

Introduction to the Git Version Control System

Sebastian Rockel

rockel@informatik.uni-hamburg.de

University of Hamburg

Faculty of Mathematics, Informatics and Natural Sciences

Department of Informatics

Technical Aspects of Multimodal Systems



November 8, 2011

Outline

Introduction

Configuration

User Setup

Repository Setup

Basic Tasks

Commonly used Commands

Branches

Special Usecases

GUI

Tags

Remote Repositories

Introduction

Git is..

- ▶ from Linus Torwalds
- ▶ local (and remote if necessary)
- ▶ fast (well..)
- ▶ Comprehensive Howto @ <http://linux.yyz.us/git-howto.html>



Introduction

Git is..

- ▶ from Linus Torwalds
- ▶ local (and remote if necessary)
- ▶ fast (well..)
- ▶ Comprehensive Howto @ <http://linux.yyz.us/git-howto.html>



Introduction

Git is..

- ▶ from Linus Torwalds
- ▶ local (and remote if necessary)
- ▶ fast (well..)
- ▶ Comprehensiv Howto @ <http://linux.yyz.us/git-howto.html>



Introduction

Git is..

- ▶ from Linus Torwalds
- ▶ local (and remote if necessary)
- ▶ fast (well..)
- ▶ Comprehensive Howto @ <http://linux.yyz.us/git-howto.html>



Insights

▶ Index

▶ under:

```
<repository dir>/ .git / ..
```

▶ Hashes

- ▶ every version in the index has its own unique SHA1 hash
- ▶ abbreviate it with the first few character



Insights

- ▶ Index

- ▶ under:

```
<repository dir>/.git /..
```

- ▶ Hashes

- ▶ every version in the index has its own unique SHA1 hash
 - ▶ abbreviate it with the first few character



Outline

Introduction

Configuration

User Setup

Repository Setup

Basic Tasks

Commonly used Commands

Branches

Special Usecases

GUI

Tags

Remote Repositories

Git per User Setup

`$HOME/.gitconfig`

```

1  [gui]
   recentrepo = /Users/sebastian/projekt090406
3  spellingdictionary = en
   [user]
5  email = rockel@informatik.uni-hamburg.de
   name = Sebastian Rockel
7  [merge]
   tool = opendiff
9  summary = true
   [color]
11 diff = auto
   status = auto
13 branch = auto
   [core]
15 autocrlf = input
  
```

Outline

Introduction

Configuration

User Setup

Repository Setup

Basic Tasks

Commonly used Commands

Branches

Special Usecases

GUI

Tags

Remote Repositories



Exclude Files

per repo: <repository dir>/.**git/info/exclude**

```
# Lines that start with '#' are comments.  
# For a project mostly in C, the following  
# would be a good set of exclude patterns  
# (uncomment them if you want to use them):  
# *.[oa]  
*~  
*.swp
```

per (sub)dir: <some repo (sub) dir>/.**gitignore**

```
.metadata/  
/doc/
```



Exclude Files

per repo: `<repository dir>/ .git/info/exclude`

```
# Lines that start with '#' are comments.  
# For a project mostly in C, the following  
# would be a good set of exclude patterns  
# (uncomment them if you want to use them):  
# *.[oa]  
*~  
*.swp
```

per (sub)dir: `<some repo (sub) dir>/ .gitignore`

```
.metadata/  
/doc/
```



Outline

Introduction

Configuration

User Setup

Repository Setup

Basic Tasks

Commonly used Commands

Branches

Special Usecases

GUI

Tags

Remote Repositories



Basic Tasks

clone a remote repository:

```
git clone  
git@github.com: buzzer/Robot-Collaboration.git
```

get the latest changes from remote repo:

```
git fetch [$branch|--all]
```

```
git pull [optional ssh/https path again]
```

Basic Tasks

clone a remote repository:

```
git clone  
git@github.com: buzzer/Robot-Collaboration.git
```

get the latest changes from remote repo:

```
git fetch [$branch|--all]
```

```
git pull [optional ssh/https path again]
```




Basic Tasks (cont'd)

stage changes for commit:

```
git add .|-i|$files
```

commit changes:

```
git commit -a|-m 'message'
```

revert version, checkout:

```
git checkout [-f] .|$branch|$hash
```



Basic Tasks (cont'd)

stage changes for commit:

```
git add .|-i|$files
```

commit changes:

```
git commit -a|-m 'message'
```

revert version, checkout:

```
git checkout [-f] .|$branch|$hash
```



Basic Tasks (cont'd)

stage changes for commit:

```
git add .|-i|$files
```

commit changes:

```
git commit -a|-m 'message'
```

revert version, checkout:

```
git checkout [-f] .|$branch|$hash
```



Basic Tasks (cont'd)

Get the current repo status

Show current branch, changes, untracked files:

```
git status
```

Show commit history:

```
git log
```

Show textual diff of current changes:

```
git diff
```



Basic Tasks (cont'd)

Get the current repo status

Show current branch, changes, untracked files:

```
git status
```

Show commit history:

```
git log
```

Show textual diff of current changes:

```
git diff
```



Basic Tasks (cont'd)

Get the current repo status

Show current branch, changes, untracked files:

```
git status
```

Show commit history:

```
git log
```

Show textual diff of current changes:

```
git diff
```



Outline

Introduction

Configuration

User Setup

Repository Setup

Basic Tasks

Commonly used Commands

Branches

Special Usecases

GUI

Tags

Remote Repositories



Branches

there might be *life* besides the master

List available branches

```
git branch [-a]
```

Create a new branch

```
git branch $newbranch
```

Switch to a new branch

```
git checkout $newbranch
```




Branches

there might be *life* besides the master

List available branches

```
git branch [-a]
```

Create a new branch

```
git branch $newbranch
```

Switch to a new branch

```
git checkout $newbranch
```



Branches

there might be *life* besides the master

List available branches

```
git branch [-a]
```

Create a new branch

```
git branch $newbranch
```

Switch to a new branch

```
git checkout $newbranch
```



Branches (cont'd)

Delete a branch

```
git branch -d|-D $newbranch
```



Outline

Introduction

Configuration

User Setup

Repository Setup

Basic Tasks

Commonly used Commands

Branches

Special Usecases

GUI

Tags

Remote Repositories

Gui

not needed but nice to have

```
git fetch [$branch|--all]
```

Display changes, prepare, execute commit

```
git gui
```

Show all commits, branches

```
gitk [--all]
```



Gui

not needed but nice to have

```
git fetch [$branch|--all]
```

Display changes, prepare, execute commit

```
git gui
```

Show all commits, branches

```
gitk [--all]
```



Merge Tool

- ▶ normally not needed
- ▶ Git automatic merge is quite efficient
- ▶ if there are merge conflicts, a (system dependent, see Git config) merge tool can be used:

```
git mergetool
```



Merge Branches

- ▶ Master branch M
- ▶ Work branch A and B
- ▶ merge A and B into M

```
git checkout M # switch to branch M  
git merge A # merge A into M  
git merge B # merge B into M
```




Outline

Introduction

Configuration

User Setup

Repository Setup

Basic Tasks

Commonly used Commands

Branches

Special Usecases

GUI

Tags

Remote Repositories



Don't forget the tags

- ▶ tag versions

```
git tag v1.3
```

- ▶ git pull / push does not take tags into account

```
git pull | fetch --tags  
git push --tags
```



Don't forget the tags

- ▶ tag versions

```
git tag v1.3
```

- ▶ git pull / push does not take tags into account

```
git pull | fetch --tags  
git push --tags
```



Outline

Introduction

Configuration

User Setup

Repository Setup

Basic Tasks

Commonly used Commands

Branches

Special Usecases

GUI

Tags

Remote Repositories



Github

`https://github.com`

add remote repo

```
git remote add origin $remoteRepo
```



Github

`https://github.com`

add remote repo

```
git remote add origin $remoteRepo
```



SVN

- ▶ Working with a remote Subversion repository (but with git locally)

```
git-svn clone [http location of an svn repository]
```

- ▶ Now you can work with the checked out directory as though it was a git repository

Thank You!

Any questions?



Further Reading

- ▶ <http://linux.yyz.us/git-howto.html>
- ▶ <http://help.github.com/git-cheat-sheets>