

dg̣it

Use the Debian archive as a git remote

Debconf 15, Heidelberg

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

- dg̣it for NMUers, downstreams, users, etc.
- dg̣it for the Debian package maintainer
- miscellany

NAME

dggit – git integration with the Debian archive

SYNOPSIS

```
dggit [dggit-opts] clone [dggit-opts] package [suite] [./dir|/dir]
dggit [dggit-opts] fetch|pull [dggit-opts] [suite]
dggit [dggit-opts] build|sbuild|build-source [build-opts]
dggit [dggit-opts] push [dggit-opts] [suite]
dggit [dggit-opts] rpush build-host:build-dir [push args...]
dggit [dggit-opts] action ...
```

DESCRIPTION

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

The usual workflow is:

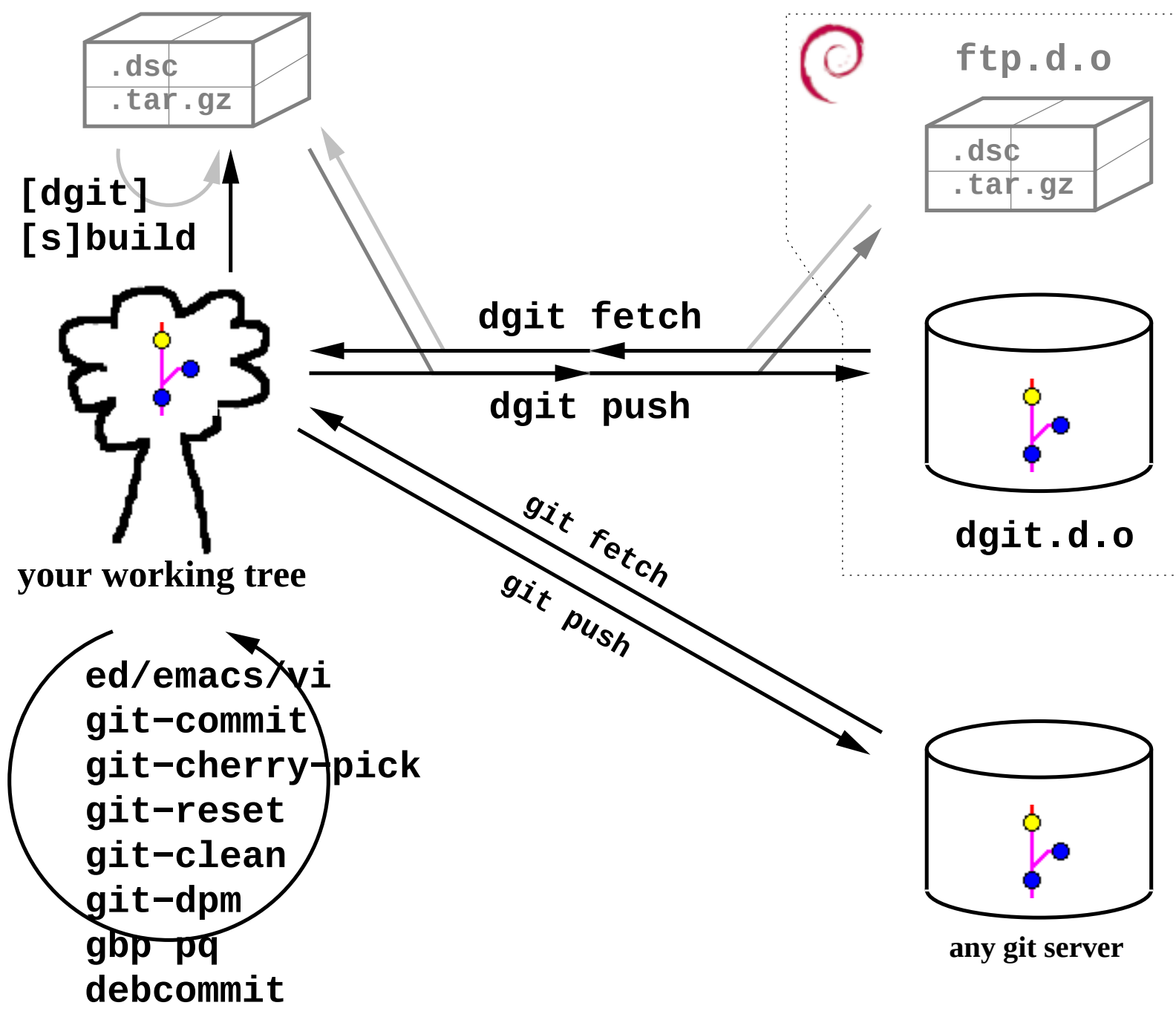
- 1. **dggit clone** or **fetch**;
- 2. make, do dev tests, and commit changes in git as desired;
- 3. build packages for upload, using e.g. **dggit sbuild**
- 4. do pre-upload tests of the proposed upload;
- 5. **dggit push**.

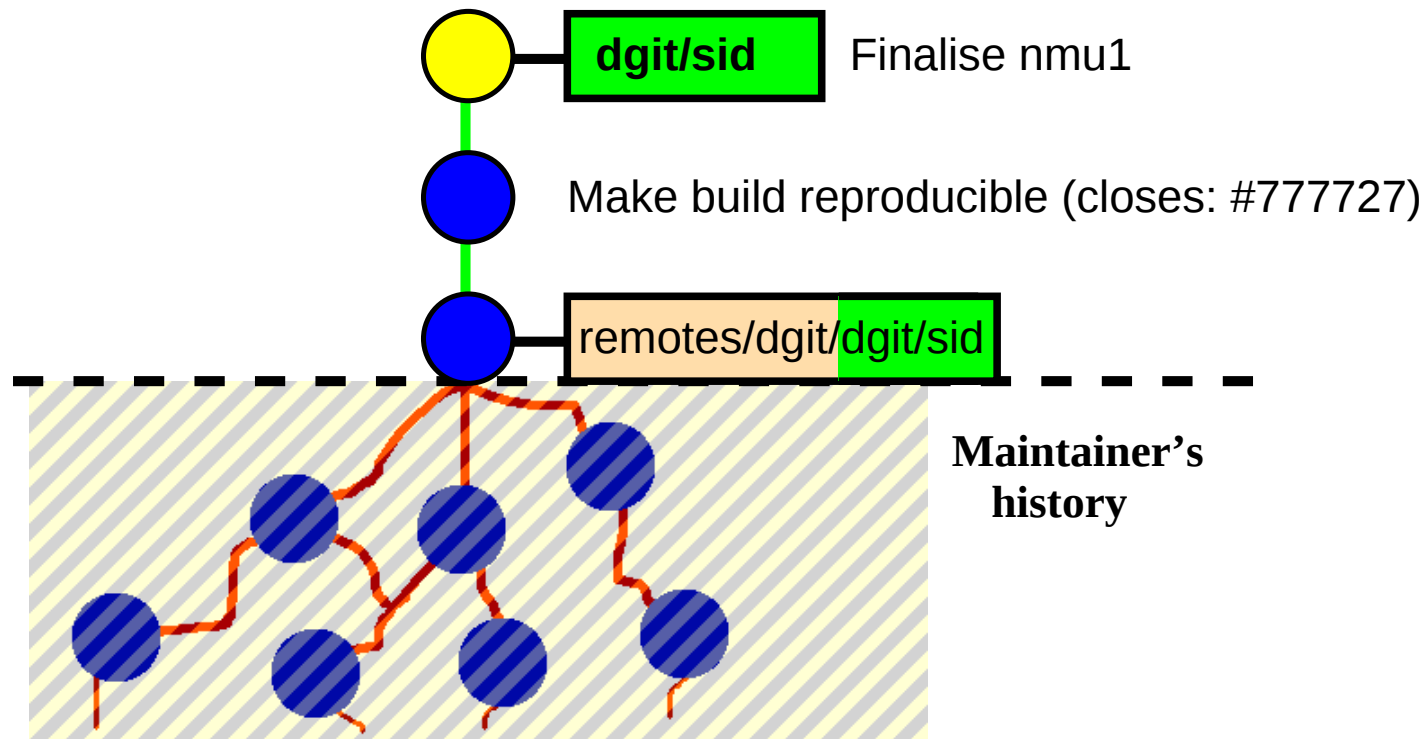
OPERATIONS

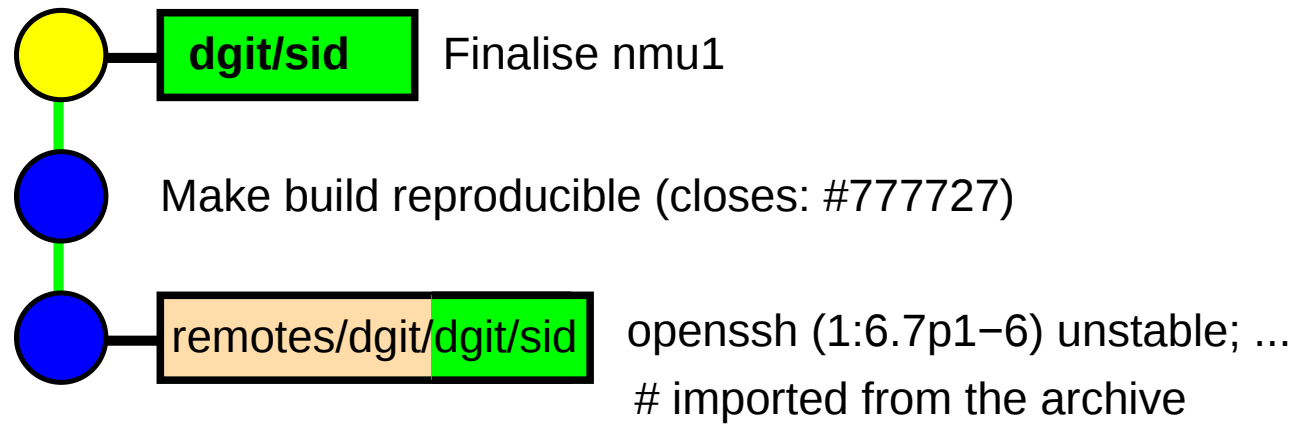
```
dggit clone package [suite] [./dir|/dir]
```

Consults the archive and dggit-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 **dggit/suite** ready for work, and on the corresponding dggit remote tracking branch. The **origin** remote will be set up to point to the package’s dggit repos

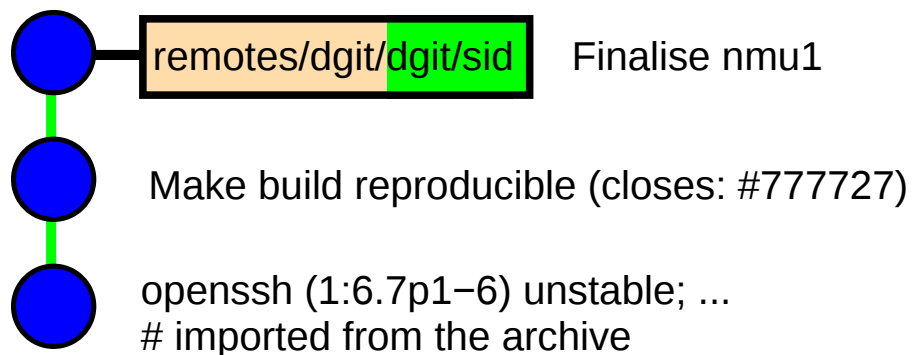




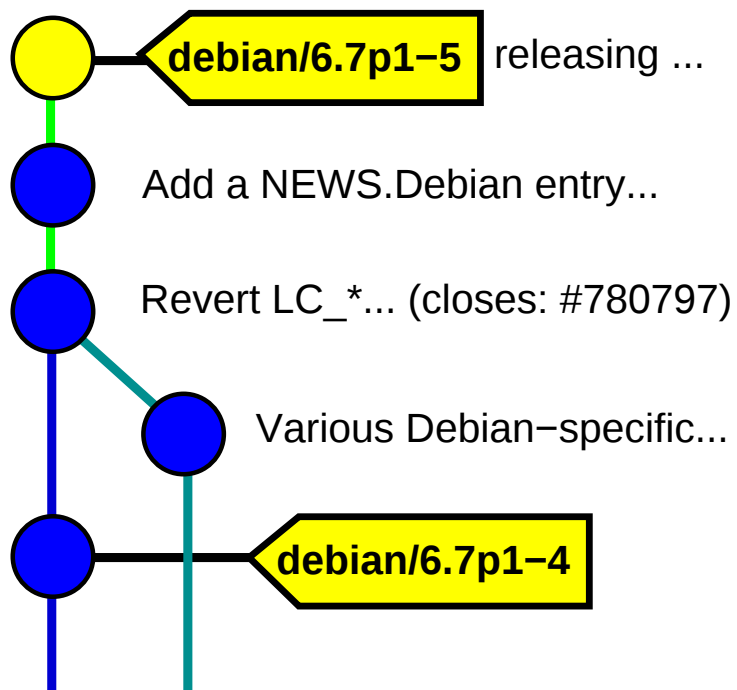


dgit synthetic history

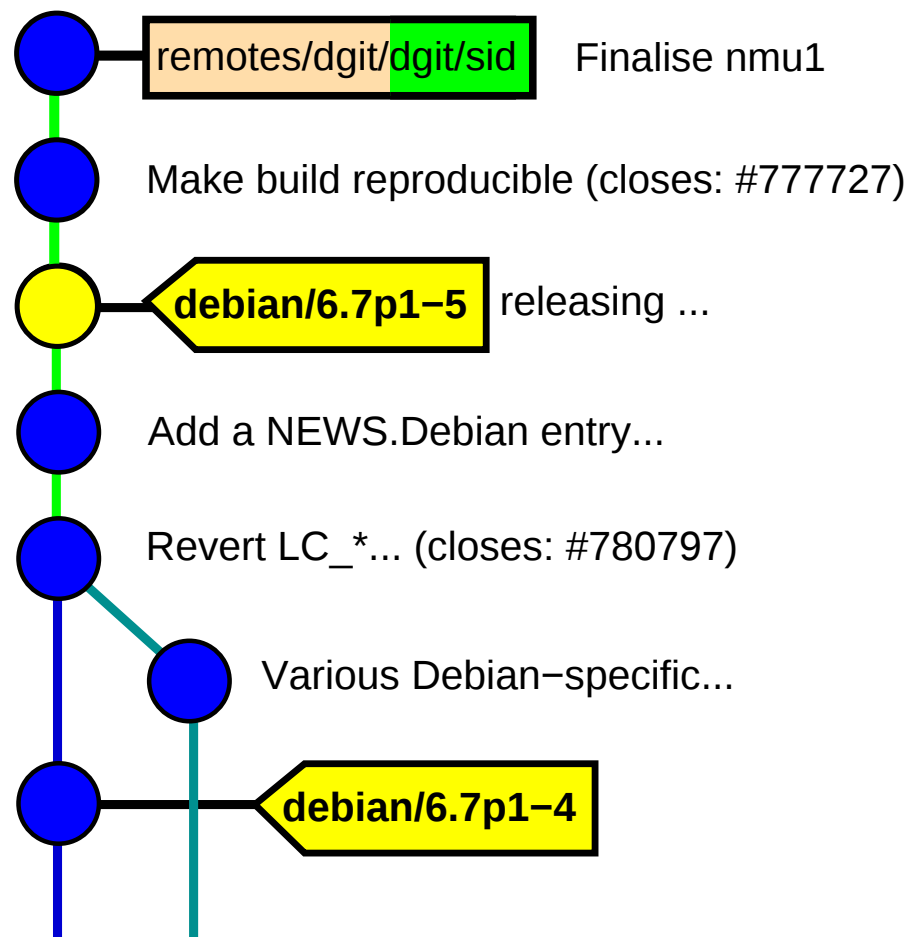
Maintainer does not use dgit

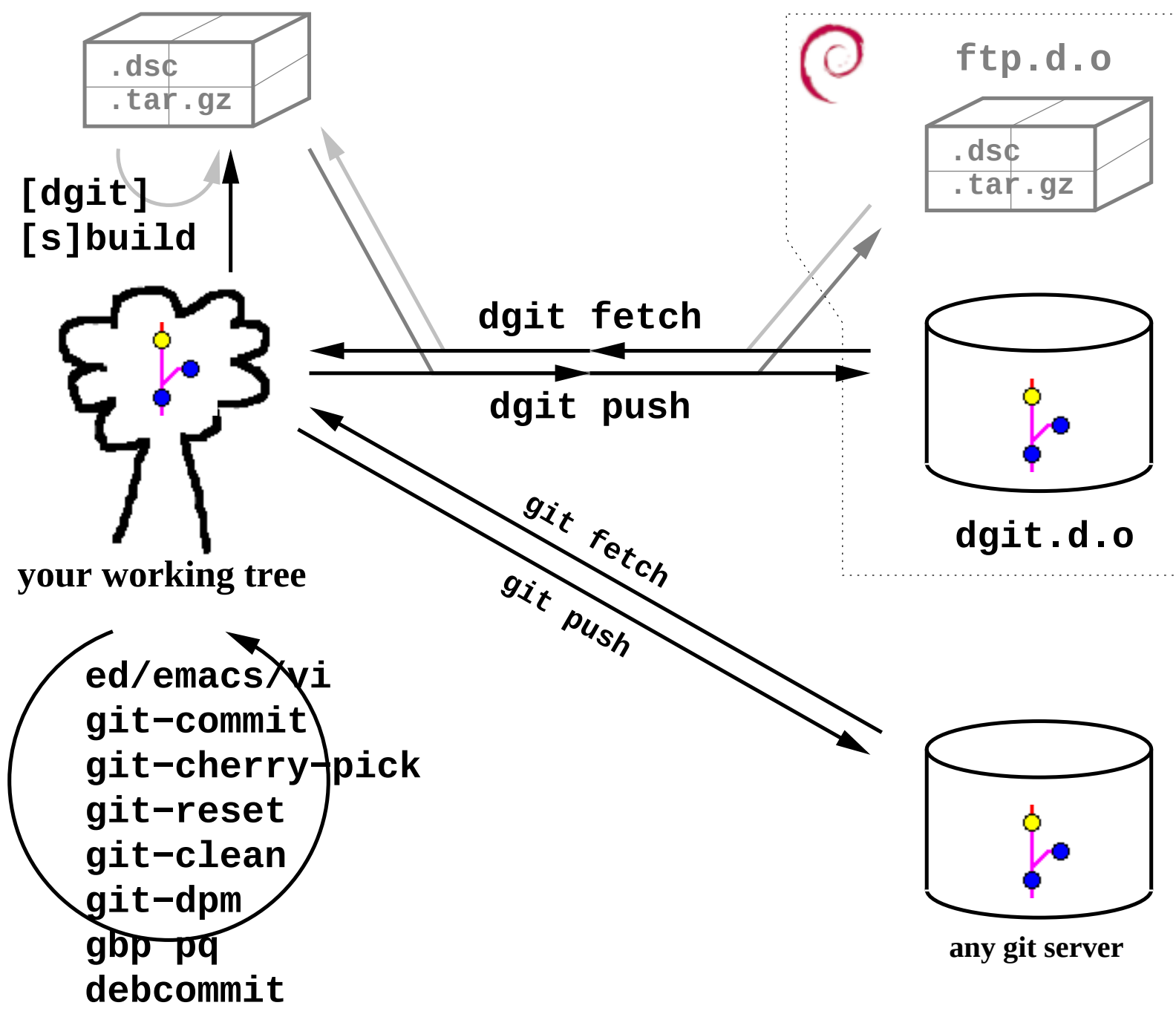


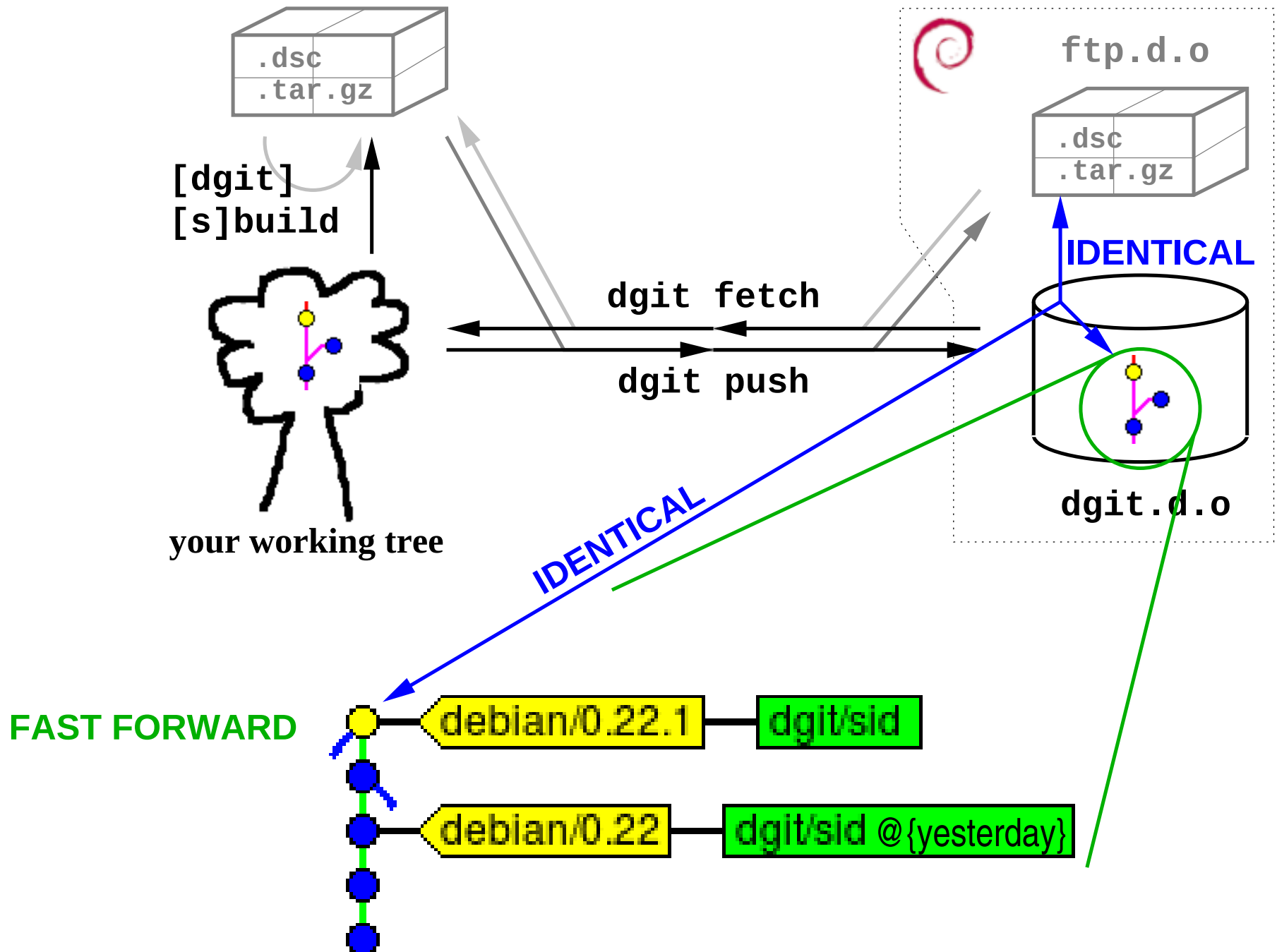
DISJOINT HISTORY



Maintainer uses dgit

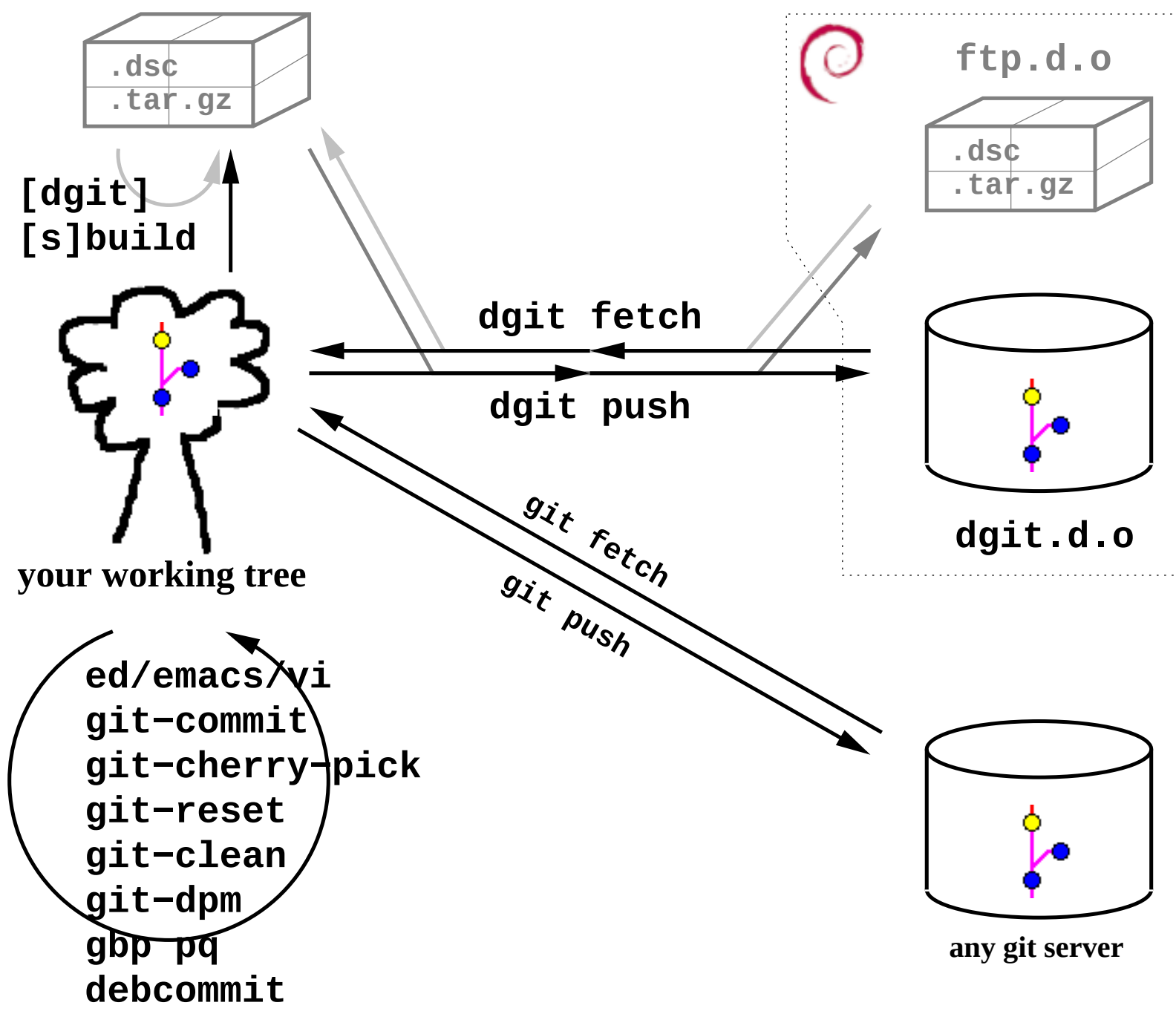


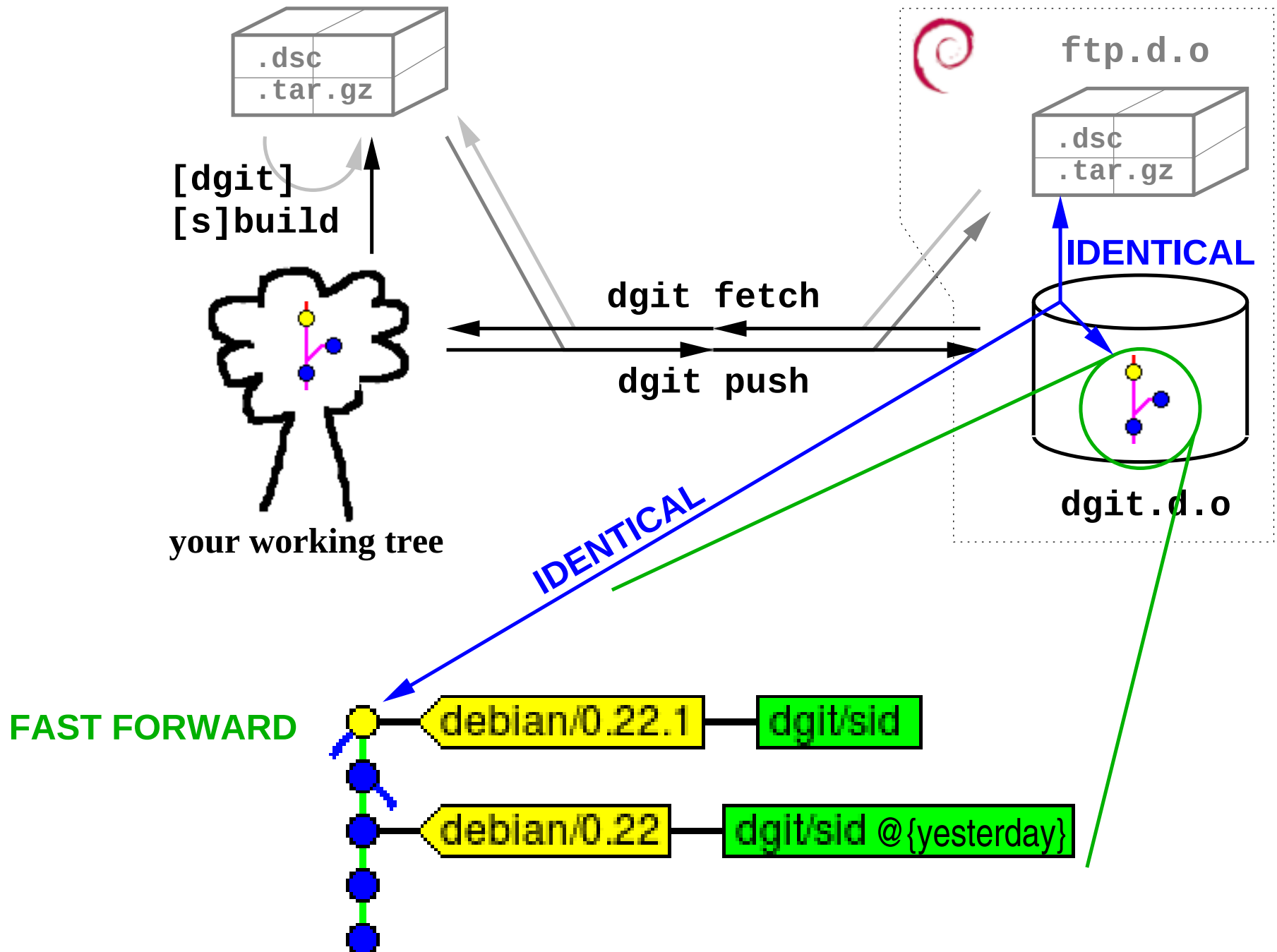




dgkit compatibility status for various maintainer workflows

source format	git workflow	push	integrate nmu
1.0 native	any	Go!	dgkit pull
1.0 diff	any (merging)	Go!	dgkit pull
1.0 diff	git-rebase or any other rebasing	Go!	dgkit fetch; git-merge -s ours (will be improved) or use git-import-dsc
3.0 (quilt)	git-buildpackage	Watch this space problems: patches applied vs. unapplied .gitignore	
3.0 (quilt)	git-dpm	Watch this space problems: .gitignore	





NAME

dggit – git integration with the Debian archive

SYNOPSIS

```
dggit [dggit-opts] clone [dggit-opts] package [suite] [./dir|/dir]
dggit [dggit-opts] fetch|pull [dggit-opts] [suite]
dggit [dggit-opts] build|sbuild|build-source [build-opts]
dggit [dggit-opts] push [dggit-opts] [suite]
dggit [dggit-opts] rpush build-host:build-dir [push args...]
dggit [dggit-opts] action ...
```

DESCRIPTION

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

The usual workflow is:

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

OPERATIONS

```
dggit clone package [suite] [./dir|/dir]
```

Consults the archive and dggit-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 **dggit/suite** ready for work, and on the corresponding dggit remote tracking branch. The **origin** remote will be set up to point to the package’s dggit repos

Specifies that we should process source package *package* rather than looking in `debian/control` or `debian/changelog`. Valid with `dgitch fetch` and `dgitch 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 `dgitch` 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 `dgitch 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

plans for my CFT

better integration with git-buildpackage

better integration with git-dpm

dggit push to DELAYED

automatically sending NMU diff to the BTS

one-step source-only push/upload

use of the dggit 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 dggit git
branches, of non-dggit 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

dggit-based sponsorship
request mechanism

dggit -d mint fetch rafaella
dggit -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/>

<http://www.chiark.greenend.org.uk/ucgi/~ijackson/git/talk-2015-debconf-dgit.git/>

Questions?