dgit

Use the Debian archive as a git remote

Debconf 15, Heidelberg

Ian Jackson  ijackson@chiark.greenend.org.uk
ian.jackson@eu.citrix.com

- dgit for NMUers, downstreams, users, etc.
- dgit for the Debian package maintainer
- miscellany
NAME
dgit – git integration with the Debian archive

SYNOPSIS


dgit [dgit−opts] fetch|pull [dgit−opts] [suite]

dgit [dgit−opts] build|sbuild|build-source [build−opts]

dgit [dgit−opts] push [dgit−opts] [suite]

dgit [dgit−opts] rpush build-host:build-dir [push args...]

dgit [dgit−opts] action ...

DESCRIPTION
dgit allows you to treat the Debian archive as if it were a git repository. See dgit(7) for detailed information about the data model, common problems likely to arise with certain kinds of package, etc.

The usual workflow is:
1. **dgit clone** or **fetch**;
2. make, do dev tests, and commit changes in git as desired;
3. build packages for upload, using e.g. **dgit sbuild**
4. do pre-upload tests of the proposed upload;
5. **dgit push**.

OPERATIONS

dgit clone package [suite] [.dir]/dir

Consults the archive and dgit-repos to construct the git view of history for *package* in *suite* (sid by default) in a new directory (named *package* by default); also, downloads any necessary orig tarballs.

The suite’s git tip is left on the local branch **dgit/suite** ready for work, and on the corresponding dgit remote tracking branch. The **origin** remote will be set up to point to the package’s dgit-repos.
your working tree

[dgit] [s]build

ed/emacs/vi

[dgit] push

git-commit

[dgit] fetch

git-fetch

git-cherry-pick

any git server

git-push

git-reset

git-clean

gbp pq

debcommit

ftp.d.o

.dsc
.tar.gz
Maintainer’s history

- Make build reproducible (closes: #777727)
- Finalise nmu1

remotes/dgit/dgit/sid
dgit/sid

Make build reproducible (closes: #777727)

remotes/dgit/dgit/sid

openssh (1:6.7p1−6) unstable; ...

# imported from the archive

dgit synthetic history
**Maintainer does not use dgit**

- Debian/6.7p1–5: Releasing ...
- Various Debian-specific...
- Revert LC_*... (closes: #780797)
- Add a NEWS.Debian entry...
- Various Debian-specific...
- Finalise nmu1

**DISJOINT HISTORY**

- Debian/6.7p1–4

**Maintainer uses dgit**

- Debian/6.7p1–5: Releasing ...
- Add a NEWS.Debian entry...
- Revert LC_*... (closes: #780797)
- Various Debian-specific...
- Finalise nmu1

- Debian/6.7p1–4

- Remotes/dgit/dgit/sid
  - Make build reproducible (closes: #777727)
  - Finalise nmu1
  - Openssh (1:6.7p1–6) unstable; ... imported from the archive
Your working tree

[dgit]
[s]build

[.dsc]
[.tar.gz]

your working tree

dgit push

dgit fetch

[debian/0.22.1] [dgit/sid]

[debian/0.22] [dgit/sid @{yesterday}]

IDENTICAL

IDENTICAL

FAST FORWARD

ftp.d.o

IDENTICAL
[.dsc]
[.tar.gz]

dgit.d.o
<table>
<thead>
<tr>
<th>source format</th>
<th>git workflow</th>
<th>push</th>
<th>integrate nmu</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0 native</td>
<td>any</td>
<td>Go!</td>
<td>dgit pull</td>
</tr>
<tr>
<td>1.0 diff</td>
<td>any (merging)</td>
<td>Go!</td>
<td>dgit pull</td>
</tr>
<tr>
<td>1.0 diff</td>
<td><code>git-rebase</code></td>
<td>Go!</td>
<td>dgit fetch;</td>
</tr>
<tr>
<td></td>
<td>or any other rebasing</td>
<td></td>
<td>git-merge -s ours</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(will be improved)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>or use <code>git-import-dsc</code></td>
</tr>
<tr>
<td>3.0 (quilt)</td>
<td><code>git-buildpackage</code></td>
<td></td>
<td>Watch this space</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>problems: patches applied vs. unapplied</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><code>.gitignore</code></td>
</tr>
<tr>
<td>3.0 (quilt)</td>
<td><code>git-dpm</code></td>
<td></td>
<td>Watch this space</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>problems: <code>.gitignore</code></td>
</tr>
</tbody>
</table>
Your working tree

`dgit push`

`dgit fetch`

`git fetch`

`git push`

Any git server

éd/emacs/vigit-commit

git-cherry-pick

git-reset

git-clean

git-dpm

gbp pq

debcommit
your working tree

[dgit]
[s]build

dgit push

dgit fetch

[build]
dgit

[debian/0.22.1]

dgit/sid

debian/0.22

dgit/sid @{yesterday}

IDENTICAL

IDENTICAL

FAST FORWARD

ftp.d.o

.dsc
.tar.gz

IDENTICAL
NAME
dgit – git integration with the Debian archive

SYNOPSIS
dgit [dgit−opts] fetch|pull [dgit−opts] [suite]
dgit [dgit−opts] build|sbuild|build-source [build−opts]
dgit [dgit−opts] push [dgit−opts] [suite]
dgit [dgit−opts] rpush build-host:build-dir [push args...]
dgit [dgit−opts] action ...

DESCRIPTION
dgit allows you to treat the Debian archive as if it were a git repository. See dgit(7) for detailed information about the data model, common problems likely to arise with certain kinds of package, etc.

The usual workflow is:

1. dgit clone or fetch;
2. make, do dev tests, and commit changes in git as desired;
3. build packages for upload, using e.g. dgit sbuild
4. do pre-upload tests of the proposed upload;
5. dgit push.

OPERATIONS
dgit clone package [suite] [/dir|/dir]
Consults the archive and dgit-repos to construct the git view of history for package in suite (sid by default) in a new directory (named .package by default); also, downloads any necessary orig tarballs.

The suite’s git tip is left on the local branch dgit/suite ready for work, and on the corresponding dgit remote tracking branch. The origin remote will be set up to point to the package’s dgit-repos
Specifies that we should process source package rather than looking in debian/control or debian/changelog. Valid with dgit fetch and dgit pull, only.

--clean=git | -wg
The source tree should be cleaned, before building a source package with one of the build options, using **git clean -xdf**. This will delete all files which are not tracked by git. Also, -wg causes dgit to pass -nc to dpkg-buildpackage, which prevents the package’s own clean target from being run.

--clean=git is useful when the package’s clean target is troublesome; the downside is simply that git clean may delete files you forgot to git add. --clean=git can also avoid needing the build-dependencies.

--clean=git-ff | -wgf
The source tree should be cleaned, before building a source package with one of the build options, using **git clean -xdff**. This is like "git clean -xdf" but it also removes any subdirectories containing different git trees (which only unusual packages are likely to create).

--clean=check | -wc
Merely check that the tree is clean (does not contain uncommitted files), before building a source package.

--clean=none | -wn
Do not clean the tree before building a source package. If there are files which are not in git, or if the build creates such files, a subsequent dgit push will fail.

--clean=dpkg-source | -wd
Use dpkg-buildpackage to do the clean, so that the source package is cleaned by dpkg-source running the package’s clean target. This is the default. It requires the package’s build dependencies.

--clean=dpkg-source-d | -wdd
Use dpkg-buildpackage -d to do the clean, so that the source package is cleaned by dpkg-source running the package’s clean target. The build-dependencies are not checked.
plans for my CFT

better integration with git–buildpackage
better integration with git–dpm
dgit push to DELAYED
one−step source–only push/upload
use of the dgit git repo server for
general purpose work (eg, maintainer’s
pre–upload branches)
better tooling assistance for use with
raw git−rebase
automatic mirroring, into the dgit git
branches, of non–dgit uploads

help needed

better workflow documentation
for integration with existing
git packaging tools (esp.
re receiving an NMU, and new
upstream versions)
empty directories problem
dgit−based sponsorship
request mechanism
dgit −d mint fetch rafaela
dgit −d ubuntu push vervet
Thanks
Joey
DSA
ftpmaster

More info

start with dgit(1), dgit(7) from manpages.debian.net, or dgit.deb in testing

ijackson@chiark.greenend.org.uk

This talk (script, slides, source code) available at
http://www.chiark.greenend.org.uk/~ijackson/2015/debconf-dgit-talk/

Questions?