

Git HOWTO

Install Git Package

In Debian or Debian-based distros (eg. Ubuntu), type at the prompt:

```
apt-get install git
```

Configure Git

Tell git what your name and email are:

```
git config --global user.name "Bernard Parent"  
git config --global user.email bernparent@gmail.com
```

Make a New Git Project

On the client, create a new directory for the project:

```
mkdir projectname
```

Within the main directory of the project type:

```
git init
```

On the client, within the main directory of the project, create a new file called `.gitignore` and put the following within it for a L^AT_EX project:

```
*.[oa]  
*.aux  
*.bbl  
*.blg  
*.dvi  
*.log  
*.ps  
*report.pdf  
*article.pdf  
*slides.pdf  
*slides-pics.pdf  
*.spl  
*~
```

On the server, create new directory and within it type:

```
git --bare init
```

On the client, set the origin on the server:

```
git remote add origin ssh://myusername@myserver.com/home/myusername/pr  
ojectname
```

Alternately, if the origin is on github, type the following:

```
git remote add origin https://bernardparent@github.com/bernardparent/p  
rojectname.git
```

Push Changes to Server

In the main directory of the project, create a new file named "gitnewversion.sh" with the following content:

```
#!/bin/sh
filecheck="gitpull.sh"
if [ -f "$filecheck" ]; then
  if [ $# -eq 1 ]; then
    if git show-ref --tags $1 ; then
      echo Version $1 already committed
      exit
    fi
    if git ls-remote --exit-code --tags origin $1 ; then
      echo Version $1 already committed on server
      exit
    fi
    echo 'cleaning latex directory..'
    make cleanall
    echo 'copy to origin using git..'
    git add -A .
    git commit -a -m "$1"
    git tag -d $1 > /dev/null 2>&1
    git tag -a $1 -m "$1"
    git push --tags origin master
    echo '[done]'  else
    echo "gitnewversion.sh needs one argument: the new version string"
  fi
else
  echo "gitnewversion.sh must be run from the project main directory."
fi
```

Make this file executable:

```
chmod u+x gitnewversion.sh
```

And push your changes to the server by typing:

```
./gitnewversion.sh 19feb18a
```

where "19feb18a" is the version number.

Clone From Server

If a git project has not yet been created on a client but exists on your server, simply type at the prompt:

```
git clone ssh://myusername@myserver.com/home/myusername/projectname
```

or, if the server is github:

```
git clone https://bernardparent@github.com/bernardparent/CFDWAREP
```

Pull Updates From Server

To download the latest updates for a project anchored on your server to your client, first create a file called "gitpull.sh" within the main directory of your project and put the following contents within it:

```
#!/bin/sh
filecheck="gitpull.sh"
if [ -f "$filecheck" ]; then
    git pull --tags origin master
else
    echo "gitpull must be run from the project main directory."
fi
```

Then, make this file executable:

```
chmod u+x gitpull.sh
```

And simply upgrade to the latest version by typing:

```
./gitpull.sh
```

Checkout Version

To checkout an older or newer version, type the following within your project directory:

```
git checkout 16may05
```

where 16may05 is the version number you wish to check out. To get back to the master branch, type:

```
git checkout master
```

To reset all the files for the current version, type:

```
git reset --hard
```

Check Project Status

To check which version is currently being looked at and the files that have been changed type:

```
git status
```

To check if files are not corrupted, type:

```
git fsck
```

To list the versions, type:

```
git tag
```