

# Build System and Version Control of the Common PWA Framework

Florian Feldbauer

Johannes Gutenberg-Universität Mainz / Helmholtz Institut Mainz

XLIII. PANDA Collaboration Meeting  
December 10, 2012

JOHANNES GUTENBERG  
UNIVERSITÄT MAINZ



- Using Boost-Build
  - Small, clearly arranged files to build a project  
“Jamroot” for main project, “Jamfile” for sub project
  - Define sub projects which are separately build
  - Dependencies between sub projects are resolved by Boost-build automatically
  - Supports two different build variants: Debug and Release
  - Only need to invoke one command (Not two like when using CMake)
- Sub projects for data interfaces, optimizer interfaces ... (c.f. talk by Mathias)
- Each sub project is build as shared library

# Example for Jamfile

```
import os ;

local ROOTSYS = [ os.environ ROOTSYS ] ;
local ROOTINC = [ os.environ ROOTINCLUDE ] ;

local rlibs = [ SHELL "$ROOTSYS/bin/root-config -libs" ] ;
ROOTLIBS = [ MATCH "(.*)[\n]" : $(rlibs) ] ;

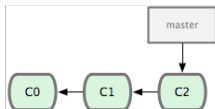
project :
    requirements <include>./
    <include>$(TOP)/Core
    <include>$(ROOTINC)
    <cxxflags>-std=c++0x
    <linkflags>$(ROOTLIBS)
    <linkflags>-IEG
    ;

lib DIFRoot2Part : [ glob *.cpp ] $(TOP)/Core//ComPWA ;
```

# Version Control for the PWA Framework

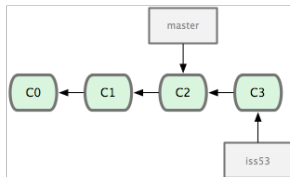
- Using git for version control
- Advantages (compared to Subversion):
  - Only additions and deletions are saved
    - ⇒ Requires less disc space
  - Everyone has a copy of complete repository
    - ⇒ Complete history with logs and diffs offline
    - ⇒ Distributed backup
  - In principal no server needed
  - Most operations are fast (only executed locally)
  - Allows branching
- Git is supported by different frontends (e.g. eclipse)

# Simple Example for Branching



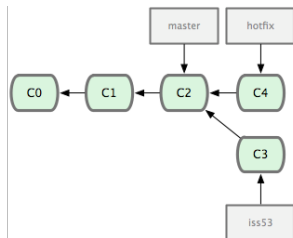
- Existing repository with some commits

# Simple Example for Branching



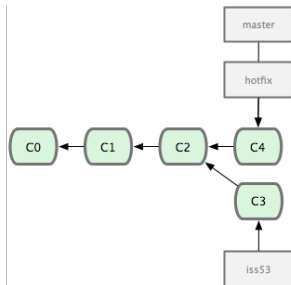
- Existing repository with some commits
- Work on issue (#53) from some ticket system

# Simple Example for Branching



- Existing repository with some commits
- Work on issue (#53) from some ticket system
- Need to apply some hotfix without adding changes made in issue53-branch

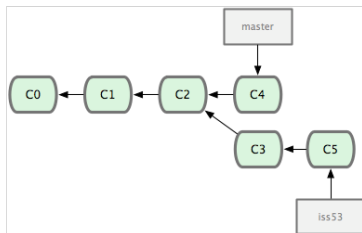
# Simple Example for Branching



- Existing repository with some commits
- Work on issue (#53) from some ticket system
- Need to apply some hotfix without adding changes made in issue53-branch
- Merge hotfix and Master branch



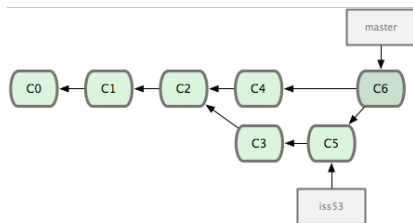
# Simple Example for Branching



- Existing repository with some commits
- Work on issue (#53) from some ticket system
- Need to apply some hotfix without adding changes made in issue53-branch
- Merge hotfix and Master branch
- Continue work on issue 53

Pictures from <http://git-scm.com/>

# Simple Example for Branching



- Existing repository with some commits
- Work on issue (#53) from some ticket system
- Need to apply some hotfix without adding changes made in issue53-branch
- Merge hotfix and Master branch
- Continue work on issue 53
- Merge issue 53 branch into master

Pictures from <http://git-scm.com/>

- Clone a remote repository  
`git clone gitosis@tau.ep1.rub.de:/var/www/git/ComPWA`
- Edit files or add new ones
- Stage your changes  
`git add changed/new file`
- Commit your changes  
`git commit -m "commit message"`
- Push your commits to remote repository  
`git push remote branch`

# Publication of the Code

- Platforms/Servers available for publication of code
- e.g. Github (<https://github.com/>)
  - Free
  - Repositories can be managed by single users or organizations
  - Public and private repositories
  - Complete version history accessible directly from the web interface
  - Wiki page(s)
  - Ticket system
  - Supports https, ssh, and git read-only access to the repositories
- Example for project on github:  
`https://github.com/ControlSystemStudio/cs-studio`