

# Git tools

---

Sylvain Bouveret, Grégory Mounié, Matthieu Moy  
2017

[first].[last]@imag.fr

<http://recherche.noiraudes.net/resources/git/git-slides.pdf>



## Git integrated graphical interface

---

Several basic tools are included with git. For portability reason (Tcl/Tk) they are ugly, yet convenient. They are good starting point to learn what are the basic operations.

**Git gui** commit edition

**Gitk** History navigation

**Gitweb** web server on the repository

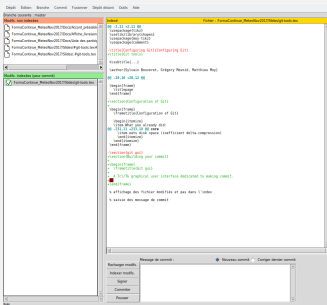
A Tcl/Tk graphical user interface  
dedicated to making commit with:

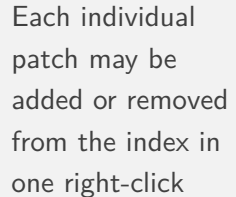
**Top left** list of modified and  
untracked files not in the  
index

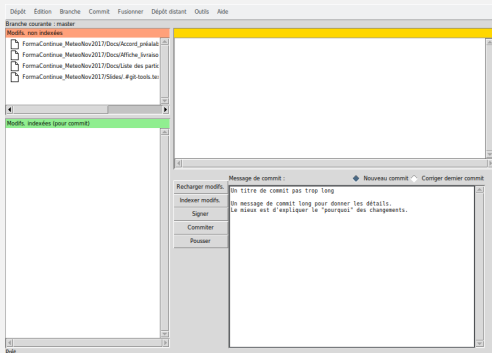
**bottom left** list of files in the index

**Up right** Patches associated with  
the selected file in one of  
the two previous area

**bottom** Message editor and  
index all/sign/commit  
buttons







Create the commit message with a very simple editor (yet still with dictionary check).

Amend previous commit in one button.

Commit may be signed with the private key of the developer.

Push in one click

It also helps to visit quickly who write the code, when and why (commit message)

```
Repository Edit Help
Commit: 03a0 03a0 11 " [--exec-path=<path>] [--html-path] [--man-path]
albe albe 12 " [-p|--paginate|--no-pager] [--no-replace-objects]
JT JT 13 " [--git-dir=<path>] [--work-tree=<path>] [--namesp
62b4 62b4 14 " <command> [<args>]";
822a 822a 15
b7d9 b7d9 16 const char git_more_info_string[] =
7390 7390 17 N_("'git help -a' and 'git help -g' lists available subcomman
PO PO 18 "concept guides. See 'git help <command>' or 'git help <co
| | 19 "to read about a specific subcommand or concept.");
b7d9 b7d9 20

commit 73903d0bcb00518e508f412a1d5c482b5094587e
Author: Philip Oakley <philipoakley@iee.org> Wed Apr 3 00:39:48 2013
Committer: Junio C Hamano <gitster@pobox.com> Wed Apr 3 03:11:08 2013

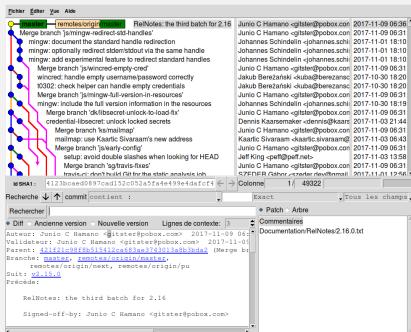
help: mention -a and -g option, and 'git help <concept>' usage.

Reword the overall help given at the end of "git help -a/-g" to
mention how to get help on individual commands and concepts.

Signed-off-by: Philip Oakley <philipoakley@iee.org>
Signed-off-by: Junio C Hamano <gitster@pobox.com>

Annotation complete.
```

# Gitk: visualize changes in repository



A Tcl/Tk graphical user interface dedicated to visualize changes in the repository:

**Top** commit graph with branches; author; date

**bottom left** message of the selected commit

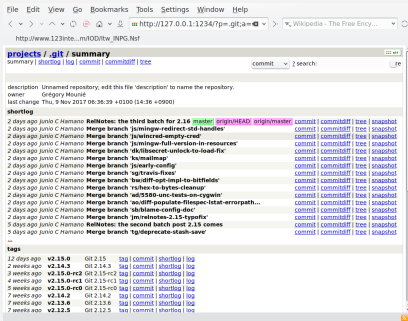
**bottom right** File list related to the selected commit

## Focus history

By default gitk shows the current branch history. `gitk --all` shows all branches. `gitk --since="2 weeks ago"`, the last two week changes.



# Gitweb: visualize changes on the web



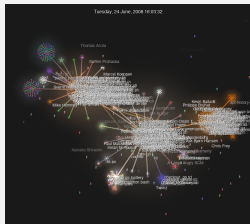
A web graphical user interface dedicated to visualize the repository content:

**Top** Description  
**Content** shortlog; tags;  
heads; remotes

To run it with lighttpd  
git instaweb

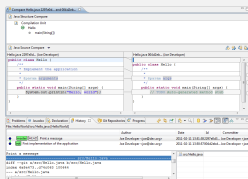
## Others interfaces

---

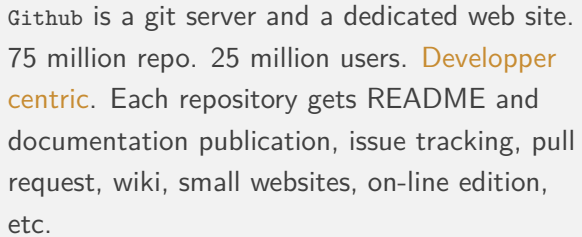


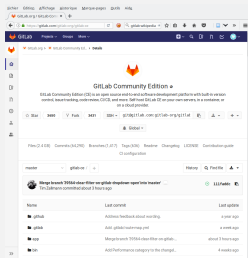
Gource animates the contribution among the time, around two points: files and contributors.

Many interfaces exist in order to fulfill different needs: eg. gitg (history exploration), git-cola (git gui on steroid), gitstats, magit (emacs mode), eget (git integration in eclipse), ...



Egit is the git plugin of Eclipse. Most functionalities are in right button "Team" menu (Add to Index, Commit, History...). Bottom picture from [http://wiki.eclipse.org/EGit/User\\_Guide](http://wiki.eclipse.org/EGit/User_Guide)





Gitlab provides similar fonctionnalités. Free private repositories at <http://gitlab.com>. It is **Open Source**. Easy to deploy at enterprise level (incl. LDAP) (eg. <http://gricad-gitlab.u-ga.fr>, <http://gitlab.inria.fr>).