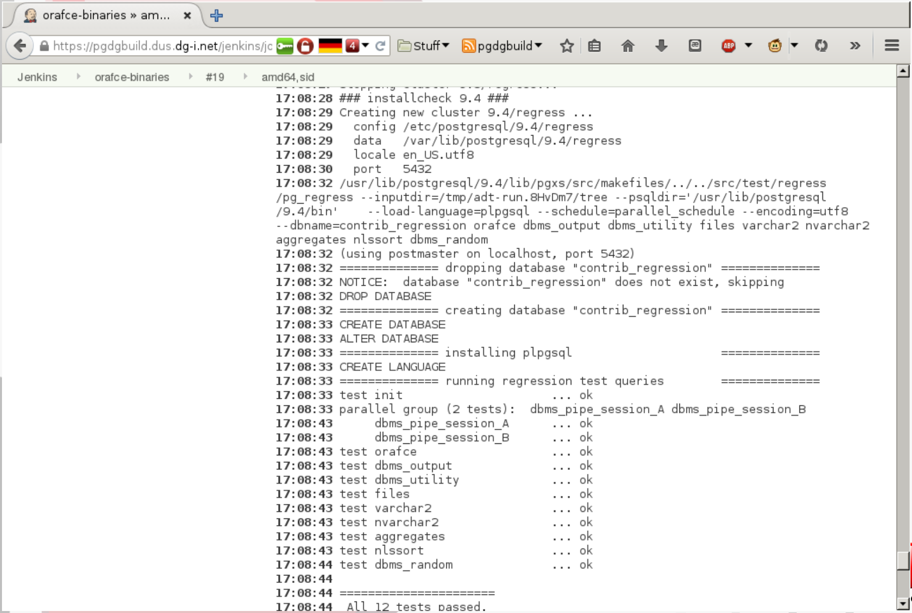
No changes.

Started by upstream project orafce-binaries build number 19
originally caused by:

- Started by upstream project orafce-source build number 20
  originally caused by:
  - Started by user Christoph Berg

Test Result (no failures)

orafce Console Output



```
17:08:28 ### installcheck 9.4 ###
17:08:29 Creating new cluster 9.4/regress ...
17:08:29   config /etc/postgresql/9.4/regress
17:08:29   data   /var/lib/postgresql/9.4/regress
17:08:29   locale en_US.utf8
17:08:30   port   5432
17:08:32 /usr/lib/postgresql/9.4/lib/pgxs/src/makefiles/../../src/test/regress
/pg_regress --inputdir=/tmp/adt-run.8HvDm7/tree --psqldir='/usr/lib/postgresql
/9.4/bin'    --load-language=plpgsql --schedule=parallel_schedule --encoding=utf8
--dbname=contrib_regression orafce dbms_output dbms_utility files varchar2 nvarchar2
aggregates nlssort dbms_random
17:08:32 (using postmaster on localhost, port 5432)
17:08:32 ============== dropping database "contrib_regression" ==============
17:08:32 NOTICE:  database "contrib_regression" does not exist, skipping
17:08:32 DROP DATABASE
17:08:32 ============== creating database "contrib_regression" ==============
17:08:33 CREATE DATABASE
17:08:33 ALTER DATABASE
17:08:33 ============== installing plpgsql                     ==============
17:08:33 CREATE LANGUAGE
17:08:33 ============== running regression test queries        ==============
17:08:33 test init                     ... ok
17:08:33 parallel group (2 tests):  dbms_pipe_session_A dbms_pipe_session_B
17:08:43      dbms_pipe_session_A      ... ok
17:08:43      dbms_pipe_session_B      ... ok
17:08:43 test orafce                   ... ok
17:08:43 test dbms_output              ... ok
17:08:43 test dbms_utility             ... ok
17:08:43 test files                    ... ok
17:08:43 test varchar2                 ... ok
17:08:43 test nvarchar2                ... ok
17:08:43 test aggregates               ... ok
17:08:43 test nlssort                  ... ok
17:08:44 test dbms_random              ... ok
17:08:44
17:08:44 =====================
17:08:44  All 12 tests passed.
```

pg_buildext

- Helps with building packages for multiple PostgreSQL versions at once
- Loops over versions listed in debian/pgversions
- Supported versions defined by /usr/share/postgresql-common/supported-versions
- Examples in pg_buildext(1)

credativ

debian/rules

```
include /usr/share/postgresql-common/pgxs_debian_control.mk

override_dh_auto_build:
	+pg_buildext build build-%v

override_dh_auto_test:
	# nothing to do here, see debian/tests/* instead

override_dh_auto_install:
	+pg_buildext install build-%v postgresql-%v-foobar

override_dh_installdocs:
	dh_installdocs --all README.*

override_dh_auto_clean:
	+pg_buildext clean build-%v

%:
	dh $@
```

Build-time testing vs. testing installed packages

- Regression tests are nice, but don't catch packaging errors
- Run tests on installed packages as well
- **autopkgtest**
- postgresql-common: 1274 tests for postgresql-9.4
- 1954 for postgresql-9.3+9.4 incl. upgrade tests

autopkgtest

- debian/tests/control

```
Tests: run-testsuite
Depends: @, build-essential, hunspell-en-us, locales, netcat-openbsd,
Restrictions: needs-root
```

- debian/tests/run-testsuite

```
#!/bin/sh
cd /usr/share/postgresql-common
./testsuite
```

```
====== Running all tests with tight umask 077 =======
=== Running test 001_packages.t ... ===
1..19
# PostgreSQL versions installed: 9.4
ok 1 - postgresql-9.4 installed
ok 2 - postgresql-plpython-9.4 installed
ok 3 - postgresql-plpython3-9.4 installed
ok 4 - postgresql-plperl-9.4 installed
ok 5 - postgresql-pltcl-9.4 installed
ok 6 - postgresql-server-dev-9.4 installed
ok 7 - postgresql-contrib-9.4 installed
ok 8 - libecpg-dev installed
ok 9 - logrotate installed
ok 10 - procps installed
ok 11 - netcat-openbsd installed
ok 12 - hunspell-en-us installed
ok 13 - system has a default UTF-8 locale
```

autopkgtest and pg_buildext

- Run “make installcheck” (via PGXS)
- Simple debian/tests/installcheck

```
#!/bin/sh
pg_buildext installcheck
```

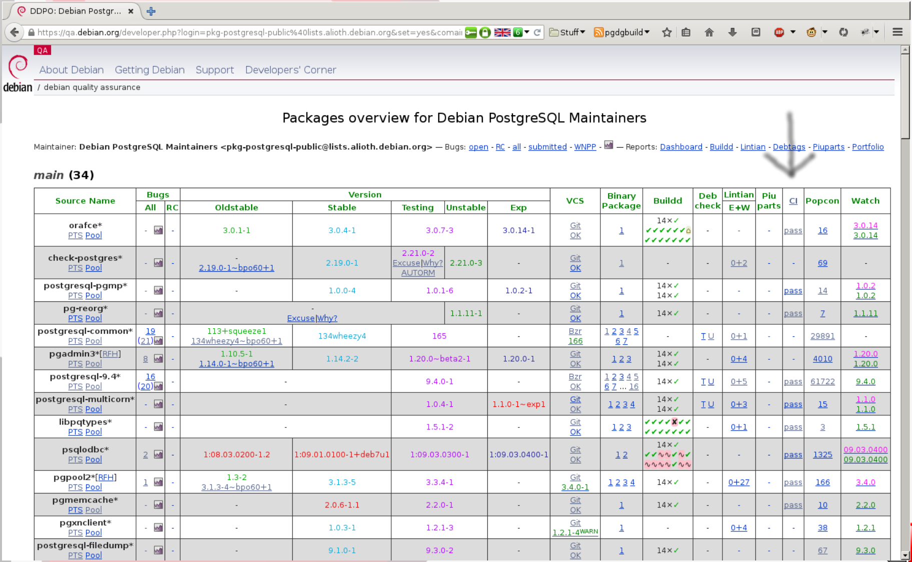
- Complex debian/tests/installcheck

```
#!/bin/sh

set -e
for v in $(pg_buildext supported-versions); do
  case $v in
    8*|9.0) # don't bother to test the extension here
      continue ;;
  esac

  (
    PATH=$PATH:/usr/lib/postgresql/$v/bin
    pg_buildext installcheck-$v
  )
done
```

ci.debian.net (qa.debian.org/developer.php)


DDPO: Debian Postgr...
https://qa.debian.org/developer.php?login=pkg-postgresql-public%40lists.alioth.debian.org&set=yes&comai
Stuff
pgdgbuild
QA
debian
About Debian
Getting Debian
Support
Developers' Corner
/ debian quality assurance

Packages overview for Debian PostgreSQL Maintainers

Maintainer: **Debian PostgreSQL Maintainers <pkg-postgresql-public@lists.alioth.debian.org>** — Bugs: open - RC - all - submitted - WNPP — Reports: Dashboard - Buildd - Lintian - Debtags - Piuparts - Portfolio

main **(34)**

Source Name	Bugs		Version					VCS	Binary Package	Buildd	Deb check	Lintian	Piu parts	CI	Popcon	Watch
	All	RC	Oldstable	Stable	Testing	Unstable	Exp					E+W				
orafce* PTS Pool	-	-	3.0.1-1	3.0.4-1	3.0.7-3		3.0.14-1	Git OK	1	14×✓ ✓✓✓✓✓✓⌂ ✓✓✓✓✓✓✓	-	-	-	pass	16	3.0.14 3.0.14
check-postgres* PTS Pool	-	-	- 2.19.0-1~bpo60+1	2.19.0-1	2.21.0-2 Excuse|Why? AUTORM	2.21.0-3	-	Git OK	1	-	-	0+2	-	-	69	-
postgresql-pgmp* PTS Pool	-	-	-	1.0.0-4	1.0.1-6		1.0.2-1	Git OK	1	14×✓ 14×✓	-	-	-	pass	14	1.0.2 1.0.2
pg-reorg* PTS Pool	-	-	- Excuse|Why?			1.1.11-1	-	Git OK	1	14×✓	-	-	-	pass	7	1.1.11
postgresql-common* PTS Pool	19 (21)	-	113+squeeze1 134wheezy4~bpo60+1	134wheezy4	165		-	Bzr 166	1 2 3 4 5 6 7	-	T U	0+1	-	-	29891	-
pgadmin3*[RFH] PTS Pool	8	-	1.10.5-1 1.14.0-1~bpo60+1	1.14.2-2	1.20.0~beta2-1		1.20.0-1	Git OK	1 2 3	14×✓ 14×✓	-	0+4	-	-	4010	1.20.0 1.20.0
postgresql-9.4* PTS Pool	16 (20)	-	-		9.4.0-1		-	Bzr OK	1 2 3 4 5 6 7 ... 16	14×✓	T U	0+5	-	pass	61722	9.4.0
postgresql-multicorn* PTS Pool	-	-	-		1.0.4-1		1.1.0-1~exp1	Git OK	1 2 3 4	14×✓ 14×✓	T U	0+3	-	pass	15	1.1.0 1.1.0
libpqtypes* PTS Pool	-	-	-		1.5.1-2		-	Git OK	1 2 3	✓✓✓✓✗✓✓ ✓✓✓✓✓✓✓	-	0+1	-	pass	3	1.5.1
psqlodbc* PTS Pool	2	-	1:08.03.0200-1.2	1:09.01.0100-1+deb7u1	1:09.03.0300-1		1:09.03.0400-1	Git OK	1 2	14×✓ ✓✓∿∿✓∿✓ ∿∿∿∿✓∿∿	-	-	-	pass	1325	09.03.0400 09.03.0400
pgpool2*[RFH] PTS Pool	1	-	1.3-2 3.1.3-4~bpo60+1	3.1.3-5	3.3.4-1		-	Git 3.4.0-1	1 2 3 4	14×✓	-	0+27	-	pass	166	3.4.0
pgmemcache* PTS Pool	-	-	-	2.0.6-1.1	2.2.0-1		-	Git OK	1	14×✓	-	-	-	pass	10	2.2.0
pgxnclient* PTS Pool	-	-	-	1.0.3-1	1.2.1-3		-	Git 1.2.1-4WARN	1	-	-	0+4	-	-	38	1.2.1
postgresql-filedump* PTS Pool	-	-	-	9.1.0-1	9.3.0-2		-	Git OK	1	14×✓	-	-	-	-	67	9.3.0

Bugs

Bugs found: peer authentication

Found by postgresql-common:

```
commit b777be0d48a042f500cac72140ffb50392973aa2
Author: Tom Lane <tgl@sss.pgh.pa.us>
Date:   Fri Mar 28 10:30:37 2014 -0400

    Un-break peer authentication.

    Commit 613c6d26bd42dd8c2dd0664315be9551475b8864 sloppily replaced a
    lookup of the UID obtained from getpeereid() with a lookup of the
    server's own user name, thus totally destroying peer authentication.
    Revert.  Per report from Christoph Berg.

    In passing, make sure get_user_name() zeroes *errstr on success on
    Windows as well as non-Windows.  I don't think any callers actually
    depend on this ATM, but we should be consistent across platforms.
```

Bugs found: stack size detection

Found by the PostgreSQL regression tests:

```
*** src/test/regress/expected/errors.out 2014-05-11 23:16:48
--- src/test/regress/results/errors.out  2014-05-13 22:16:05
***************
*** 444,447 ****
  'select infinite_recurse()' language sql;
  \set VERBOSITY terse
  select infinite_recurse();
! ERROR:  stack depth limit exceeded
--- 444,447 ----
  'select infinite_recurse()' language sql;
  \set VERBOSITY terse
  select infinite_recurse();
! connection to server was lost
```

General Linux ASLR issue on 32bit archs, mitigated by disabling -pie

Bugs found: non-reproducible EXPLAIN output

Found by the multicorn regression tests on 9.4:
During the 9.4 development, EXPLAIN grew an extra "Planning time" output line that made it impossible to use this in regression tests:

```
explain select * from testmulticorn m1 inner join testmulticorn m2 on m1.te
                                     QUERY PLAN
---------------------------------------------------------------------------
 Nested Loop  (cost=20.00..806.05 rows=2 width=128)
   Join Filter: ((m1.test1)::text = (m2.test1)::text)
   ->  Foreign Scan on testmulticorn m1  (cost=10.00..400.00 rows=20 width=
   ->  Materialize  (cost=10.00..400.10 rows=20 width=20)
         ->  Foreign Scan on testmulticorn m2  (cost=10.00..400.00 rows=20
(5 rows)
```

Resolved by showing the Planning time for EXPLAIN ANALYZE only

Bugs found: timezone mess

First spotted on Solaris, but it was a Debian problem as well:
When using `--with-system-tzdata`:

```
1937c1937
<  Sat Oct 25 22:00:00 2014 UTC
---
>  Sat Oct 25 21:00:00 2014 UTC
1943c1943
<  Sat Oct 25 22:00:01 2014 UTC
---
>  Sat Oct 25 21:00:01 2014 UTC
```

(Blame the Russians)

Resolved by using testcases for similar TZ changes from 2012 instead

Bugs found: shm_mq problems

```
../../src/test/regress/pg_regress --inputdir=/«PKGBUILDDIR»/build/../contri
============== creating temporary installation        ==============
============== initializing database system           ==============
============== starting postmaster                    ==============
running on port 3133 with PID 29665
============== creating database "contrib_regression" ==============
CREATE DATABASE
ALTER DATABASE
============== running regression test queries        ==============
E: Caught signal ‘Terminated’: terminating immediately
make[4]: *** [check] Terminated
make[3]: *** [check-test_shm_mq-recurse] Terminated
test test_shm_mq              ... /«PKGBUILDDIR»/build/../src/makefiles/pgx
[...]
GNUmakefile:69: recipe for target 'check-world-contrib-recurse' failed
Build killed with signal TERM after 300 minutes of inactivity
```

Issue on some mips(el) build hosts, a similar issue on Solaris was fixed last month, but it's probably something else

Bugs found: psqlodbc problems

Found by the psqlodbc regression tests on s390x:

```
 struct IPDFields_
 {
-#if (ODBCVER >= 0x0300)
-       SQLUINTEGER             *param_processed_ptr;
-#else
-       SQLULEN                 *param_processed_ptr; /* SQLParamOptions */
-#endif /* ODBCVER */
+       SQLULEN                 *param_processed_ptr;
        SQLUSMALLINT            *param_status_ptr;
```

Bugs found: extensions not ported to 9.x yet

Dozens of cases where extensions were not ported to ~~9.3~~ ~~9.4~~ 9.5 yet:

```
*** /tmp/adt-run.PNuMnw/tree/test/expected/mpq.out 2015-01-16 01:47:29
--- /tmp/adt-run.PNuMnw/tree/results/mpq.out 2015-01-17 21:45:27
***************
*** 486,491 ****
--- 486,492 ----
  insert into test_mpq_idx select generate_series(1, 10000);
  create index test_mpq_btree_idx on test_mpq_idx using btree (q);
  create index test_mpq_hash_idx on test_mpq_idx using hash (q);
+ WARNING:  hash indexes are not WAL-logged and their use is discouraged
  -- Hash is compatible with mpz
  select mpq_hash(0) = mpz_hash(0);
  t
```

Regression failures usually easy to work around, code changes more involved

Bugs NOT found: SSL problems in libpq

One of our servers is currently running on postgres 9.2 using the 9.2.9-1.pgdg70+1 packages from pgdg.
After an apt update this morning which brought in the libpq5 package version 9.4.0-1.pgdg70+1, connections to the tabase started failing with SSL errors logged on the server:

```
    [unknown] [unknown] LOG:  could not accept SSL connection:
```

Rolling back the server and client to libpq5 version 9.3.5-2.pgdg70+1 fixed it.
This is running on an otherwise up-to-date Debian Wheezy. The SSL certificate is locally issued using an internal CA which has been added to the local trust store. SSL-related config options are left set to the defaults.

9.4's libpq only supports TLS, not SSL, for which 512bit keys are too small

Bugs NOT found: libpq and NSS lookups

In minimal chroots, 9.4's libpq is unusable:

```
Jan 10 00:11:40 lehmann postfix/trivial-rewrite[29960]:
  warning: connect to pgsql server localhost:5432:
    out of memory?
Jan 10 00:11:40 lehmann postfix/trivial-rewrite[29960]:
  warning: pgsql:/etc/postfix/pgsqltest lookup error for "*"
```

/etc/passwd lookup failures are non-fatal again in 9.4.1

TODO

Things not automated yet

- Some packages don't have tests (meh)
- Some packages have, well, complicated testsuites (someone needs to untangle them)
- Triggering builds of updated packages (wanna-build NIH)
- Nice webpages for packages in the repository (done, but hasn't made it to www.pg.o yet)
- `dh --with=pg_buildext`
- `dh_make_pgxs`
- More maintainers!