

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and

1.

- 1.1 - .
- 1.2 - Git
- 1.3 - Git
- 1.4 - Git
- 1.5 - Git
- 1.6 - .
- 1.7 - .

2. Git

2.1 - Git

2.2 - .

2.3 - .

2.4 - .

2.5 - .

2.6 - .

2.7 - .

2.8 - .

3. Git

3.1 - .

3.2 - .

3.3 - .

3.4 - .

3.5 - .

3.6 - .

3.7 - .

4. Git

- 4.1 - .
- 4.2 - Git
- 4.3 - SSH
- 4.4 - .
- 4.5 - .
- 4.6 - GitWeb
- 4.7 - Gitosis
- 4.8 - Git
- 4.9 - Git
- 4.10 - .

5. Git

5.1 - .

5.2 - .

5.3 - .

5.4 - .

6. Git

6.1 - Revision

6.2 - .

6.3 - Stashing

6.4 - .

6.5 - Git

6.6 - .

6.7 - .

6.8 - .

7. Git

7.1 - Git

7.2 - Git

7.3 - Git

7.4 - An Example Git-Enforced Policy

7.5 - Summary

8. Git

8.1 - Git Subversion

8.2 - Git

8.3 - .

9. Git

9.1 - [\(Plumbing\) \(Porcelain\)](#)

9.2 - [Git](#)

9.3 - [Git References](#)

9.4 - [Packfiles](#)

9.5 - [The Refspec](#)

9.6 - .

9.7 - .

9.8 - .

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and

Chapter 1

Git Git Git

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).
Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

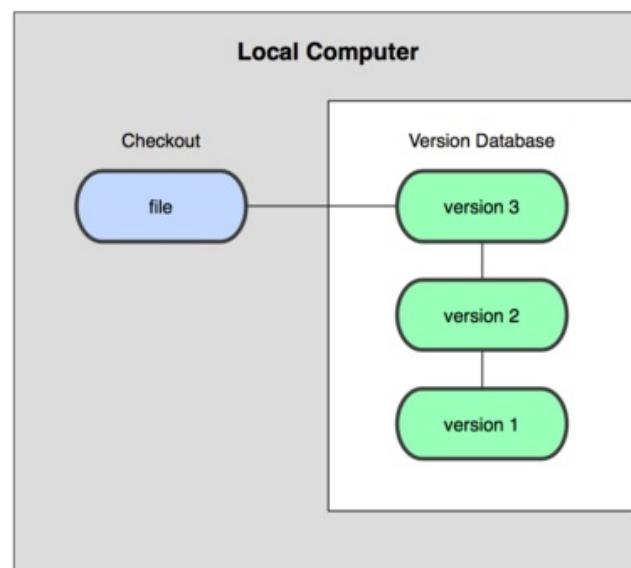
[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

VCS

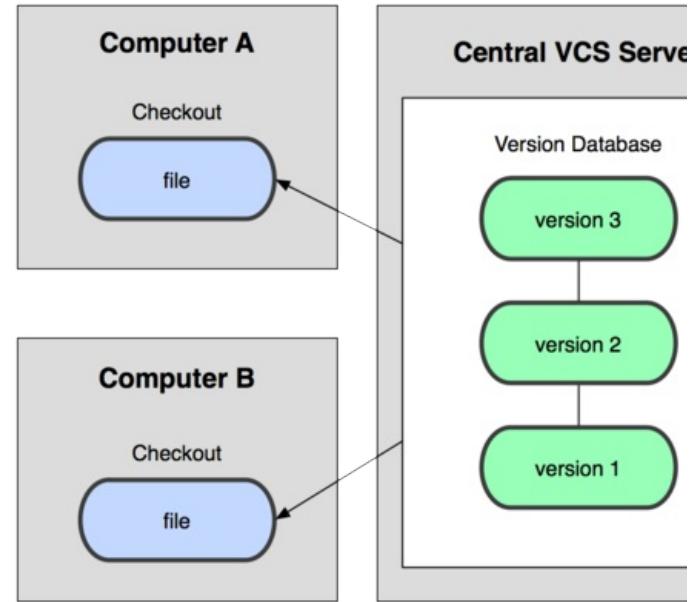
1-1



1-1.

rcs Mac OS X rcs patchrcs

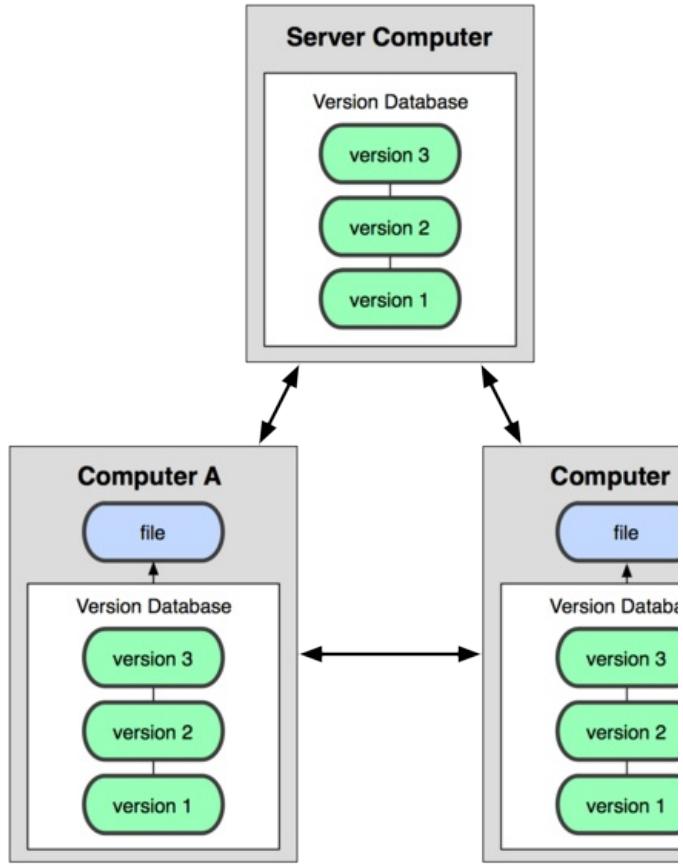
Centralized Version Control Systems CVCS CVSSubversion Perfor
1-2



1-2.

VCS CVCS

Distributed Version Control System DVCS GitMercurialBazaar Darc
1-3



1-3.

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).
 Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .
 All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).
 Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

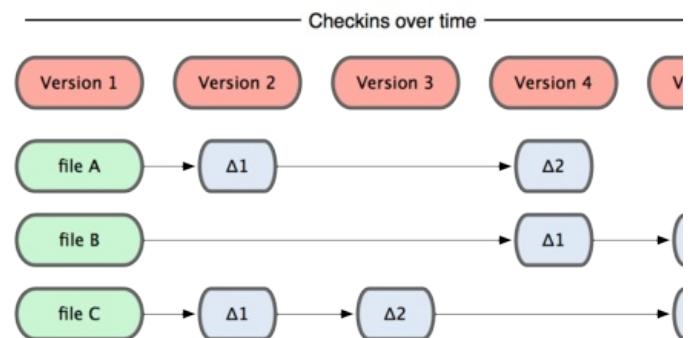
This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

Git

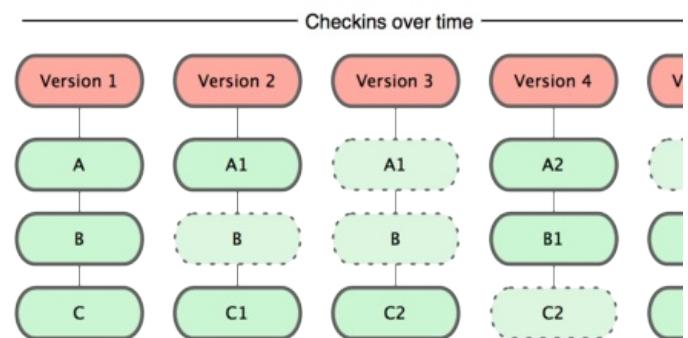
Git Git Git Subversion Perforce Git
Git

Git Git CVSSubversionPerforceBazaar 1-4



1-4.

Git Git Git Git 1-5



1-5. Git

Git Git VCS Git

Git CVCS Git

Git Git

CVCS VPN Git VPN Perforce
Perforce Subversion CVS

Git checksumGit Git Git

Git SHA-1 SHA-1 40 0-9 a-f

```
24b9da6552252987aa493b52f8696cd6d3b00373
```

Git Git

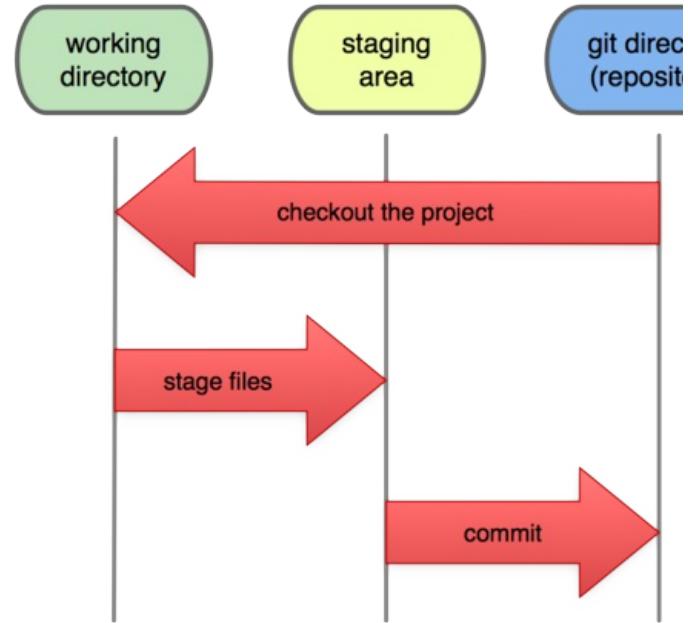
Git VCS Git

Git ""

Git committedmodifiedstaged

Git Git

Local Operations



1-6. git

git Git

git

git

Git

1. 2. 3. git

git

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#).

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and

Git

Git

Git Linux distro backports

Git curlzlibopensslxpatlibleconv yum Fedora apt-get Debian

```
$ yum install curl-devel expat-devel gettext-devel \
openssl-devel zlib-devel

$ apt-get install curl-devel expat-devel gettext-devel \
openssl-devel zlib-devel
```

Git

<http://git-scm.com/download>

```
$ tar -zxf git-1.6.0.5.tar.gz
$ cd git-1.6.0.5
$ make prefix=/usr/local all
$ sudo make prefix=/usr/local install
```

git git Git

```
$ git clone git://git.kernel.org/pub/scm/git/git.git
```

Linux

Linux Git Fedora yum

```
$ yum install git-core
```

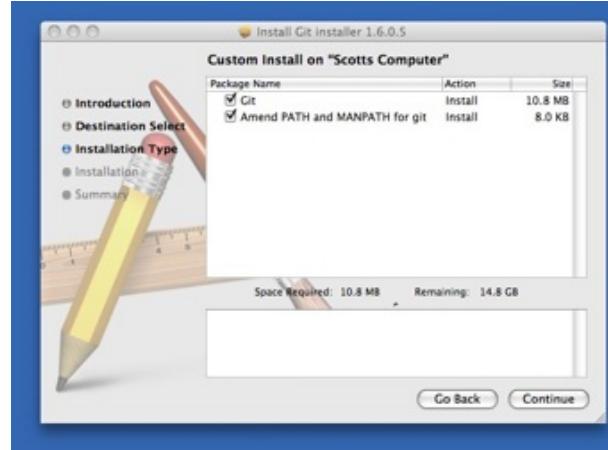
Ubuntu Debian apt-get

```
$ apt-get instal git-core
```

Mac

Mac Git Git 1-7

```
http://code.google.com/p/git-osx-installer
```



1-7. Git OS X

MacPorts (<http://www.macports.org>) MacPorts Git

```
$ sudo port install git-core +svn +doc +bash_completion +gitweb
```

Macports Git Subversion +svn

Windows

Windows Git msysGit Google Code .exe

<http://code.google.com/p/msysgit>

git ssh Git

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

Git

Git

Git git config git-config git Git

- /etc/gitconfig git config --system
- ~/.gitconfig git config --global
- git .git/config .git/config /etc/gitcon

Windows Git .gitconfig \$HOME C:\Documents and ...
/etc/gitconfig Git

Git

```
$ git config --global user.name "John Doe"  
$ git config --global user.email johndoe@example.com
```

--global --global .git/config

Git Vi Vim Emacs

```
$ git config --global core.editor emacs
```

vimdiff

```
$ git config --global merge.tool vimdiff
```

Git kdiff3tkdiffmeldxxdiffmergevimdiffgvimdiffecmerge opendiff

```
git config --list
```

```
$ git config --list
user.name=Scott Chacon
user.email=schacon@gmail.com
color.status=auto
color.branch=auto
color.interactive=auto
color.diff=auto
...
```

/etc/gitconfig ~/.gitconfig Git

```
$ git config user.name
Scott Chacon
```

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#).

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).
Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and

Git

```
$ git help <verb>
$ git <verb> --help
$ man git-<verb>
```

config

```
$ git help config
```

Frenode IRC irc.freenode.net

#git #github git

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#).

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

Git CVCS Git Git

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and

Chapter 2

Git

Git Git Git

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).
Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

Git

Git Git Git

Git

```
$ git init
```

.git Git

.git

git add Git

```
$ git add *.c  
$ git add README  
$ git commit -m 'initial project version'
```

Git

Git git clone VCS Subversion clone checkoutGit

git clone [url] Ruby Git Grit

```
$ git clone git://github.com/schacon/grit.git
```

“grit”

.git

grit

```
$ git clone git://github.com/schacon/grit.git mygrit
```

mygrit

Git <git://> [http\(s\)://](http://) user@server:/path.git SSH

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and

```
git log
```

simplegit

```
git clone git://github.com/schacon/simplegit-progit.git
```

```
git log
```

```
$ git log
commit ca82a6dff817ec66f44342007202690a93763949
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Mon Mar 17 21:52:11 2008 -0700

    changed the verison number

commit 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Sat Mar 15 16:40:33 2008 -0700

    removed unnecessary test code

commit a11bef06a3f659402fe7563abf99ad00de2209e6
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Sat Mar 15 10:31:28 2008 -0700

    first commit
```

```
git log SHA-1
```

```
git log
```

```
-p      -2
```

```
$ git log -p -2
commit ca82a6dff817ec66f44342007202690a93763949
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Mon Mar 17 21:52:11 2008 -0700
```

```
changed the verison number

diff --git a/Rakefile b/Rakefile
index a874b73..8f94139 100644
--- a/Rakefile
+++ b/Rakefile
@@ -5,7 +5,7 @@ require 'rake/gempackagetask'
spec = Gem::Specification.new do |s|
-  s.version    = "0.1.0"
+  s.version    = "0.1.1"
  s.author     = "Scott Chacon"

commit 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7
Author: Scott Chacon <schacon@gee-mail.com>
Date: Sat Mar 15 16:40:33 2008 -0700

removed unnecessary test code

diff --git a/lib/simplegit.rb b/lib/simplegit.rb
index a0a60ae..47c6340 100644
--- a/lib/simplegit.rb
+++ b/lib/simplegit.rb
@@ -18,8 +18,3 @@ class SimpleGit
end

end
-
-if $0 == __FILE__
-  git = SimpleGit.new
-  puts git.show
-end
\ No newline at end of file
```

--stat

```
$ git log --stat
commit ca82a6dff817ec66f44342007202690a93763949
Author: Scott Chacon <schacon@gee-mail.com>
Date: Mon Mar 17 21:52:11 2008 -0700

changed the verison number

Rakefile |    2 ++
1 files changed, 1 insertions(+), 1 deletions(-)

commit 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7
Author: Scott Chacon <schacon@gee-mail.com>
Date: Sat Mar 15 16:40:33 2008 -0700

removed unnecessary test code

lib/simplegit.rb |    5 -----
1 files changed, 0 insertions(+), 5 deletions(-)

commit a11bef06a3f659402fe7563abf99ad00de2209e6
Author: Scott Chacon <schacon@gee-mail.com>
Date: Sat Mar 15 10:31:28 2008 -0700

first commit
```

```
README |      6 ++++++
Rakefile |     23 ++++++*****+++++++
lib/simplegit.rb |   25 ++++++*****+++++++
3 files changed, 54 insertions(+), 0 deletions(-)
```

--pretty oneline shortfull fuller

```
$ git log --pretty=oneline
ca82a6dff817ec66f44342007202690a93763949 changed the verison number
085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7 removed unnecessary test code
a11bef06a3f659402fe7563abf99ad00de2209e6 first commit
```

format

```
$ git log --pretty=format:"%h - %an, %ar : %s"
ca82a6d - Scott Chacon, 11 months ago : changed the verison number
085bb3b - Scott Chacon, 11 months ago : removed unnecessary test code
a11bef0 - Scott Chacon, 11 months ago : first commit
```

2-1

```
%H        commit
%h
%T        tree
%t
%P        parent
%p
%an       author
%ae
%ad       -date=
%ar
%cn       (committer)
%ce
%cd
%cr
%s
```

_author__committer_

oneline format --graph ASCII Grit

```
$ git log --pretty=format:"%h %s" --graph
* 2d3acf9 ignore errors from SIGCHLD on trap
* 5e3ee11 Merge branch 'master' of git://github.com/dustin/grit
 \\
| * 420eac9 Added a method for getting the current branch.
* | 30e367c timeout code and tests
* | 5a09431 add timeout protection to grit
* | e1193f8 support for heads with slashes in them
|
* d6016bc require time for xmlschema
* 11d191e Merge branch 'defunkt' into local
```

git log 2-2

```
-p
--stat
--shortstat --stat
--name-only
--name-status
--abbrev-commit SHA-1 40
--relative-date "2 weeks ago"
--graph ASCII
--pretty onelinershortfullfuller format
```

git log -2 -<n> n Git pager

--since --until

```
$ git log --since=2.weeks
```

“2008-01-15”“2 years 1 day 3 minutes ago”

--author --grep --all-match

git log --

2-3

```
-(n)      n
--since, --after
--until, --before
--author
--committer
```

Git 2008 10 Junio Hamano t/

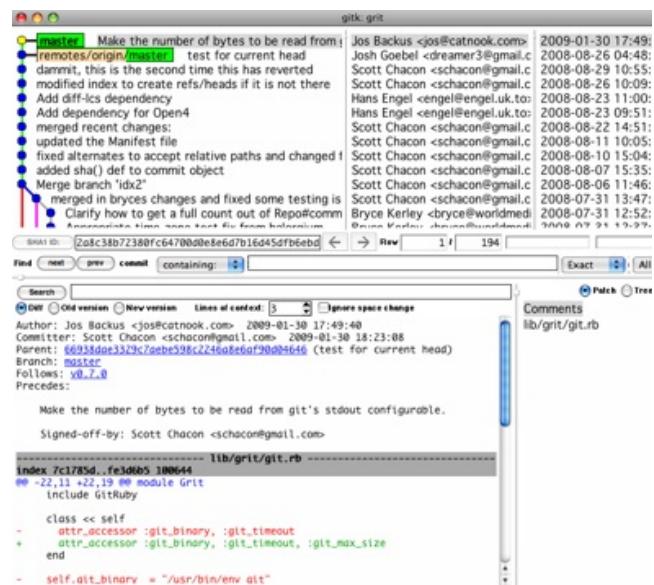
```
$ git log --pretty="%h:%s" --author=gitster --since="2008-10-01" \
--before="2008-11-01" --no-merges -- t/
5610e3b - Fix testcase failure when extended attribute
acd3b9e - Enhance hold_lock_file_for_{update,append}()
f563754 - demonstrate breakage of detached checkout wi
d1a43f2 - reset --hard/read-tree --reset -u: remove un
51a94af - Fix "checkout --track -b newbranch" on detac
b0ad11e - pull: allow "git pull origin $something:$cur
```

Git 20,000 6

Git gitk Tcl/Tk

git log

git log gitk gitk 2-2



2-2. gitk

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

--amend

```
$ git commit --amend
```

--amend

```
$ git commit -m 'initial commit'  
$ git add forgotten_file  
$ git commit --amend
```

git add * git status

```
$ git add .  
$ git status  
# On branch master  
# Changes to be committed:  
#   (use "git reset HEAD <file>..." to unstage)  
#  
#       modified: README.txt  
#       modified: benchmarks.rb  
#
```

“Changes to be committed”

git reset HEAD <file>... be

```
$ git reset HEAD benchmarks.rb
benchmarks.rb: locally modified
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       modified:   README.txt
#
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   benchmarks.rb
#
```

benchmarks.rb

benchmarks.rb

git status

```
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   benchmarks.rb
#
```

Git 1.6.1

```
$ git checkout -- benchmarks.rb
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       modified:   README.txt
#
```

stashing

Git

-- amend Git

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#).

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and

Git Git

Bash shell Git Git contrib/completion git-compl
cp git-completion.bash ~/.git-completion.bash

```
source ~/.git-completion.bash
```

Mac /opt/local/etc/bash_completion.d Linux /etc/bash_c
Bash

Windows msysGit Git Bash

Git Tab

```
$ git co<tab><tab>  
commit config
```

git co Tab commit config m<tab> git committ

git log Tab

```
$ git log --s<tab>  
--shortstat --since= --src-prefix= --stat --summary
```

Git

Git git config

```
$ git config --global alias.co checkout  
$ git config --global alias.br branch  
$ git config --global alias.ci commit  
$ git config --global alias.st status
```

git commit git ci Git

```
$ git config --global alias.unstage 'reset HEAD --'
```

```
$ git unstage fileA  
$ git reset HEAD fileA
```

last

```
$ git config --global alias.last 'log -1 HEAD'
```

```
$ git last  
commit 66938dae3329c7aebe598c2246a8e6af90d04646  
Author: Josh Goebel <dreamer3@example.com>  
Date: Tue Aug 26 19:48:51 2008 +0800  
  
test for current head  
  
Signed-off-by: Scott Chacon <schacon@example.com>
```

Git Git

! Git

git visual

```
$ git config --global alias.visual "!gitk"
```

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

Git Git

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and

Chapter 3

Git

Git “” Git Git Git Git Git

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and

Git Git Git

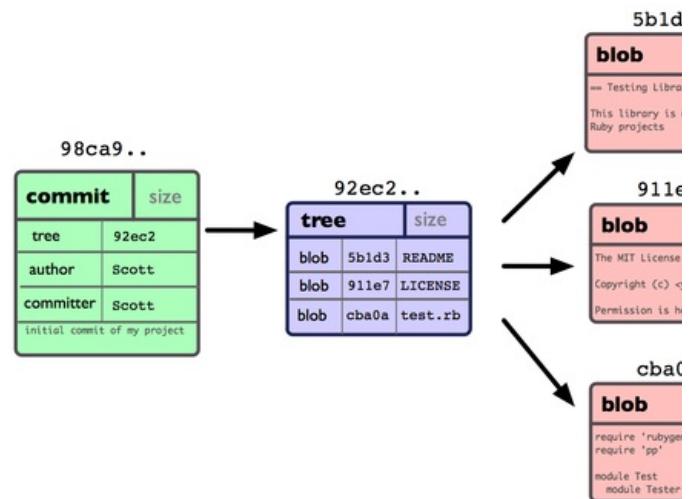
Git commit

SHA-1 Git Git blob

```
$ git add README test.rb LICENSE2  
$ git commit -m 'initial commit of my project'
```

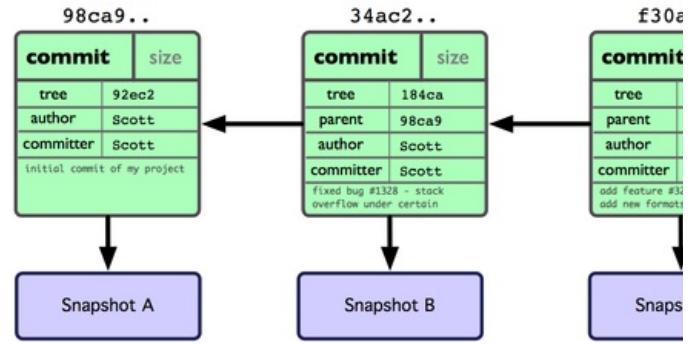
git commit Git Git tree Git

Git blob blob tree tree commit 3-1



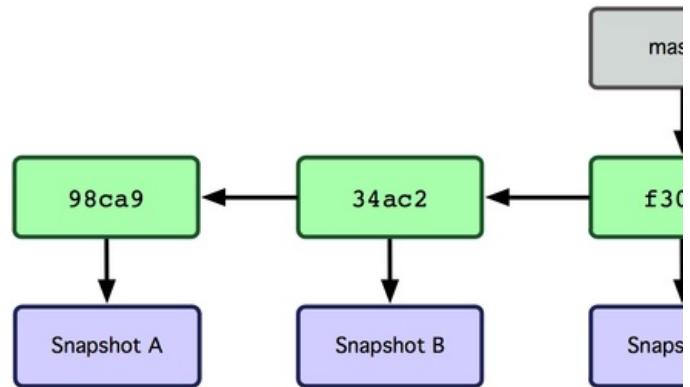
3-1.

parent 3-2



3-2. Git

Git commit Git master master

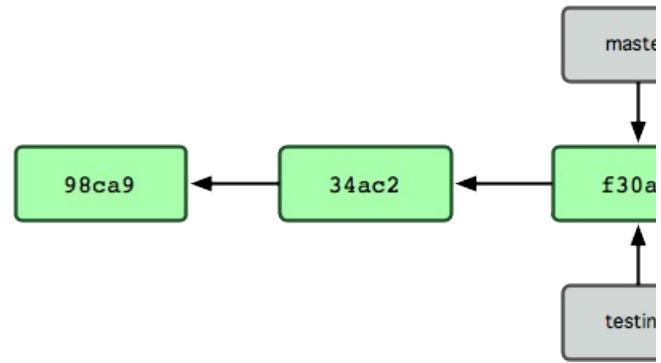


3-3.

Git testing git branch

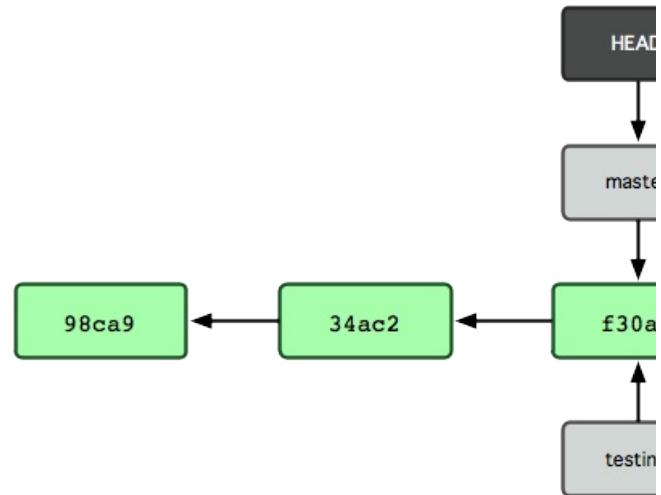
```
$ git branch testing
```

commit 3-4



3-4.

Git HEAD Subversion CVS HEAD Git
master 3-5

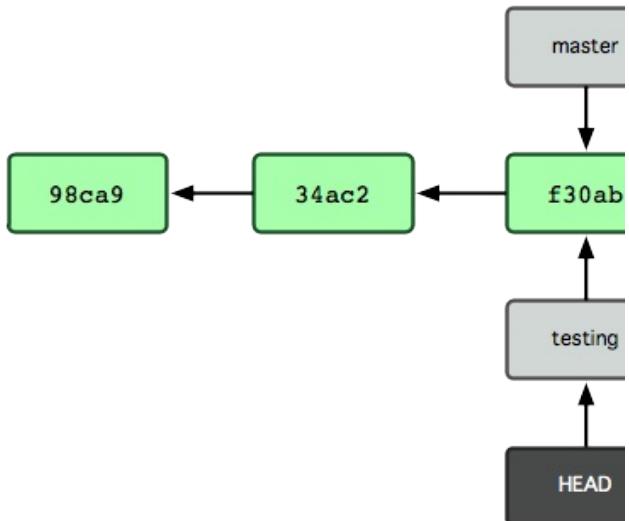


3-5. HEAD

git checkout testing

```
$ git checkout testing
```

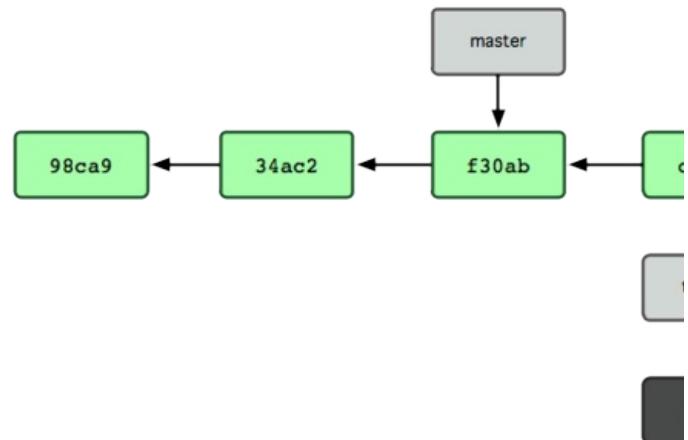
HEAD testing 3-6



3-6. HEAD

```
$ vim test.rb  
$ git commit -a -m 'made a change'
```

3-7



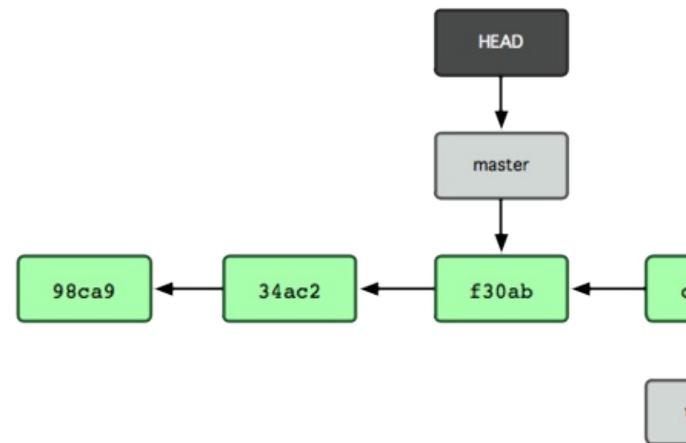
3-7. HEAD

testing master

git checkout commit master

```
$ git checkout master
```

3-8



3-8. HEAD checkout

HEAD master master testing

```
$ vim test.rb  
$ git commit -a -m 'made other changes'
```

3-9

branch checkout



3-9.

Git 40 SHA-1 41

Git parent Git

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

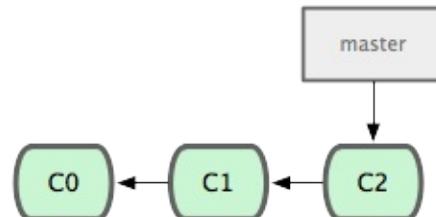
This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

1. 2. 3.

1. 2. 3. 4.

3-10



3-10.

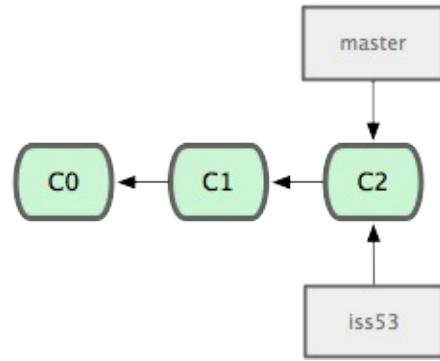
#53 Git iss53

git checkout -b

```
$ git checkout -b iss53
Switched to a new branch "iss53"
```

```
$ git branch iss53
$ git checkout iss53
```

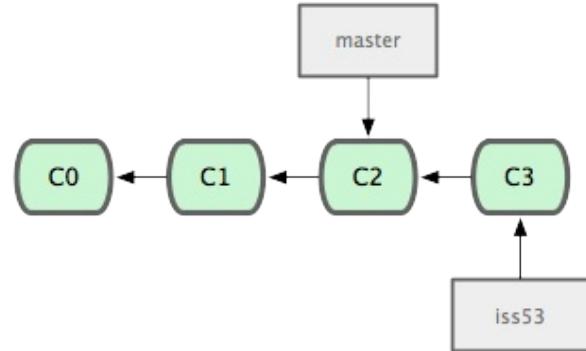
3-11



3-11.

iss53 HEAD 3-12

```
$ vim index.html
$ git commit -a -m 'added a new footer [issue 53]'
```



3-12. iss53

Git iss53 master

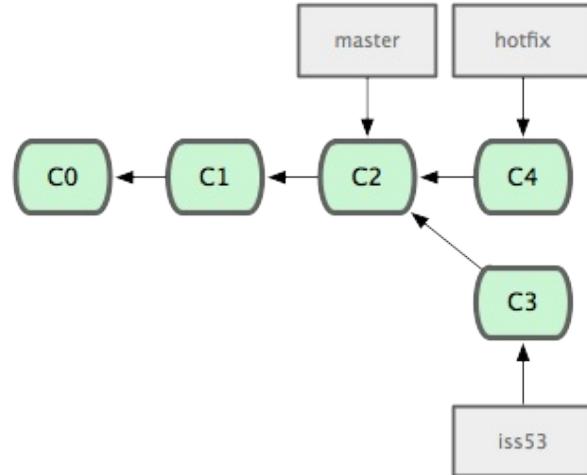
Git stashing amending master

```
$ git checkout master
Switched to branch "master"
```

#53 Git commit

hotfix 3-13

```
$ git checkout -b 'hotfix'  
Switched to a new branch "hotfix"  
$ vim index.html  
$ git commit -a -m 'fixed the broken email address'  
[hotfix]: created 3a0874c: "fixed the broken email address"  
1 files changed, 0 insertions(+), 1 deletions(-)
```



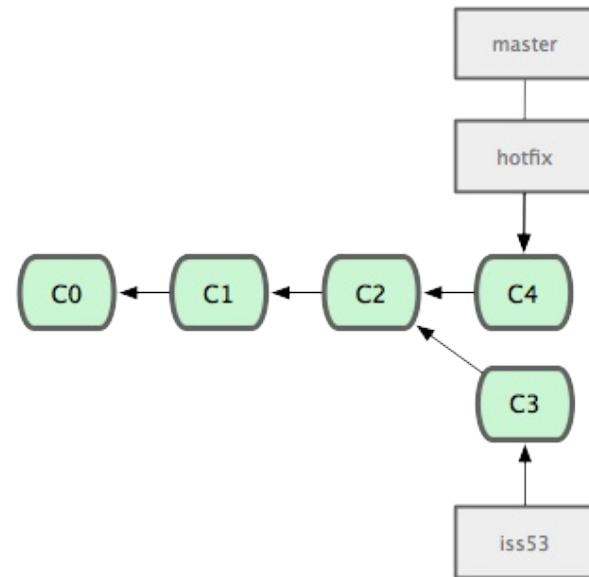
3-13. hotfix master

master git merge

```
$ git checkout master  
$ git merge hotfix  
Updating f42c576..3a0874c  
Fast forward  
 README | 1 -  
 1 files changed, 0 insertions(+), 1 deletions(-)
```

“Fast forward” master commit hotfix Git Git
Fast forward

master commit 3-14



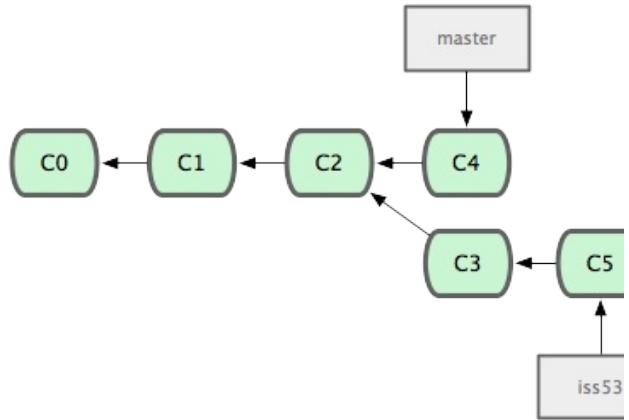
3-14. master hotfix

hotfix master git branch -d

```
$ git branch -d hotfix
Deleted branch hotfix (3a0874c).
```

#53 3-15

```
$ git checkout iss53
Switched to branch "iss53"
$ vim index.html
$ git commit -a -m 'finished the new footer [issue 53]'
[iss53]: created ad82d7a: "finished the new footer [issue 53]"
 1 files changed, 1 insertions(+), 0 deletions(-)
```



3-15. iss53

hotfix iss53 git merge master master iss53 i

#53 master hotfix master git merge

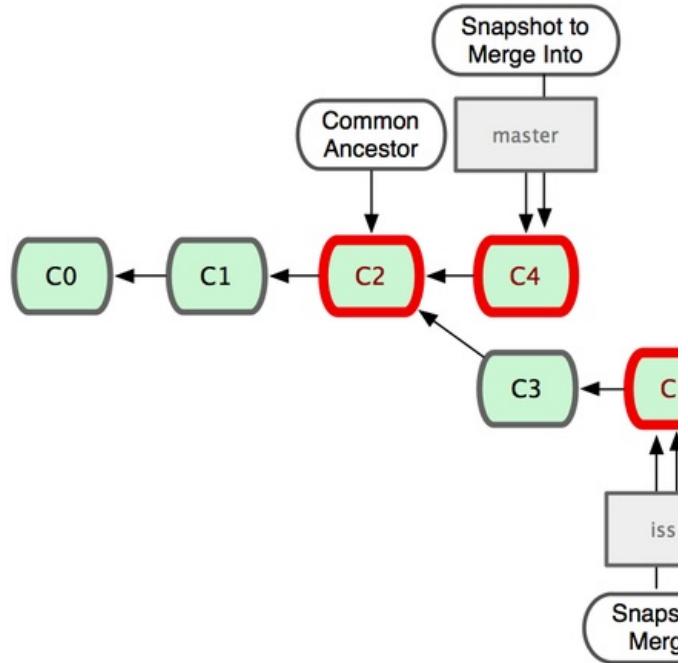
```

$ git checkout master
$ git merge iss53
Merge made by recursive.
 README | 1 +
 1 files changed, 1 insertions(+), 0 deletions(-)

```

hotfix master commit (C4)iss53Git Git C4 C5C2

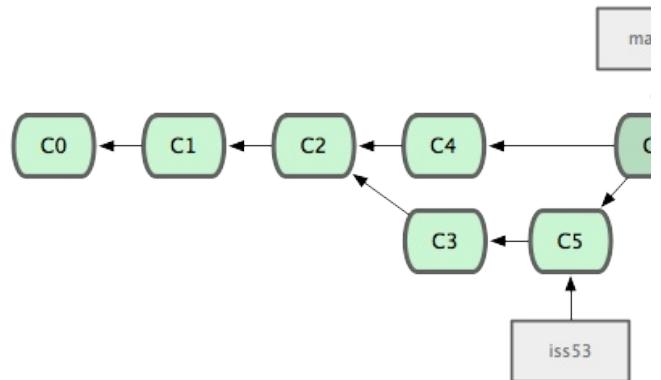
3-16 Git



3-16. Git

Git commit
C6 3-17 commit merge commit

Git CVS Subversion1.5 Git



3-17. Git commit

iss53

```
$ git branch -d iss53
```

Git #53

hotfix

```
$ git merge iss53
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.
```

Git

git status

```
[master*]$ git status
index.html: needs merge
# On branch master
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       unmerged:    index.html
#
```

unmergedGit

```
<<<<< HEAD:index.html
<div id="footer">contact : email.support@github.com</div>
=====
<div id="footer">
  please contact us at support@github.com
</div>
>>>>> iss53:index.html
```

===== HEAD master merge

iss53

```
<div id="footer">
  please contact us at email.support@github.com
</div>
```

<<<<<=====>>>>>

git add resolved

```
$ git mergetool
merge tool candidates: kdiff3 tkdiff xxdiff meld gvimdiff opendiff emerge vimdiff
```

```
Merging the files: index.html

Normal merge conflict for 'index.html':
{local}: modified
{remote}: modified
Hit return to start merge resolution tool (opendiff):
```

Git opendiff Mac "merge tool candidates"

Git

```
git status
```

```
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       modified:   index.html
#
```

git commit

```
Merge branch 'iss53'

Conflicts:
  index.html
#
# It looks like you may be committing a MERGE.
# If this is not correct, please remove the file
# .git/MERGE_HEAD
# and try again.
#
```

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).
Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .
All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).
Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and

```
git branch
```

```
$ git branch  
iss53  
* master  
testing
```

```
master * master commit git branch -v
```

```
$ git branch -v  
iss53 93b412c fix javascript issue  
* master 7a98805 Merge branch 'iss53'  
          testing 782fd34 add scott to the author list in the readmes
```

```
--merge --no-merged Git 1.5.6 git branch -me
```

```
$ git branch --merged  
iss53  
* master
```

```
iss53 * git branch -d
```

```
git branch --no-merged
```

```
$ git branch --no-merged  
testing
```

```
git branch -d
```

```
$ git branch -d testing  
error: The branch 'testing' is not an ancestor of your current HEAD.
```

-D `git branch -D testing`

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).
Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

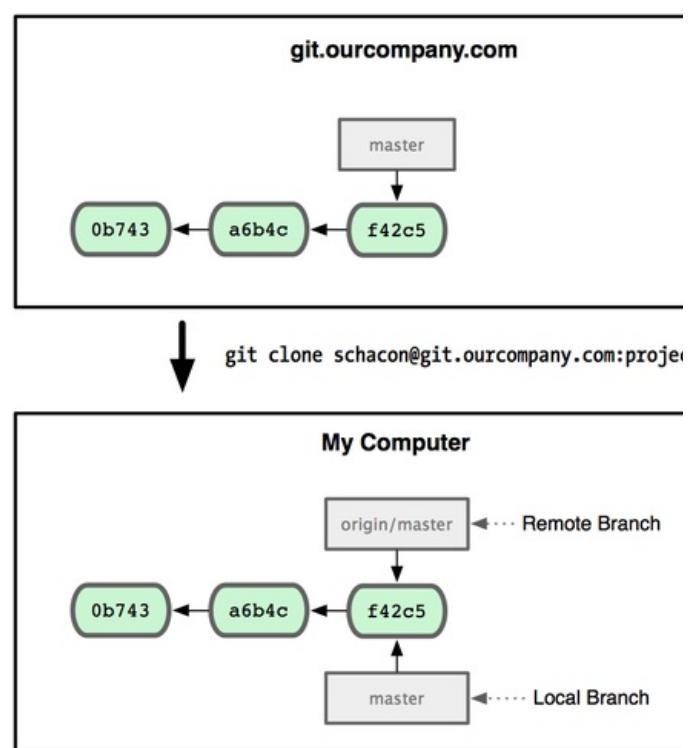
[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

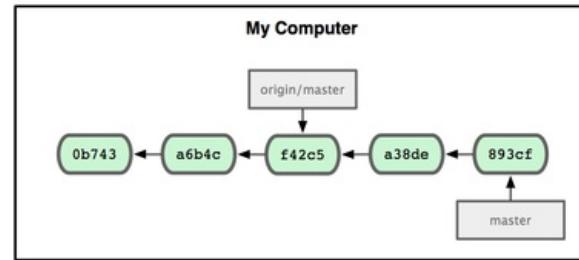
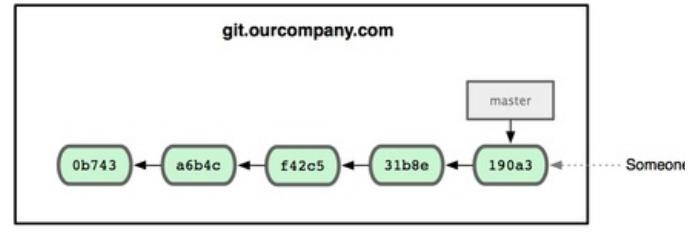
remote branch Git

```
( )/( )          origin  master      origin/master      iss5  
origin/iss53  
  
git.ourcompany.com  Git  Git      origin      master  
master      origin  master  3-22
```



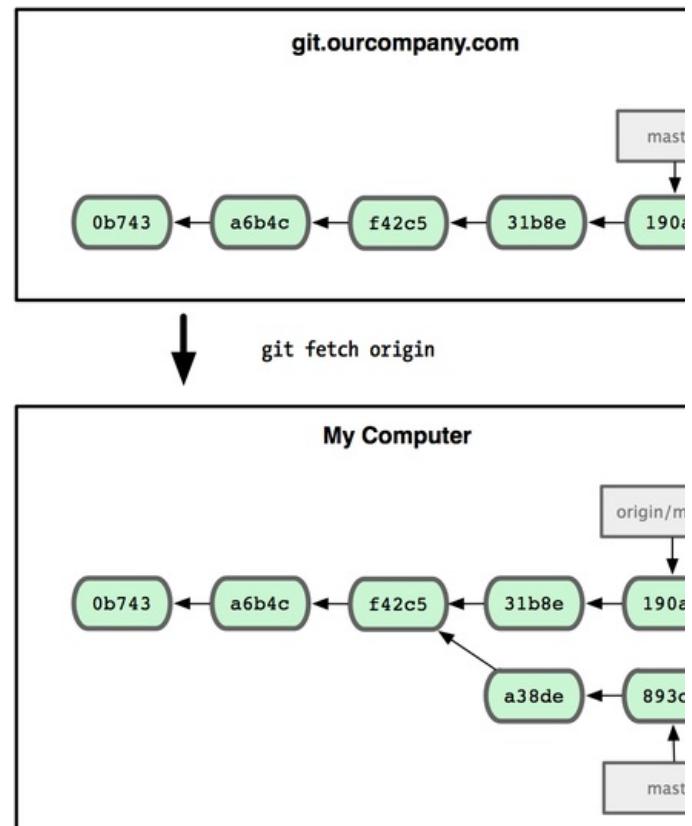
3-22. Git master origin/master origin/master

```
master      git.ourcompany.com  master      origin
```



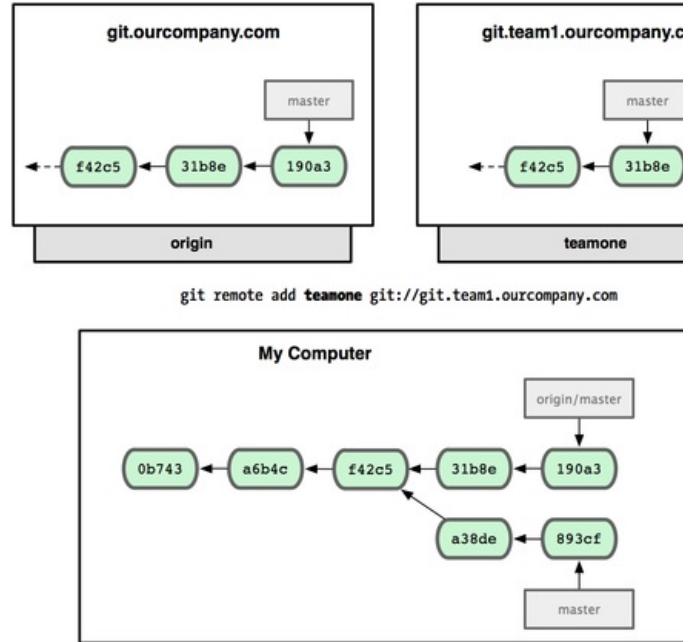
3-23.

```
git fetch origin          origin      git.ourcompany.com
```



3-24. git fetch remote

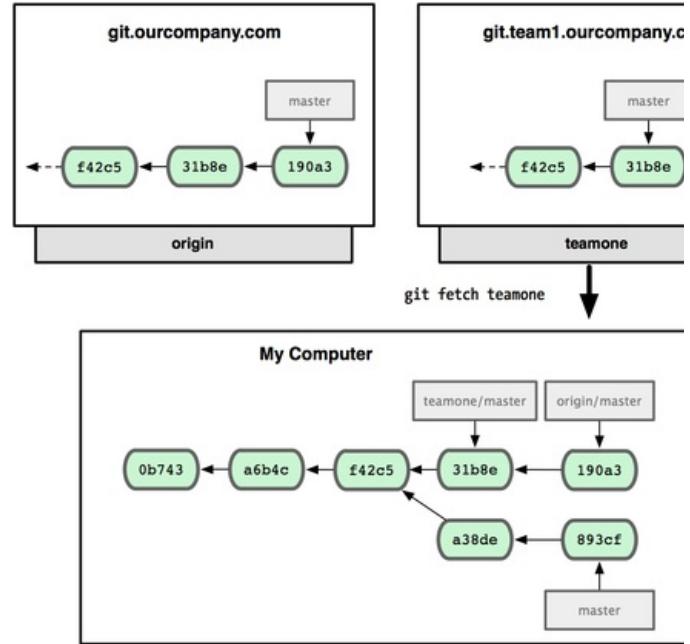
```
git.team1.ourcompany.com  git remote add      teamon
```



3-25.

```
git fetch teamone      origin Git      teamone/master   team
```

3-26



3-26. teamone master

```
serverfix      git push () ()
```

```
$ git push origin serverfix
Counting objects: 20, done.
Compressing objects: 100% (14/14), done.
Writing objects: 100% (15/15), 1.74 KiB, done.
Total 15 (delta 5), reused 0 (delta 0)
To git@github.com:schacon/simplegit.git
 * [new branch]      serverfix -> serverfix
```

```
Git      serverfix    refs/heads/serverfix:refs/heads/serverfix" S
"
          refs/heads/      git push origin serve
serverfix"
```

origin/serverfix

```
$ git fetch origin
```

```
remote: Counting objects: 20, done.  
remote: Compressing objects: 100% (14/14), done.  
remote: Total 15 (delta 5), reused 0 (delta 0)  
Unpacking objects: 100% (15/15), done.  
From git@github.com:schacon/simplegit  
 * [new branch]      serverfix    -> origin/serverfix
```

fetch

serverfix

origin/serverfix

```
git merge origin/serverfix  serverfix
```

```
$ git checkout -b serverfix origin/serverfix  
Branch serverfix set up to track remote branch refs/remotes/origin/serverfix.  
Switched to a new branch "serverfix"
```

serverfix origin/serverfix

(tracking branch)

git push

git pull

Git master origin/master git push git pull
checkout -b [] []/[] 1.6.2 Git --track

```
$ git checkout --track origin/serverfix  
Branch serverfix set up to track remote branch refs/remotes/origin/serverfix.  
Switched to a new branch "serverfix"
```

```
$ git checkout -b sf origin/serverfix  
Branch sf set up to track remote branch refs/remotes/origin/serverfix.  
Switched to a new branch "sf"
```

sf origin/serverfix

master

git push [] :[]

serverfix

```
$ git push origin :serverfix
To git@github.com:schacon/simplegit.git
 - [deleted]           serverfix
```

git push [] []:[] [] “ [] ”

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#).

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).
Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

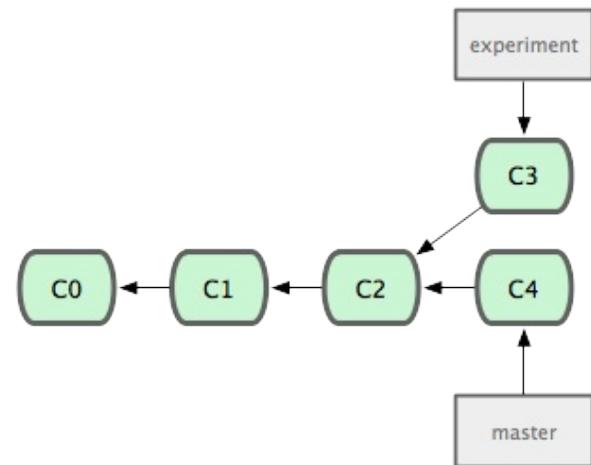
This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

merge

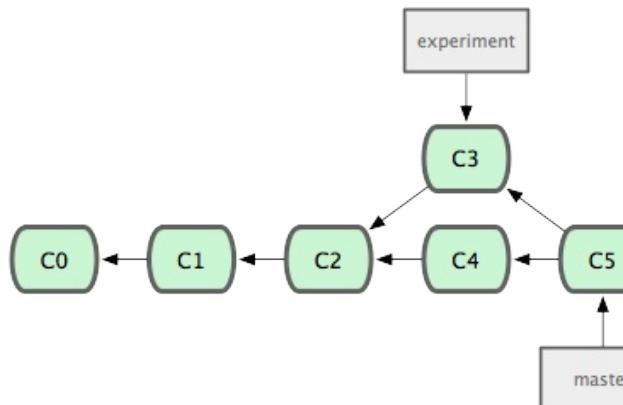
rebase

3-27



3-27.

merge C3 C4C2 3-28



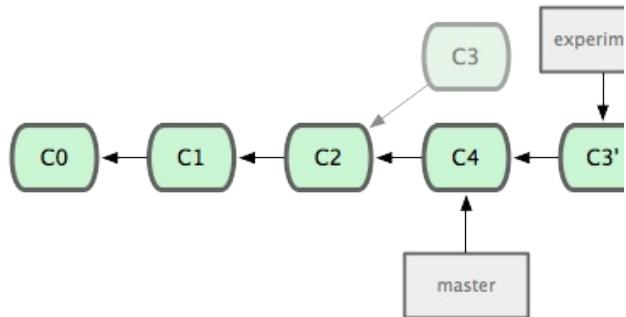
3-28.

C3 C4 Git _rebase_

rebase

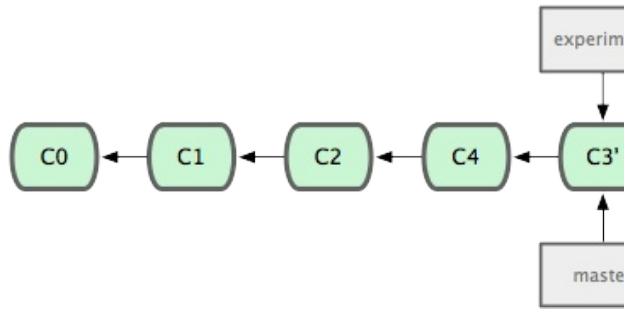
```
$ git checkout experiment
$ git rebase master
First, rewinding head to replay your work on top of it...
  Applying: added staged command
```

diff 3-29



3-29. C3 C4

master 3-30



3-30. master

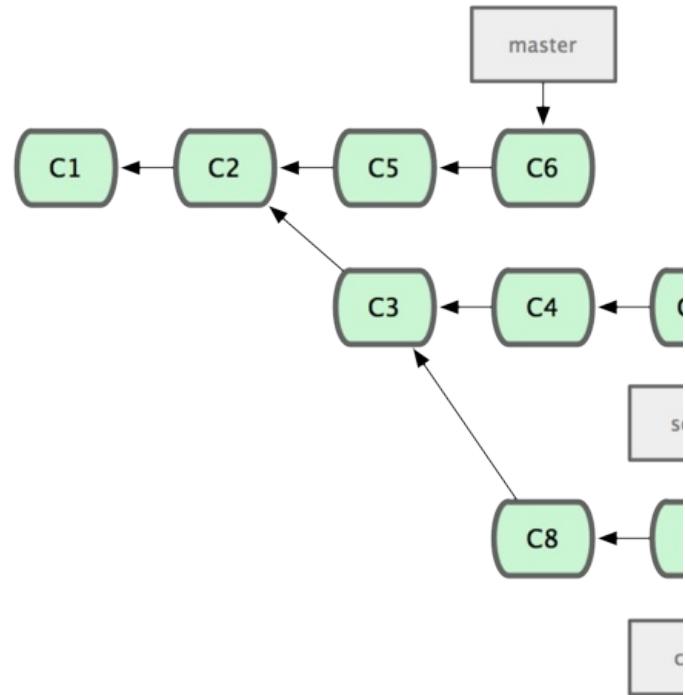
C3 C3' C5

origin/master

3-31

server C3 C4 C3

client C8 C9



3-31.

C8 C9

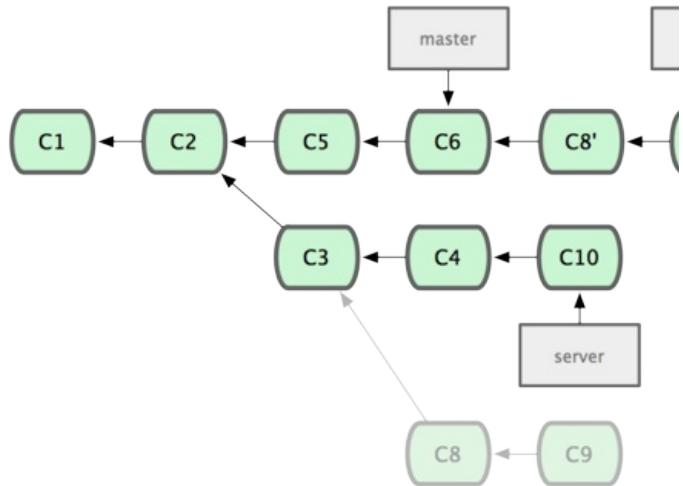
git rebase --onto master

```
$ git rebase --onto master server client
```

“client

client server

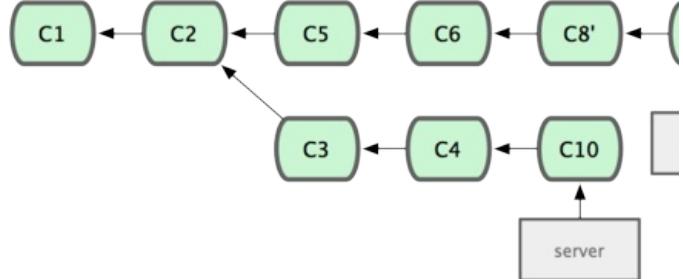
master” 3-32



3-32.

master 3-33

```
$ git checkout master
$ git merge client
```

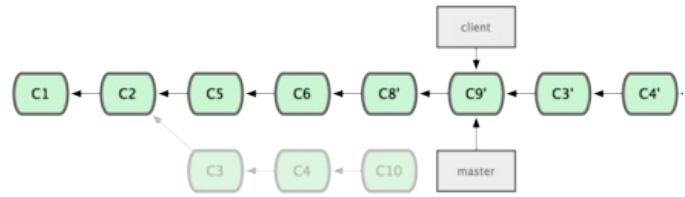


3-33. master client

server server master server git rebase [] []

```
$ git rebase master server
```

server master 3-34



3-34. master server

master

```
$ git checkout master
$ git merge server
```

client server 3-35

```
$ git branch -d client
$ git branch -d server
```



3-35.

commit commitcommit

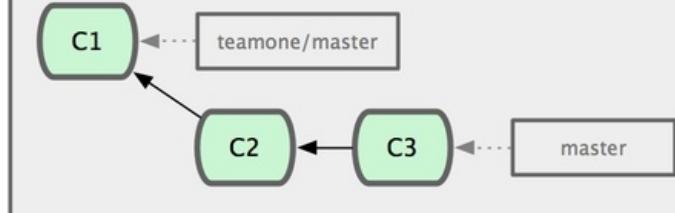
git rebase commit

3-36

git.team1.ourcompany.com



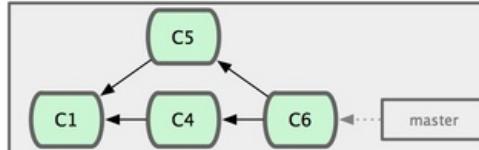
My Computer



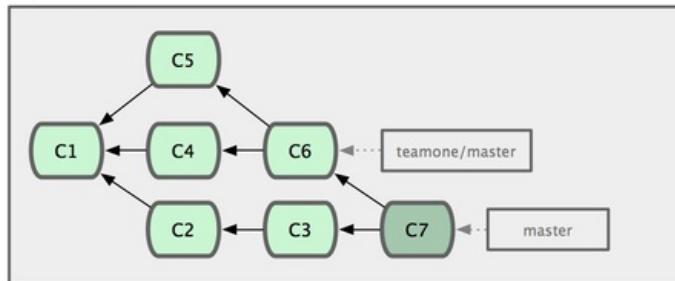
3-36.

C6 3-37

git.team1.ourcompany.com

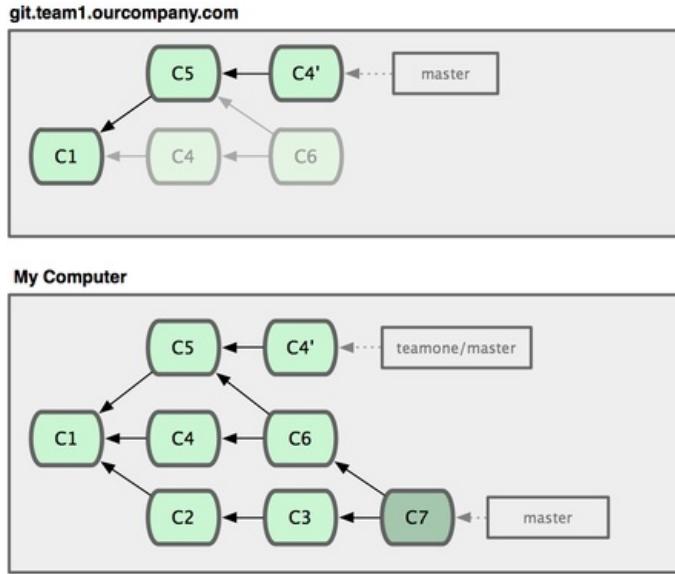


My Computer



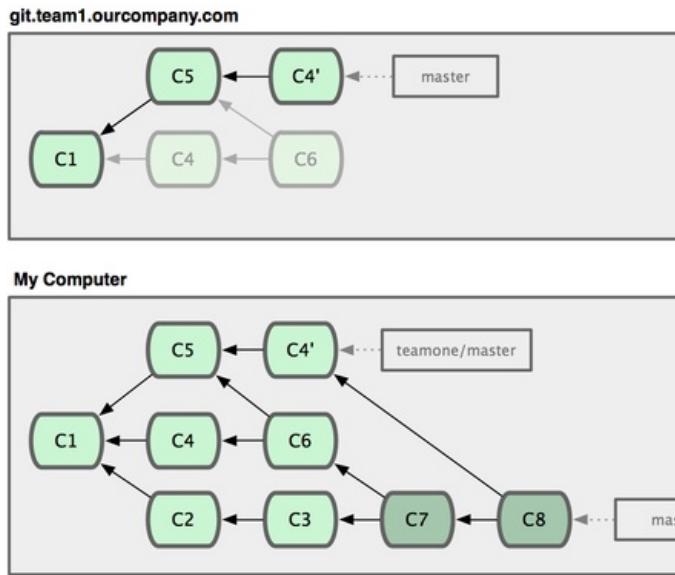
3-37.

C6 git push --force C4'



3-38. C4' C6

commit SHA-1 Git commit C4 3-39:



3-39. C8

C4 C4' SHA-1

commit commit commit

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and

Chapter 4

Git

Git Git —— “Git ” Git

Git

(bare repository) —— Git .git

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).
Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

Git SSH Git HTTP

HTTP Git

(Local protocol) (NFS)

URL

```
$ git clone /opt/git/project.git
```

```
$ git clone file:///opt/git/project.git
```

URL file:// Git Git

file:// Git

file:// —

Git

```
$ git remote add local_proj /opt/git/project.git
```

“ Git ”

```
git pull /home/john/project
```

Git NFS SSH

SSH

Git SSH SSH —— SSH HTTP Git SSHSSH

SSH Git ssh:// URL

```
$ git clone ssh://user@server:project.git
```

—— Git SSH

```
$ git clone user@server:project.git
```

Git

SSH SSHSSH —— SSH SSH —— Git SSH

SSH SSH SSH SSH SSH

Git

Git Git SSH 9418 Git
URL

git-export-daemo

Git Git SSH

Git Git SSH
9418

git:// Git — “Gitosis” —

HTTP/S

HTTP HTTP HTTPS Git HTTP
Git web HTTP

post-update

```
$ cd /var/www/htdocs/  
$ git clone --bare /path/to/git_project gitproject.git  
$ cd gitproject.git  
$ mv hooks/post-update.sample hooks/post-update  
$ chmod a+x hooks/post-update
```

Git post-update git update-server-info HTTP SSH

```
$ git clone http://example.com/gitproject.git
```

Apache /var/www/htdocs web — Git 9

HTTP WebDAV HTTP

<http://www.kernel.org/pub/software/scm/git/docs/howto/setup-c>

HTTP WebDAV Git WebDAV

HTTP HTTP HTTP Apache ——

HTTPS SSL SSH HTTPS SSL HTTP

HTTP HTTP

HTTP HTTP ——— HTTP *(dumb)* HTTP

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

Git

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

Git

Git — clone --bare .git

```
$ git clone --bare my_project my_project.git
Initialized empty Git repository in /opt/projects/my_project.git/
```

```
clone  git init  git fetch  git init          my_proc
```

```
$ cp -Rf my_project/.git my_project.git
```

Git Git

git.example.com SSH Git /opt/git

```
$ scp -r my_project.git user@git.example.com:/opt/git
```

SSH /opt/git

```
$ git clone user@git.example.com:/opt/git/my_project.git
```

SSH /opt/git/my_project.git git init --shared Git

```
$ ssh user@git.example.com  
$ cd /opt/git/my_project.git  
$ git init --bare --shared
```

Git SSH

Git — SSH

Gitosis SSH

Git Git

SSH

SSH

SSH SSH

adduser

git SSH git ~/.ssh/authorized_keys git —

SSH LDAP shell SSH

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).
Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

SSH

Git SSH SSH

~/.ssh

```
$ cd ~/.ssh  
$ ls  
authorized_keys2  id_dsa      known_hosts  
config           id_dsa.pub
```

.pub id_dsa id_rsa .pub .ssh

Linux/Mac SSH Windows MSysGit

```
$ ssh-keygen  
Generating public/private rsa key pair.  
Enter file in which to save the key (/Users/schacon/.ssh/id_rsa):  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /Users/schacon/.ssh/id_rsa.  
Your public key has been saved in /Users/schacon/.ssh/id_rsa.pub.  
The key fingerprint is:  
43:c5:5b:5f:b1:f1:50:43:ad:20:a6:92:6a:1f:9a:3a schacon@agadorlaptop.local
```

.ssh/id_rsa

Git SSH

.put e-email

```
$ cat ~/.ssh/id_rsa.pub  
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEAk10UpkDHrfHY17SbrmTIpNLTGK9Tjom/BWDSU  
GPL+nafzlHDTYW7hdI4yZ5ew18JH4JW9jbhUFrvjQzM7x1ELEVf4h9lFX5QVkbPppSwg0cda3  
Pbv7k0dJ/MTyB1WXFCR+HAo3FXRitBqxiX1nKhXpHAZsMcilq8V6RjsNAQwdsdMFvS1VK/7XA  
t3FaoJoAsncM1Q9x5+3V0Ww68/eIFmb1zuUF1jQJKprrx88XypNDvjYNby6vw/Pb0rwert/En  
mZ+AW40ZPnPPI89ZPmVMLuayrD2cE86Z/il8b+gw3r3+1nKatmIkjn2so1d01QraT1MqVSsbx  
NrRFi9wrf+M7Q== schacon@agadorlaptop.local
```

SSH GitHub SSH

<http://github.com/guides/providing>

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

SSH —

web Git post-update /opt/git Apache web
Apache

```
$ cd project.git
$ mv hooks/post-update.sample hooks/post-update
$ chmod a+x hooks/post-update
```

Git 1.6 mv —— Git .sample

post-update

```
$ cat .git/hooks/post-update
#!/bin/sh
exec git-update-server-info
```

SSH Git HTTP

Apache VirtualHost DocumentRoot Git DNS

```
<VirtualHost *:80>
    ServerName git.gitserver
    DocumentRoot /opt/git
    <Directory /opt/git/>
        Order allow, deny
        allow from all
    </Directory>
</VirtualHost>
```

/opt/git Unix www-data web Apache CGI

```
$ chgrp -R www-data /opt/git
```

Apache URL

```
$ git clone http://git.gitserver/project.git
```

HTTP Git —

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#).

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

GitWeb

Git GitWeb CGI

<http://git.kernel.org> GitWeb

The screenshot shows the GitWeb interface for the 'git' repository at <http://git.kernel.org/pub/scm/git/git.git>. The page title is '/pub/scm/git/git.git / summary'. It includes a navigation bar with links for 'summary', 'shortlog', 'log', 'commit', 'commitdiff', and 'tree'. A search bar and a 'commit' button are also present. The main content area displays the repository's metadata: description ('The core git plumbing'), owner ('Junio C Hamano'), last change ('Fri, 20 Feb 2009 07:44:07 +0000'), and URL ('git://git.kernel.org/pub/scm/git/git.git' and 'http://www.kernel.org/pub/scm/git/git.git'). Below this is the 'shortlog' section, which lists commits from 34 hours ago to 2 days ago. Each commit entry includes the author, date, subject, and a link to the commit details.

Figure 4-1. GitWeb

GitWeb	Git	lighttpd	webrick	Linux	lighttpd	(
Leopard	Ruby		webrick	lighttpd	git	instaweb

```
$ git instaweb --httpd=webrick
[2009-02-21 10:02:21] INFO  WEBrick 1.3.1
[2009-02-21 10:02:21] INFO  ruby 1.8.6 (2008-03-03) [universal-darwin9.0]
```

1234 HTTPD

--stop

```
$ git instaweb --httpd=webrick --stop
```

web	CGI	Linux	apt	yum	gitweb	GitWeb	Git
GitWeb CGI							

```
$ git clone git://git.kernel.org/pub/scm/git/git.git
$ cd git/
$ make GITWEB_PROJECTROOT="/opt/git" \
    prefix=/usr gitweb/gitweb.cgi
$ sudo cp -Rf gitweb /var/www/
```

GITWEB_PROJECTROOT Git Apache CGI VirtualHost

```
<VirtualHost *:80>
  ServerName gitserver
  DocumentRoot /var/www/gitweb
  <Directory /var/www/gitweb>
    Options ExecCGI +FollowSymLinks +SymLinksIfOwnerMatch
    AllowOverride All
    order allow,deny
    Allow from all
    AddHandler cgi-script cgi
    DirectoryIndex gitweb.cgi
  </Directory>
</VirtualHost>
```

GitWeb CGI Apache

<http://gitserver>

HTTP Again, GitWeb can be served with any CGI capable web server. If you have something else, it shouldn't be difficult to set up. At this point, you should be able to type <http://gitserver/> to view your repositories online, and you can use the `git` command to fetch your repositories over HTTP.

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#), [Macedonian](#), [Polish](#), [Thai](#) and [Russian](#).

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

Gitosis

authorized_keys shell——

Gitosis Gitosis

authorized_keys Git Gitosis

Gitosis Linux —— Ubuntu 8.10

Gitosis Python Python setuptools Ubuntu python-setuptools

```
$ apt-get install python-setuptools
```

Gitosis

```
$ git clone git://eagain.net/gitosis.git
$ cd gitosis
$ sudo python setup.py install
```

Gitosis Gitosis

/home/git

/opt/git

```
$ ln -s /opt/git /home/git/repositories
```

Gitosis Gitosis

authorized_keys

authorized_k

```
$ mv /home/git/.ssh/authorized_keys /home/git/.ssh/ak.bak
```

'git' shell

git-shell Gitosis

/etc/passwd

```
git:x:1000:1000::/home/git:/usr/bin/git-shell
```

```
git:x:1000:1000::/home/git:/bin/sh
```

Gitosis gitosis-init

```
$ sudo -H -u git gitosis-init < /tmp/id_dsa.pub  
Initialized empty Git repository in /opt/git/gitosis-admin.git/  
Reinitialized existing Git repository in /opt/git/gitosis-admin.git/
```

Gitosis Git post-update

```
$ sudo chmod 755 /opt/git/gitosis-admin.git/hooks/post-update
```

Gitosis SSH

```
$ ssh git@gitserver  
PTY allocation request failed on channel 0  
fatal: unrecognized command 'gitosis-serve schacon@quaternion'  
Connection to gitserver closed.
```

Gitosis Git —— Gitosis

```
#  
$ git clone git@gitserver:gitosis-admin.git
```

gitosis-admin

```
$ cd gitosis-admin  
$ find .  
.gitosis.conf  
.keydir  
.keydir/scott.pub
```

gitosis.conf keydir —— keydir scott.pub—— Gitosis

gitosis.conf gitosis-admin

```
$ cat gitosis.conf
[gitosis]

[group gitosis-admin]
writable = gitosis-admin
members = scott
```

scott —— Gitosis —— gitosis-admin

mobile 'SCOTT' iphone_project

```
[group mobile]
writable = iphone_project
members = scott
```

gitosis-admin

```
$ git commit -am 'add iphone_project and mobile group'
[master]: created 8962da8: "changed name"
 1 files changed, 4 insertions(+), 0 deletions(-)
$ git push
Counting objects: 5, done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 272 bytes, done.
Total 3 (delta 1), reused 0 (delta 0)
To git@gitserver:/opt/git/gitosis-admin.git
 fb27aec..8962da8 master -> master
```

iphone_project remote —— Gitosis

```
$ git remote add origin git@gitserver:iphone_project.git
$ git push origin master
Initialized empty Git repository in /opt/git/iphone_project.git/
Counting objects: 3, done.
Writing objects: 100% (3/3), 230 bytes, done.
Total 3 (delta 0), reused 0 (delta 0)
To git@gitserver:iphone_project.git
 * [new branch]      master -> master
```

—— Gitosis

```
~/ssh/authorized_keys      keydir      gitosis.conf  Jo
```

```
$ cp /tmp/id_rsa.john.pub keydir/john.pub
$ cp /tmp/id_rsa.josie.pub keydir/josie.pub
$ cp /tmp/id_rsa.jessica.pub keydir/jessica.pub
```

'mobile' iphone_project

```
[group mobile]
writable = iphone_project
members = scott john josie jessica
```

Gitosis John

```
[group mobile]
writable = iphone_project
members = scott josie jessica

[group mobile_ro]
readonly = iphone_project
members = john
```

John Gitosis

```
loglevel=DEBUG  [gitosis]          /home/git/
gitosis.conf    gitosis-admin
```

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#).

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).
Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

Git

HTTP Git Git HTTP

SSH

Git

```
git daemon --reuseaddr --base-path=/opt/git/ /opt/git/
```

--reuseaddr --base-path Git 9418

Ubuntu Upstart

```
/etc/event.d/local-git-daemon
```

```
start on startup
stop on shutdown
exec /usr/bin/git daemon \
    --user=git --group=git \
    --reuseaddr \
    --base-path=/opt/git/ \
    /opt/git/
respawn
```

— git-ro Gitosis ‘git’

Git

```
initctl start local-git-daemon
```

xinetd sysinit ——

Gitosis Git Git iphone

gitosis.conf

```
[repo iphone_project]
daemon = yes
```

9418

Gitosis Git Git

```
$ cd /path/to/project.git
$ touch git-daemon-export-ok
```

git-deamon-export-ok Git

Gitosis GitWeb

/etc/gitweb.conf

```
$projects_list = "/home/git/gitosis/projects.list";
$projectroot = "/home/git/repositories";
$export_ok = "git-daemon-export-ok";
@git_base_url_list = ('git://gitserver');
```

Gitosis gitweb GitWeb iphone GitWeb

rep

```
[repo iphone_project]
daemon = yes
gitweb = yes
```

GitWeb iphone

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).
Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#).

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).
Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

Git

Git —

Git wiki Githosting

<http://git.or.cz/gitwiki/GitHosting>

Scott Chacon GitHub

GitHub Git GitHub GitHub

GitHub

GitHub GitHub GitHub grit github.com/grit git
GitHub shacon

GitHub

Pricing and Signup http://github.com/plans Free acc

4-2 The first thing you need to do is set up a free user account. If you Signup page at <http://github.com/plans> and click the “Sign Up” button (Figure 4-2), you’re taken to the signup page.



Figure 4-2. GitHub

4-3

A screenshot of the GitHub user sign-up form. It has fields for "Username", "Email Address" (which is highlighted in yellow), "Password", and "Confirm Password". Below these is a field for "SSH Public Key (explain ssh keys)" with a note: "Please enter one key only. You may add more later. This field is not required to sign up." At the bottom, there's a note: "You're signing up for the free plan. If you have any questions please email support." followed by "By signing up, you agree to the Terms of Service, Privacy, and Refund policies." A "I agree, sign me up!" button is at the very bottom.

Figure 4-3. The GitHub user signup form

SSH ”” SSH Public Key SSH “explain ssh keys” “I agree
sign me up ” 4-4

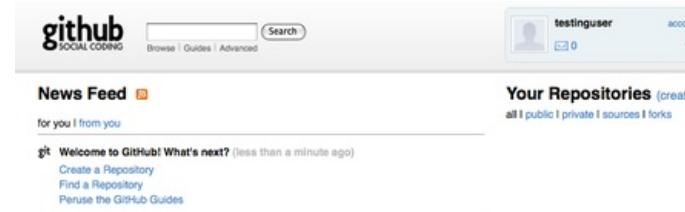


Figure 4-4. GitHub

“create a new one” Create a New Repository 4-5

The screenshot shows the 'Create a New Repository' page on GitHub. The form fields are as follows:

- Project Name:** iphone_project
- Description:** iphone project for our mobile group
- Homepage URL:** (empty field)
- Access Control:** Radio buttons for 'Anyone' (selected) and 'Upgrade your plan to create more private repositories.'
- Create Repository:** A button at the bottom.

Figure 4-5. GitHub

“Create Repository” 4-6

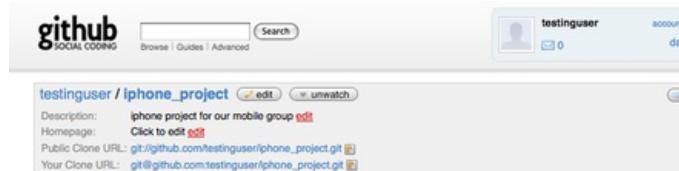


Figure 4-6. GitHub

GitHub Subversion google code sourceforge 4-7

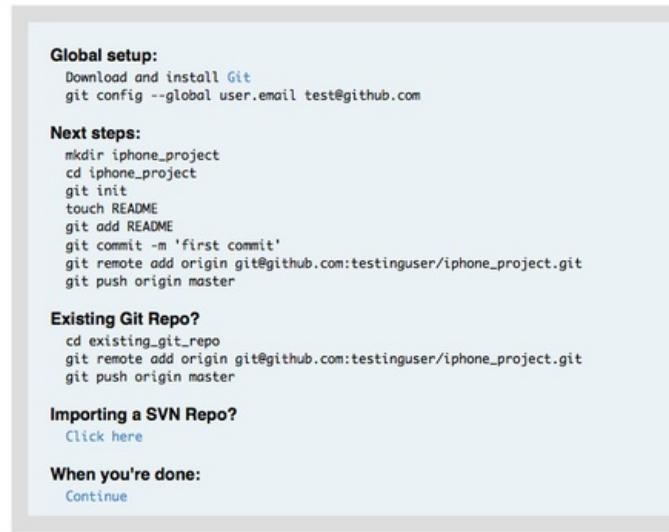


Figure 4-7.

Git Git

```
$ git init  
$ git add .  
$ git commit -m 'initial commit'
```

Git GitHub master

```
$ git remote add origin git@github.com:testinguser/iphone_project.git  
$ git push origin master
```

GitHub URL URL

http://github.com/testinguser/iphone_project

8

The screenshot shows a GitHub repository page for 'testinguser / iphone_project'. The page displays the repository name, a brief description ('iphone project for our mobile group'), a homepage link ('Click to edit'), and two clone URLs: 'Public Clone URL' and 'Your Clone URL', both linking to git://github.com/testinguser/iphone_project.git.

Figure 4-8. URL URL

Public Clone URL URL Git URL URL

Your Clone URL URL SSH URL SSH URL——

Subversion

Subversion GitGitHub Subversion Subversion 4-9

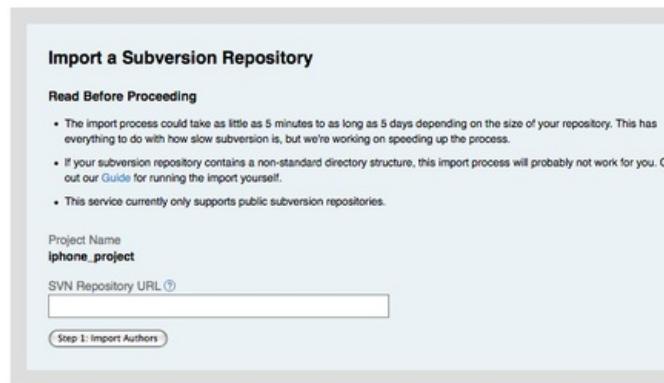


Figure 4-9. Subversion

JohnJosie Jessica GitHub

“edit” Admin 4-10

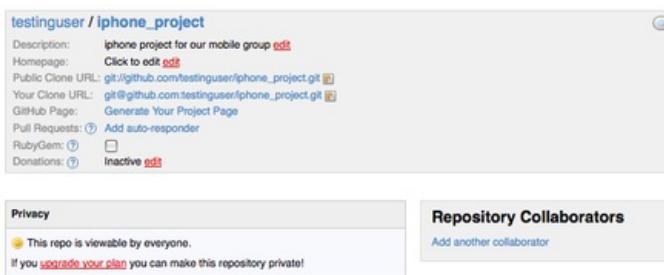


Figure 4-10. GitHub

“Add another collaborator” Add 4-11

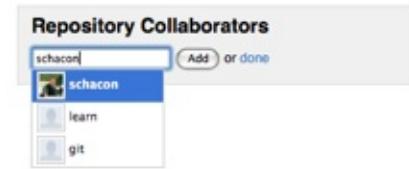


Figure 4-11.

Repository Collaborators 4-12



Figure 4-12.

"revoke "

Subversion 4-13

A screenshot of a Subversion repository page on GitHub. The repository is named 'testinguser/iphone_project'. The commit history shows an 'initial commit' by 'schacon' 5 days ago. The commit message is 'initial commit [schacon]'. The commit hash is '8ce0cc66be796384c1241232cf15' and the tree hash is 'c3a8f5888d80f608c71f0775fc58'. Below the commit details is a table of files and their commit information. The table has columns for 'name', 'age', and 'message'. The files listed are 'Classes/', 'Info.plist', 'MainWindow.xib', 'build/', 'iGit.xcodeproj/', 'iGitViewController.xib', 'iGit_Prefix.pch', and 'main.m'. All files were committed 5 days ago with the message 'initial commit [schacon]'.

Figure 4-13. GitHub

Commits commit git log Network fork Downloads
Wiki wikiGraphs Source README

forking

GitHub forking Hack “fork” GitHub

mojombo/chronic “fork” 4-14

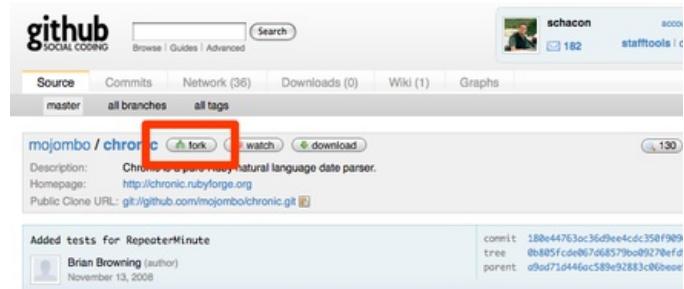


Figure 4-14. “fork”

4-15

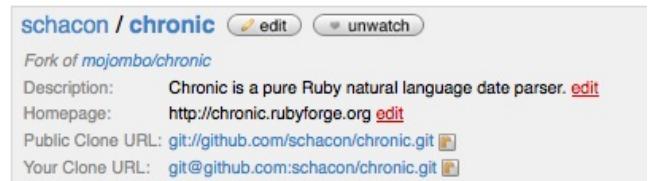


Figure 4-15.

GitHub

GitHub Git

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and

Chapter 5

Git

Git Git

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

Git Git

peer-review

Git Git

Documentation/SubmittingPatches

whitespaceGit

git diff --check

X

```
$ git diff --check
lib/simplegit.rb:5: trailing whitespace.
+ @git_dir = File.expand_path(git_dir)XX
lib/simplegit.rb:7: trailing whitespace.
+ XXXXXXXXXXXX
lib/simplegit.rb:26: trailing whitespace.
+ def command(git_cmd)XXXX
```

git add --patch

50 Git “I added tests for” “Adding tests for”
“Add tests for” tpope.net Tim Pope

```
50  
72  
  
rebase  
  
-  
-
```

```
Git Git          git log --no-merges git      g.  
git://git.kernel.org/pub/scm/git/git.git  
  
-m  git commit
```

Subversion Git John

```
# John's Machine  
$ git clone john@githost:simplegit.git  
Initialized empty Git repository in /home/john/simplegit/.git/  
...  
$ cd simplegit/  
$ vim lib/simplegit.rb  
$ git commit -am 'removed invalid default value'  
[master 738ee87] removed invalid default value  
 1 files changed, 1 insertions(+), 1 deletions(-)
```

Jessica

```
# Jessica's Machine  
$ git clone jessica@githost:simplegit.git  
Initialized empty Git repository in /home/jessica/simplegit/.git/  
...  
$ cd simplegit/  
$ vim TODO  
$ git commit -am 'add reset task'  
[master fbfff5bc] add reset task  
 1 files changed, 1 insertions(+), 0 deletions(-)
```

Jessica

```
# Jessica's Machine
$ git push origin master
...
To jessica@githost:simplegit.git
  1edee6b..fbff5bc  master -> master
```

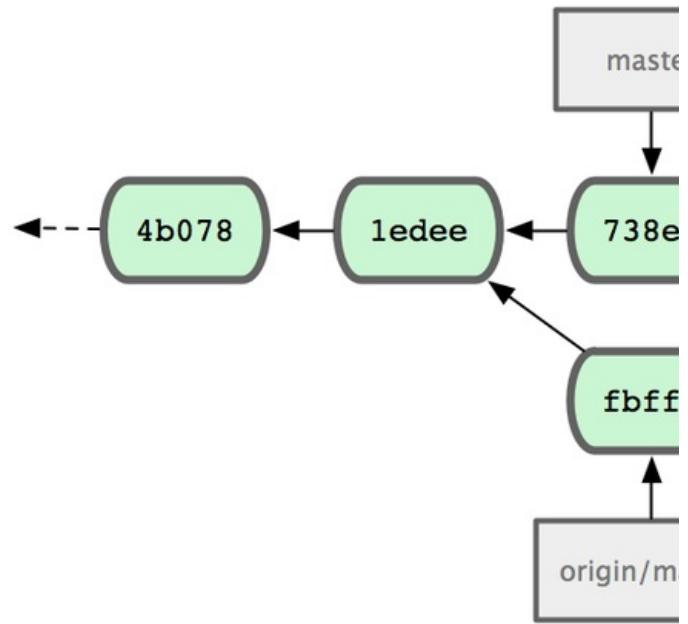
John

```
# John's Machine
$ git push origin master
To john@githost:simplegit.git
 ! [rejected]      master -> master (non-fast forward)
error: failed to push some refs to 'john@githost:simplegit.git'
```

John Jessica Subversion Subversion Git John
Jessica

```
$ git fetch origin
...
From john@githost:simplegit
 + 049d078...fbff5bc master    -> origin/master
```

John 5-4

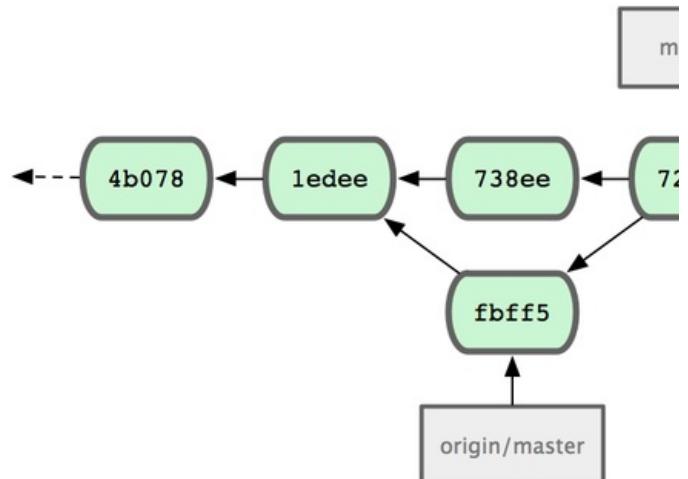


5-4. John

John Jessica fbff5 origin/master master 738ee

```
$ git merge origin/master
Merge made by recursive.
 TODO | 1 +
 1 files changed, 1 insertions(+), 0 deletions(-)
```

John 5-5

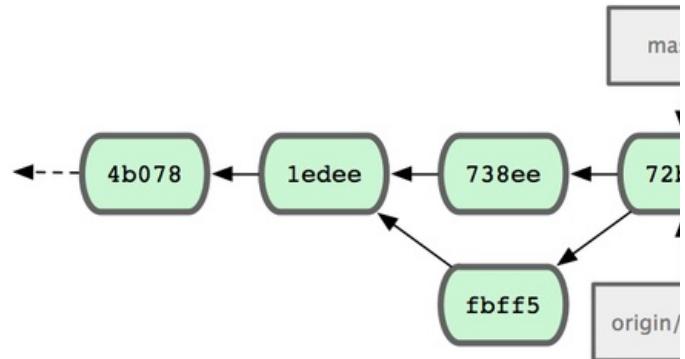


5-5. origin/master John

John 72bbc

```
$ git push origin master
...
To john@githost:simplegit.git
  fbff5bc..72bbc59  master -> master
```

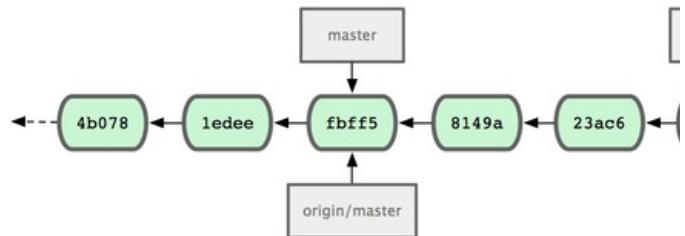
John 5-6



5-6. John

Jessica

issue54 John 5-7

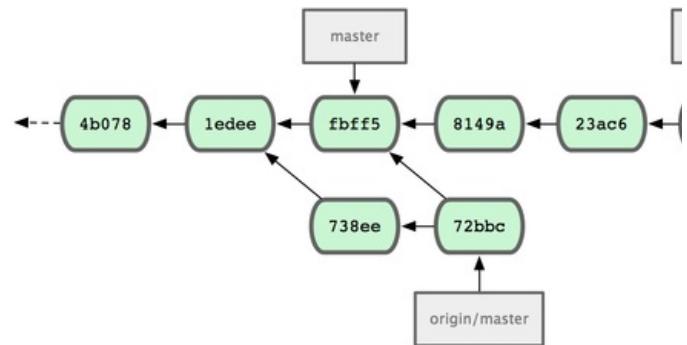


5-7. Jessica

Jessica

```
# Jessica's Machine
$ git fetch origin
...
From jessica@githost:simplegit
  fbff5bc..72bbc59  master      -> origin/master
```

Jessica John 738ee 72bbc 5-8



5-8. John Jessica

Jessica

git log

```
$ git log --no-merges origin/master ^issue54
commit 738ee872852dfa9d6634e0dea7a324040193016
Author: John Smith <jsmith@example.com>
Date:   Fri May 29 16:01:27 2009 -0700

    removed invalid default value
```

Jessica master John origin/master master

```
$ git checkout master
Switched to branch "master"
Your branch is behind 'origin/master' by 2 commits, and can be fast-forwarded.
```

origin/master issue54 upstream upstream Jessica
issue54

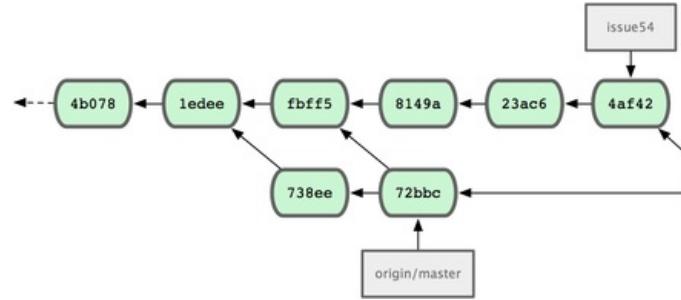
```
$ git merge issue54
Updating fbfff5bc..4af4298
Fast forward
 README      |    1 +
 lib/simplegit.rb |    6 +++++-
 2 files changed, 6 insertions(+), 1 deletions(-)
```

Jessica John

origin/master

```
$ git merge origin/master
Auto-merging lib/simplegit.rb
Merge made by recursive.
 lib/simplegit.rb |    2 ++
 1 files changed, 1 insertions(+), 1 deletions(-)
```

Jessica 5-9

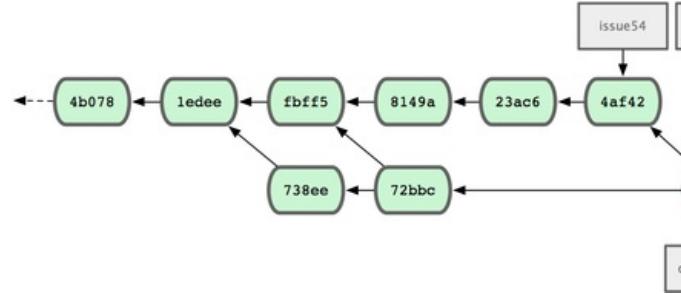


5-9. John Jessica

Jessica master origin/master John

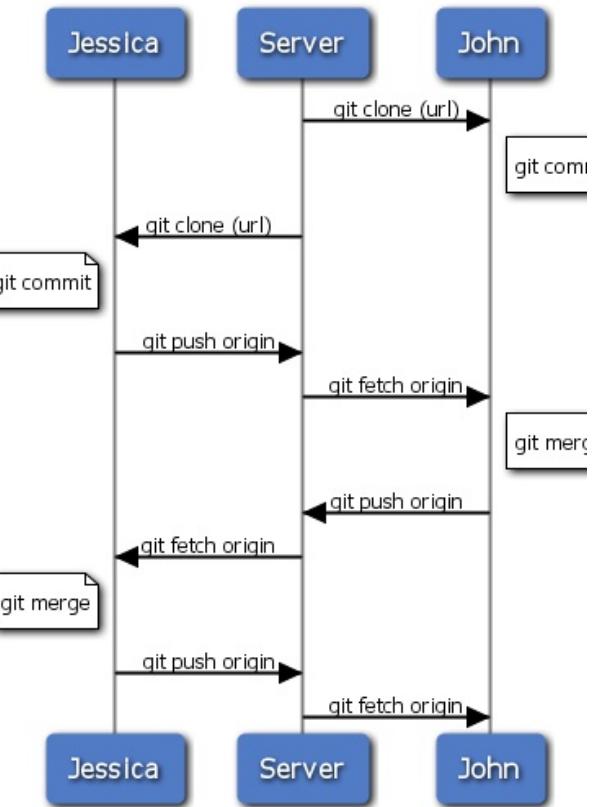
```
$ git push origin master
...
To jessica@githost:simplegit.git
 72bbc59..8059c15  master -> master
```

5-10



5-10. Jessica

master origin/master 5-11



5-11.

John Jessica A Jessica Josie B master

Jessica A featureA

```
# Jessica's Machine
$ git checkout -b featureA
Switched to a new branch "featureA"
$ vim lib/simplegit.rb
$ git commit -am 'add limit to log function'
[featureA 3300904] add limit to log function
 1 files changed, 1 insertions(+), 1 deletions(-)
```

John featureA Jessica master John

```
$ git push origin featureA
...
To jessica@githost:simplegit.git
 * [new branch]      featureA -> featureA
```

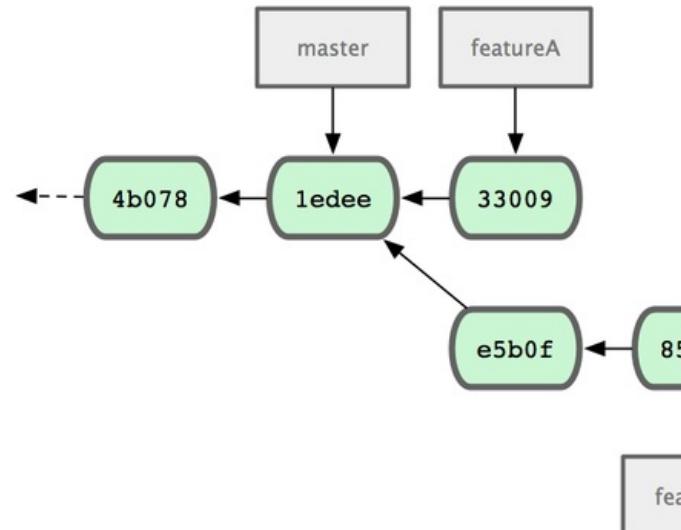
Jessica John featureA Jessica Josie featureB B

```
# Jessica's Machine
$ git fetch origin
$ git checkout -b featureB origin/master
Switched to a new branch "featureB"
```

Jessica featureB

```
$ vim lib/simplegit.rb
$ git commit -am 'made the ls-tree function recursive'
[featureB e5b0fdc] made the ls-tree function recursive
 1 files changed, 1 insertions(+), 1 deletions(-)
$ vim lib/simplegit.rb
$ git commit -am 'add ls-files'
[featureB 8512791] add ls-files
 1 files changed, 5 insertions(+), 0 deletions(-)
```

Jessica 5-12



5-12. Jessica

Jessica Josie

featureBee Jessica Josie

```
$ git fetch origin
...
From jessica@githost:simplegit
 * [new branch]      featureBee -> origin/featureBee
```

Jessica git merge

```
$ git merge origin/featureBee
Auto-merging lib/simplegit.rb
Merge made by recursive.
 lib/simplegit.rb |    4 +---
 1 files changed, 4 insertions(+), 0 deletions(-)
```

featureB featureBee :

```
$ git push origin featureB:featureBee
...
To jessica@githost:simplegit.git
 fba9af8..cd685d1  featureB -> featureBee
```

refspec Git refs

John Jessica

featureA Jessica

git fetch

```
$ git fetch origin
...
From jessica@githost:simplegit
 3300904..aad881d  featureA -> origin/featureA
```

git log

```
$ git log origin/featureA ^featureA
commit aad881d154acdaeb2b6b18ea0e827ed8a6d671e6
Author: John Smith <jsmith@example.com>
Date:   Fri May 29 19:57:33 2009 -0700

changed log output to 30 from 25
```

John

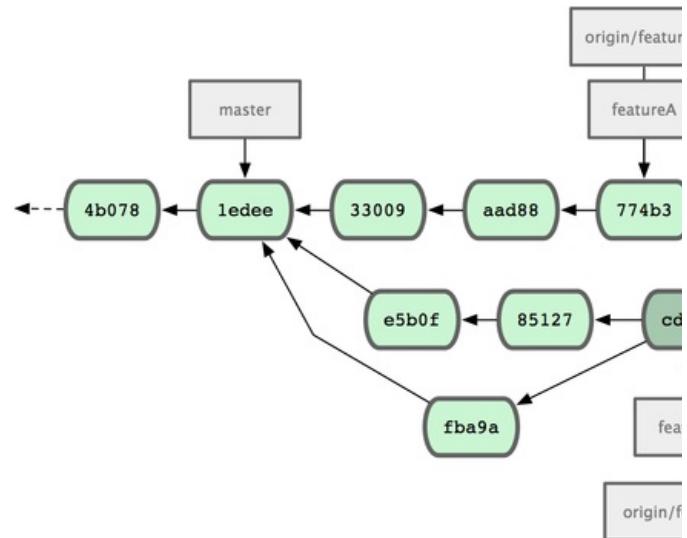
featureA

```
$ git checkout featureA
Switched to branch "featureA"
$ git merge origin/featureA
Updating 3300904..aad881d
Fast forward
 lib/simplegit.rb |    10 ++++++----
 1 files changed, 9 insertions(+), 1 deletions(-)
```

Jessica

```
$ git commit -am 'small tweak'
[featureA ed774b3] small tweak
 1 files changed, 1 insertions(+), 1 deletions(-)
$ git push origin featureA
...
To jessica@githost:simplegit.git
 3300904..ed774b3  featureA -> featureA
```

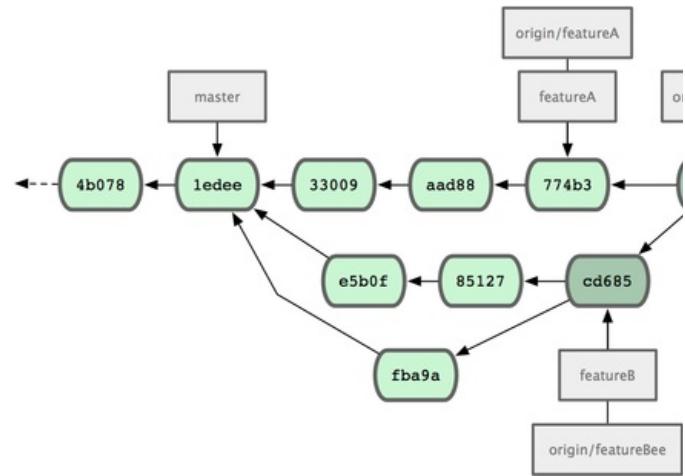
Jessica 5-13



5-13.

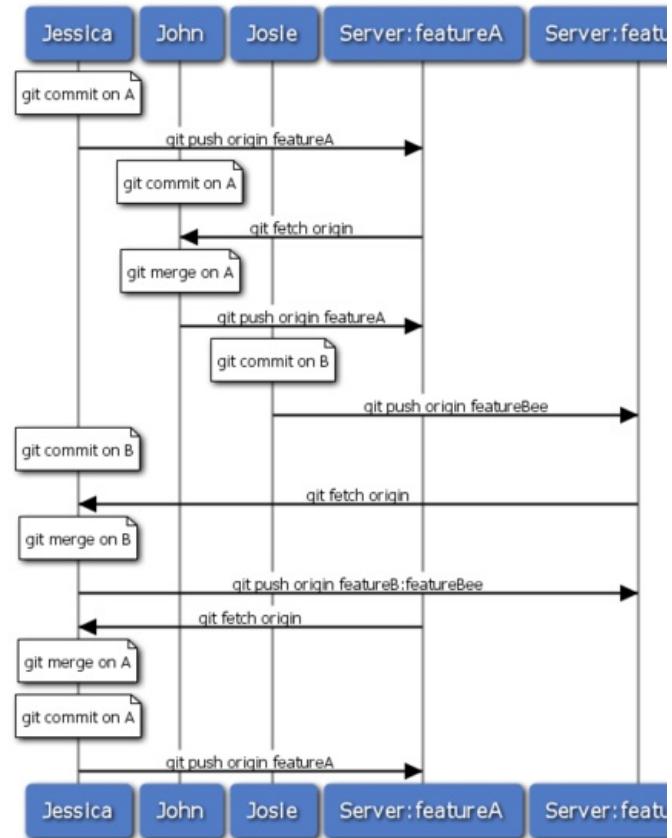
Jessica Josie John

featureA featureBee 5399e fetch 5-14



5-14. Jessica

Git Git 5-15



5-15.

git fork repo.or.cz GitHub

```
$ git clone (url)
$ cd project
$ git checkout -b featureA
$(work)
$ git commit
$(work)
$ git commit
```

rebase -i -

“Fork” url

git://githost/simplegit.gi

```
$ git remote add myfork (url)
```

master cherry pickrewind master cherry-pick
featureA

```
$ git push myfork featureA
```

pull request GitHub “pull request”

git request-pull

request-pull Git URL

myfork Jessica John pull request

```
$ git request-pull origin/master myfork
The following changes since commit 1edee6b1d61823a2de3b09c160d7080b8d1b3a40:
  John Smith (1):
    added a new function

are available in the git repository at:

  git://githost/simplegit.git featureA

  Jessica Smith (2):
    add limit to log function
    change log output to 30 from 25
```

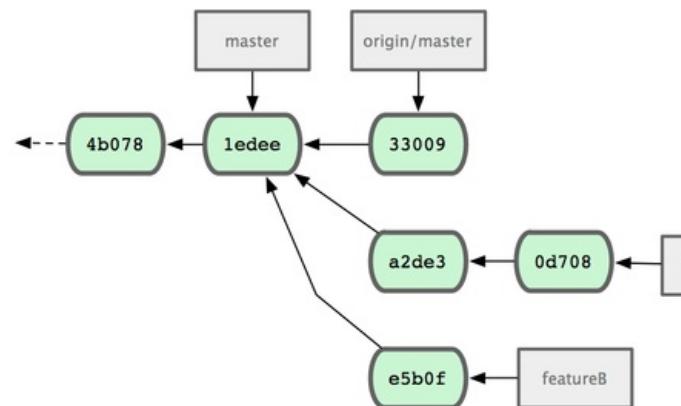
```
lib/simplegit.rb | 10 ++++++--  
1 files changed, 9 insertions(+), 1 deletions(-)
```

master origin/master

master

```
$ git checkout -b featureB origin/master  
$ (work)  
$ git commit  
$ git push myfork featureB  
$ (email maintainer)  
$ git fetch origin
```

AB 5-16

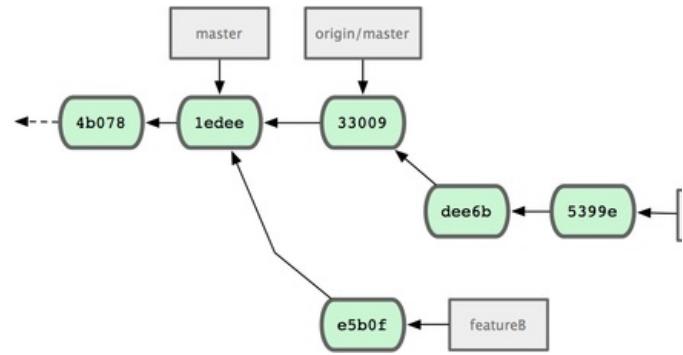


5-16. featureB

origin/master

```
$ git checkout featureA  
$ git rebase origin/master  
$ git push -f myfork featureA
```

5-17



5-17. featureA

-f force

featureA commit

feature

origin/master

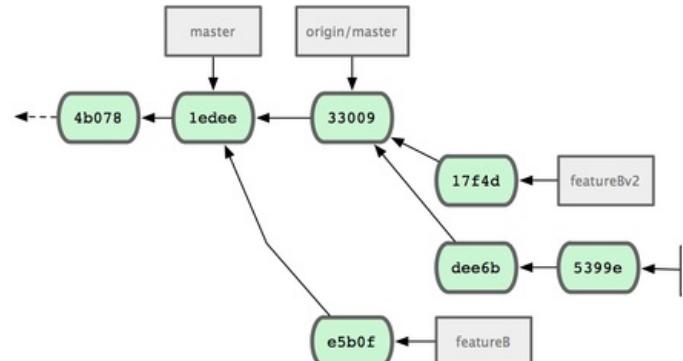
featureBv2

featureB

```
$ git checkout -b featureBv2 origin/master
$ git merge --no-commit --squash featureB
$ (change implementation)
$ git commit
$ git push myfork featureBv2
```

--squash --no-commit Git

featureBv2 5-18



5-18. featureBy2

```
$ git checkout -b topicA  
$ (work)  
$ git commit  
$ (work)  
$ git commit
```

git format-patch mbox .patch mbox Git

```
$ git format-patch -M origin/master  
0001-add-limit-to-log-function.patch  
0002-changed-log-output-to-30-from-25.patch
```

format-patch -M Git

```
$ cat 0001-add-limit-to-log-function.patch  
From 330090432754092d704da8e76ca5c05c198e71a8 Mon Sep 17 00:00:00 2001  
From: Jessica Smith <jessica@example.com>  
Date: Sun, 6 Apr 2008 10:17:23 -0700  
Subject: [PATCH 1/2] add limit to log function  
  
Limit log functionality to the first 20  
  
---  
lib/simplegit.rb |    2 +-  
1 files changed, 1 insertions(+), 1 deletions(-)  
  
diff --git a/lib/simplegit.rb b/lib/simplegit.rb  
index 76f47bc..f9815f1 100644  
--- a/lib/simplegit.rb  
+++ b/lib/simplegit.rb  
@@ -14,7 +14,7 @@ class SimpleGit  
end  
  
def log(treeish = 'master')  
-   command("git log #{treeish}")  
+   command("git log -n 20 #{treeish}")  
end  
  
def ls_tree(treeish = 'master')  
--  
1.6.2.rc1.20.g8c5b.dirty
```

--- Limit log functionality to the first 20

Git IMAP Gmail IMAP Git

Documentation/SubmittingPatches

```
~/.gitconfig imap          git config
```

```
[imap]
  folder = "[Gmail]/Drafts"
  host = imaps://imap.gmail.com
  user = user@gmail.com
  pass = p4ssw0rd
  port = 993
  sslverify = false
```

```
IMAP SSL host          imap:// s  imaps://      git send-email
Gmail  [Gmail]/Drafts Drafts
```

```
$ git send-email *.patch
0001-added-limit-to-log-function.patch
0002-changed-log-output-to-30-from-25.patch
Who should the emails appear to be from? [Jessica Smith <jessica@example.com>]
Emails will be sent from: Jessica Smith <jessica@example.com>
Who should the emails be sent to? jessica@example.com
Message-ID to be used as In-Reply-To for the first email? y
```

Git

```
(mbox) Adding cc: Jessica Smith <jessica@example.com> from
 \line 'From: Jessica Smith <jessica@example.com>'
OK. Log says:
Sendmail: /usr/sbin/sendmail -i jessica@example.com
From: Jessica Smith <jessica@example.com>
To: jessica@example.com
Subject: [PATCH 1/2] added limit to log function
Date: Sat, 30 May 2009 13:29:15 -0700
Message-Id: <1243715356-61726-1-git-send-email-jessica@example.com>
X-Mailer: git-send-email 1.6.2.rc1.20.g8c5b.dirty
In-Reply-To: <y>
References: <y>

Result: OK
```

Gmail Drafts Cc

Git Git

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and

GitGitGit

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

Chapter 6

Git

Git

Git

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

Revision

Git

SHA-1

SHA

Git SHA-1 —— SHA-1

git log

```
$ git log
commit 734713bc047d87bf7eac9674765ae793478c50d3
Author: Scott Chacon <schacon@gmail.com>
Date:   Fri Jan 2 18:32:33 2009 -0800

    fixed refs handling, added gc auto, updated tests

commit d921970aadf03b3cf0e71becdaab3147ba71cdef
Merge: 1c002dd... 35cfb2b...
Author: Scott Chacon <schacon@gmail.com>
Date:   Thu Dec 11 15:08:43 2008 -0800

    Merge commit 'phedders/rdocs'

commit 1c002dd4b536e7479fe34593e72e6c6c1819e53b
Author: Scott Chacon <schacon@gmail.com>
Date:   Thu Dec 11 14:58:32 2008 -0800

    added some blame and merge stuff
```

1c002dd.... git show

```
$ git show 1c002dd4b536e7479fe34593e72e6c6c1819e53b
$ git show 1c002dd4b536e7479f
$ git show 1c002d
```

```
Git SHA-1      --abbrev-commit  git log SHA-1
```

```
$ git log --abbrev-commit --pretty=oneline
ca82a6d changed the verison number
085bb3b removed unnecessary test code
a11bef0 first commit
```

SHA-1 Git Linux 40 12

SHA-1

SHA-1

SHA-1 Git Git

SHA-1 20 160 50% 2^{80} $p = n(r)$
 $1.2 \times 10^{24} 1200$

SHA-1 65 Linux Git Git 5 50% SHA-1

SHA-1 Git topic1 ca82a6d

```
$ git show ca82a6dff817ec66f44342007202690a93763949  
$ git show topic1
```

SHA SHA-1 rev-parse Git 9 rev-parse Git

```
$ git rev-parse topic1  
ca82a6dff817ec66f44342007202690a93763949
```

Git — HEAD

```
git reflog
```

```
$ git reflog  
734713b... HEAD@{0}: commit: fixed refs handling, added gc auto, updated  
d921970... HEAD@{1}: merge phedders/rdocs: Merge made by recursive.  
1c002dd... HEAD@{2}: commit: added some blame and merge stuff  
1c36188... HEAD@{3}: rebase -i (squash): updating HEAD  
95df984... HEAD@{4}: commit: # This is a combination of two commits.  
1c36188... HEAD@{5}: rebase -i (squash): updating HEAD  
7e05da5... HEAD@{6}: rebase -i (pick): updating HEAD
```

Git HEAD

@{n}

```
$ git show HEAD@{5}
```

master

```
$ git show master@{yesterday}
```

git log git log -g

```
$ git log -g master  
commit 734713bc047d87bf7eac9674765ae793478c50d3  
Reflog: master@{0} (Scott Chacon <schacon@gmail.com>)  
Reflog message: commit: fixed refs handling, added gc auto, updated  
Author: Scott Chacon <schacon@gmail.com>  
Date: Fri Jan 2 18:32:33 2009 -0800  
  
        fixed refs handling, added gc auto, updated tests  
  
commit d921970aadf03b3cf0e71becdaab3147ba71cdef  
Reflog: master@{1} (Scott Chacon <schacon@gmail.com>)  
Reflog message: merge phedders/rdocs: Merge made by recursive.  
Author: Scott Chacon <schacon@gmail.com>  
Date: Thu Dec 11 15:08:43 2008 -0800  
  
        Merge commit 'phedders/rdocs'
```

— git show HEAD@{2.months.ago} —

^Git

```
$ git log --pretty=format:'%h %s' --graph
* 734713b fixed refs handling, added gc auto, updated tests
*   d921970 Merge commit 'phedders/rdocs'
|\ 
| * 35cfb2b Some rdoc changes
| | 1c002dd added some blame and merge stuff
| / 
* 1c36188 ignore *.gem
* 9b29157 add open3_detach to gemspec file list
```

HEAD^“HEAD”

```
$ git show HEAD^
commit d921970aadf03b3cf0e71becdaab3147ba71cdef
Merge: 1c002dd... 35cfb2b...
Author: Scott Chacon <schacon@gmail.com>
Date: Thu Dec 11 15:08:43 2008 -0800

Merge commit 'phedders/rdocs'
```

^ —— d921970^2 “d921970”

```
$ git show d921970^
commit 1c002dd4b536e7479fe34593e72e6c6c1819e53b
Author: Scott Chacon <schacon@gmail.com>
Date: Thu Dec 11 14:58:32 2008 -0800

    added some blame and merge stuff

$ git show d921970^2
commit 35cfb2b795a55793d7cc56a6cc2060b4bb732548
Author: Paul Hedderly <paul+git@mjr.org>
Date: Wed Dec 10 22:22:03 2008 +0000

    Some rdoc changes
```

~ HEAD~ HEAD^ HEAD~2 “”—— HE

```
$ git show HEAD~3
commit 1c3618887afb5fbcbbea25b7c013f4e2114448b8d
Author: Tom Preston-Werner <tom@mojombo.com>
Date: Fri Nov 7 13:47:59 2008 -0500
```

```
ignore *.gem
```

HEAD^^^

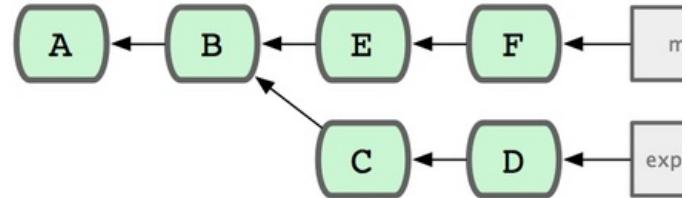
```
$ git show HEAD^^^
commit 1c3618887afb5fbcbea25b7c013f4e2114448b8d
Author: Tom Preston-Werner <tom@mojombo.com>
Date:   Fri Nov 7 13:47:59 2008 -0500

ignore *.gem
```

—— HEAD~3^2

—— “”

Git 6-1



6-1.

master..experiment Git —“experimentmaster”

```
$ git log master..experiemnt
D
C
```

—— master experiment —— experiment..master master

```
$ git log experiment..master  
F  
E
```

experiment

```
$ git log origin/master..HEAD
```

origin git push origin/mastergit log origin/master..HEAD
git log origin/master.. — Git HEAD

Git ^--not

```
$ git log refA..refB  
$ git log ^refA refB  
$ git log refB --not refA
```

refArefBrefC

```
$ git log refA refB ^refC  
$ git log refA refB --not refC
```

6-1 masterexperiment

```
$ git log master...experiment  
F  
E  
D  
C
```

log

`log--left-right`

```
$ git log --left-right master...experiment
< F
< E
> D
> C
```

Git

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and

Git Git stash ——

```
$ git commit --amend
```

```
git addgit rmgit commit --amend
```

SHA-1rebase——

Git rebase HEAD rebase

git rebase -i rebase

```
git rebase -i          HEAD~2HEAD~3~3
```

```
$ git rebase -i HEAD~3
```

——HEAD~3..HEAD——

```
pick f7f3f6d changed my name a bit
pick 310154e updated README formatting and added blame
pick a5f4a0d added cat-file

# Rebase 710f0f8..a5f4a0d onto 710f0f8
#
# Commands:
#   p, pick = use commit
#   e, edit = use commit, but stop for amending
#   s, squash = use commit, but meld into previous commit
#
# If you remove a line here THAT COMMIT WILL BE LOST.
# However, if you remove everything, the rebase will be aborted.
#
```

log log

```
$ git log --pretty=format:"%h %s" HEAD~3..HEAD
a5f4a0d added cat-file
310154e updated README formatting and added blame
f7f3f6d changed my name a bit
```

rebase(HEAD~3)

pickedit

```
edit f7f3f6d changed my name a bit
pick 310154e updated README formatting and added blame
pick a5f4a0d added cat-file
```

Git

```
$ git rebase -i HEAD~3
Stopped at 7482e0d... updated the gemspec to hopefully work better
You can amend the commit now, with

    git commit --amend

Once you're satisfied with your changes, run

    git rebase --continue
```

```
$ git commit --amend
```

```
$ git rebase --continue
```

pick edit Git

"added cat-file"rebase

```
pick f7f3f6d changed my name a bit
pick 310154e updated README formatting and added blame
pick a5f4a0d added cat-file
```

```
pick 310154e updated README formatting and added blame
pick f7f3f6d changed my name a bit
```

Git 310154ef7f3f6d"added cat-file"

(Squashing)

rebase

```
#
# Commands:
# p, pick = use commit
# e, edit = use commit, but stop for amending
# s, squash = use commit, but meld into previous commit
#
# If you remove a line here THAT COMMIT WILL BE LOST.
# However, if you remove everything, the rebase will be aborted.
#
```

"pick""edit""squash"Git

```
pick f7f3f6d changed my name a bit
squash 310154e updated README formatting and added blame
squash a5f4a0d added cat-file
```

Git

```
# This is a combination of 3 commits.
# The first commit's message is:
changed my name a bit

# This is the 2nd commit message:

updated README formatting and added blame

# This is the 3rd commit message:

added cat-file
```

"updated README formatting and added blame""updated README f
"added blame" rebase -i"edit"

```
pick f7f3f6d changed my name a bit
edit 310154e updated README formatting and added blame
pick a5f4a0d added cat-file
```

Git f7f3f6d 310154e git reset HEAD^
continue

```
$ git reset HEAD^
$ git add README
$ git commit -m 'updated README formatting'
$ git add lib/simplegit.rb
$ git commit -m 'added blame'
$ git rebase --continue
```

Gita5f4a0d

```
$ git log -4 --pretty=format:"%h %s"  
1c002dd added cat-file  
9b29157 added blame  
35cfb2b updated README formatting  
f3cc40e changed my name a bit
```

SHA

: filter-branch

— filter-branch

```
git add .           filter-branch password.txt      filter-b
```

```
$ git filter-branch --tree-filter 'rm -f passwords.txt' HEAD  
Rewrite 6b9b3cf04e7c5686a9cb838c3f36a8cb6a0fc2bd (21/21)  
Ref 'refs/heads/master' was rewritten
```

--tree-filter password.txt git filter-branch --t

Git hard-reset

filter-branch --all

trunk, tags

trunk filter-branch

```
$ git filter-branch --subdirectory-filter trunk HEAD  
Rewrite 856f0bf61e41a27326cdae8f09fe708d679f596f (12/12)  
Ref 'refs/heads/master' was rewritten
```

trunkGit

```
git config           filter-branch          --commit-filter
```

```
$ git filter-branch --commit-filter '  
    if [ "$GIT_AUTHOR_EMAIL" = "schacon@localhost" ];  
    then  
        GIT_AUTHOR_NAME="Scott Chacon";  
        GIT_AUTHOR_EMAIL="schacon@example.com";  
        git commit-tree "$@";  
    else  
        git commit-tree "$@";  
    fi' HEAD
```

SHA-1

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#).

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and

Git

Git Git

git blame -L1222

```
$ git blame -L 12,22 simplegit.rb
^4832fe2 (Scott Chacon 2008-03-15 10:31:28 -0700 12) def show(tree = 'master')
^4832fe2 (Scott Chacon 2008-03-15 10:31:28 -0700 13)   command("git show #{tree}")
^4832fe2 (Scott Chacon 2008-03-15 10:31:28 -0700 14) end
^4832fe2 (Scott Chacon 2008-03-15 10:31:28 -0700 15)
9f6560e4 (Scott Chacon 2008-03-17 21:52:20 -0700 16) def log(tree = 'master')
79eaf55d (Scott Chacon 2008-04-06 10:15:08 -0700 17)   command("git log #{tree}")
9f6560e4 (Scott Chacon 2008-03-17 21:52:20 -0700 18) end
9f6560e4 (Scott Chacon 2008-03-17 21:52:20 -0700 19)
42cf2861 (Magnus Chacon 2008-04-13 10:45:01 -0700 20) def blame(path)
42cf2861 (Magnus Chacon 2008-04-13 10:45:01 -0700 21)   command("git blame #{path}")
42cf2861 (Magnus Chacon 2008-04-13 10:45:01 -0700 22) end
```

SHA-1 —————

^4832fe2Git

^SHA

Git
blame

git blame -C

GITServerHandler.m GITPackUpload.m

```
$ git blame -C -L 141,153 GITPackUpload.m
f344f58d GITServerHandler.m (Scott 2009-01-04 141)
f344f58d GITServerHandler.m (Scott 2009-01-04 142) - (void) gatherObjectShasFromC
f344f58d GITServerHandler.m (Scott 2009-01-04 143) {
70befddd GITServerHandler.m (Scott 2009-03-22 144)           //NSLog(@"%@",@"GATHER COMM
ad11ac80 GITPackUpload.m (Scott 2009-03-24 145)
ad11ac80 GITPackUpload.m (Scott 2009-03-24 146)           NSString *parentSha;
ad11ac80 GITPackUpload.m (Scott 2009-03-24 147)           GITCommit *commit = [g
ad11ac80 GITPackUpload.m (Scott 2009-03-24 148)
ad11ac80 GITPackUpload.m (Scott 2009-03-24 149)           //NSLog(@"%@",@"GATHER COMM
ad11ac80 GITPackUpload.m (Scott 2009-03-24 150)
56ef2caf GITServerHandler.m (Scott 2009-01-05 151)           if(commit) {
56ef2caf GITServerHandler.m (Scott 2009-01-05 152)               [refDict setOb
56ef2caf GITServerHandler.m (Scott 2009-01-05 153)
```

Git

```
bisect    bisect  
  
bisect          git bisect start  git bisect badbisect  
[good_commit]
```

```
$ git bisect start  
$ git bisect bad  
$ git bisect good v1.0  
Bisection: 6 revisions left to test after this  
[ecb6e1bc347ccecc5f9350d878ce677feb13d3b2] error handling on repo
```

```
Git (v1.0)12          git bisect good Git
```

```
$ git bisect good  
Bisection: 3 revisions left to test after this  
[b047b02ea83310a70fd603dc8cd7a6cd13d15c04] secure this thing
```

```
git bisect badGit
```

```
$ git bisect bad  
Bisection: 1 revisions left to test after this  
[f71ce38690acf49c1f3c9bea38e09d82a5ce6014] drop exceptions table
```

```
Git SHA-1
```

```
$ git bisect good  
b047b02ea83310a70fd603dc8cd7a6cd13d15c04 is first bad commit  
commit b047b02ea83310a70fd603dc8cd7a6cd13d15c04  
Author: PJ Hyett <pjhyett@example.com>  
Date:   Tue Jan 27 14:48:32 2009 -0800  
  
        secure this thing  
  
:040000 040000 40ee3e7821b895e52c1695092db9bdc4c61d1730  
f24d3c6ebcf639b1a3814550e62d60b8e68a8e4 M config
```

```
git bisect resetHEAD
```

```
$ git bisect reset
```

00 git bisect bisect start

```
$ git bisect start HEAD v1.0  
$ git bisect run test-error.sh
```

test-error.shGit makemake tests

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).
Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and

Git Git__Git_____

__ rack

Git____

Rack Rack

```
$ git remote add rack_remote git@github.com:schacon/rack.git
$ git fetch rack_remote
warning: no common commits
remote: Counting objects: 3184, done.
remote: Compressing objects: 100% (1465/1465), done.
remote: Total 3184 (delta 1952), reused 2770 (delta 1675)
Receiving objects: 100% (3184/3184), 677.42 KiB | 4 KiB/s, done.
Resolving deltas: 100% (1952/1952), done.
From git@github.com:schacon/rack
 * [new branch]      build      -> rack_remote/build
 * [new branch]      master     -> rack_remote/master
 * [new branch]      rack-0.4   -> rack_remote/rack-0.4
 * [new branch]      rack-0.9   -> rack_remote/rack-0.9
$ git checkout -b rack_branch rack_remote/master
Branch rack_branch set up to track remote branch refs/remotes/rack_remote/master.
Switched to a new branch "rack_branch"
```

rack_branch Rack master

```
$ ls
AUTHORS      KNOWN-ISSUES    Rakefile      contrib      lib
COPYING      README         bin           example      test
$ git checkout master
Switched to branch "master"
$ ls
README
```

Rack master Git git read-tree9 read-tree master

```
$ git read-tree --prefix=rack/ -u rack_branch
```

Rack——tarball Rack

```
$ git checkout rack_branch  
$ git pull
```

master git merge -s subtree Git -s subtree

```
$ git checkout master  
$ git merge --squash -s subtree --no-commit rack_branch  
Squash commit -- not updating HEAD  
Automatic merge went well; stopped before committing as requested
```

Rack —— rackrack_branch

rackrack_branch——diff git diff-tree

```
$ git diff-tree -p rack_branch
```

rackmaster

```
$ git diff-tree -p rack_remote/master
```

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#).

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).
Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

Git

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

Chapter 7

Git

Git Git Git

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).
Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

Git

git config Git

```
$ git config --global user.name "John Doe"  
$ git config --global user.email johndoe@example.com
```

Git

Git Git /etc/gitconfig gitconfig --systemgit

Git ~/.gitconfig --global Git

Git Git .git/config .git/config/etc/gitconfig
config

Git Git

```
$ git config --help
```

git configman

core.editor

GiteditorVi core.editor

```
$ git config --global core.editor emacs
```

editorGit Emacs

commit.template

Git \$HOME/.gitmessage.txt

```
subject line  
what happened  
[ticket: X]
```

commit.templategit commit Git commit.template

```
$ git config --global commit.template $HOME/.gitmessage.txt  
$ git commit
```

```
subject line  
what happened  
[ticket: X]  
# Please enter the commit message for your changes. Lines starting  
# with '#' will be ignored, and an empty message aborts the commit.  
# On branch master  
# Changes to be committed:  
#   (use "git reset HEAD <file>..." to unstage)  
#  
# modified:   lib/test.rb  
#  
~  
~  
.git/COMMIT_EDITMSG" 14L, 297C
```

Git

core.pager

core.pager Git logdiff more less

```
$ git config --global core.pager ''
```

user.signingkey

GPGID

```
$ git config --global user.signingkey <gpg-key-id>
```

git tag

```
$ git tag -s <tag-name>
```

core.excludesfile

.gitignore Git

git add

core.excludesfile Git

help.autocorrect

Git 1.6.1 Git 1.6

```
$ git com
git: 'com' is not a git-command. See 'git --help'.
Did you mean this?
    commit
```

help.autocorrect 1 Git

Git

Git

color.ui

Git color.ui true

```
$ git config --global color.ui true
```

Git false always false always Git Git 1.5.5

color.ui = always --color Git color.ui = true

color.*

Git true false always

```
color.branch  
color.diff  
color.interactive  
color.status
```

diff

```
$ git config --global color.diff.meta "blue black bold"
```

normal black red green yellow blue magenta cyan white bold dim underline blink reverse

git config

Git diff, P4Merge Perforce

P4Merge Mac Linux Windows

/usr/local/bin

P4Merge

```
http://www.perforce.com/perforce/downloads/component.html
```

Mac

p4merge merge

extMerge p4merge

```
$ cat /usr/local/bin/extMerge
#!/bin/sh
/Applications/p4merge.app/Contents/MacOS/p4merge $*
```

diff72merge Git diff

```
path old-file old-hex old-mode new-file new-hex new-mode
```

old-file new-file diff

```
$ cat /usr/local/bin/extDiff
#!/bin/sh
[ $# -eq 7 ] && /usr/local/bin/extMerge "$2" "$5"
```

```
$ sudo chmod +x /usr/local/bin/extMerge
$ sudo chmod +x /usr/local/bin/extDiff
```

merge.tool Git mergetool.*.cmd mergetool.trustExitCode
Git 4

```
$ git config --global merge.tool extMerge
$ git config --global mergetool.extMerge.cmd \
    'extMerge "$BASE" "$LOCAL" "$REMOTE" "$MERGED"'
$ git config --global mergetool.trustExitCode false
$ git config --global diff.external extDiff
```

~/.gitconfig

```
[merge]
  tool = extMerge
[mergetool "extMerge"]
  cmd = extMerge "$BASE" "$LOCAL" "$REMOTE" "$MERGED"
  trustExitCode = false
[diff]
  external = extDiff
```

diff

```
$ git diff 32d1776b1^ 32d1776b1
```

diffGit P4Merge7-1

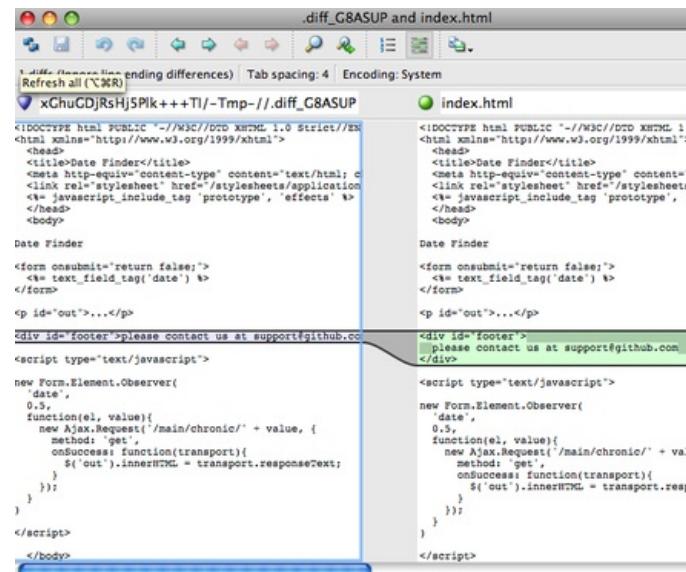


Figure 7-1. P4Merge.

git mergetoolGit P4Merge

diffmerge extDiff extMerge KDiff3 extMerge

```
$ cat /usr/local/bin/extMerge
#!/bin/sh
/Applications/kdiff3.app/Contents/MacOS/kdiff3 $*
```

Git KDiff3

Gitcmd kdiff3opendifftkdiffmeldxxdiffmergevimdiffgvimdiffKDiff3 kdiff:

```
$ git config --global merge.tool kdiff3
```

extMergeextDiffGit KDiff3

Windows Git

core.autorl

WindowsWindows WindowsMacLinux

GitCRLFLF CRLF core.autorl Windows trueLFCRLF

```
$ git config --global core.autorl true
```

LinuxMacLF Git CRLF core.autorl input Git CRLFLF

```
$ git config --global core.autorl input
```

WindowsCRLFMacLinuxLF

WindowsWindows false

```
$ git config --global core.autorl false
```

core.whitespace

Git422

2trailing-space space-before-tab trailing-space space-before-tab

2indent-with-non-tab cr-at-eol indent-with-non-tab 8 cr-at

core.whitespace - cr-at-eol

```
$ git config --global core.whitespace \
    trailing-space,space-before-tab,indent-with-non-tab
```

git diff Git git apply Git

```
$ git apply --whitespace=warn <patch>
```

Git

```
$ git apply --whitespace=fix <patch>
```

--whitespace=fix rebase Git

Git

receive.fsckObjects

GitSHA-1 Git Git Git receive.fsc

```
$ git config --system receive.fsckObjects true
```

Git

receive.denyNonFastForwards

push -f

receive.denyNonFastForwards

```
$ git config --system receive.denyNonFastForwards true
```

receive.denyDeletes

denyNonFastForwards Git 1.6.1

receive.denyDeletes Git 1.6.1

```
$ git config --system receive.denyDeletes true
```

—

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#).

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

Git

Git Git

Git hooks .git/hooks Git shellPerl — RubyPythonGit 1.6
.sampleGit 1.6

Git hooks Git

4 pre-commitGit git commit --no-verify

prepare-commit-msgSHA-1

commit-msgGit

post-commit git log -1 HEAD

clone

E-mail

3e-mail git ame-mail git format-patch

applypatch-msg Git

git am pre-applypatch

git am

git am post-applypatch

pre- rebaseGit

pre- rebasenextnext

git checkout post-checkout

merge post-merge Git Git

pre-receive post-receive

The first script to run when handling a push from a client is pre-receive. If any of the references being pushed are being pushed from stdin; if it exits non-zero, none of them are accepted. This hook is useful for things like make sure none of the updated references are non-fast-forward or that the user doing the pushing has create, delete, or push access or access to push to a specific branch before modifying with the push.

The post-receive hook runs after the entire process is completed and can be used to update services or notify users. It takes the same stdin data as the pre-receive hook and can be used for mailing a list, notifying a continuous integration server, or updating a ticket database. It can even parse the commit messages to see if any tickets need to be opened. Note that this script can't stop the push process, but the client doesn't disconnect until the push is complete, so you can do anything you want when you try to do anything that may take a long time.

update

The update script is very similar to the pre-receive script, except that pusher is trying to update. If the pusher is trying to push to multiple branches, whereas update runs once per branch they're pushing to. Instead, update takes three arguments: the name of the reference (branch), the SHA-1 of the push, and the SHA-1 the user is trying to push. If the update script rejects the reference, the update fails and the reference is rejected; other references can still be updated.

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#), [Macedonian](#), [Polish](#), [Thai](#) and [Russian](#).

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).
Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and

An Example Git-Enforced Policy

In this section, you'll use what you've learned to establish a Git workflow commit message format, enforces fast-forward-only pushes, and allows certain subdirectories in a project. You'll build client scripts that help to be rejected and server scripts that actually enforce the policies.

I used Ruby to write these, both because it's my preferred scripting language and most pseudocode-looking of the scripting languages; thus you should even if you don't use Ruby. However, any language will work fine. All examples with Git are in either Perl or Bash scripting, so you can also see plenty of other languages by looking at the samples.

Server-Side Hook

All the server-side work will go into the update file in your hooks directory. This file is run whenever a branch is being pushed and takes the reference being pushed to, the old revision and the new revision being pushed. You also have access to the user who is connecting over SSH. If you've allowed everyone to connect with a single password authentication, you may have to give that user a shell wrapper that does not require a password based on the public key, and set an environment variable specifying that the connecting user is in the \$USER environment variable, so your update hook has all the information you need:

```
#!/usr/bin/env ruby

$refname = ARGV[0]
$oldrev = ARGV[1]
$newrev = ARGV[2]
$user = ENV['USER']

puts "Enforcing Policies... \n(#{refname}) (#{$oldrev[0,6]}) (#{$newrev[0,6]})"
```

Yes, I'm using global variables. Don't judge me — it's easier to demo!

Enforcing a Specific Commit-Message Format

Your first challenge is to enforce that each commit message must adl have a target, assume that each message has to include a string that want each commit to link to a work item in your ticketing system. You pushed up, see if that string is in the commit message, and, if the stri commits, exit non-zero so the push is rejected.

You can get a list of the SHA-1 values of all the commits that are bein and \$oldrev values and passing them to a Git plumbing command ca the git log command, but by default it prints out only the SHA-1 valu get a list of all the commit SHAs introduced between one commit SH/ something like this:

```
$ git rev-list 538c33..d14fc7  
d14fc7c847ab946ec39590d87783c69b031bdfb7  
9f585da4401b0a3999e84113824d15245c13f0be  
234071a1be950e2a8d078e6141f5cd20c1e61ad3  
dfa04c9ef3d5197182f13fb5b9b1fb7717d2222a  
17716ec0f1ff5c77eff40b7fe912f9f6cf0e475
```

You can take that output, loop through each of those commit SHAs, g that message against a regular expression that looks for a pattern.

You have to figure out how to get the commit message from each of t raw commit data, you can use another plumbing command called git plumbing commands in detail in Chapter 9; but for now, here's what tl

```
$ git cat-file commit ca82a6  
tree cfda3bf379e4f8dba8717dee55aab78aef7f4daf  
parent 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7  
author Scott Chacon <schacon@gmail.com> 1205815931 -0700  
committer Scott Chacon <schacon@gmail.com> 1240030591 -0700  
  
changed the version number
```

A simple way to get the commit message from a commit when you have a first blank line and take everything after that. You can do so with the `sed` command:

```
$ git cat-file commit ca82a6 | sed '1,/^\$/d'  
changed the version number
```

You can use that incantation to grab the commit message from each commit and exit if you see anything that doesn't match. To exit the script and make the whole method look like this:

```
$regex = /^[ref: (\d+)]/  
  
# enforced custom commit message format  
def check_message_format  
  missed_revs = `git rev-list #{$oldrev}...#{$newrev}`.split("\n")  
  missed_revs.each do |rev|  
    message = `git cat-file commit #{rev} | sed '1,/^\$/d'`  
    if !$regex.match(message)  
      puts "[POLICY] Your message is not formatted correctly"  
      exit 1  
    end  
  end  
end  
check_message_format
```

Putting that in your update script will reject updates that contain commits that don't adhere to your rule.

Enforcing a User-Based ACL System

Suppose you want to add a mechanism that uses an access control list to determine who is allowed to push changes to which parts of your projects. Some people might only have access to push changes to certain subdirectories or specific files. You can add those rules to a file named `acl` that lives in your bare Git repository or `hooks/post-push`. This hook looks at those rules, sees what files are being introduced for all the pushes, and then determines whether the user doing the push has access to update all those files.

The first thing you'll do is write your ACL. Here you'll use a format very similar to the one you used for the commit message checker:

mechanism: it uses a series of lines, where the first field is avail or ur delimited list of the users to which the rule applies, and the last field is (blank meaning open access). All of these fields are delimited by a pi

In this case, you have a couple of administrators, some documentation directory, and one developer who only has access to the lib and test looks like this:

```
avail|nickh,pjhyett,defunkt,tpw
avail|us Sinclair,cdickens,ebronte|doc
avail|schacon|lib
avail|schacon|tests
```

You begin by reading this data into a structure that you can use. In thi simple, you'll only enforce the avail directives. Here is a method that where the key is the user name and the value is an array of paths to \

```
def get_acl_access_data(acl_file)
  # read in ACL data
  acl_file = File.read(acl_file).split("\n").reject { |line| line == '' }
  access = {}
  acl_file.each do |line|
    avail, users, path = line.split('|')
    next unless avail == 'avail'
    users.split(',').each do |user|
      access[user] ||= []
      access[user] << path
    end
  end
  access
end
```

On the ACL file you looked at earlier, this get_acl_access_data method looks like this:

```
{"defunkt"=>[nil],
 "tpw"=>[nil],
 "nickh"=>[nil],
 "pjhyett"=>[nil],
 "schacon"=>["lib", "tests"],
 "cdickens"=>["doc"],
 "us Sinclair"=>["doc"],
 "ebronte"=>["doc"]}
```

Now that you have the permissions sorted out, you need to determine what files have been modified, so you can make sure the user who's pushing has the right permissions.

You can pretty easily see what files have been modified in a single command by piping the output of the `git log` command (mentioned briefly in Chapter 2):

```
$ git log -1 --name-only --pretty=format:'' 9f585d  
README  
lib/test.rb
```

If you use the ACL structure returned from the `get_acl_access_data` method and iterate over the listed files in each of the commits, you can determine whether the user has access to those files in those commits:

```
# only allows certain users to modify certain subdirectories in a project
def check_directory_perms
  access = get_acl_access_data('acl')

  # see if anyone is trying to push something they can't
  new_commits = `git rev-list #{$oldrev}..#{$newrev}`.split("\n")
  new_commits.each do |rev|
    files_modified = `git log -1 --name-only --pretty=format:'' #{rev}`.split("\n")
    files_modified.each do |path|
      next if path.size == 0
      has_file_access = false
      access[$user].each do |access_path|
        if !access_path # user has access to everything
          || (path.index(access_path) == 0) # access to this path
          has_file_access = true
        end
      end
      if !has_file_access
        puts "[POLICY] You do not have access to push to #{path}"
        exit 1
      end
    end
  end
end

check_directory_perms
```

Most of that should be easy to follow. You get a list of new commits by running `git rev-list`. Then, for each of those, you find which files are modified and check if the user has access to those files.

pushing has access to all the paths being modified. One Rubyism that path.index(access_path) == 0, which is true if path begins with access_path is not just in one of the allowed paths, but an allowed path

Now your users can't push any commits with badly formed messages in their designated paths.

Enforcing Fast-Forward-Only Pushes

The only thing left is to enforce fast-forward-only pushes. In Git versions 1.8.4 and later, you can set the receive.denyDeletes and receive.denyNonFastForwards settings. But this is not available in older versions of Git, and you can modify it to do so only for certain paths by setting up a hook script with later.

The logic for checking this is to see if any commits are reachable from the newer commit. If there are none, then it was a fast-forward push; if there are some, then it was a non-fast-forward push.

```
# enforces fast-forward only pushes
def check_fast_forward
  missed_refs = `git rev-list #{$newrev}..#{$oldrev}`
  missed_ref_count = missed_refs.split("\n").size
  if missed_ref_count > 0
    puts "[POLICY] Cannot push a non fast-forward reference"
    exit 1
  end
end

check_fast_forward
```

Everything is set up. If you run chmod u+x .git/hooks/update, which is what we did earlier, have put all this code, and then try to push a non-fast-forwarded reference, you will get an error message.

```
$ git push -f origin master
Counting objects: 5, done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 323 bytes, done.
Total 3 (delta 1), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
Enforcing Policies...
(refs/heads/master) (8338c5) (c5b616)
[POLICY] Cannot push a non-fast-forward reference
error: hooks/update exited with error code 1
```

```
error: hook declined to update refs/heads/master
To git@gitserver:project.git
  ! [remote rejected] master -> master (hook declined)
error: failed to push some refs to 'git@gitserver:project.git'
```

There are a couple of interesting things here. First, you see this when:

```
Enforcing Policies...
(refs/heads/master) (fb8c72) (c56860)
```

Notice that you printed that out to stdout at the very beginning of your note that anything your script prints to stdout will be transferred to the

The next thing you'll notice is the error message.

```
[POLICY] Cannot push a non fast-forward reference
error: hooks/update exited with error code 1
error: hook declined to update refs/heads/master
```

The first line was printed out by you, the other two were Git telling you zero and that is what is declining your push. Lastly, you have this:

```
To git@gitserver:project.git
  ! [remote rejected] master -> master (hook declined)
error: failed to push some refs to 'git@gitserver:project.git'
```

You'll see a remote rejected message for each reference that your hook was declined specifically because of a hook failure.

Furthermore, if the ref marker isn't there in any of your commits, you'll printing out for that.

```
[POLICY] Your message is not formatted correctly
```

Or if someone tries to edit a file they don't have access to and push a something similar. For instance, if a documentation author tries to pus

in the `lib` directory, they see

```
[POLICY] You do not have access to push to lib/test.rb
```

That's all. From now on, as long as that `update` script is there and executed, the user will never have a commit message without your pattern, and it will be sandboxed.

Client-Side Hooks

The downside to this approach is the whining that will inevitably result when users try to push their changes. Having their carefully crafted work rejected at the last minute is frustrating, confusing, and confusing; and furthermore, they will have to edit their history to correct it, which is a faint of heart.

The answer to this dilemma is to provide some client-side hooks that tell the user what they're doing something that the server is likely to reject. That way, the user can fix the problem before committing and before those issues become more difficult to fix. When you transfer a project or a clone of a project, you must distribute these scripts to your users copy them to their `.git/hooks` directory and make them executable. You can't hook within the project or in a separate project, but there is no way to prevent it.

To begin, you should check your commit message just before each commit. If the server won't reject your changes due to badly formatted commit messages, you can use a `commit-msg` hook. If you have it read the message from the file passed to it, compare that to the pattern, you can force Git to abort the commit if the message doesn't match.

```
#!/usr/bin/env ruby
message_file = ARGV[0]
message = File.read(message_file)

$regex = /^[ref: (\d+)]/

if !$regex.match(message)
  puts "[POLICY] Your message is not formatted correctly"
  exit 1
end
```

If that script is in place (in `.git/hooks/commit-msg`) and executable, and isn't properly formatted, you see this:

```
$ git commit -am 'test'  
[POLICY] Your message is not formatted correctly
```

No commit was completed in that instance. However, if your message allows you to commit:

```
$ git commit -am 'test [ref: 132]'  
[master e05c914] test [ref: 132]  
 1 files changed, 1 insertions(+), 0 deletions(-)
```

Next, you want to make sure you aren't modifying files that are outside the `.git` directory. If the `.git/hooks/commit-msg` file contains a copy of the ACL file you used previously, the script will enforce those constraints for you:

```
#!/usr/bin/env ruby  
  
$user      = ENV['USER']  
  
# [ insert acl_access_data method from above ]  
  
# only allows certain users to modify certain subdirectories in a project  
def check_directory_perms  
  access = get_acl_access_data('.git/acl')  
  
  files_modified = `git diff-index --cached --name-only HEAD`.split("\n")  
  files_modified.each do |path|  
    next if path.size == 0  
    has_file_access = false  
    access[$user].each do |access_path|  
      if !access_path || (path.index(access_path) == 0)  
        has_file_access = true  
      end  
      if !has_file_access  
        puts "[POLICY] You do not have access to push to #{path}"  
        exit 1  
      end  
    end  
  end  
end  
  
check_directory_perms
```

This is roughly the same script as the server-side part, but with two in file is in a different place, because this script runs from your working directory. You have to change the path to the ACL file from this

```
access = get_acl_access_data('acl')
```

to this:

```
access = get_acl_access_data('.git/acl')
```

The other important difference is the way you get a listing of the files — the server-side method looks at the log of commits, and, at this point, yet, you must get your file listing from the staging area instead. Instead

```
files_modified = `git log -1 --name-only --pretty=format:'' #{ref}`
```

you have to use

```
files_modified = `git diff-index --cached --name-only HEAD`
```

But those are the only two differences — otherwise, the script works the same. It expects you to be running locally as the same user you push as to the server. If you’re pushing under a different, you must set the \$user variable manually.

The last thing you have to do is check that you’re not trying to push non-fast-forward changes. That is, if you’re pushing up to a branch that is a bit less common. To get a reference that isn’t a fast-forward, you can either push up to a branch that you’ve already pushed up or try pushing a different local branch.

Because the server will tell you that you can’t push a non-fast-forward changes, the only accidental thing you can try to catch is rebasing a local branch that has been pushed.

Here is an example pre-rebase script that checks for that. It gets a list of rewrite and checks whether they exist in any of your remote references from one of your remote references, it aborts the rebase:

```
#!/usr/bin/env ruby

base_branch = ARGV[0]
if ARGV[1]
  topic_branch = ARGV[1]
else
  topic_branch = "HEAD"
end

target_shas = `git rev-list #{base_branch}..#{topic_branch}`.split("\n")
remote_refs = `git branch -r`.split("\n").map { |r| r.strip }

target_shas.each do |sha|
  remote_refs.each do |remote_ref|
    shas_pushed = `git rev-list ^#{sha}^@ refs/remotes/#{remote_ref}`
    if shas_pushed.split("\n").include?(sha)
      puts "[POLICY] Commit #{sha} has already been pushed to #{remote_ref}"
      exit 1
    end
  end
end
```

This script uses a syntax that wasn't covered in the Revision Selection list of commits that have already been pushed up by running this:

```
git rev-list ^#{sha}^@ refs/remotes/#{remote_ref}
```

The SHA^{^@} syntax resolves to all the parents of that commit. You're looking for something reachable from the last commit on the remote and that isn't reachable from the SHAs you're trying to push up — meaning it's a fast-forward.

The main drawback to this approach is that it can be very slow and is likely to fail if you try to force the push with -f, the server will warn you and not accept the push. This is an interesting exercise and can in theory help you avoid a rebase that you'll need to fix.

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

Summary

You've covered most of the major ways that you can customize your workflow and projects. You've learned about all sorts of configuration event hooks, and you've built an example policy-enforcing server. You fit nearly any workflow you can dream up.

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and

Git Subversion

Subversion Subversion CVS

Git git svn Subversion Git Subversion Git Subversion
Subversion Git Git Subversion DVCS, Distributed
VCS

git svn

Git Subversion git svn

git svn Subversion Git Git

Git Git Subversion SVN Git SVN ——

SVN svnsync Subversion

Subversion

```
$ mkdir /tmp/test-svn
$ svnadmin create /tmp/test-svn
```

revprop —— 0 pre-revprop-change

```
$ cat /tmp/test-svn/hooks/pre-revprop-change
#!/bin/sh
exit 0;
$ chmod +x /tmp/test-svn/hooks/pre-revprop-change
```

```
svnsync init
```

```
$ svnsync init file:///tmp/test-svn http://progit-example.googlecode.com/svn/
```

```
$ svnsync sync file:///tmp/test-svn  
Committed revision 1.  
Copied properties for revision 1.  
Committed revision 2.  
Copied properties for revision 2.  
Committed revision 3.  
...
```

100 Subversion —

Subversion git svn clone Subversion Git Subversion
file:///tmp/test-svn Subversion URL

```
$ git svn clone file:///tmp/test-svn -T trunk -b branches -t tags  
Initialized empty Git repository in /Users/schacon/projects/testsvnsync/svn/.git/  
r1 = b4e387bc68740b5af56c2a5faf4003ae42bd135c (trunk)  
A m4/acx_pthread.m4  
A m4/stl_hash.m4  
...  
r75 = d1957f3b307922124eec6314e15bcda59e3d9610 (trunk)  
Found possible branch point: file:///tmp/test-svn/trunk => \  
  file:///tmp/test-svn /branches/my-calc-branch, 75  
Found branch parent: (my-calc-branch) d1957f3b307922124eec6314e15bcda59e3d9610  
Following parent with do_switch  
Successfully followed parent  
r76 = 8624824ecc0badd73f40ea2f01fce51894189b01 (my-calc-branch)  
Checked out HEAD:  
  file:///tmp/test-svn/branches/my-calc-branch r76
```

URL — git svn init gitsvn fetch 75 Git

-T trunk -b branches -t tags Git Subversion (trunkmaster
s Standard layout

```
$ git svn clone file:///tmp/test-svn -s
```

Git

```
$ git branch -a
* master
  my-calc-branch
  tags/2.0.2
  tags/release-2.0.1
  tags/release-2.0.2
  tags/release-2.0.2rc1
  trunk
```

Git

origin/[branch] ——

git svn Git

```
$ git show-ref
1cbd4904d9982f386d87f88fce1c24ad7c0f0471 refs/heads/master
ae1ecc26318164f355a883f5d99cff0c852d3c4 refs/remotes/my-calc-branch
03d09b0e2aad427e34a6d50ff147128e76c0e0f5 refs/remotes/tags/2.0.2
50d02cc0adc9da4319eeba0900430ba219b9c376 refs/remotes/tags/release-2.0.1
4caaa711a50c77879a91b8b90380060f672745cb refs/remotes/tags/release-2.0.2
1c4cb508144c513ff1214c3488abe66dcb92916f refs/remotes/tags/release-2.0.2rc1
1cbd4904d9982f386d87f88fce1c24ad7c0f0471 refs/remotes/trunk
```

Git

```
$ git show-ref
83e38c7a0af325a9722f2fdc56b10188806d83a1 refs/heads/master
3e15e38c198baac84223acf6224bb8b99ff2281 refs/remotes/gitserver/master
0a30dd3b0c795b80212ae723640d4e5d48cabdff refs/remotes/origin/master
25812380387fdd55f916652be4881c6f11600d6f refs/remotes/origin/testing
```

gitserver master origin master testing

git svn Subversion) Git Subversion tags

Subversion

Git SVN Git Subversion

```
$ git commit -am 'Adding git-svn instructions to the README'  
[master 97031e5] Adding git-svn instructions to the README  
1 files changed, 1 insertions(+), 1 deletions(-)
```

Subversion —— Subversion Subversion

git svn

```
$ git svn dcommit  
Committing to file:///tmp/test-svn/trunk ...  
M README.txt  
Committed r79  
M README.txt  
r79 = 938b1a547c2cc92033b74d32030e86468294a5c8 (trunk)  
No changes between current HEAD and refs/remotes/trunk  
Resetting to the latest refs/remotes/trunk
```

Subversion commit Subversion Git commit commit SHA-1 Git Subversion commit

```
$ git log -1  
commit 938b1a547c2cc92033b74d32030e86468294a5c8  
Author: schacon <schacon@4c93b258-373f-11de-be05-5f7a86268029>  
Date: Sat May 2 22:06:44 2009 +0000  
  
    Adding git-svn instructions to the README  
  
    git-svn-id: file:///tmp/test-svn/trunk@79 4c93b258-373f-11de-be05-5f7a86268029
```

97031e5 SHA-1 938b1a5 Git Subversion Subversion

git svn

```
$ git svn dcommit  
Committing to file:///tmp/test-svn/trunk ...  
Merge conflict during commit: Your file or directory 'README.txt' is probably \  
out-of-date: resource out of date; try updating at /Users/schacon/libexec/git-\  
core/git-svn line 482
```

git svn rebase

```
$ git svn rebase
      M      README.txt
r80 = ff829ab914e8775c7c025d741beb3d523ee30bc4 (trunk)
First, rewinding head to replay your work on top of it...
Applying: first user change
```

dcommit

```
$ git svn dcommit
Committing to file:///tmp/test-svn/trunk ...
      M      README.txt
Committed r81
      M      README.txt
r81 = 456cbe6337abe49154db70106d1836bc1332deed (trunk)
No changes between current HEAD and refs/remotes/trunk
Resetting to the latest refs/remotes/trunk
```

Git git svn dcommit

```
$ git svn dcommit
Committing to file:///tmp/test-svn/trunk ...
      M      configure.ac
Committed r84
      M      autogen.sh
r83 = 8aa54a74d452f82eee10076ab2584c1fc424853b (trunk)
      M      configure.ac
r84 = cdbac939211ccb18aa744e581e46563af5d962d0 (trunk)
W: d2f23b80f67aaaa1f6f5aaef48fce3263ac71a92 and refs/remotes/trunk differ, \
    using rebase:
:100755 100755 efa5a59965fbbb5b2b0a12890f1b351bb5493c18 \
    015e4c98c482f0fa71e4d5434338014530b37fa6 M  autogen.sh
First, rewinding head to replay your work on top of it...
Nothing to do.
```

Git —— Git SVN

Subversion sit svn fetch git svn rebase

```
$ git svn rebase
      M      generate_descriptor_proto.sh
r82 = bd16df9173e424c6f52c337ab6efa7f7643282f1 (trunk)
First, rewinding head to replay your work on top of it...
Fast-forwarded master to refs/remotes/trunk.
```

git svn rebase

git svn rebase ——

Git

Git git svn Subversion Subversion Git Git svn
Subversion commit

experiment master dcommit

```
$ git svn dcommit
Committing to file:///tmp/test-svn/trunk ...
    M      CHANGES.txt
Committed r85
    M      CHANGES.txt
r85 = 4bfebeec434d156c36f2bcd18f4e3d97dc3269a2 (trunk)
No changes between current HEAD and refs/remotes/trunk
Resetting to the latest refs/remotes/trunk
COPYING.txt: locally modified
INSTALL.txt: locally modified
    M      COPYING.txt
    M      INSTALL.txt
Committed r86
    M      INSTALL.txt
    M      COPYING.txt
r86 = 2647f6b86ccfcaad4ec58c520e369ec81f7c283c (trunk)
No changes between current HEAD and refs/remotes/trunk
Resetting to the latest refs/remotes/trunk
```

dcommit Git experiment commit — SVN commit

commit

Subversion

Subversion Git git svn Subversion

SVN

Subversion git svn branch [] To create a new branch in S
branch [branchname]:

```
$ git svn branch opera
Copying file:///tmp/test-svn/trunk at r87 to file:///tmp/test-svn/branches/opera..
Found possible branch point: file:///tmp/test-svn/trunk => \
file:///tmp/test-svn/branches/opera, 87
```

```
Found branch parent: (opera) 1f6bfe471083cbca06ac8d4176f7ad4de0d62e5f
Following parent with do_switch
Successfully followed parent
r89 = 9b6fe0b90c5c9adf9165f700897518dbc54a7cbf (opera)
```

Subversion svn copy trunk branches/opera Subversion

Git Subversion dcommit —— git-svn-id

Subversion commit dcommit opera

```
$ git branch opera remotes/opera
```

opera trunk master git merge -m Merge
git merge Subversion Git Git commit Subversion
commit Git
Subversion

Subversion

git svn Subversion Git Subversion

SVN

Subversion SVN git svn log SVN

```
$ git svn log
-----
r87 | schacon | 2009-05-02 16:07:37 -0700 (Sat, 02 May 2009) | 2 lines
autogen change

-----
r86 | schacon | 2009-05-02 16:00:21 -0700 (Sat, 02 May 2009) | 2 lines
Merge branch 'experiment'
```

```
r85 | schacon | 2009-05-02 16:00:09 -0700 (Sat, 02 May 2009) | 2 lines  
updated the changelog
```

git svn log svn log Subversion Subversion commit
Git Subversion Subversion

SVN

```
git svn log    git log      svn annotate    git svn blame []
```

```
$ git svn blame README.txt  
2 temporal Protocol Buffers - Google's data interchange format  
2 temporal Copyright 2008 Google Inc.  
2 temporal http://code.google.com/apis/protocolbuffers/  
2 temporal  
22 temporal C++ Installation - Unix  
22 temporal =====  
2 temporal  
79 schacon Committing in git-svn.  
78 schacon  
2 temporal To build and install the C++ Protocol Buffer runtime and the Proto  
2 temporal Buffer compiler (protoc) execute the following:  
2 temporal
```

Git Subversion

SVN

```
git svn info    svn info
```

```
$ git svn info  
Path: .  
URL: https://schacon-test.googlecode.com/svn/trunk  
Repository Root: https://schacon-test.googlecode.com/svn  
Repository UUID: 4c93b258-373f-11de-be05-5f7a86268029  
Revision: 87  
Node Kind: directory  
Schedule: normal  
Last Changed Author: schacon  
Last Changed Rev: 87  
Last Changed Date: 2009-05-02 16:07:37 -0700 (Sat, 02 May 2009)
```

blame log Subversion

Subversion

svn:ignore Subversion .gitignore git svn git sv
.gitignore

git svn show-ignore .gitignore

```
$ git svn show-ignore > .git/info/exclude
```

.gitignore Subversion Git .gitignore

Git-Svn

git svn Subversion Subversion Git

- git merge commit
- Git git-svn-id pre-receive git-svn-id comm

Subversion Git

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#), [Macedonian](#), [Polish](#), [Thai](#) and [Russian](#).

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

Git

Git Git

— Subversion Perforce — Git Git

Subversion

git svn git svn clone Subversion Git server Subversion

Subversion

schacon blame git svn

Git

user.txt

```
schacon = Scott Chacon <schacon@geemail.com>
selse = Someo Nelse <selse@geemail.com>
```

SVN

```
$ svn log --xml | grep author | sort -u | perl -pe 's/.>(.?)<./$1 = /'
```

XML — XML

grepsort perl. user.txt Git

git svn clone init --no-metadata git svn Subversio

```
$ git-svn clone http://my-project.googlecode.com/svn/ \
--authors-file=users.txt --no-metadata -s my_project
```

my_project Subversion commit

```
commit 37efa680e8473b615de980fa935944215428a35a
Author: schacon <schacon@4c93b258-373f-11de-be05-5f7a86268029>
Date:   Sun May 3 00:12:22 2009 +0000

    fixed install - go to trunk

git-svn-id: https://my-project.googlecode.com/svn/trunk@94 4c93b258-373f-11de-
be05-5f7a86268029
```

```
commit 03a8785f44c8ea5cdb0e8834b7c8e6c469be2ff2
Author: Scott Chacon <schacon@geemail.com>
Date:   Sun May 3 00:12:22 2009 +0000

    fixed install - go to trunk
```

git-svn-id

post-import git svn

Git

```
$ cp -Rf .git/refs/remotes/tags/* .git/refs/tags/
$ rm -Rf .git/refs/remotes/tags
```

tag/

refs/remotes

```
$ cp -Rf .git/refs/remotes/* .git/refs/heads/
$ rm -Rf .git/refs/remotes
```

Git Git Git

```
$ git push origin --all
```

Git

Perforce

Perforce. Git Perforce contrib — git svn Git git.kernel.org

```
$ git clone git://git.kernel.org/pub/scm/git/git.git
$ cd git/contrib/fast-import
```

fast-import git-p4 Python Python p4 Perforce DepotPerforce Jam P4PORT export Perforce

```
$ export P4PORT=public.perforce.com:1666
```

git-p4 clone Perforce Jam

```
$ git-p4 clone //public/jam/src@all /opt/p4import
Importing from //public/jam/src@all into /opt/p4import
Reinitialized existing Git repository in /opt/p4import/.git/
Import destination: refs/remotes/p4/master
Importing revision 4409 (100%)
```

/opt/p4import git log

```
$ git log -2
commit 1fd4ec126171790efd2db83548b85b1bbbc07dc2
Author: Perforce staff <support@perforce.com>
Date:   Thu Aug 19 10:18:45 2004 -0800

    Drop 'rc3' moniker of jam-2.5. Folded rc2 and rc3 RELNOTES into
    the main part of the document. Built new tar/zip balls.

    Only 16 months later.

    [git-p4: depot-paths = "//public/jam/src/": change = 4409]

commit ca8870db541a23ed867f38847eda65bf4363371d
Author: Richard Geiger <rmg@perforce.com>
Date:   Tue Apr 22 20:51:34 2003 -0800

    Update derived jamgram.c
```

```
[git-p4: depot-paths = "//public/jam/src/": change = 3108]
```

commit git-p4 Perforce — git filter

```
$ git filter-branch --msg-filter '
    sed -e "/^#[git-p4]/d"
'
Rewrite 1fd4ec126171790efd2db83548b85b1bbbc07dc2 (123/123)
Ref 'refs/heads/master' was rewritten
```

git log commit SHA-1 git-p4

```
$ git log -2
commit 10a16d60cffca14d454a15c6164378f4082bc5b0
Author: Perforce staff <support@perforce.com>
Date:   Thu Aug 19 10:18:45 2004 -0800

Drop 'rc3' moniker of jam-2.5. Folded rc2 and rc3 RELNOTES into
the main part of the document. Built new tar/zip balls.

Only 16 months later.

commit 2b6c6db311dd76c34c66ec1c40a49405e6b527b2
Author: Richard Geiger <rmg@perforce.com>
Date:   Tue Apr 22 20:51:34 2003 -0800

Update derived jamgram.c
```

Git

Subversion Perforce — CVS Clear Case Visual Source Safe
import Git Git Git git fast-import

back_YY_MM_DD Git

```
$ ls /opt/import_from
back_2009_01_02
back_2009_01_04
back_2009_01_14
back_2009_02_03
current
```

Git Git Git commit fast-import commit co
commit

“Git ” Ruby —— Windows carriage returns Windows\r ——
Git fast-import LF Windows CRLF

commit

```
last_mark = nil

#
Dir.chdir(ARGV[0]) do
  Dir.glob("**").each do |dir|
    next if File.file?(dir)

    #
    Dir.chdir(dir) do
      last_mark = print_export(dir, last_mark)
    end
  end
end
```

print_export "mark" fast-import commit commit co
print_export

```
mark = convert_dir_to_mark(dir)
```

```
$marks = []
def convert_dir_to_mark(dir)
  if !$marks.include?(dir)
    $marks << dir
  end
  ($marks.index(dir) + 1).to_s
end
```

commit print_export

```
date = convert_dir_to_date(dir)
```

convert_dir_to_date

```
def convert_dir_to_date(dir)
  if dir == 'current'
    return Time.now().to_i
  else
    dir = dir.gsub('back_', '')
    (year, month, day) = dir.split('_')
    return Time.local(year, month, day).to_i
  end
end
```

```
$author = 'Scott Chacon <schacon@example.com>'
```

commit commit

```
#  
puts 'commit refs/heads/master'  
puts 'mark :' + mark  
puts "committer #{$author} #{date} -0700"  
export_data('imported from ' + dir)  
puts 'from :' + last_mark if last_mark
```

-0700

```
data (size)\n(contents)
```

data

export_data

```
def export_data(string)
  print "data #{string.size}\n#{string}"
end
```

—— deleteall Git

```
puts 'deleteall'  
Dir.glob("**/*").each do |file| next if !File.file?(file)  
  inline_data(file)  
end
```

commit commit fast-import —— Git

```
M 644 inline path/to/file  
data (size)  
(file contents)
```

644 755 inline

inline_data

```
def inline_data(file, code = 'M', mode = '644')  
  content = File.read(file)  
  puts "#{code} #{mode} inline #{file}"  
  export_data(content)  
end
```

export_data

```
return mark
```

WindowsWindows CRLF Git fast-import LF git fast-import ruby LF CRLF

```
$stdout.binmode
```

```
$ ruby import.rb /opt/import_from  
commit refs/heads/master  
mark :1  
committer Scott Chacon <schacon@geemail.com> 1230883200 -0700  
data 29  
imported from back_2009_01_02deleteall  
M 644 inline file.rb  
data 12  
version two  
commit refs/heads/master  
mark :2  
committer Scott Chacon <schacon@geemail.com> 1231056000 -0700  
data 29  
imported from back_2009_01_04from :1  
deleteall  
M 644 inline file.rb  
data 14  
version three  
M 644 inline new.rb  
data 16  
new version one  
(...)
```

git fast-import git init

```
$ git init  
Initialized empty Git repository in /opt/import_to/.git/  
$ ruby import.rb /opt/import_from | git fast-import  
git-fast-import statistics:  
-----  
Alloc'd objects:      5000  
Total objects:       18 (      1 duplicates )  
    blobs :          7 (      1 duplicates )  
    trees :          6 (      0 duplicates )  
    commits:         5 (      0 duplicates )  
    tags  :          0 (      0 duplicates )  
Total branches:      1 (      1 loads )  
    marks:        1024 (      5 unique )  
    atoms:          3  
Memory total:        2255 KiB  
    pools:          2098 KiB  
    objects:        156 KiB  
-----  
pack_report: getpagesize() = 4096  
pack_report: core.packedGitWindowSize = 33554432  
pack_report: core.packedGitLimit = 268435456  
pack_report: pack_used_ctr = 9  
pack_report: pack_mmap_calls = 5  
pack_report: pack_open_windows = 1 / 1  
pack_report: pack_mapped = 1356 / 1356  
-----
```

```
$ git log -2
commit 10bfe7d22ce15ee25b60a824c8982157ca593d41
Author: Scott Chacon <schacon@example.com>
Date:   Sun May 3 12:57:39 2009 -0700

        imported from current

commit 7e519590de754d079dd73b44d695a42c9d2df452
Author: Scott Chacon <schacon@example.com>
Date:   Tue Feb 3 01:00:00 2009 -0700

        imported from back_2009_02_03
```

— Git —

master

```
$ ls
$ git reset --hard master
HEAD is now at 10bfe7d imported from current
$ ls
file.rb  lib
```

fast-import — Git

contib/fast-import

g

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#).

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

Subversion Git Git Git

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and

Chapter 9

Git

Git Git

Git (content-addressable) VCS

Git (1.5) VCS UI Git UI

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and

(Plumbing) (Porcelain)

checkout, branch, remote 30 Git Git VCS VCS UNIX “plumbing”
“porcelain”

Git Git

git init Git .git Git

```
$ ls
HEAD
branches/
config
description
hooks/
index
info/
objects/
refs/
```

git init Git	branches	description	GitWeb
.gitignore (ignored patterns)			hooks
HEAD index objects refs Git	objects	refs	()
Git			

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#).

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

Git

Git

Git key-value hash-object .git Git

```
$ mkdir test
$ cd test
$ git init
Initialized empty Git repository in /tmp/test/.git/
$ find .git/objects
.git/objects
.git/objects/info
.git/objects/pack
$ find .git/objects -type f
$
```

Git objects pack info Git

```
$ echo 'test content' | git hash-object -w --stdin  
d670460b4b4aece5915caf5c68d12f560a9fe3e4
```

```
-w      hash-object ()          --stdin (stdin) 40 SHA-1 — G
```

```
$ find .git/objects -type f  
.git/objects/d6/70460b4b4aece5915caf5c68d12f560a9fe3e4
```

objects Git — SHA-1 () 38 ()

`cat-file` `Git` -p

```
$ git cat-file -p d670460b4b4aece5915caf5c68d12f560a9fe3e4  
test content
```

Git

```
$ echo 'version 1' > test.txt  
$ git hash-object -w test.txt  
83baae61804e65cc73a7201a7252750c76066a30
```

```
$ echo 'version 2' > test.txt  
$ git hash-object -w test.txt  
1f7a7a472abf3dd9643fd615f6da379c4acb3e3a
```

```
$ find .git/objects -type f  
.git/objects/1f/7a7a472abf3dd9643fd615f6da379c4acb3e3a  
.git/objects/83/baae61804e65cc73a7201a7252750c76066a30  
.git/objects/d6/70460b4b4aece5915caf5c68d12f560a9fe3e4
```

```
$ git cat-file -p 83baae61804e65cc73a7201a7252750c76066a30 > test.txt  
$ cat test.txt  
version 1
```

```
$ git cat-file -p 1f7a7a472abf3dd9643fd615f6da379c4acb3e3a > test.txt  
$ cat test.txt  
version 2
```

SHA-1 blob SHA-1

cat-file -t Git

```
$ git cat-file -t 1f7a7a472abf3dd9643fd615f6da379c4acb3e3a  
blob
```

tree ()

tree tree Git UNIX tree blob tree UNIX blob inodes
tree tree blob tree SHA-1 (mode) simplegit tree

```
$ git cat-file -p master^{tree}  
100644 blob a906cb2a4a904a152e80877d4088654daad0c859 README  
100644 blob 8f94139338f9404f26296befa88755fc2598c289 Rakefile  
040000 tree 99f1a6d12cb4b6f19c8655fca46c3ecf317074e0 lib
```

master^{tree} branch tree lib blob tree

```
$ git cat-file -p 99f1a6d12cb4b6f19c8655fca46c3ecf317074e0  
100644 blob 47c6340d6459e05787f644c2447d2595f5d3a54b simplegit.rb
```

Git 9-1

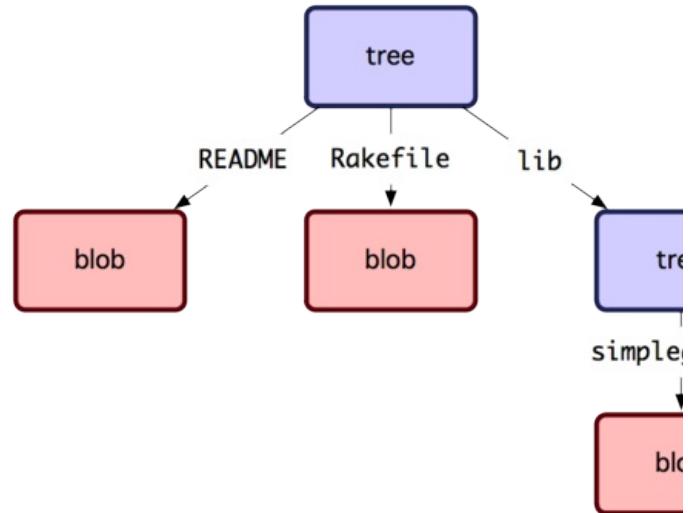


Figure 9-1. Git

tree Git index tree tree index plumbing update-index
— index test.txt ()

```
$ git update-index --add --cacheinfo 100644 \  
83baae61804e65cc73a7201a7252750c76066a30 test.txt
```

100644 100755 120000 UNIX — Git (blobs) ()

write-tree tree -w — tree write-tree index tree

```
$ git write-tree  
d8329fc1cc938780ffdd9f94e0d364e0ea74f579  
$ git cat-file -p d8329fc1cc938780ffdd9f94e0d364e0ea74f579  
100644 blob 83baae61804e65cc73a7201a7252750c76066a30 test.txt
```

tree

```
$ git cat-file -t d8329fc1cc938780ffdd9f94e0d364e0ea74f579  
tree
```

test.txt tree

```
$ echo 'new file' > new.txt  
$ git update-index test.txt  
$ git update-index --add new.txt
```

test.txt new.txt () tree (index tree)

```
$ git write-tree  
0155eb4229851634a0f03eb265b69f5a2d56f341  
$ git cat-file -p 0155eb4229851634a0f03eb265b69f5a2d56f341  
100644 blob fa49b077972391ad58037050f2a75f74e3671e92 new.txt  
100644 blob 1f7a7a472abf3dd9643fd615f6da379c4acb3e3a test.txt
```

tree test.txt SHA "" (1f7a7a) tree tree
-prefix read-tree tree tree

```
$ git read-tree --prefix=bak d8329fc1cc938780ffdd9f94e0d364e0ea74f579  
$ git write-tree  
3c4e9cd789d88d8d89c1073707c3585e41b0e614  
$ git cat-file -p 3c4e9cd789d88d8d89c1073707c3585e41b0e614  
040000 tree d8329fc1cc938780ffdd9f94e0d364e0ea74f579 bak  
100644 blob fa49b077972391ad58037050f2a75f74e3671e92 new.txt  
100644 blob 1f7a7a472abf3dd9643fd615f6da379c4acb3e3a test.txt
```

tree bak test.txt Git 9-2

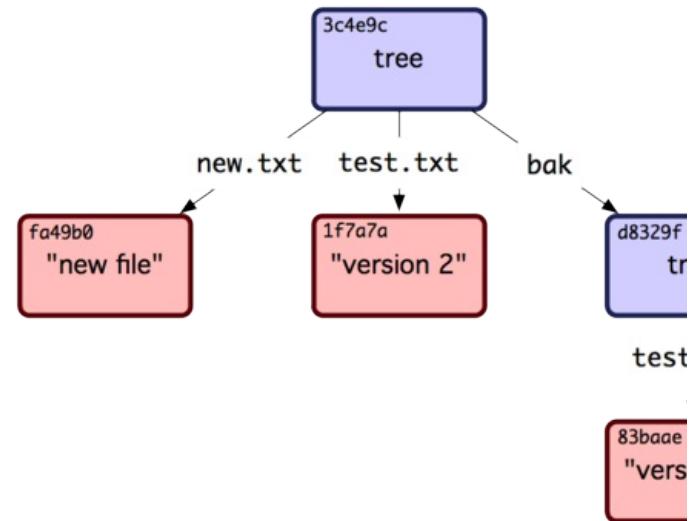


Figure 9-2. Git

commit ()

tree SHA-1 commit

commit commit-tree tree SHA-1 tree

```
$ echo 'first commit' | git commit-tree d8329f  
fdf4fc3344e67ab068f836878b6c4951e3b15f3d
```

cat-file commit

```
$ git cat-file -p fdf4fc3  
tree d8329fc1cc938780ffdd9f94e0d364e0ea74f579  
author Scott Chacon <schacon@gmail.com> 1243040974 -0700  
committer Scott Chacon <schacon@gmail.com> 1243040974 -0700  
  
first commit
```

commit / Git user.name user.email)

commit commit

```
$ echo 'second commit' | git commit-tree 0155eb -p fdf4fc3  
cac0cab538b970a37ea1e769cbbde608743bc96d  
$ echo 'third commit' | git commit-tree 3c4e9c -p cac0cab  
1a410efbd13591db07496601ebc7a059dd55cfe9
```

commit Git

git log commit SHA-1

```
$ git log --stat 1a410e  
commit 1a410efbd13591db07496601ebc7a059dd55cfe9  
Author: Scott Chacon <schacon@gmail.com>  
Date:   Fri May 22 18:15:24 2009 -0700  
  
        third commit  
  
        bak/test.txt |    1 +  
        1 files changed, 1 insertions(+), 0 deletions(-)  
  
commit cac0cab538b970a37ea1e769cbbde608743bc96d  
Author: Scott Chacon <schacon@gmail.com>  
Date:   Fri May 22 18:14:29 2009 -0700  
  
        second commit  
  
        new.txt |    1 +  
        test.txt |    2 +-  
        2 files changed, 2 insertions(+), 1 deletions(-)  
  
commit fdf4fc3344e67ab068f836878b6c4951e3b15f3d  
Author: Scott Chacon <schacon@gmail.com>  
Date:   Fri May 22 18:09:34 2009 -0700  
  
        first commit  
  
        test.txt |    1 +  
        1 files changed, 1 insertions(+), 0 deletions(-)
```

Git git add git commit Git — blob tree commit commit tree commit Git — blobtree tree —

```
$ find .git/objects -type f  
.git/objects/01/55eb4229851634a0f03eb265b69f5a2d56f341 # tree 2  
.git/objects/1a/410efbd13591db07496601ebc7a059dd55cfe9 # commit 3  
.git/objects/1f/7a7a472abf3dd9643fd615f6da379c4acb3e3a # test.txt v2  
.git/objects/3c/4e9cd789d88d8d89c1073707c3585e41b0e614 # tree 3  
.git/objects/83/baae61804e65cc73a7201a7252750c76066a30 # test.txt v1  
.git/objects/ca/c0cab538b970a37ea1e769cbbde608743bc96d # commit 2  
.git/objects/d6/70460b4b4aece5915caf5c68d12f560a9fe3e4 # 'test content'  
.git/objects/d8/329fc1cc938780ffdd9f94e0d364e0ea74f579 # tree 1  
.git/objects/fa/49b077972391ad58037050f2a75f74e3671e92 # new.txt  
.git/objects/fd/f4fc3344e67ab068f836878b6c4951e3b15f3d # commit 1
```

9-3

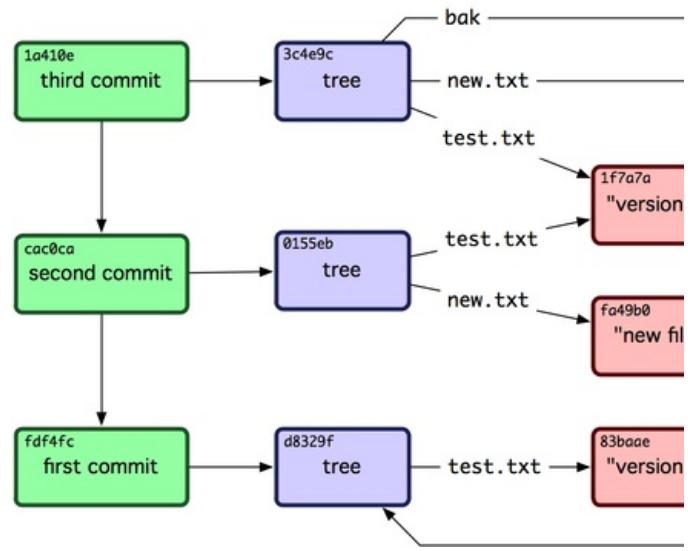


Figure 9-3. Git

Git Ruby blob (“what is up, doc?”)

irb R

```
$ irb
>> content = "what is up, doc?"
=> "what is up, doc?"
```

Git blob (null byte)

```
>> header = "blob #{content.length}\0"
=> "blob 16\000"
```

Git SHA-1 Ruby

require SHA1 digest

Digest::S

```
>> store = header + content
=> "blob 16\000what is up, doc?"
>> require 'digest/sha1'
=> true
>> sha1 = Digest::SHA1hexdigest(store)
```

```
=> "bd9dbf5aae1a3862dd1526723246b20206e5fc37"
```

Git zlib Ruby zlib

Zlib::Deflate.deflate()

```
>> require 'zlib'  
=> true  
>> zlib_content = Zlib::Deflate.deflate(store)  
=> "x\234K\312\3110R04c(\317H,Q\310,V(-\320QH\3110\266\000_\034\0235"
```

zlib (SHA-1 38) Ruby

FileUtils.mkdir_p()

write()

```
>> path = '.git/objects/' + sha1[0,2] + '/' + sha1[2,38]  
=> ".git/objects/bd/9dbf5aae1a3862dd1526723246b20206e5fc37"  
>> require 'fileutils'  
=> true  
>> FileUtils.mkdir_p(File.dirname(path))  
=> ".git/objects/bd"  
>> File.open(path, 'w') { |f| f.write zlib_content }  
=> 32
```

— blob Git — blob commit tree blob commit tree

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#), [Macedonian](#), [Polish](#), [Thai](#) and [Russian](#).

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

Git References

```
git log 1a410e      1a410e SHA-1 SHA-1
```

```
Git "references refs" .git/refs SHA-1
```

```
$ find .git/refs  
.git/refs  
.git/refs/heads  
.git/refs/tags  
$ find .git/refs -type f  
$
```

```
$ echo "1a410efbd13591db07496601ebc7a059dd55cfe9" > .git/refs/heads/master
```

Git SHA-1

```
$ git log --pretty=oneline master  
1a410efbd13591db07496601ebc7a059dd55cfe9 third commit  
cac0cab538b970a37ea1e769ccbde608743bc96d second commit  
fdf4fc3344e67ab068f836878b6c4951e3b15f3d first commit
```

```
Git update-ref
```

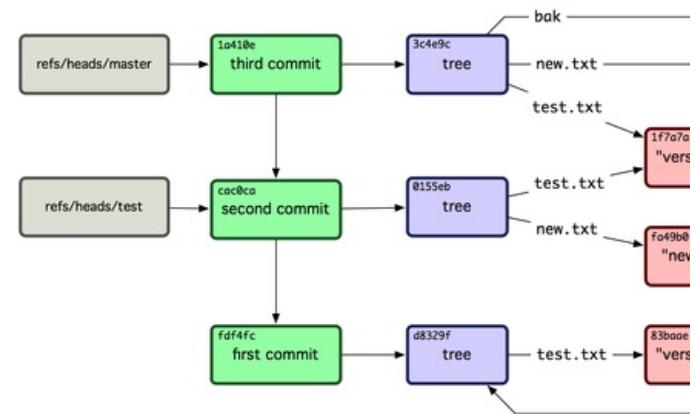
```
$ git update-ref refs/heads/master 1a410efbd13591db07496601ebc7a059dd55cfe9
```

Git HEAD

```
$ git update-ref refs/heads/test cac0ca
```

```
$ git log --pretty=oneline test  
cac0cab538b970a37ea1e769ccbde608743bc96d second commit  
fdf4fc3344e67ab068f836878b6c4951e3b15f3d first commit
```

Git 9-4



9-4. Git

```
git branch () Git update-ref SHA-1
```

HEAD

```
git branch () Git SHA-1 HEAD HEAD ----- SHA-1
```

```
$ cat .git/HEAD  
ref: refs/heads/master
```

```
git checkout testGit
```

```
$ cat .git/HEAD  
ref: refs/heads/test
```

git commit commit commit HEAD SHA-1

symbolic-ref HEAD

```
$ git symbolic-ref HEAD  
refs/heads/master
```

HEAD

```
$ git symbolic-ref HEAD refs/heads/test  
$ cat .git/HEAD  
ref: refs/heads/test
```

refs

```
$ git symbolic-ref HEAD test  
fatal: Refusing to point HEAD outside of refs/
```

Tags

Git Tag commit —— Tag commit tree—— commit

Tag annotated lightweight lightweight tag

```
$ git update-ref refs/tags/v1.0 cac0cab538b970a37ea1e769ccbde608743bc96d
```

lightweight tag —— annotated tag annotated tag
Git tag commit
reference annotated tag -a annotated tag

```
$ git tag -a v1.1 1a410efbd13591db07496601ebc7a059dd55cf9 -m 'test tag'
```

SHA-1

```
$ cat .git/refs/tags/v1.1  
9585191f37f7b0fb9444f35a9bf50de191beadc2
```

cat-file SHA-1

```
$ git cat-file -p 9585191f37f7b0fb9444f35a9bf50de191beadc2  
object 1a410efbd13591db07496601ebc7a059dd55cfe9  
type commit  
tag v1.1  
tagger Scott Chacon <schacon@gmail.com> Sat May 23 16:48:58 2009 -0700  
  
test tag
```

commit SHA-1 commit Git Git GPG blob

```
$ git cat-file blob junio-gpg-pub
```

Git . Linux kernel commit tag — tag tree initial tree

Remotes

```
reference remote reference remote Git remote  
origin remote master
```

```
$ git remote add origin git@github.com:schacon/simplegit-progit.git  
$ git push origin master  
Counting objects: 11, done.  
Compressing objects: 100% (5/5), done.  
Writing objects: 100% (7/7), 716 bytes, done.  
Total 7 (delta 2), reused 4 (delta 1)  
To git@github.com:schacon/simplegit-progit.git  
 a11bef0..ca82a6d master -> master
```

```
refs/remotes/origin/master      origin remote      master
```

```
$ cat .git/refs/remotes/origin/master  
ca82a6dff817ec66f44342007202690a93763949
```

Remote check out Git

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

Packfiles

test Git 11 — 4 blob3 tree3 commit tag

```
$ find .git/objects -type f
.git/objects/01/55eb4229851634a0f03eb265b69f5a2d56f341 # tree 2
.git/objects/1a/410efbd13591db07496601ebc7a059dd55cfe9 # commit 3
.git/objects/1f/7a7a472abf3dd9643fd615f6da379c4acb3e3a # test.txt v2
.git/objects/3c/4e9cd789d88d8d89c1073707c3585e41b0e614 # tree 3
.git/objects/83/baae61804e65cc73a7201a7252750c76066a30 # test.txt v1
.git/objects/95/85191f37f7b0fb9444f35a9bf50de191beadc2 # tag
.git/objects/ca/c0cab538b970a37ea1e769cbbde608743bc96d # commit 2
.git/objects/d6/70460b4b4aece5915caf5c68d12f560a9fe3e4 # 'test content'
.git/objects/d8/329fc1cc938780ffdd9f94e0d364e0ea74f579 # tree 1
.git/objects/fa/49b077972391ad58037050f2a75f74e3671e92 # new.txt
.git/objects/fd/f4fc3344e67ab068f836878b6c4951e3b15f3d # commit 1
```

Git zlib 925 Git Grit repo.rb — 12K

```
$ curl http://github.com/mojombo/grit/raw/master/lib/grit/repo.rb > repo.rb
$ git add repo.rb
$ git commit -m 'added repo.rb'
[master 484a592] added repo.rb
 3 files changed, 459 insertions(+), 2 deletions(-)
 delete mode 100644 bak/test.txt
 create mode 100644 repo.rb
 rewrite test.txt (100%)
```

tree repo.rb blob SHA-1

```
$ git cat-file -p master^{tree}
100644 blob fa49b077972391ad58037050f2a75f74e3671e92      new.txt
100644 blob 9bc1dc421dcd51b4ac296e3e5b6e2a99cf44391e      repo.rb
100644 blob e3f094f522629ae358806b17daf78246c27c007b      test.txt
```

git cat-file

```
$ git cat-file -s 9bc1dc421dcd51b4ac296e3e5b6e2a99cf44391e
12898
```

```
$ echo '# testing' >> repo.rb
$ git commit -am 'modified repo a bit'
[master ab1afef] modified repo a bit
 1 files changed, 1 insertions(+), 0 deletions(-)
```

commit tree

```
$ git cat-file -p master^{tree}
100644 blob fa49b077972391ad58037050f2a75f74e3671e92      new.txt
100644 blob 05408d195263d853f09dca71d55116663690c27c      repo.rb
100644 blob e3f094f522629ae358806b17daf78246c27c007b      test.txt
```

blob 400 Git

```
$ git cat-file -s 05408d195263d853f09dca71d55116663690c27c
12908
```

12K Git

Git Git (loose object) Git packfile git gc Git
Git

```
$ git gc
Counting objects: 17, done.
Delta compression using 2 threads.
Compressing objects: 100% (13/13), done.
Writing objects: 100% (17/17), done.
Total 17 (delta 1), reused 10 (delta 0)
```

objects

```
$ find .git/objects -type f
.git/objects/71/08f7ecb345ee9d0084193f147cdad4d2998293
.git/objects/d6/70460b4b4aece5915caf5c68d12f560a9fe3e4
.git/objects/info/packs
.git/objects/pack/pack-7a16e4488ae40c7d2bc56ea2bd43e25212a66c45.idx
.git/objects/pack/pack-7a16e4488ae40c7d2bc56ea2bd43e25212a66c45.pack
```

commit blob — “what is up, doc?” “test content” blob commit Git packfile

packfile packfile packfile gc 12K packfile 6K

Git Git packfile git verify-pack

```
$ git verify-pack -v \
.git/objects/pack/pack-7a16e4488ae40c7d2bc56ea2bd43e25212a66c45.idx
0155eb4229851634a0f03eb265b69f5a2d56f341 tree 71 76 5400
05408d195263d853f09dca71d55116663690c27c blob 12908 3478 874
09f01cea547666f58d6a8d809583841a7c6f0130 tree 106 107 5086
1a410efbd13591db07496601ebc7a059dd55cfe9 commit 225 151 322
1f7a7a472abf3dd9643fd615f6da379c4acb3e3a blob 10 19 5381
3c4e9cd789d88d8d89c1073707c3585e41b0e614 tree 101 105 5211
484a59275031909e19aadb7c92262719cfcdf19a commit 226 153 169
83baae61804e65cc73a7201a7252750c76066a30 blob 10 19 5362
9585191f37f7b0fb9444f35a9bf50de191beadc2 tag 136 127 5476
9bc1dc421dc51b4ac296e3e5b6e2a99cf44391e blob 7 18 5193 1
05408d195263d853f09dca71d55116663690c27c \
ab1afef80fac8e34258ff41fc1b867c702daa24b commit 232 157 12
cac0cab538b970a37ea1e769cbbde608743bc96d commit 226 154 473
d8329fc1cc938780ffdd9f94e0d364e0ea74f579 tree 36 46 5316
e3f094f522629ae358806b17daf78246c27c007b blob 1486 734 4352
f8f51d7d8a1760462eca26eebafde32087499533 tree 106 107 749
fa49b077972391ad58037050f2a75f74e3671e92 blob 9 18 856
fdf4fc3344e67ab068f836878b6c4951e3b15f3d commit 177 122 627
chain length = 1: 1 object
pack-7a16e4488ae40c7d2bc56ea2bd43e25212a66c45.pack: ok
```

, 9bc1d blob repo.rb blob 05408 blob 05408

—
Git git gc

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#), [Macedonian](#), [Polish](#), [Thai](#) and [Russian](#).

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and

The Refspec

```
$ git remote add origin git@github.com:schacon/simplegit-progit.git
```

.git/config (origin), URL Refspec:

```
[remote "origin"]
  url = git@github.com:schacon/simplegit-progit.git
  fetch = +refs/heads/*:refs/remotes/origin/*
```

Refspec + <src>:<dst> <src> <dst> + Git

refspec git remote add Git refs/heads/ refs/remotes/

```
$ git log origin/master
$ git log remotes/origin/master
$ git log refs/remotes/origin/master
```

Git refs/remotes/origin/master.

Git master fetch

```
fetch = +refs/heads/master:refs/remotes/origin/master
```

git fetch refspec refspec. master origin/mymaster

```
$ git fetch origin master:refs/remotes/origin/mymaster
```

refspec.

```
$ git fetch origin master:refs/remotes/origin/mymaster \
  topic:refs/remotes/origin/topic
From git@github.com:schacon/simplegit
 ! [rejected]      master      -> origin/mymaster  (non fast forward)
 * [new branch]    topic       -> origin/topic
```

master refspec +
refspec. master experiment

```
[remote "origin"]
  url = git@github.com:schacon/simplegit-progit.git
  fetch = +refs/heads/master:refs/remotes/origin/master
  fetch = +refs/heads/experiment:refs/remotes/origin/experiment
```

QA master QA

```
[remote "origin"]
  url = git@github.com:schacon/simplegit-progit.git
  fetch = +refs/heads/master:refs/remotes/origin/master
  fetch = +refs/heads/qa/*:refs/remotes/origin/qa/*
```

QA

Refspec

QA1 qa/ refspec

QA master qa/master

```
$ git push origin master:refs/heads/qa/master
```

Git git push origin push

```
[remote "origin"]
  url = git@github.com:schacon/simplegit-progit.git
  fetch = +refs/heads/*:refs/remotes/origin/*
  push = refs/heads/master:refs/heads/qa/master
```

git push origin master qa/master

refspec

```
$ git push origin :topic
```

refspec <src>:<dst>, <src> topic

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

Git HTTP file://, ssh://, git://

Git HTTP Git GETGit simplegit http-fetch

```
$ git clone http://github.com/schacon/simplegit-progit.git
```

```
1 info/refs      update-server-info HTTP      post-receive
```

```
=> GET info/refs  
ca82a6dff817ec66f44342007202690a93763949 refs/heads/master
```

SHAHEAD

```
=> GET HEAD  
ref: refs/heads/master
```

master info/refs ca82a6 commit

=> GET objects/ca/82a6dff817ec66f44342007202690a93763949
(179 bytes of binary data)

HTTP GET zlib commit

```
$ git cat-file -p ca82a6dff817ec66f44342007202690a93763949
tree cfda3bf379e4f8dba8717dee55aab78aef7f4daf
parent 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7
author Scott Chacon <schacon@gmail.com> 1205815931 -0700
committer Scott Chacon <schacon@gmail.com> 1240030591 -0700
```

changed the version number

cfda3b commit tree 085bb3

```
=> GET objects/08/5bb3bcb608e1e8451d4b2432f8ecbe6306e7e7  
(179 bytes of data)
```

commit tree

```
=> GET objects/cf/da3bf379e4f8dba8717dee55aab78aef7f4daf  
(404 - Not Found)
```

Oops - tree 404HTTP Git

```
=> GET objects/info/http-alternates  
(empty file)
```

objects/info/packs update

```
=> GET objects/info/packs  
P pack-816a9b2334da9953e530f27bcac22082a9f5b835.pack
```

```
=> GET objects/pack/pack-816a9b2334da9953e530f27bcac22082a9f5b835.idx  
(4k of binary data)
```

SHA

```
=> GET objects/pack/pack-816a9b2334da9953e530f27bcac22082a9f5b835.pack  
(13k of binary data)
```

tree commit HEAD

master Git

```
$ git clone http://github.com/schacon/simplegit-progit.git
Initialized empty Git repository in /private/tmp/simplegit-progit/.git/
got ca82a6dff817ec66f44342007202690a93763949
walk ca82a6dff817ec66f44342007202690a93763949
got 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7
Getting alternates list for http://github.com/schacon/simplegit-progit.git
Getting pack list for http://github.com/schacon/simplegit-progit.git
Getting index for pack 816a9b2334da9953e530f27bcac22082a9f5b835
Getting pack 816a9b2334da9953e530f27bcac22082a9f5b835
  which contains cfda3bf379e4f8dba8717dee55aab78aef7f4daf
walk 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7
walk a11bef06a3f659402fe7563abf99ad00de2209e6
```

HTTPGit

```
git push origin master,    origin SSHURL Git      send-pack
```

```
$ ssh -x git@github.com "git-receive-pack 'schacon/simplegit-progit.git'"  
005bca82a6df817ec66f4437202690a93763949 refs/heads/master report-status delete-re  
003e085bb3bcb608e1e84b2432f8ecbe6306e7e7 refs/heads/topic  
0000
```

```
git-receive-pack      master SHA1      report-status  
41005b91191003e62620000  
  
send-pack commit      receive-pack      master  
pack
```

'0'SHA-1 experiment '0'

Git SHASHA1()

```
000Aunpack ok
```

```
fetch-pack    upload-pack      fetch-pack    upload-pack  
upload-pack    receive-pack SSH Git 9418    fetch-|
```

```
003fgit-upload-pack schacon/simplegit-progit.git\0host=myserver.com\0
```

4 Git upload-pack

SSH fetch-pack

```
$ ssh -x git@github.com "git-upload-pack 'schacon/simplegit-progit.git'"
```

fetch-pack upload-pack

```
0088ca82a6dff817ec66f44342007202690a93763949 HEAD\0multi_ack thin-pack \  
side-band side-band-64k ofs-delta shallow no-progress include-tag  
003fcfa82a6dff817ec66f44342007202690a93763949 refs/heads/master  
003e085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7  
0000
```

receive-pack HEAD

fetch-pack “want” SHA “have” SHA “done”

```
0054want ca82a6dff817ec66f44342007202690a93763949 ofs-delta  
0032have 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7  
0000
```

0009done

multi_ack side-band

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

Git Git plumbing — porcelain Git Git

Git content-addressable VCS Git Git

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#).

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).
Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

Git

Git Linux Linux 19912002 2002 BitKeeper

2005 BitKeeper Linux BitKeeper Linux Linux Linus Torvalds

* * * * * Linux

2005 Git

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

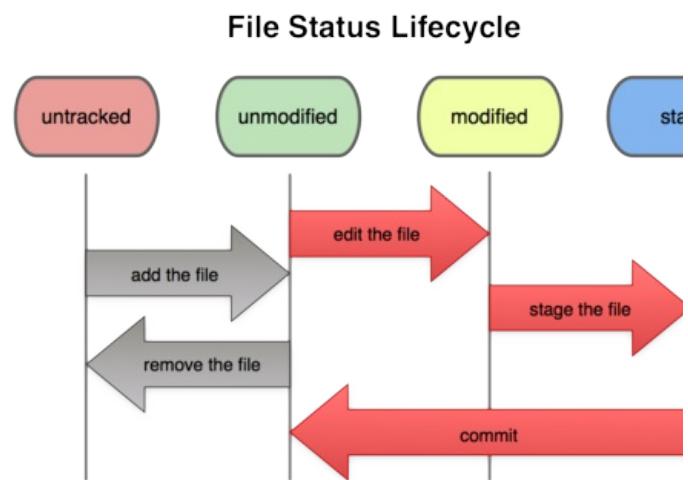
[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

Git

Git Git 2-1



2-1.

git status

```
$ git status
# On branch master
nothing to commit (working directory clean)
```

Git master

vim README

git status

```
$ vim README
$ git status
# On branch master
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       README
nothing added to commit but untracked files present (use "git add" to track)
```

“Untracked files”Git README

git add README

```
$ git add README
```

git status README

```
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       new file:   README
#
```

“Changes to be committed”

git init git add git

benchmarks.rb status

```
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       new file:   README
#
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#
```

```
#      modified: benchmarks.rb  
#
```

benchmarks.rb “Changed but not updated”

git a

```
git add benchmarks.rb      git status
```

```
$ git add benchmarks.rb  
$ git status  
# On branch master  
# Changes to be committed:  
#   (use "git reset HEAD <file>..." to unstage)  
#  
#       new file: README  
#       modified: benchmarks.rb  
#
```

benchmarks.rb

git status

```
$ vim benchmarks.rb  
$ git status  
# On branch master  
# Changes to be committed:  
#   (use "git reset HEAD <file>..." to unstage)  
#  
#       new file: README  
#       modified: benchmarks.rb  
#  
# Changed but not updated:  
#   (use "git add <file>..." to update what will be committed)  
#  
#       modified: benchmarks.rb  
#
```

benchmarks.rb Git git add

```
$ git add benchmarks.rb  
$ git status  
# On branch master  
# Changes to be committed:  
#   (use "git reset HEAD <file>..." to unstage)  
#  
#       new file: README  
#       modified: benchmarks.rb  
#
```

Git .gitignore

```
$ cat .gitignore
*[oa]
*~
```

Git .o .a Git

~ Emacs logtmp pid .gitignore

.gitignore

- Git
- glob * / * !

glob shell * [abc] a b c ? [0-9]

.gitignore

```
# - Git
*.a      # .a
!lib.a   # lib.a
/TODO    # TODO subdir/TODO
build/   # build/
doc/*.txt # doc/notes.txt doc/server/arch.txt
```

git status

git diff

git diff

git c

README benchmarks.rb

status

```
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       new file: README
```

```
# Changed but not updated:  
#   (use "git add <file>..." to update what will be committed)  
#  
#       modified:   benchmarks.rb  
#
```

git diff

```
$ git diff  
diff --git a/benchmarks.rb b/benchmarks.rb  
index 3cb747f..da65585 100644  
--- a/benchmarks.rb  
+++ b/benchmarks.rb  
@@ -36,6 +36,10 @@ def main  
    @commit.parents[0].parents[0].parents[0]  
end  
  
+    run_code(x, 'commits 1') do  
+        git.commits.size  
+    end  
  
+    run_code(x, 'commits 2') do  
+        log = git.commits('master', 15)  
+        log.size
```

git diff --cached Git 1.6.1

git diff --staged

```
$ git diff --cached  
diff --git a/README b/README  
new file mode 100644  
index 0000000..03902a1  
--- /dev/null  
+++ b/README2  
@@ -0,0 +1,5 @@  
+grit  
+ by Tom Preston-Werner, Chris Wanstrath  
+ http://github.com/mojombo/grit  
+  
+Grit is a Ruby library for extracting information from a Git repository
```

git diff

git diff

benchmarks.rb

git status

```
$ git add benchmarks.rb
```

```
$ echo '# test line' >> benchmarks.rb
$ git status
# On branch master
#
# Changes to be committed:
#
#       modified:   benchmarks.rb
#
# Changed but not updated:
#
#       modified:   benchmarks.rb
#
```

git diff

```
$ git diff
diff --git a/benchmarks.rb b/benchmarks.rb
index e445e28..86b2f7c 100644
--- a/benchmarks.rb
+++ b/benchmarks.rb
@@ -127,3 +127,4 @@ end
main()

##pp Grit::GitRuby.cache_client.stats
+# test line
and git diff --cached to see what you've staged so far:
$ git diff --cached
diff --git a/benchmarks.rb b/benchmarks.rb
index 3cb747f..e445e28 100644
--- a/benchmarks.rb
+++ b/benchmarks.rb
@@ -36,6 +36,10 @@ def main
    @commit.parents[0].parents[0]
end

+      run_code(x, 'commits 1') do
+        git.commits.size
+      end
+
+      run_code(x, 'commits 2') do
+        log = git.commits('master', 15)
+        log.size
```

git add

git status

git commit

```
$ git commit
```

shell \$EDITOR vim emacs

git config --g

Vim

```
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       new file:   README
#       modified:  benchmarks.rb
~
~
~
".git/COMMIT_EDITMSG" 10L, 283C
```

git status

-v Git

-m

```
$ git commit -m "Story 182: Fix benchmarks for speed"
[master]: created 463dc4f: "Fix benchmarks for speed"
 2 files changed, 3 insertions(+), 0 deletions(-)
 create mode 100644 README
```

master SHA-1

463dc4f

Git

git commit -a Git

git add

```
$ git status
# On branch master
#
# Changed but not updated:
#
#       modified:  benchmarks.rb
#
$ git commit -a -m 'added new benchmarks'
[master 83e38c7] added new benchmarks
 1 files changed, 5 insertions(+), 0 deletions(-)
```

```
git add benchmarks.rb
```

Git

```
git rm
```

```
git status "Changed but not updated" __
```

```
$ rm grit.gemspec
$ git status
# On branch master
#
# Changed but not updated:
#   (use "git add/rm <file>..." to update what will be committed)
#
#       deleted:    grit.gemspec
#
```

```
git rm
```

```
$ git rm grit.gemspec
rm 'grit.gemspec'
$ git status
# On branch master
#
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       deleted:    grit.gemspec
#
```

-f force

Git

.a

.gitignore

--cach

```
$ git rm --cached readme.txt
```

glob

```
$ git rm log/*.log
```

* \ Git shell shell

1

```
$ git rm \*~
```

~

VCS Git Git Git

Git mv Git

```
$ git mv file_from file_to
```

```
$ git mv README.txt README
$ git status
# On branch master
# Your branch is ahead of 'origin/master' by 1 commit.
#
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       renamed:    README.txt -> README
#
```

git mv

```
$ mv README.txt README
$ git rm README.txt
$ git add README
```

Git git mv

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and

Git

git remote origin Git

```
$ git clone git://github.com/schacon/ticgit.git
Initialized empty Git repository in /private/tmp/ticgit/.git/
remote: Counting objects: 595, done.
remote: Compressing objects: 100% (269/269), done.
remote: Total 595 (delta 255), reused 589 (delta 253)
Receiving objects: 100% (595/595), 73.31 KiB | 1 KiB/s, done.
Resolving deltas: 100% (255/255), done.
$ cd ticgit
$ git remote
origin
```

-v –verbose

```
$ git remote -v
origin  git://github.com/schacon/ticgit.git
```

Grit

```
$ cd grit
$ git remote -v
bakkdoor  git://github.com/bakkdoor/grit.git
cho45     git://github.com/cho45/grit.git
defunkt   git://github.com/defunkt/grit.git
koke      git://github.com/koke/grit.git
origin    git@github.com:mojombo/grit.git
```

origin SSH URL

```
git remote add [shortname] [url]
```

```
$ git remote  
origin  
$ git remote add pb git://github.com/paulboone/ticgit.git  
$ git remote -v  
origin  git://github.com/schacon/ticgit.git  
pb      git://github.com/paulboone/ticgit.git
```

pb Paul

git fetch pb

```
$ git fetch pb  
remote: Counting objects: 58, done.  
remote: Compressing objects: 100% (41/41), done.  
remote: Total 44 (delta 24), reused 1 (delta 0)  
Unpacking objects: 100% (44/44), done.  
From git://github.com/paulboone/ticgit  
* [new branch]      master    -> pb/master  
* [new branch]      ticgit     -> pb/ticgit
```

Paul master

pb/master

```
$ git fetch [remote-name]
```

origin

git fetch origin fetch fetch

git pull

git clone master master ma

pull

git push [remote-name] [branch-name] master ori

```
$ git push origin master
```

```
git remote show [remote-name]      origin
```

```
$ git remote show origin
* remote origin
  URL: git://github.com/schacon/ticgit.git
  Remote branch merged with 'git pull' while on branch master
    master
  Tracked remote branches
    master
    ticgit
```

```
master      git pull
```

```
git remote show
```

```
$ git remote show origin
* remote origin
  URL: git@github.com:defunkt/github.git
  Remote branch merged with 'git pull' while on branch issues
    issues
  Remote branch merged with 'git pull' while on branch master
    master
  New remote branches (next fetch will store in remotes/origin)
    caching
  Stale tracking branches (use 'git remote prune')
    libwalker
    walker2
  Tracked remote branches
    acl
    apiv2
    dashboard2
    issues
    master
    postgres
  Local branch pushed with 'git push'
    master:master
```

```
git push caching Stale tracking branches  
issues master
```

Git git remote rename pb paul

```
$ git remote rename pb paul  
$ git remote  
origin  
paul
```

pb/master paul/master

git remote rm

```
$ git remote rm paul  
$ git remote  
origin
```

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).
Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .
All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).
Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

VCS Git v1.0

git tag

```
$ git tag  
v0.1  
v1.3
```

Git 240 1.4.2

```
$ git tag -l 'v1.4.2.*'  
v1.4.2.1  
v1.4.2.2  
v1.4.2.3  
v1.4.2.4
```

Git lightweight annotated GNU Privacy Guard (GPG)

-a annotated

```
$ git tag -a v1.4 -m 'my version 1.4'  
$ git tag  
v0.1
```

```
v1.3  
v1.4
```

-m Git Git

```
git show
```

```
$ git show v1.4  
tag v1.4  
Tagger: Scott Chacon <schacon@gee-mail.com>  
Date: Mon Feb 9 14:45:11 2009 -0800  
  
my version 1.4  
commit 15027957951b64cf874c3557a0f3547bd83b3ff6  
Merge: 4a447f7... a6b4c97...  
Author: Scott Chacon <schacon@gee-mail.com>  
Date: Sun Feb 8 19:02:46 2009 -0800  
  
Merge branch 'experiment'
```

GPG -a -s Signed

```
$ git tag -s v1.5 -m 'my signed 1.5 tag'  
You need a passphrase to unlock the secret key for  
user: "Scott Chacon <schacon@gee-mail.com>"  
1024-bit DSA key, ID F721C45A, created 2009-02-09
```

```
git show GPG
```

```
$ git show v1.5  
tag v1.5  
Tagger: Scott Chacon <schacon@gee-mail.com>  
Date: Mon Feb 9 15:22:20 2009 -0800  
  
my signed 1.5 tag  
-----BEGIN PGP SIGNATURE-----  
Version: GnuPG v1.4.8 (Darwin)  
  
iEYEABECAAYFAkmQuriIACgkQON3DxfchxFr5cACeIMN+ZxLKggJQf0QYiQBwgySN  
Ki0An2JeAVUCAiJ70x6ZEtK+NvZAj82/  
=WryJ
```

```
-----END PGP SIGNATURE-----  
commit 15027957951b64cf874c3557a0f3547bd83b3ff6  
Merge: 4a447f7... a6b4c97...  
Author: Scott Chacon <schacon@gee-mail.com>  
Date: Sun Feb 8 19:02:46 2009 -0800  
  
Merge branch 'experiment'
```

-a-S -m

```
$ git tag v1.4-lw  
$ git tag  
v0.1  
v1.3  
v1.4  
v1.4-lw  
v1.5
```

git show

```
$ git show v1.4-lw  
commit 15027957951b64cf874c3557a0f3547bd83b3ff6  
Merge: 4a447f7... a6b4c97...  
Author: Scott Chacon <schacon@gee-mail.com>  
Date: Sun Feb 8 19:02:46 2009 -0800  
  
Merge branch 'experiment'
```

git tag -v [tag-name] verify GPG keyring

```
$ git tag -v v1.4.2.1  
object 883653babd8ee7ea23e6a5c392bb739348b1eb61  
type commit  
tag v1.4.2.1  
tagger Junio C Hamano <junkio@cox.net> 1158138501 -0700  
  
GIT 1.4.2.1  
  
Minor fixes since 1.4.2, including git-mv and git-http with alternates.
```

```
gpg: Signature made Wed Sep 13 02:08:25 2006 PDT using DSA key ID F3119B9A
gpg: Good signature from "Junio C Hamano <junkio@cox.net>"
gpg:                               aka "[jpeg image of size 1513]"
Primary key fingerprint: 3565 2A26 2040 E066 C9A7 4A7D C0C6 D9A4 F311 9B9A
```

```
gpg: Signature made Wed Sep 13 02:08:25 2006 PDT using DSA key ID F3119B9A
gpg: Can't check signature: public key not found
error: could not verify the tag 'v1.4.2.1'
```

```
$ git log --pretty=oneline
15027957951b64cf874c3557a0f3547bd83b3ff6 Merge branch 'experiment'
a6b4c97498bd301d84096da251c98a07c7723e65 beginning write support
0d52aaab4479697da7686c15f77a3d64d9165190 one more thing
6d52a271eda8725415634dd79daabbc4d9b6008e Merge branch 'experiment'
0b7434d86859cc7b8c3d5e1dddfed66ff742fcbe added a commit function
4682c3261057305bdd616e23b64b0857d832627b added a todo file
166ae0c4d3f420721acbb115cc33848dfcc2121a started write support
9fce02d0ae598e95dc970b74767f19372d61af8 updated rakefile
964f16d36dfccde844893cac5b347e7b3d44abbc commit the todo
8a5cbc430f1a9c3d00faaeffd07798508422908a updated readme
```

“updated rakefile” v1.2

```
$ git tag -a v1.2 9fce02
```

```
$ git tag
v0.1
v1.2
v1.3
v1.4
v1.4-lw
v1.5

$ git show v1.2
tag v1.2
Tagger: Scott Chacon <schacon@gee-mail.com>
Date:   Mon Feb 9 15:32:16 2009 -0800
```

```
version 1.2
commit 9fcceb02d0ae598e95dc970b74767f19372d61af8
Author: Magnus Chacon <mchacon@gee-mail.com>
Date:   Sun Apr 27 20:43:35 2008 -0700

        updated rakefile
...
```

`git push` `git push origin [tagname]`

```
$ git push origin v1.5
Counting objects: 50, done.
Compressing objects: 100% (38/38), done.
Writing objects: 100% (44/44), 4.56 KiB, done.
Total 44 (delta 18), reused 8 (delta 1)
To git@github.com:schacon/simplegit.git
 * [new tag]      v1.5 -> v1.5
```

`--tags`

```
$ git push origin --tags
Counting objects: 50, done.
Compressing objects: 100% (38/38), done.
Writing objects: 100% (44/44), 4.56 KiB, done.
Total 44 (delta 18), reused 8 (delta 1)
To git@github.com:schacon/simplegit.git
 * [new tag]      v0.1 -> v0.1
 * [new tag]      v1.2 -> v1.2
 * [new tag]      v1.4 -> v1.4
 * [new tag]      v1.4-1w -> v1.4-1w
 * [new tag]      v1.5 -> v1.5
```

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).
Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .
All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).
Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

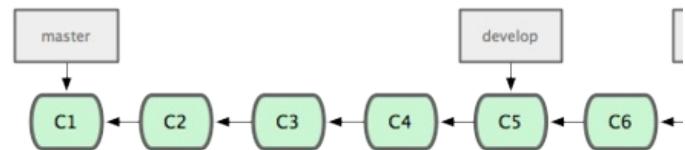
To help translate the book, please fork the book [at GitHub](#) and

Git

Git master develop next ——

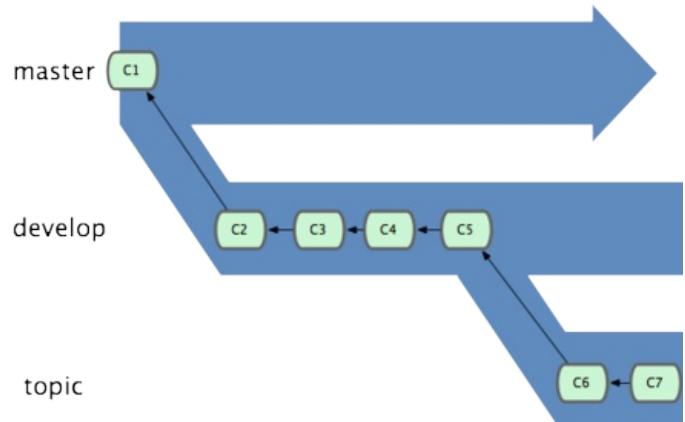
iss53

commit 3-18



3-18.

commit 3-19



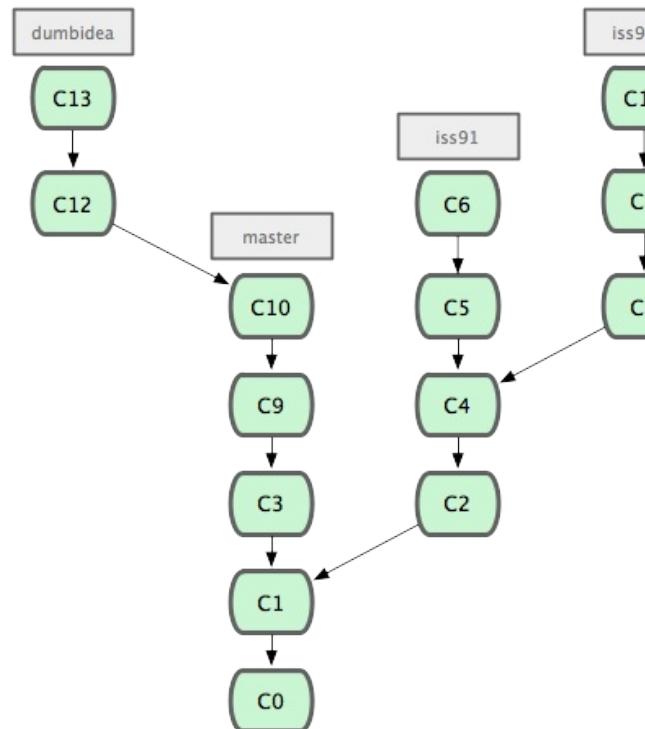
3-19.

proposed proposed updates next master

Topic Git

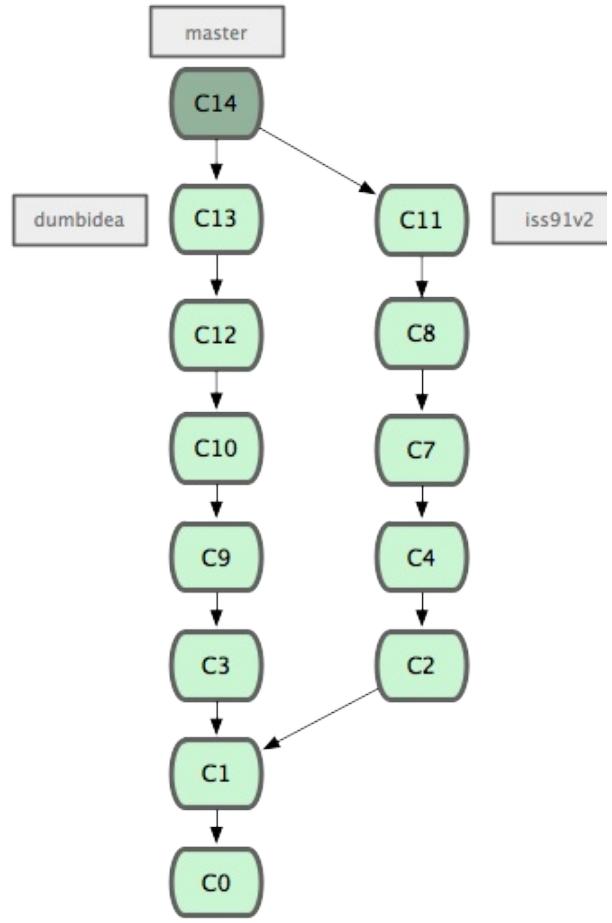
iss53 hotfix —

3-20 master C1 iss91 91 C6 C4 iss
C10 iss91v2 C11 master C10 dumbidea



3-20.

iss91v2 dumbidea iss91 C5 C6 3-2



3-21. dumbidea iss91v2

Git —

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#).

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).
Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

SSH authorized_keys Ubuntu Linux 'git' .9

```
$ sudo adduser git  
$ su git  
$ cd  
$ mkdir .ssh
```

SSH authorized_keys e-mail

```
$ cat /tmp/id_rsa.john.pub  
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCB007n/ww+ouN4gSLKssMxXnB0vf9LGt4L  
ojG6rs6hPB09j9R/T17/x4lhJA0F3FR1rP6kYBRsWj2aThGw6HXLm9/5zytK6Ztg3RPKK+4k  
Yjh6541NYsnEAZuXz0jTTyAUfrtu3Z5E003C4ox0j6H0rfIF1kKI9MAQLMdppGW1GYEIgS9Ez  
Sdfd8AcCIicTDWbqLAcU4UpkaX8KyG1LwsNuuGztobF8m72ALC/nLF6JLtPofwFBlgc+myiv  
07TCUSBdLQlgMV0Fq1I2uPWQ0kOWQAHukE0mfjy2jctxSDBQ220ymjaNsHT4kgtZg2AYYgPq  
dAv8JggJICUvax2T9va5 gsg-keypair
```

authorized_keys

```
$ cat /tmp/id_rsa.john.pub >> ~/.ssh/authorized_keys  
$ cat /tmp/id_rsa.josie.pub >> ~/.ssh/authorized_keys  
$ cat /tmp/id_rsa.jessica.pub >> ~/.ssh/authorized_keys
```

--bare git init

```
$ cd /opt/git  
$ mkdir project.git  
$ cd project.git  
$ git --bare init
```

JoinJosie Jessica shell gitserver gi
gitserver

```
# John
$ cd myproject
$ git init
$ git add .
$ git commit -m 'initial commit'
$ git remote add origin git@gitserver:/opt/git/project.git
$ git push origin master
```

```
$ git clone git@gitserver:/opt/git/project.git
$ vim README
$ git commit -am 'fix for the README file'
$ git push origin master
```

Git

Git	git-shell	git	Git	git	shell	shell	shell
csh			/etc/passwd				

```
$ sudo vim /etc/passwd
```

```
git:x:1000:1000::/home/git:/bin/sh
```

bin/sh	/usr/bin/git-shell	which git-shell
--------	--------------------	-----------------

```
git:x:1000:1000::/home/git:/usr/bin/git-shell
```

git SSH Git shell

```
$ ssh git@gitserver
fatal: What do you think I am? A shell? shell)
Connection to gitserver closed. gitserver
```

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

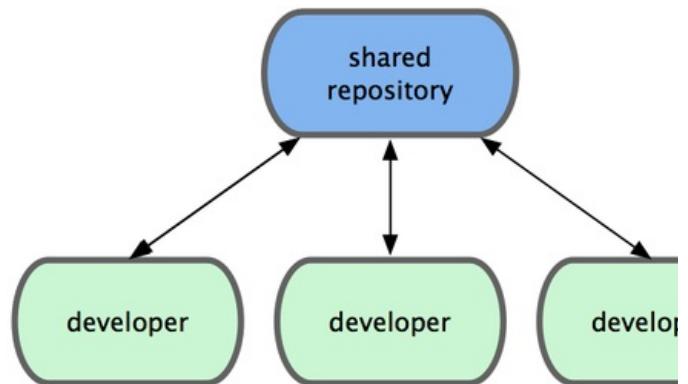
[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and

CVCS Git Git Git

5-1



5-1.

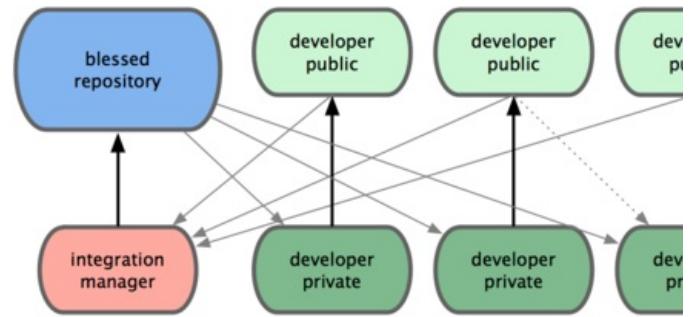
Git Subversion CVCS

Git fast-forward

Git blessed repository developer public integration manager

5-2

1. blessed repository
2. developer public
3. integration manger
4. blessed repository

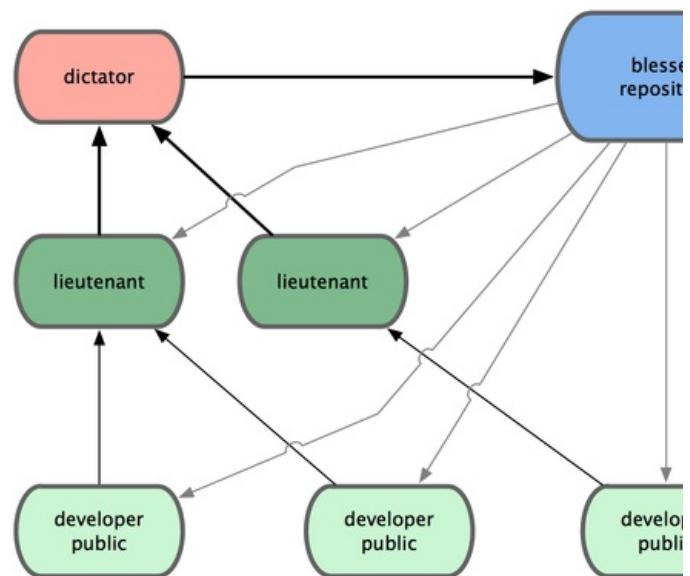


5-2.

GitHub fork

Linux lieutenantdictator 5-3

1. dictator master
2. lieutenant master
3. dictator master master
4. dictator master blessed repository



5-3.

Git

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

format-patch

ruby_client Git

sc/ruby_cl

```
$ git branch sc/ruby_client master
```

checkout -b

```
$ git checkout -b sc/ruby_client master
```

git apply git am

apply

git diff Unix diff git apply /tmp/patch-ruby-cl

```
$ git apply /tmp/patch-ruby-client.patch
```

patch -p1 git diff patch
patch git apply patch

```
git apply --check
```

```
$ git apply --check 0001-seeing-if-this-helps-the-gem.patch
error: patch failed: ticgit.gemspec:1
error: ticgit.gemspec: patch does not apply
```

shell

am

```
Git      format-patch          format-patch      diff
```

```
format-patch      git am      git am mbox From
```

```
From 330090432754092d704da8e76ca5c198e71a8 Mon Sep 17 00:00:00 2001
From: Jessica Smith <jessica@example.com>
Date: Sun, 6 Apr 2008 10:17:23 -0700
Subject: [PATCH 1/2] add limit to log function
```

```
Limit log functionality to the first 20
```

```
format-patch mbox      git send-email      git am mk
```

```
format-patch Request Ticket      git am
```

```
$ git am 0001-limit-log-function.patch
Applying: add limit to log function
```

From Date Subject mbox

```
$ git log --pretty=fuller -1
commit 6c5e70b984a60b3cecd395edd5b48a7575bf58e0
Author: Jessica Smith <jessica@example.com>
AuthorDate: Sun Apr 6 10:17:23 2008 -0700
Commit: Scott Chacon <schacon@gmail.com>
CommitDate: Thu Apr 9 09:19:06 2009 -0700
```

```
add limit to log function
```

```
Limit log functionality to the first 20
```

Commit Author

git am

```
$ git am 0001-seeing-if-this-helps-the-gem.patch
Applying: seeing if this helps the gem
error: patch failed: ticgit.gemspec:1
error: ticgit.gemspec: patch does not apply
Patch failed at 0001.
When you have resolved this problem run "git am --resolved".
If you would prefer to skip this patch, instead run "git am --skip".
To restore the original branch and stop patching run "git am --abort".
```

Git git am --resolved

```
$ (fix the file)
$ git add ticgit.gemspec
$ git am --resolved
Applying: seeing if this helps the gem
```

Git -3

```
$ git am -3 0001-seeing-if-this-helps-the-gem.patch
Applying: seeing if this helps the gem
error: patch failed: ticgit.gemspec:1
error: ticgit.gemspec: patch does not apply
Using index info to reconstruct a base tree...
Falling back to patching base and 3-way merge...
No changes -- Patch already applied.
```

-3

mbox am -i

```
$ git am -3 -i mbox
Commit Body is:
-----
seeing if this helps the gem
-----
Apply? [y]es/[n]o/[e]dit/[v]iew patch/[a]ccept all
```

Git

Jessica ruby-client

```
$ git remote add jessica git://github.com/jessica/myproject.git
$ git fetch jessica
$ git checkout -b rubyclient jessica/ruby-client
```

-3

git pull

```
$ git pull git://github.com/onetimeguy/project.git
From git://github.com/onetimeguy/project
 * branch            HEAD      -> FETCH_HEAD
Merge made by recursive.
```

contrib --not master

```
$ git log contrib --not master
commit 5b6235bd297351589efc4d73316f0a68d484f118
Author: Scott Chacon <schacon@gmail.com>
Date:   Fri Oct 24 09:53:59 2008 -0700
```

```
seeing if this helps the gem

commit 7482e0d16d04bea79d0dba8988cc78df655f16a0
Author: Scott Chacon <schacon@gmail.com>
Date:   Mon Oct 22 19:38:36 2008 -0700

updated the gemspec to hopefully work better
```

```
git log      -p
```

```
$ git diff master
```

```
master      diff      master
master      master
master
```

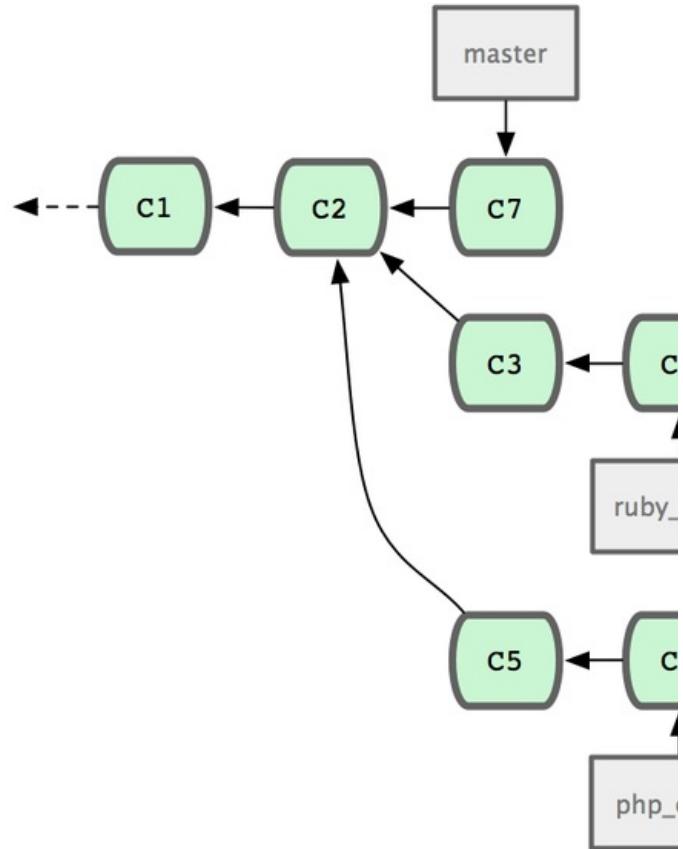
```
$ git merge-base contrib master
36c7dba2c95e6bbb78dfa822519ecfec6e1ca649
$ git diff 36c7db
```

```
Git      ...      diff      ...
```

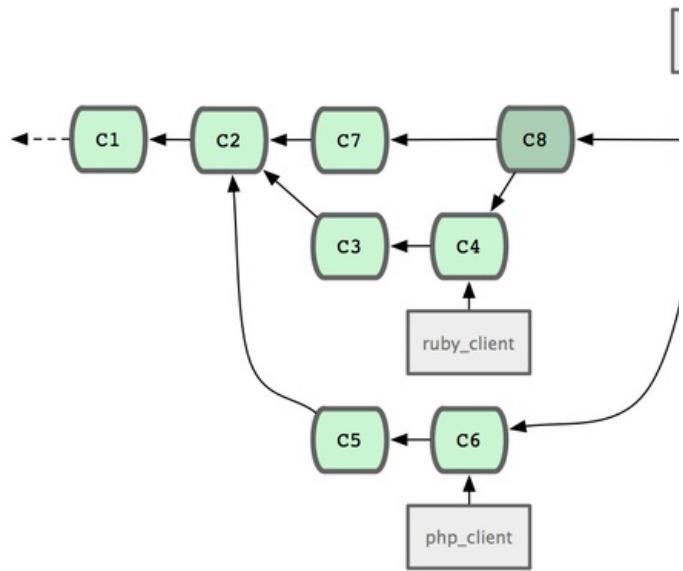
```
$ git diff master...contrib
```

master
php_client 5-20

ruby_client php_client !

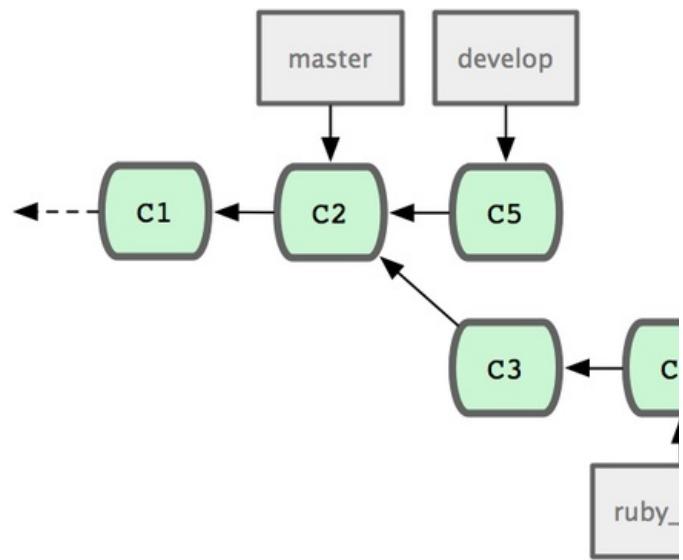


5-19.

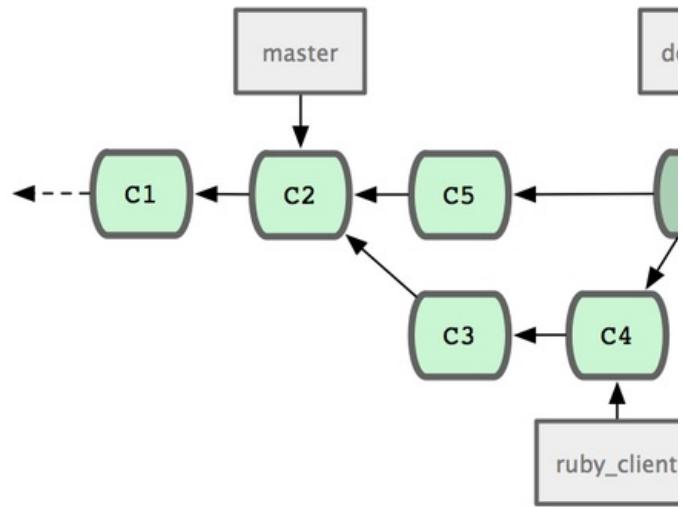


5-20.

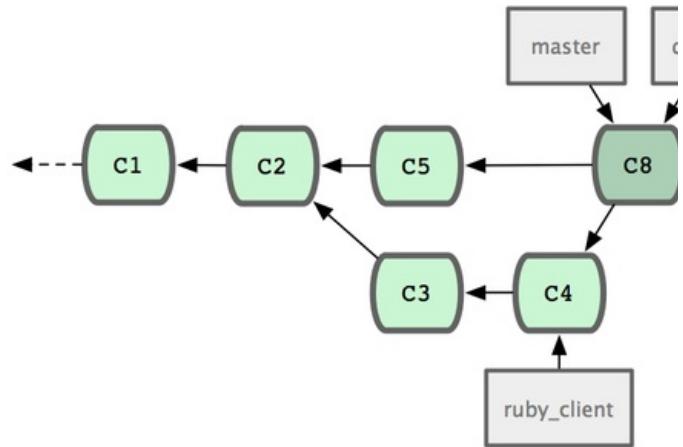
master	develop	5-21	ruby_client	develop	5-22	C
master	5-23		C8			



5-21.



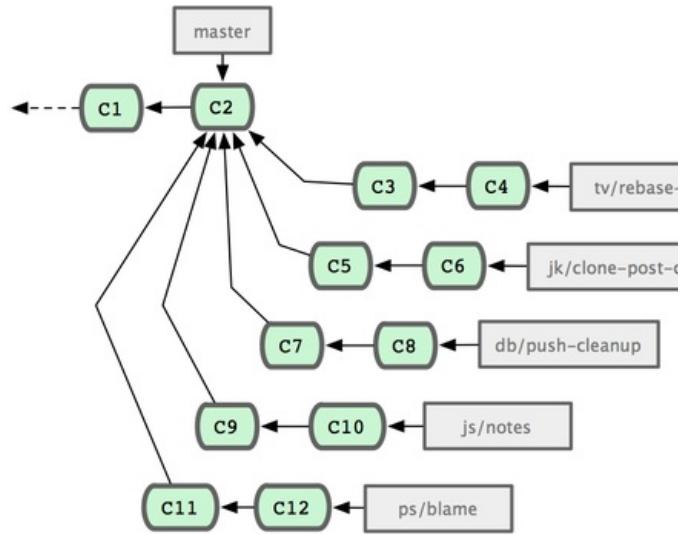
5-22.



5-23.

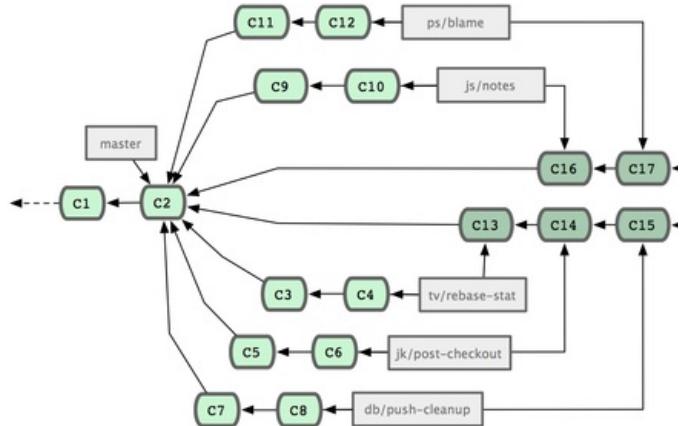
develop

Git	master	next	pu	pu	proposed updates	
maintenance	5-24					



5-24.

pu master next master master next pu 5-25



5-25.

master Git maint Git

cherry-pick

develop master
commits 5-26

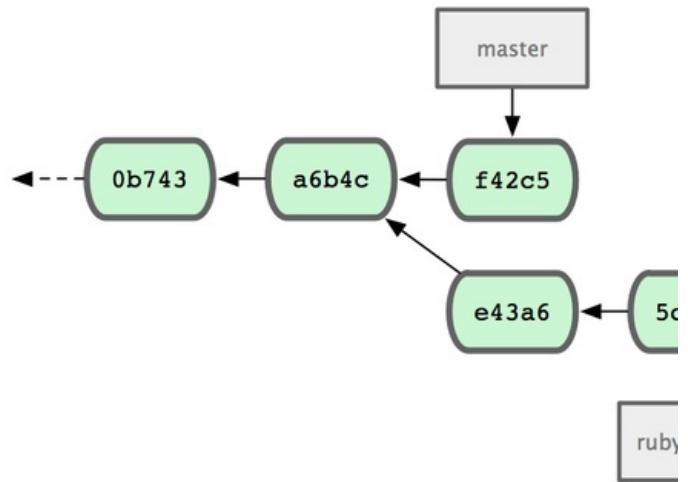


Figure 5-26. cherry-pick

e43a6

```
$ git cherry-pick e43a6fd3e94888d76779ad79fb568ed180e5fcdf
Finished one cherry-pick.
[master]: created a0a41a9: "More friendly message when locking the index fails."
 3 files changed, 17 insertions(+), 3 deletions(-)
```

e43a6SHA-1 5-27.

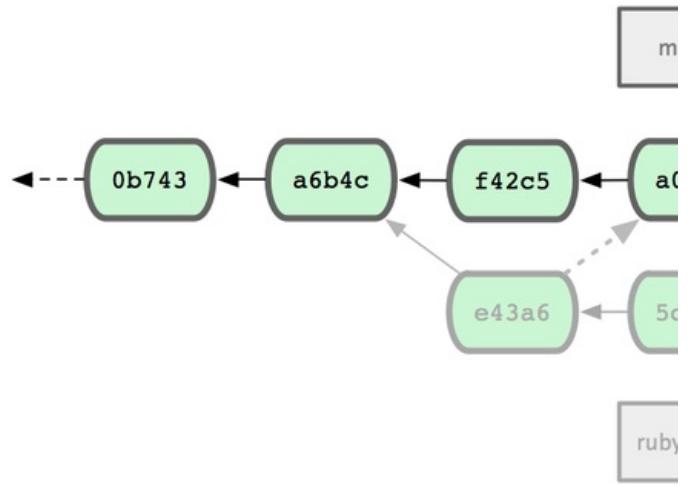


Figure 5-27. cherry-pick

commit

```
$ git tag -s v1.5 -m 'my signed 1.5 tag'  
You need a passphrase to unlock the secret key for  
user: "Scott Chacon <schacon@gmail.com>"  
1024-bit DSA key, ID F721C45A, created 2009-02-09
```

PGPpublic keyGitkeyblobGit

```
$ gpg --list-keys  
/Users/schacon/.gnupg/pubring.gpg  
-----  
pub 1024D/F721C45A 2009-02-09 [expires: 2010-02-09]  
uid Scott Chacon <schacon@gmail.com>  
sub 2048g/45D02282 2009-02-09 [expires: 2010-02-09]
```

key git hash-objectblobGitblobSHA-1

```
$ gpg -a --export F721C45A | git hash-object -w --stdin  
659ef797d181633c87ec71ac3f9ba29fe5775b92
```

GitkeySHA-1key

```
$ git tag -a maintainer-pgp-pub 659ef797d181633c87ec71ac3f9ba29fe5775b92
```

git push --tags maintainer-pgp-pubkey

```
$ git show maintainer-pgp-pub | gpg --import
```

key git show <tag>

Git'v123' git describeGitSHA-1of the commit you're describing

```
$ git describe master  
v1.6.2-rc1-20-g8c5b85c
```

Git git --version describe

git describe -a-s git describe checkoutshowSHA-1SHA/
git describe

Git git archive

```
$ git archive master --prefix='project/' | gzip > `git describe master`.tar.gz  
$ ls *.tar.gz  
v1.6.2-rc1-20-g8c5b85c.tar.gz
```

zip git archive--format=zip

```
$ git archive master --prefix='project/' --format=zip > `git describe master`.zip
```

tar.gzzipe-mail

git shortlogchangelogbug;v1.0.1

```
$ git shortlog --no-merges master --not v1.0.1  
Chris Wanstrath (8):  
    Add support for annotated tags to Grit::Tag  
    Add packed-refs annotated tag support.  
    Add Grit::Commit#to_patch  
    Update version and History.txt  
    Remove stray `puts`  
    Make ls_tree ignore nils  
  
Tom Preston-Werner (4):
```

fix dates in history
dynamic version method
Version bump to 1.0.2
Regenerated gemspec for version 1.0.2

v1.0.1e-mail

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).
Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

Git

```
git add -i --interactive
```

```
$ git add -i
      staged      unstaged path
 1: unchanged      +0/-1 TODO
 2: unchanged      +1/-1 index.html
 3: unchanged      +5/-1 lib/simplegit.rb

*** Commands ***
 1: status      2: update      3: revert      4: add untracked
 5: patch       6: diff        7: quit        8: help
What now>
```

—git status

What now>2u

```
What now> 2
      staged      unstaged path
 1: unchanged      +0/-1 TODO
 2: unchanged      +1/-1 index.html
 3: unchanged      +5/-1 lib/simplegit.rb
Update>>
```

TODOindex.html

```
Update>> 1,2
      staged      unstaged path
 * 1: unchanged      +0/-1 TODO
 * 2: unchanged      +1/-1 index.html
```

```
3: unchanged      +5/-1 lib/simplegit.rb  
Update>>
```

* update>>Git

```
Update>>  
updated 2 paths  
  
*** Commands ***  
1: status      2: update      3: revert      4: add untracked  
5: patch       6: diff        7: quit        8: help  
What now> 1  
          staged      unstaged path  
1:           +0/-1      nothing TODO  
2:           +1/-1      nothing index.html  
3: unchanged      +5/-1 lib/simplegit.rb
```

TODOindex.htmlsimplegit.rbTODO 3rrevert

```
*** Commands ***  
1: status      2: update      3: revert      4: add untracked  
5: patch       6: diff        7: quit        8: help  
What now> 3  
          staged      unstaged path  
1:           +0/-1      nothing TODO  
2:           +1/-1      nothing index.html  
3: unchanged      +5/-1 lib/simplegit.rb  
Revert>> 1  
          staged      unstaged path  
* 1:           +0/-1      nothing TODO  
2:           +1/-1      nothing index.html  
3: unchanged      +5/-1 lib/simplegit.rb  
Revert>> [enter]  
reverted one path
```

GitTODO

```
*** Commands ***  
1: status      2: update      3: revert      4: add untracked  
5: patch       6: diff        7: quit        8: help  
What now> 1  
          staged      unstaged path  
1: unchanged      +0/-1 TODO  
2:           +1/-1      nothing index.html  
3: unchanged      +5/-1 lib/simplegit.rb
```

6ddiff git diff --cached

```

*** Commands ***
1: status      2: update      3: revert      4: add untracked
5: patch       6: diff        7: quit        8: help
What now> 6
          staged      unstaged path
1:           +1/-1      nothing index.html
Review diff>> 1
diff --git a/index.html b/index.html
index 4d07108..4335f49 100644
--- a/index.html
+++ b/index.html
@@ -16,7 +16,7 @@ Date Finder

<p id="out">...</p>

-<div id="footer">contact : support@github.com</div>
+<div id="footer">contact : email.support@github.com</div>

<script type="text/javascript">

```

Gitsimplegit.rbGit

5ppatchGit

```

diff --git a/lib/simplegit.rb b/lib/simplegit.rb
index dd5ecc4..57399e0 100644
--- a/lib/simplegit.rb
+++ b/lib/simplegit.rb
@@ -22,7 +22,7 @@ class SimpleGit
end

def log(treeish = 'master')
-  command("git log -n 25 #{treeish}")
+  command("git log -n 30 #{treeish}")
end

def blame(path)
Stage this hunk [y,n,a,d/,j,J,g,e,?]?

```

?

```

Stage this hunk [y,n,a,d/,j,J,g,e,?]?
y - stage this hunk
n - do not stage this hunk
a - stage this and all the remaining hunks in the file
d - do not stage this hunk nor any of the remaining hunks in the file
g - select a hunk to go to
/ - search for a hunk matching the given regex

```

```
j - leave this hunk undecided, see next undecided hunk  
J - leave this hunk undecided, see next hunk  
k - leave this hunk undecided, see previous undecided hunk  
K - leave this hunk undecided, see previous hunk  
s - split the current hunk into smaller hunks  
e - manually edit the current hunk  
? - print help
```

yn

```
What now> 1
      staged      unstaged  path
 1:   unchanged      +0/-1 TODO
 2:           +1/-1      nothing index.html
 3:           +1/-1      +4/-0 lib/simplegit.rb
```

simplegit.rb git commit

——git add -pgit add --patch

prev | next

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#), [Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).
Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

Stashing

```
git stash
```

```
~~~~~
```

```
git status
```

```
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       modified:   index.html
#
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#
#       modified:   lib/simplegit.rb
#
```

```
git stash
```

```
$ git stash
Saved working directory and index state \
  "WIP on master: 049d078 added the index file"
HEAD is now at 049d078 added the index file
(To restore them type "git stash apply")
```

```
$ git status
# On branch master
nothing to commit (working directory clean)
```

git stash list

```
$ git stash list
stash@{0}: WIP on master: 049d078 added the index file
stash@{1}: WIP on master: c264051... Revert "added file_size"
stash@{2}: WIP on master: 21d80a5... added number to log
```

stash

git stash apply

git stash apply

```
$ git stash apply
# On branch master
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#
#       modified:   index.html
#       modified:   lib/simplegit.rb
#
```

Git —— Git

git stash apply --index

```
$ git stash apply --index
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       modified:   index.html
#
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#
#       modified:   lib/simplegit.rb
#
```

apply ——

git stash drop

```
$ git stash list
stash@{0}: WIP on master: 049d078 added the index file
stash@{1}: WIP on master: c264051... Revert "added file_size"
stash@{2}: WIP on master: 21d80a5... added number to log
$ git stash drop stash@{0}
Dropped stash@{0} (364e91f3f268f0900bc3ee613f9f733e82aaed43)
```

```
git stash pop
```

```
git stash branch
```

```
$ git stash branch testchanges
Switched to a new branch "testchanges"
# On branch testchanges
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       modified:   index.html
#
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#
#       modified:   lib/simplegit.rb
#
Dropped refs/stash@{0} (f0dfc4d5dc332d1cee34a634182e168c4efc3359)
```

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).
Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

AtomAtomCPAN installRuby gem

Git Git Git

Rack Ruby web

git submodule add

```
$ git submodule add git://github.com/chneukirchen/rack.git rack
Initialized empty Git repository in /opt/subtest/rack/.git/
remote: Counting objects: 3181, done.
remote: Compressing objects: 100% (1534/1534), done.
remote: Total 3181 (delta 1951), reused 2623 (delta 1603)
Receiving objects: 100% (3181/3181), 675.42 KiB | 422 KiB/s, done.
Resolving deltas: 100% (1951/1951), done.
```

rack Rack

git status

```
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       new file:   .gitmodules
#       new file:   rack
#
```

.gitmodules URL

```
$ cat .gitmodules
[submodule "rack"]
    path = rack
    url = git://github.com/chneukirchen/rack.git
```

```
.gitignore
```

```
git status rack git diff
```

```
$ git diff --cached rack
diff --git a/rack b/rack
new file mode 160000
index 0000000..08d709f
--- /dev/null
+++ b/rack
@@ -0,0 +1 @@
+Subproject commit 08d709f78b8c5b0fbebe7821e37fa53e69afc433
```

rack Git Git HEAD

master

```
$ git commit -m 'first commit with submodule rack'
[master 0550271] first commit with submodule rack
 2 files changed, 4 insertions(+), 0 deletions(-)
 create mode 100644 .gitmodules
 create mode 160000 rack
```

rack 160000 Git

rackGit

```
$ git log -1
commit 0550271328a0038865aad6331e620cd7238601bb
Author: Scott Chacon <schacon@gmail.com>
Date:   Thu Apr 9 09:03:56 2009 -0700

    first commit with submodule rack
$ cd rack/
$ git log -1
commit 08d709f78b8c5b0fbebe7821e37fa53e69afc433
Author: Christian Neukirchen <chneukirchen@gmail.com>
Date:   Wed Mar 25 14:49:04 2009 +0100
```

Document version change

```
$ git clone git://github.com/schacon/myproject.git
Initialized empty Git repository in /opt/myproject/.git/
remote: Counting objects: 6, done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 0), reused 0 (delta 0)
Receiving objects: 100% (6/6), done.
$ cd myproject
$ ls -l
total 8
-rw-r--r-- 1 schacon admin 3 Apr 9 09:11 README
drwxr-xr-x 2 schacon admin 68 Apr 9 09:11 rack
$ ls rack/
$
```

rack git submodule init git submodule update

```
$ git submodule init
Submodule 'rack' (git://github.com/chneukirchen/rack.git) registered for path 'rack'
$ git submodule update
Initialized empty Git repository in /opt/myproject/rack/.git/
remote: Counting objects: 3181, done.
remote: Compressing objects: 100% (1534/1534), done.
remote: Total 3181 (delta 1951), reused 2623 (delta 1603)
Receiving objects: 100% (3181/3181), 675.42 KiB | 173 KiB/s, done.
Resolving deltas: 100% (1951/1951), done.
Submodule path 'rack': checked out '08d709f78b8c5b0fbebe7821e37fa53e69afcf433'
```

rack rack

```
$ git merge origin/master
Updating 0550271..85a3eee
Fast forward
  rack | 2 ++
  1 files changed, 1 insertions(+), 1 deletions(-)
[master*]$ git status
# On branch master
# Changed but not updated:
#   (use "git add <file>..." to update what will be committed)
#     (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   rack
#
```

```
$ git diff
diff --git a/rack b/rack
index 6c5e70b..08d709f 160000
--- a/rack
+++ b/rack
@@ -1 +1 @@
-Subproject commit 6c5e70b984a60b3cecd395edd5b48a7575bf58e0
+Subproject commit 08d709f78b8c5b0fbebe7821e37fa53e69afcf433
```

git submodule update

```
$ git submodule update
remote: Counting objects: 5, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 2 (delta 0)
Unpacking objects: 100% (3/3), done.
From git@github.com:schacon/rack
 08d709f..6c5e70b  master      -> origin/master
Submodule path 'rack': checked out '6c5e70b984a60b3cecd395edd5b48a7575bf58e0'
```

git submodule update

```
$ git submodule update
fatal: reference isn't a tree: 6c5e70b984a60b3cecd395edd5b48a7575bf58e0
Unable to checkout '6c5e70b984a60b3cecd395edd5ba7575bf58e0' in submodule path 'rac
```

```
$ git log -1 rack
commit 85a3eee996800fcfa91e2119372dd4172bf76678
Author: Scott Chacon <schacon@gmail.com>
Date:   Thu Apr 9 09:19:14 2009 -0700

    added a submodule reference I will never make public. hahahahaha!
```

CVS Subversion

Git Git Git

```
git submodule update—— HEAD  
sub  
updateGit
```

```
git checkout -b
```

```
$ git checkout -b rack  
Switched to a new branch "rack"  
$ git submodule add git@github.com:schacon/rack.git rack  
Initialized empty Git repository in /opt/myproj/rack/.git/  
...  
Receiving objects: 100% (3184/3184), 677.42 KiB | 34 KiB/s, done.  
Resolving deltas: 100% (1952/1952), done.  
$ git commit -am 'added rack submodule'  
[rack cc49a69] added rack submodule  
 2 files changed, 4 insertions(+), 0 deletions(-)  
 create mode 100644 .gitmodules  
 create mode 160000 rack  
$ git checkout master  
Switched to branch "master"  
$ git status  
# On branch master  
# Untracked files:  
#   (use "git add <file>..." to include in what will be committed)  
#  
#       rack/
```

Git rack

submodule addGit

```
$ rm -Rf rack/  
$ git submodule add git@github.com:schacon/rack.git rack  
'rack' already exists in the index
```

```
rack
```

```
$ git rm -r rack
```

```
$ git submodule add git@github.com:schacon/rack.git rack
Initialized empty Git repository in /opt/testsub/rack/.git/
remote: Counting objects: 3184, done.
remote: Compressing objects: 100% (1465/1465), done.
remote: Total 3184 (delta 1952), reused 2770 (delta 1675)
Receiving objects: 100% (3184/3184), 677.42 KiB | 88 KiB/s, done.
Resolving deltas: 100% (1952/1952), done.
```

```
$ git checkout master
error: Untracked working tree file 'rack/AUTHORS' would be overwritten by merge.
```

rack

```
$ mv rack /tmp/
$ git checkout master
Switched to branch "master"
$ ls
README  rack
```

```
rack    git submodule update    /tmp/rack
```

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

Git

Git Git .gitattributes .git/info/attrib

Git Git

Git Git — Git

MacXcode .pbxproj IDEJSON javascript ASCII — 2

Git pbxproj .gitattributes

```
*.pbxproj -crlf -diff
```

Git CRLF git showgit diffGit 1.6 -crlf -diff

```
*.pbxproj binary
```

Git 1.6 Git Git diff

WordWord Git git diff

```
$ git diff  
diff --git a/chapter1.doc b/chapter1.doc  
index 88839c4..4afcb7c 100644
```

```
Binary files a/chapter1.doc and b/chapter1.doc differ
```

Word Git

.gitattributes

```
*.doc diff=word
```

"doc"Git "word""word" Git

strings Word

```
$ git config diff.word.textconv strings
```

.docGit "word"Word

Git

git diff

```
$ git diff
diff --git a/chapter1.doc b/chapter1.doc
index c1c8a0a..b93c9e4 100644
--- a/chapter1.doc
+++ b/chapter1.doc
@@ -8,7 +8,8 @@ re going to cover Version Control Systems (VCS) and Git basics
 re going to cover how to get it and set it up for the first time if you don
 t already have it on your system.
 In Chapter Two we will go over basic Git usage - how to use Git for the 80%
-s going on, modify stuff and contribute changes. If the book spontaneously
+s going on, modify stuff and contribute changes. If the book spontaneously
+Let's see if this works.
```

Git "Let's see if this works"Word

strings MacLinu

JPEGEXIF —

exiftool

```
$ echo '*.png diff=exif' >> .gitattributes
$ git config diff.exif.textconv exiftool
```

git diff

```
diff --git a/image.png b/image.png
index 88839c4..4afcb7c 100644
```

```
--- a/image.png
+++ b/image.png
@@ -1,12 +1,12 @@
    ExifTool Version Number      : 7.74
    -File Size                 : 70 KB
    -File Modification Date/Time: 2009:04:21 07:02:45-07:00
    +File Size                 : 94 KB
    +File Modification Date/Time: 2009:04:21 07:02:43-07:00
    File Type                  : PNG
    MIME Type                  : image/png
    -Image Width               : 1058
    -Image Height              : 889
    +Image Width               : 1056
    +Image Height              : 827
    Bit Depth                  : 8
    Color Type                 : RGB with Alpha
```

SVNCVS Git Git Git 2

blobSHA-1 \$Id\$Git blobSHA-1SHAblob

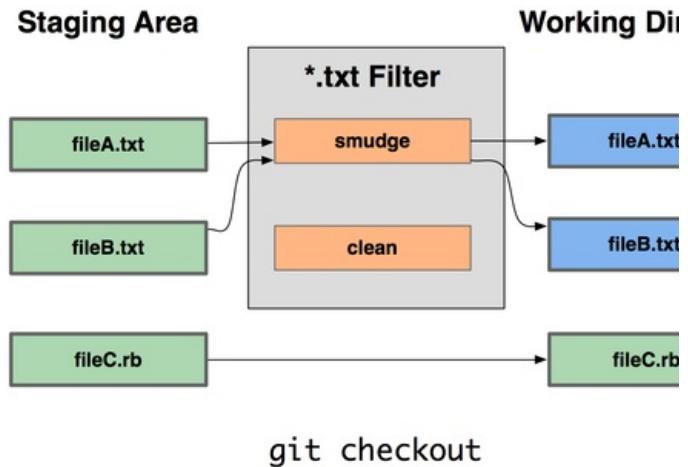
```
$ echo '* .txt ident' >> .gitattributes
$ echo '$Id$' > test.txt
```

Git blobSHA

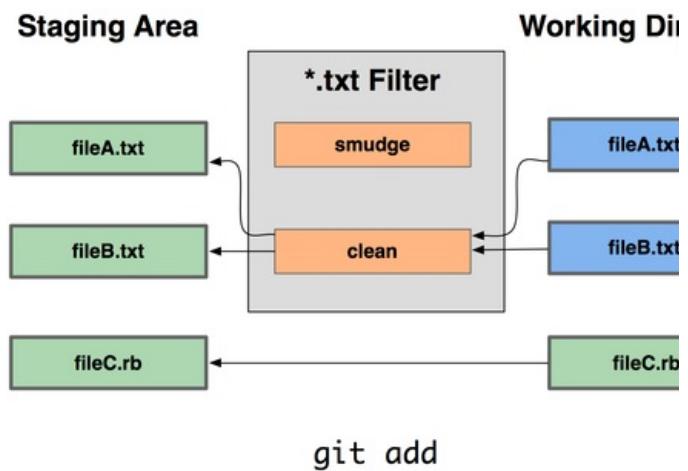
```
$ rm text.txt
$ git checkout -- text.txt
$ cat test.txt
$Id: 42812b7653c7b88933f8a9d6cad0ca16714b9bb3 $
```

SHACVSSubversion

2"clean""smudge" .gitattributes"smudge" 7-2"cle



7-2. "smudge"



7-3. "clean"

```
indentC          .gitattributes"indent"      * .c
```

```
* .c      filter=indent
```

Git "indent""smudge""clean"

```
$ git config --global filter.indent.clean indent
$ git config --global filter.indent.smudge cat
```

```
* .c      indent      cat      catC      indent
```

RCS\$Date\$Ruby

```
#!/usr/bin/env ruby
data = STDIN.read
last_date = `git log --pretty=format:"%ad" -1`
puts data.gsub('$Date$', '$Date: ' + last_date.to_s + '$')
```

git log \$Date\$ \$Date\$ — expand_date Git dater

```
$ git config filter.dater.smudge expand_date
$ git config filter.dater.clean 'perl -pe "s/\$\$Date[^\\\$]*\\\$/\$\$Date\\\$/"'
```

Perl\$Date\$ \$Date\$ \$Date\$ Git

```
$ echo '# $Date$' > date_test.txt
$ echo 'date*.txt filter=dater' >> .gitattributes
```

```
$ git add date_test.txt .gitattributes
$ git commit -m "Testing date expansion in Git"
$ rm date_test.txt
$ git checkout date_test.txt
$ cat date_test.txt
# $Date: Tue Apr 21 07:26:52 2009 -0700$
```

.gitattributes dater

Git

export-ignore

Git export-ignore

test/ Git

```
test/ export-ignore
```

git archive

export-subst

```
2      --pretty=format$Format:$           LAST_COMMIT    git archive
```

```
$ echo 'Last commit date: $Format:%cd$' > LAST_COMMIT
$ echo "LAST_COMMIT export-subst" >> .gitattributes
$ git add LAST_COMMIT .gitattributes
$ git commit -am 'adding LAST_COMMIT file for archives'
```

git archive

```
$ cat LAST_COMMIT
Last commit date: $Format:Tue Apr 21 08:38:48 2009 -0700$
```

Git Git

database.xml2

```
database.xml merge=ours
```

database.xml

```
$ git merge topic
Auto-merging database.xml
Merge made by recursive.
```

database.xml

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and

Chapter 8

Git

Git VCS, Version Control System Subversion

g

Git Git Subversion Perforce

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#) .

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).

Pro Git professional version

[Home](#) [Book](#) [Blog](#) [About](#) [Support Us](#)

This is an in-progress translation.

To help translate the book, please fork the book [at GitHub](#) and contribute!

Git “auto gc” (loose object, packfile) packfileGit
(garbage collect) packfile packfile commit ()

auto gc

```
$ git gc --auto
```

7,000 50 packfileGit gc gc.auto gc.autopac
gc (references)

```
$ find .git/refs -type f  
.git/refs/heads/experiment  
.git/refs/heads/master  
.git/refs/tags/v1.0  
.git/refs/tags/v1.1
```

git gc, refs Git .git/packed-refs

```
$ cat .git/packed-refs  
# pack-refs with: peeled  
cac0cab538b970a37ea1e769cbbde608743bc96d refs/heads/experiment  
ab1afef80fac8e34258ff41fc1b867c702daa24b refs/heads/master  
cac0cab538b970a37ea1e769cbbde608743bc96d refs/tags/v1.0  
9585191f37f7b0fb9444f35a9bf50de191beadc2 refs/tags/v1.1  
^1a410efbd13591db07496601ebc7a059dd55cfe9
```

Git refs/heads SHA Git refs packed-refs

refs

^ annotated commit

Git commit hard-reset commit commit

test hard-reset commit commit

```
$ git log --pretty=oneline
ab1afeb80fac8e34258ff41fc1b867c702daa24b modified repo a bit
484a59275031909e19aaedb7c92262719cfcd19a added repo.rb
1a410efbd13591db07496601ebc7a059dd55cfe9 third commit
cac0cab538b970a37ea1e769cbbde608743bc96d second commit
fdf4fc3344e67ab068f836878b6c4951e3b15f3d first commit
```

master commit

```
$ git reset --hard 1a410efbd13591db07496601ebc7a059dd55cf9e9
HEAD is now at 1a410ef third commit
$ git log --pretty=oneline
1a410efbd13591db07496601ebc7a059dd55cf9e9 third commit
cac0cab538b970a37ea1e769cbbde608743bc96d second commit
fdf4fc3344e67ab068f836878b6c4951e3b15f3d first commit
```

commit — commit commit SHA commit SHA — SHA

git reflog () Git HEAD reflog git update-
References" SHA ref

```
$ git reflog  
1a410ef HEAD@{0}: 1a410efbd13591db07496601ebc7a059dd55cfe9: updating HEAD  
ab1afef HEAD@{1}: ab1afef80fac8e34258ff41fc1b867c702daa24b: updating HEAD
```

commit git log -g reflog

```
$ git log -g
```

```
commit 1a410efbd13591db07496601ebc7a059dd55cfe9
Reflog: HEAD@{0} (Scott Chacon <schacon@gmail.com>)
Reflog message: updating HEAD
Author: Scott Chacon <schacon@gmail.com>
Date: Fri May 22 18:22:37 2009 -0700
```

third commit

```
commit ab1afef80fac8e34258ff41fc1b867c702daa24b
Reflog: HEAD@{1} (Scott Chacon <schacon@gmail.com>)
Reflog message: updating HEAD
Author: Scott Chacon <schacon@gmail.com>
Date: Fri May 22 18:15:24 2009 -0700
```

modified repo a bit

commit commit commit (ab1afef)

recover-branch

```
$ git branch recover-branch ab1afef
$ git log --pretty=oneline recover-branch
ab1afef80fac8e34258ff41fc1b867c702daa24b modified repo a bit
484a59275031909e19aadb7c92262719cfcdf19a added repo.rb
1a410efbd13591db07496601ebc7a059dd55cfe9 third commit
cac0cab538b970a37ea1e769ccbde608743bc96d second commit
fdf4fc3344e67ab068f836878b6c4951e3b15f3d first commit
```

master recover-branch commit

commit reflog —

recover-branch reflog commit

```
$ git branch -D recover-branch
$ rm -Rf .git/logs/
```

reflog .git/logs/ reflog commit

git fsck

```
$ git fsck --full
dangling blob d670460b4b4aece5915caf5c68d12f560a9fe3e4
dangling commit ab1afef80fac8e34258ff41fc1b867c702daa24b
dangling tree aea790b9a58f6cf6f2804eeac9f0abbe9631e4c9
dangling blob 7108f7ecb345ee9d0084193f147cdad4d2998293
```

dangling commit commit SHA

Git git clone Git

Subversion Perforce Git () ()

tree commit — () commit

test

```
$ curl http://kernel.org/pub/software/scm/git/git-1.6.3.1.tar.bz2 > git.tbz2
$ git add git.tbz2
$ git commit -am 'added git tarball'
[master 6df7640] added git tarball
 1 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 git.tbz2
```

tar

```
$ git rm git.tbz2
rm 'git.tbz2'
$ git commit -m 'oops - removed large tarball'
[master da3f30d] oops - removed large tarball
 1 files changed, 0 insertions(+), 0 deletions(-)
 delete mode 100644 git.tbz2
```

gc

```
$ git gc
Counting objects: 21, done.
Delta compression using 2 threads.
Compressing objects: 100% (16/16), done.
Writing objects: 100% (21/21), done.
Total 21 (delta 3), reused 15 (delta 1)
```

count-objects

```
$ git count-objects -v
count: 4
size: 16
in-pack: 21
packs: 1
size-pack: 2016
prune-packable: 0
garbage: 0
```

size-pack packfiles 2MB 2K — 2MB

() git gc packfile git verify-pack

```
$ git verify-pack -v .git/objects/pack/pack-3f8c0...bb.idx | sort -k 3 -n | tail -e3f094f522629ae358806b17daf78246c27c007b blob 1486 734 4667 05408d195263d853f09dca71d55116663690c27c blob 12908 3478 1189 7a9eb2fba2b1811321254ac360970fc169ba2330 blob 2056716 2056872 5401
```

2MB 7 rev-list rev-list --objects commit SHA
blob

```
$ git rev-list --objects --all | grep 7a9eb2fb 7a9eb2fba2b1811321254ac360970fc169ba2330 git.tbz2
```

tree commit

```
$ git log --pretty=oneline -- git.tbz2 da3f30d019005479c99eb4c3406225613985a1db oops - removed large tarball 6df764092f3e7c8f5f94cbe08ee5cf42e92a0289 added git tarball
```

6df76 commit Git 6 filter-branch

```
$ git filter-branch --index-filter \'git rm --cached --ignore-unmatch git.tbz2\' -- 6df7640^.. Rewrite 6df764092f3e7c8f5f94cbe08ee5cf42e92a0289 (1/2)rm 'git.tbz2' Rewrite da3f30d019005479c99eb4c3406225613985a1db (2/2) Ref 'refs/heads/master' was rewritten
```

--index-filter 6 --tree-filter rm file git r

— Git filter

filter-branch 6df7640 commit

reflog filter-branch Git .git/refs/original refs repacl
commits

```
$ rm -Rf .git/refs/original  
$ rm -Rf .git/logs/  
$ git gc  
Counting objects: 19, done.  
Delta compression using 2 threads.  
Compressing objects: 100% (14/14), done.  
Writing objects: 100% (19/19), done.  
Total 19 (delta 3), reused 16 (delta 1)
```

```
$ git count-objects -v  
count: 8  
size: 2040  
in-pack: 19  
packs: 1  
size-pack: 7  
prune-packable: 0  
garbage: 0
```

repack 7K 2MB size

git prune --expir

[prev](#) | [next](#)

Book translated into [German](#), [Chinese](#), [Japanese](#) and [Dutch](#).

Partial translations available in [Arabic](#), [Czech](#), [Spanish](#), [French](#), [Indonesian](#), [Italian](#),
[Macedonian](#), [Polish](#), [Thai](#) and [Russian](#).

All content under [Creative Commons Attribution-Non Commercial-Share Alike 3.0 license](#).

Theme and code by [Michael Bleigh](#). Hosted by [GitHub](#) and powered by [Jekyll](#).