

# git

*an introduction & explanation by*

*Nikit “Nefari0uss” Malkan*



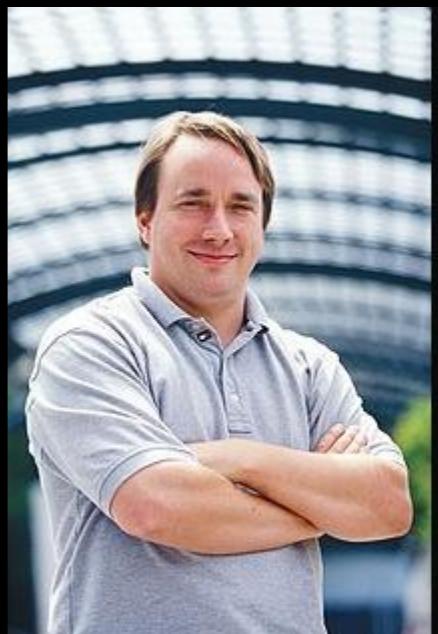
# Table of Contents

- ❖ What is it?
  - ❖ Source control analogy
- ❖ Why use source control?
  - ❖ Specifically Git?
  - ❖ Terminal vs GUI interface

# what is a git

git: [informal] [British] an unpleasant or contemptible person.

# what is a git



I'm an egotistical bastard, and I name all my projects after myself. First Linux, now git.

(Linus Torvalds)

# what is git

git: A distributed revision control system with...support for distributed, non-linear workflows. [Wikipedia – Git (software)]

# what is git

...

try this:

a version control system that allows a user to  
store, update, and maintain files and projects.

# why use version control

let's talk workflow

# let's talk workflow

imagine writing an essay

what's the process like

the writing process

# the writing process

write the whole thing at once

# the writing process

write the whole thing at once

CLG are NALCS champs. TSM is trash.

# the writing process

write the whole thing at once

unlikely

# the writing process

write the whole thing at once

unlikely

write bits and pieces in multiple settings

# the writing process

write

CLG are NALCS champs.

# the writing process

write → save

CLG are NALCS champs.

# the writing process

write → save → s5secretsgg.docx

CLG are NALCS champs.

# the writing process

write → save → s5secretsgg.docx

procrastinate

CLG are NALCS champs.

# the writing process

write → save → write

CLG are NALCS champs. TSM is trash.

# the writing process

write → save → write → save

CLG are NALCS champs. TSM is trash.

# the writing process

write → save → write → save → s5secretsgg.docx

CLG are NALCS champs. TSM is trash.

# the writing process

write →

:g.docx



# the writing process

write → save → write → save → s5secretsgg.docx  
rip progress

CLG are NALCS champs. TSM is trash.

# the writing process (trial 2)

write → save → s5secretsgg.docx

CLG are NALCS champs. TSM is trash.

# the writing process (trial 2)

write → save → s5secretsgg.docx

save backup → s5secretsgg\_backup.docx

CLG are NALCS champs. TSM is trash.

# the writing process (trial 2)

write → save → s5secretsgg.docx

save backup

s5secretsgg.docx

CLG are NALCS champs. TSM is trash.

s5secretsgg\_backup.docx

CLG are NALCS champs. TSM is trash.

# the writing process (trial 2)

s5secretsgg.docx

CLG are NALCS champs. TSM is trash.

s5secretsgg\_backup.docx

CLG are NALCS champs. TSM is trash.

# the writing process (trial 2)

s5secretsgg.docx

CLG are NALCS champs. TSM is trash.

s5secretsgg\_backup.docx

CLG are NALCS champs. TSM is trash.

what if we mess up our active copy

# the writing process (trial 2)

s5secretsgg.docx

TSM are NALCS champs. No thread is safe.

s5secretsgg\_backup.docx

CLG are NALCS champs. TSM is trash.

what if we mess up our active copy

# the writing process (trial 2)

s5secretsgg.docx

s5secretsgg\_backup.docx

TSM are NALCS champs. No thread is safe.

CLG are NALCS champs. TSM is trash.

what if we mess up our active copy  
restore from backup

# the writing process (trial 2)

s5secretsgg.docx

s5secretsgg\_backup.docx

TSM are NALCS champs. No thread is safe.

CLG are NALCS champs. TSM is trash.

what if we mess up our active copy  
restore from backup

# the writing process (trial 2)

s5secretsgg.docx

CLG are NALCS champs. TSM is trash.

s5secretsgg\_backup.docx

CLG are NALCS champs. TSM is trash.

what if we mess up our active copy  
restore from backup

# the writing process (trial 2)

s5secretsgg.docx

CLG are NALCS champs. TSM is trash.

s5secretsgg\_backup.docx

CLG are NALCS champs. TSM is trash.



# the writing process (trial 2)

s5secretsgg.docx

s5secretsgg\_backup.docx

what if we mess up active copy and overwrite the backup

# the writing process (trial 2)

s5secretsgg.docx

→ save →

s5secretsgg\_backup.docx

TSM are NALCS champs. No thread is safe.

→ TSM are NALCS champs. No thread is safe.

# the writing process (trial 2)

s5secretsgg.docx

s5secretsgg\_backup.docx

uh-oh

# the writing process (trial 2)

s5secretsgg.docx      s5secretsgg\_backup.docx

uh-oh

we overwrote our backup with the mess up

# the writing process (trial 3)

write → save → s5secretsgg.docx

CLG are NALCS champs. TSM is trash.

# the writing process (trial 3)

write → save → s5secretsgg.docx

make iterative backups

CLG are NALCS champs. TSM is trash.

# the writing process (trial 3)

s5secretsgg

s5secretsgg\_2015-09-01

s5secretsgg\_2015-09-10

s5secretsgg\_2015-09-19    s5secretsgg\_2015-09-28

this gets messy very quickly

however, this system allows us to restore from any given point

# new problem

which backup has the info we need

s5secretsgg

s5secretsgg\_2015-09-01

s5secretsgg\_2015-09-10

s5secretsgg\_2015-09-19

s5secretsgg\_2015-09-28

ponder that question for a bit...

let's shift gears for a minute...

let's talk *git*

# why use git

you have all these iterative backups

how do you manage them all

# why use git

git pros

git cons

# git con

considered very hard to learn

lots of weird syntax

**understanding** git internals helps remedy this tremendously

# terminal vs GUI

terminal way is best way

but nefari0uss – that gui application looks so nice and pretty

and that terminal looks scary and hard

I don't care

gitg - linux-git

File Edit View Help

History Commit

Branch: Select branch

Subject Date

Subject	Author	Date
Merge branch 'gb/gitweb-opml'	Junio C Hamano	Sun Jan 18 08:07:19 2009
Merge branch 'mv/apply-parse-opt'	Junio C Hamano	Sun Jan 18 08:06:53 2009
Merge branch 'tr/rebase-root'	Junio C Hamano	Sun Jan 18 08:06:38 2009
Merge branch 'gb/gitweb-patch'	Junio C Hamano	Sun Jan 18 08:06:19 2009
Merge branch 'ap/clone-into-empty'	Junio C Hamano	Sun Jan 18 08:05:54 2009
Merge branch 'jc/maint-format-patch'	Junio C Hamano	Sun Jan 18 08:05:50 2009
Merge branch 'tr/maint-no-index-fixes'	Junio C Hamano	Sun Jan 18 08:05:38 2009
Merge branch 'as/autocorrect-alias'	Junio C Hamano	Sun Jan 18 08:05:34 2009
Merge branch 'rs/fgrep'	Junio C Hamano	Sun Jan 18 08:05:28 2009
Merge branch 'rs/maint-shortlog-foldline'	Junio C Hamano	Sun Jan 18 08:05:23 2009
Merge branch 'mh/maint-commit-color-status'	Junio C Hamano	Sun Jan 18 08:05:19 2009
Merge branch 'maint'	Junio C Hamano	Sun Jan 18 08:04:40 2009
Update draft release notes for 1.6.1.1	Junio C Hamano	Sun Jan 18 08:04:35 2009
bundle: allow the same ref to be given more than once	Junio C Hamano	Sun Jan 18 07:27:08 2009
Merge branch 'maint-1.6.0' into maint	Junio C Hamano	Sun Jan 18 07:39:49 2009
builtin-fsck: fix off by one head count	Christian Couder	Sun Jan 18 04:46:09 2009
revision walker: include a detached HEAD in --all	Johannes Schindelin	Fri Jan 16 13:52:53 2009
Fix aitdir_detection when in subdir of aitdir	SZEDER Gábor	Fri Jan 16 16:37:33 2009

Details Tree

SHA: b2a6d1c6868b6d5e7d2b4fa9129341220ale848a  
 Author: Junio C Hamano  
 Date: Sun Jan 18 07:27:08 2009  
 Subject: **bundle: allow the same ref to be given more than once**  
 Parent: [f0298cf1c6a7b5cc8b79d84a03b0ce07df2d9e6b](#) (revision walker: include a detached HEAD in --all)

```

bundle: allow the same ref to be given more than once

"git bundle create x master master" used to create a bundle that lists
the same branch (master) twice. Cloning from such a bundle resulted in
a needless warning "warning: Duplicated ref: refs/remotes/origin/master".

Signed-off-by: Junio C Hamano <gitster@pobox.com>

diff --git a/bundle.c b/bundle.c
index daecd8e..b20f210 100644
--- a/bundle.c
+++ b/bundle.c
@@ -240,6 +240,8 @@ int create_bundle(struct bundle_header *header, const char *path,
240         return error("unrecognized argument: %s'", argv[i]);
241     }
242 
243     object_array_remove_duplicates(&revs.pending);
244+
245     for (i = 0; i < revs.pending.nr; i++) {
246         struct object_array_entry *e = revs.pending.objects + i;
247         unsigned char_shall[20];
diff --git a/object.c b/object.c
index 50b6528..7e6a92c 100644
--- a/object.c
+++ b/object.c

```

Loaded 17336 revisions in 0.75s

Terminal

```
2012-11-23 10:51 James Winters      M [master] [origin/HEAD] [origin/master] Merge branch 'atlassian_jira_6_0_0~
2012-11-23 10:49 James Winters      M Merge commit 'a6710b1b1148a3f66816df1ddde193d6a6d7d6f2' from 6.0_OD1 rele~
2012-11-23 10:42 James Winters      o [origin/atlassian_jira_6_0_OD_01_release_branch] ROTP-609: Now store th~
2012-11-23 10:25 Geoffrey Wong     o Update Baseline Images - changes from ROTP-614: show custom content lin~
2012-11-23 09:57 Jeroen De Raedt   M [origin/JRADEV-15983] Merge branch 'master' into JRADEV-15983
2012-11-22 16:35 Filip Rogaczewski o Revert merged from stable causing functional test break.
2012-11-22 18:02 Eric Dalglish    M Merged from stable.
2012-11-22 16:28 Martin Meinholt  o ROTP-578 Fix WebDriver test when common header is on
2012-11-22 16:28 Graeme Smith     o Fixed merge weirdness. Somehow this got removed.
2012-11-22 16:23 Graeme Smith     o Revert "Revert "Merge branch 'JRADEV-15983'"""
[main] 647478e4b89612425641d3102b6feadeb5137fc8 - commit 8 of 47622 (0%)
commit 647478e4b89612425641d3102b6feadeb5137fc8
Refs: atlassian_jira_5_2-773-g647478e
Author: Martin Meinholt <mmeinholt@atlassian.com>
AuthorDate: Thu Nov 22 16:28:23 2012 +1100
Commit: Martin Meinholt <mmeinholt@atlassian.com>
CommitDate: Thu Nov 22 16:28:23 2012 +1100

ROTP-578 Fix WebDriver test when common header is on

* Add common header specific issues menu
* Fix wrong selector in JiraCommonHeader page object
---
.../jira/pageobjects/components/JiraCommonHeader.java |  2 +-+
.../jira/pageobjects/components/JiraHeader.java       | 17 ++++++-----+
.../components/menu/CommonHeaderIssuesMenu.java     | 14 ++++++-----+
.../jira/pageobjects/components/menu/IssuesMenu.java |  5 +++++
4 files changed, 34 insertions(+), 4 deletions(-)

diff --git a/jira-page-objects/src/main/java/com/atlassian/jira/pageobjects/components/JiraCommonHeader.java b/ji~
index 6b551d6..8f3c3b0 100644
--- a/jira-page-objects/src/main/java/com/atlassian/jira/pageobjects/components/JiraCommonHeader.java
[diff] 647478e4b89612425641d3102b6feadeb5137fc8 - line 1 of 119 (17%)
```

# terminal vs GUI

gui applications hide what's happening

terminal way forces you to learn what's happening

Murphy's law *will* happen

your code base will become borked

knowing how git works will help you fix it

THIS IS GIT. IT TRACKS COLLABORATIVE WORK  
ON PROJECTS THROUGH A BEAUTIFUL  
DISTRIBUTED GRAPH THEORY TREE MODEL.

COOL. HOW DO WE USE IT?

NO IDEA. JUST MEMORIZIZE THESE SHELL  
COMMANDS AND TYPE THEM TO SYNC UP.  
IF YOU GET ERRORS, SAVE YOUR WORK  
ELSEWHERE, DELETE THE PROJECT,  
AND DOWNLOAD A FRESH COPY.



<https://xkcd.com/1597/>

# terminal vs GUI

gui may not always be available

terminal will

↙ ↗ ↘ ↙ *terminal way is best way* ↙ ↗ ↘ ↙

going back...

remember this?

# new problem

which backup has the info we need

s5secretsgg

s5secretsgg\_2015-09-01

s5secretsgg\_2015-09-10

s5secretsgg\_2015-09-19

s5secretsgg\_2015-09-28

# new problem

which backup has the info we need

answer is to look at the *metadata*

# git internals

this section is **very** important to understand  
seriously – pay attention to this section

# metadata

a set of data that describes and gives information about other data

backups should contain metadata

# metadata

who

what

when

# metadata

who: author of the change

what: what was changed

when: date and time of change

# fin

very this is so short

the full presentation is over 400 slides...

it is freely available on my GitHub profile: Nefari0uss

<https://goo.gl/uyzCs9>

I hope you learned something!

# in case of fire

1. git commit
2. git push
3. leave building

# Sources

- ❖ Git-SCM book: <https://git-scm.com/book>
  - ❖ Daniel “paradigm” Thau from Open Source Club at OSU on Git:
    - ❖ <https://opensource.osu.edu/au10/git>
  - ❖ git – the simple guide: <https://rogerdudler.github.io/git-guide/>
  - ❖ Atlassian Git Tutorial: <https://www.atlassian.com/git/tutorials/>
  - ❖ Understanding Git Conceptually: <http://www.sbf5.com/~cduan/technical/git/>
  - ❖ Wikiepda – Git (software): [https://en.wikipedia.org/wiki/Git\\_\(software\)](https://en.wikipedia.org/wiki/Git_(software))
  - ❖ Git Reference: <http://gitref.org/basic/>
- 
- ❖ Lots of pictures from Git-SCM Book, Wikipedia and Google/Bing Image searches