
Applications Documentation

Release 0.1.0

Joon Ro

November 30, 2013

Contents

1	BitBucket	3
1.1	Wiki	3
2	Command-line interface	5
2.1	find	5
2.2	perl	5
2.3	grep	5
3	Emacs	7
3.1	org-mode	7
3.2	CUA-Mode	7
3.3	Clipboard	7
3.4	Auto Complete	8
3.5	Windows specific	8
4	foobar2000	9
4.1	Componenets	9
4.2	External Programs	9
4.3	Useful Scripts	9
5	Git	11
5.1	Local	11
5.2	Remote	12
5.3	Commit Message	12
5.4	Workflow: gitflow	13
5.5	Workflow: Paper	13
6	Kile	15
6.1	Live Preview	15
7	LyX	17
7.1	View/Updating PDF	17
7.2	Beamer	17
7.3	Impressive	18

7.4	Windows	18
7.5	GNU/Linux	20
7.6	Screen Fonts	20
8	Qualtrics	21
8.1	Change .css settings per page	21
9	Read the Docs	23
9.1	BitBucket Settings	23
10	Sync	25
10.1	Dropbox	25
10.2	AeroFS	25
11	Vim	27
11.1	File marks	27
12	Indices and tables	29

Contents:

BitBucket

1.1 Wiki

1.1.1 Clone the wiki

Use the following URL for wiki:

```
git@bitbucket.org:username/repo.git/wiki
```

1.1.2 Adding Images

Use absolute path:

```
.. image:: ./images/Dead-Rising-2.png
```

Note: It seems it does not work if you have spaces in the image name.

Command-line interface

2.1 find

2.1.1 Find files with filename and exec command against them

Delete found files:

```
$ find ./ -maxdepth 1 -name "8*" -delete
$ find ./ -maxdepth 1 -name "8*" -exec rm {} \;
```

Move found files:

```
$ find ./ -maxdepth 1 -name "8*" -exec mv {} tmp/ \;
```

2.2 perl

2.2.1 Perl Pie

```
$ perl -pi -e '<some replacement regular expression>' <file or files>
```

2.3 grep

2.3.1 Frequently Used Options

-P, --perl-regexp	PATTERN is a Perl regular expression
-H, --with-filename	print the filename for each match
-R, -r, --recursive	equivalent to --directories=recurse
-B, --before-context=NUM	print NUM lines of leading context
-A, --after-context=NUM	print NUM lines of trailing context
-C, --context=NUM	print NUM lines of output context
-i, --ignore-case	ignore case distinctions

```
$ grep -r -i -P -B 3 -A 3 "orbitz" *
```

2.3.2 Find duplicate words

The following will print out duplicate words cases such as the the:

```
$ grep -E "\b(\w+)\b\b\1\b" *.lyx
```

2.3.3 Searching in all files recursively using `grep -r`

```
$ grep -r "ramesh" *
```

2.3.4 From shell history, find items with `rename` in it

```
$ history | grep 'rename'
```

Emacs

3.1 org-mode

```
#+TITLE:      Project: Name
#+AUTHOR:     Joon Ro
#+EMAIL:      joonhyoung.ro@gmail.com
#+DESCRIPTION: org file for the project
#+CATEGORY:   CATEGORY
#+STARTUP:    overview
```

- * Literature
- * Feedback
- * Tasks
- * Logs
- * Schedules
- * Notes
- * Events

3.2 CUA-Mode

```
(cua-mode t)
(setq cua-auto-tabify-rectangles nil) ; Don't tabify after rectangle commands
(setq cua-keep-region-after-copy t) ; Standard Windows behavior
```

3.3 Clipboard

```
(transient-mark-mode 1) ; no region when it is not highlighted
(setq select-active-regions nil) ; stops active region automatically setting primary selection

; OS dependent settings
(cond ((eq system-type 'gnu/linux)
      (progn
        (setq interprogram-paste-function 'x-cut-buffer-or-selection-value)
      ))
      (t)))
```

```
((eq system-type 'windows-nt)
  (progn
    (setq interprogram-paste-function 'x-selection-value)
  ))
)
```

3.4 Auto Complete

3.4.1 Tab completion does not work in org-mode

<http://lists.gnu.org/archive/html/emacs-orgmode/2011-02/msg00180.html>

org mode somehow (I have no idea how or why) manages to distinguish between TAB and C-i and this distinction is inherited by minor modes operating within an org mode buffer. For auto-complete, the fix is easy:

```
(require 'auto-complete)
(require 'auto-complete-config)
(ac-config-default)
;; (define-key ac-complete-mode-map "\t" 'ac-expand)
(define-key ac-complete-mode-map [tab] 'ac-expand) ; <<--- add this line
;; (define-key ac-complete-mode-map "\r" 'ac-complete)
```

I've included two commented out lines: they are the defaults set by auto-complete. I found, before adding the annotated line, that C-i worked as expected within an auto-complete pop-up but the tab key didn't. With this key binding, it does.

3.5 Windows specific

3.5.1 Aspell

<http://www.emacswiki.org/emacs/AspellWindows>

1. Download Aspell executable (Full installer) and at least one dictionary (aspell-en-0.50-2-3.exe) from <http://aspell.net/win32/>.
2. Install Aspell-0-50-3-3-Setup and then Aspell-en-0.50-2-3.
3. Add the following to your ~/.emacs file:

```
(add-to-list 'exec-path "C:/Program Files (x86)/Aspell/bin/")
(setq ispell-program-name "aspell")
(setq ispell-personal-dictionary "~/ .emacs/.ispell")
(require 'ispell)
```

3.5.2 org-mode: Google calendar sync

1. Install awk
2. Install cron for windows
3. Install cURL:

```
$ cinst curl
```

foobar2000

4.1 Componentets

- Columns UI
 - You can install this to your %appdata%\foobar2000\user-components\foo_ui_columns
- Quick Tagger
- Windows 7 Shell Integration

4.2 External Programs

- flac

4.3 Useful Scripts

<http://nujufoobar.blogspot.com/2013/01/useful-scripts-foobar2000.html>

4.3.1 Rating column

```
$rgb(0,128,255,0,128,255)$repeat(;Ú,$meta(rating))  
$rgb(192,192,192,128,128,128)$repeat(;Û,$sub(5,%rating%))
```

Git

5.1 Local

5.1.1 Cancel Last Commit

<http://superuser.com/questions/35267/how-can-i-roll-back-1-commit>

If you have committed junk but not pushed:

```
git reset --hard HEAD~1
```

Note that when using `--hard` any changes to tracked files in the working tree since the commit before head are lost.

If you don't want to wipe out the work you have done, you can use `--soft` option that will delete the commit but it will leave all your changed files "Changes to be committed", as `git status` would put it:

```
git reset --soft "HEAD^"
```

5.1.2 Dealing with line endings

<https://help.github.com/articles/dealing-with-line-endings>

if `git diff` shows the whole file, you should set the following option:

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

5.1.3 Ignore mode changes (chmod)?

How do I make git ignore mode changes (chmod)?:

```
$ git config core.fileMode false
```

5.1.4 Export snapshot of previous revision

```
git archive rev > file
```

```
--prefix=dir/      Nest all files in the snapshot in directory
--format=[tar|zip] Specify archive format to use: tar or zip
```

for example:

```
$ git archive 778b252 -o snapshot.tar --prefix=snapshot/
```

another example (with automatic extraction):

```
$ git archive 778b252 -o snapshot.tar --prefix=snapshot/ && ex snapshot.tar && rm snapshot.tar && cd
```

the same example in PowerShell:

```
$ git archive 778b252 -o snapshot.tar --prefix=snapshot\; Expand-Archive .\snapshot.tar; rm snapshot
```

5.1.5 GUI frontends

SmartGit

Good inline diff for LaTeX source.

SourceTree

Pleasant GUI

git-cola

Very easy to select lines and stage.

There is Windows installer at: <https://github.com/git-cola/git-cola/downloads>

5.2 Remote

5.2.1 Delete remote branches

```
git push origin :branchname
```

5.3 Commit Message

```
! changed
- removed
+ added
* fixed
```


5.4 Workflow: gitflow

<http://nvie.com/posts/a-successful-git-branching-model/>

- Have `master` and `devel`
- Work on a feature branch for each task, branching out from `develop`. Also make the branch name self-exploratory
- Work on the branch, make commits
- When the task is completed, merge the branch into `devel`, with the following options:
 - `Include messages from commits being merged in merge commit`
This will list the commit message titles in the merge commit message
 - `Create a new commit even if fast-forward is possible`
This will make another commit explicitly saying Merge branch “whatever”, so it is visually explicit. In cli, this is equivalent to:


```
$ git merge --no-ff
```
- Then your `devel` branch’s commit history would nicely reflect your workflow - and also it will have commits only when there is significant changes so it is much easier to look at.
- When it is ready to release, make another branch with `release`.
- You have to merge in changes from `devel` often to get the newest changes from other branches.
- You can rebase instead of merge

5.4.1 Main branches

`master`

`develop`

5.4.2 Supporting branches

`feature`

`release`

`hotfix`

5.5 Workflow: Paper

- Since a paper has many sections, I think the feature branches should be divided into section names. For example, instead of using `feature/` for everything, use actual section names such as `estimation/` and `results/`.
- Or, it might be better to have `edition`, `writing` since sometimes an edition is across many sections.

Kile

6.1 Live Preview

The Live Preview feature only works with developmental version of Kile. in openSUSE, you have to install the one at `KDE:Unstable:Playground` repo.

Also, if you have previous version of Kile installed, it might miss the settings for Live Preview. Go to `Settings > Configure Kile .. > Tools > Build` and select `Restore Default Tools..` to refresh settings.

7.1 View/Updating PDF

- When you have sections in separate files, View/Update PDF will only compile the section file separately, and View/Update Master Document will compile everything. The references will only work when you compile the master.

7.2 Beamer

Lyx color schemes for the beamer title/frametitle/etc

<http://www.mail-archive.com/lyx-users@lists.lyx.org/msg88941.html>

You have to change the beamer . layout file. In the `Style BeginFrame` layout definition, change:

```
Font
  Series      Bold
  Size       Largest
  Color      Blue
EndFont
```

to whatever you need.

For single documents, you can also paste something like:

```
Style Title
  Font
  Color      Green
  EndFont
End
```

```
Style Author
  Font
  Color      Green
  EndFont
End
```

```
Style Institute
  Font
```

```
    Color          Green
EndFont
End

Style BeginFrame
  Font
    Color          Green
  EndFont
End
```

to Document > Settings > Local Layout, hit **Validate** and **Apply**.

7.3 Impressive

pdftk Version 1.44-16.1.1 worked, but 1.45 version did not work.

7.3.1 openSUSE

- If you have pdftk 1.45 version, downgrade it.
- Use numpu version from devel:python repo. Otherwise it raises an error.

7.3.2 Windows

- You can download older version of pdftk from <http://www.pdflabs.com/docs/pdftk-version-history/>. Make sure you get 1.44.

7.4 Windows

7.4.1 Inverse Search

<http://wiki.lyx.org/LyX/SyncTeX#toc3>

1. Set Tools > Preferences > Paths > LyxServer pipe to `\\.\pipe\lyxpipe`.
2. In Document > Settings > Output, **check** Synchronize with output.
3. Create a batch-file named `lyxeditor.cmd` and save it to one of the locations in your PATH Windows environmental variable:

```
@echo off
SETLOCAL enabledelayedexpansion
set file=%1
set row=%2
REM remove quotes from variables
set file=!file:!="!
set row=!row:!="!
%comspec% /q /c (@echo LYXCMD:revdvi:server-goto-file-row:%file% %row%)> \\.\pipe\lyxpipe.in&&ty
endlocal
```

AutoHotkey Script

The above cmd works well, but it shows annoying black cmd window everytime when you invoke the script. With a simple AutoHotkey script, one can not only suppress this window, but also activate LyX windows after the inverse search.

Create a AutoHotkey script named `lyx-inverse-search.ahk` with the following code and save it to the same location to `lyxeditor.cmd` and compile it with AutoHotkey to generate `lyx-inverse-search.exe`:

```
SetTitleMatchMode, RegEx
Run, lyxeditor.cmd "%1%" "%2%",, Hide
WinActivate, ^LyX:,,,
```

If you don't have AutoHotkey installed, you can download the compiled `exe`.

SumatraPDF

<http://wiki.lyx.org/Windows/LyXWinTips#toc6>

1. Download and install **Sumatra PDF**.
2. In LyX go to **Tools > Preferences > Paths** and add the install location to **PATH** prefix. Most likely this is `C:\Program Files\SumatraPDF`. In my case, it is `C:\Users\Programs\SumatraPDF`.
3. In **Tools > Preferences > File Handling > File Formats** select **PDF (pdflatex)** from the list of formats and modify **Viewer** to the following (include everything after the colon, including quotes:

```
SumatraPDF -inverse-search "%f" "%1"
```

4. If you have the compiled AutoHotkey script from above, use the following:

```
SumatraPDF -inverse-search "lyx-inverse-search.exe" "%f" "%1"
```

Okular

1. In LyX go to **Tools > Preferences > Paths** and add the install location to **PATH** prefix. In my case, it is `D:\Users\Joon\Programs\KDE\bin`.
2. In **Okular**, **Settings > Configure Okular > Editor**, choose **Custom Text Editor** and input the following:

```
lyxeditor.cmd "%f" "%1"
```

If you have the compiled AutoHotkey script from above, use the following:

```
lyx-inverse-search.exe "%f" "%1"
```

Note: If `lyxeditor.cmd` or `lyx-inverse-search.exe` is not in your **PATH**, then you have to specify the full path here.

7.5 GNU/Linux

7.6 Screen Fonts

For the document in the LyX editor windows to look like an article, use:

Latin Modern Roman
Latin Modern Sans
Latin Modern Typewriter

7.6.1 Okular and Inverse Search

<http://wiki.lyx.org/LyX/SyncTeX#toc3>

Qualtrics

8.1 Change `.css` settings per page

<http://www.qualtrics.com/university/researchsuite/coders-corner/html-and-css>

Common Qualtrics CSS Classes

The following are commonly used Qualtrics classes and IDs which can be modified through CSS. Additional classes and IDs can be found using the Inspect Element option in your browser, or through a third party extension such as Firebug.

- `#SurveyEngineBody`: The background of the entire survey page.
- `.Skin .SkinInner`: The container for the questions area as well as the header and footer.
- `.Skin #SkinContent`: The background for the questions area of the survey.
- `.Skin #LogoBar`: The container for the logo on your survey.
- `html .Skin #Logo`: The logo for your survey.
- `.Skin #Buttons`: The next and previous buttons.
- `.Skin .QuestionText`: The question text.

Note: In [Example Style Sheet](#) page, *Can Customize Width, Font* example is wrong (`.SkinInner` should be `.Skin .SkinInner`) - 04/23/2013

Read the Docs

9.1 BitBucket Settings

9.1.1 BitBucket

- Make sure you uncheck `private` from the repo
- Repo settings -> Hooks -> Add Read the Docs hook

9.1.2 Read the Docs

1. Import
2. For Repo:, use:

`https://bitbucket.org/username/project`

Sync

10.1 Dropbox

10.2 AeroFS

- It seems there is no way to share just one file without installing AeroFS client. - 2013-04-24

Vim

11.1 File marks

11.1.1 GNU/Linux

~/viminfo:

```
# File marks:
'B 25 0 ~/.bashrc
'E 366 0 ~/.emacs
'G 42 15 ~/.gvimrc
'U 5 8 ~/.unison/settings.prf
'V 199 8 ~/.vimrc
'Z 188 11 ~/.zshrc
```

11.1.2 Windows

~_viminfo:

```
# File marks:
'A 31 0 ~\Documents\AutoHotkey\AutoHotkey.ahk
'B 36 0 ~\.bashrc
'E 50 0 ~\.emacs
'G 62 0 ~\.gvimrc
'P 77 0 ~\Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps1
'U 1 0 ~\.unison\Settings-win.prf
'V 146 0 ~\.vimrc
```

Indices and tables

- *genindex*
- *modindex*
- *search*