
Dgplug Summer Training Documentation

Release 0.1

Kushal Das

Jun 22, 2017

1	Communication	3
1.1	Clients	3
1.2	How to use IRC?	3
1.3	Register your nickname	3
1.4	Rules to follow	3
1.5	Mailing list	4
1.6	Rules for the sessions	4
1.7	How to ask a question?	5
1.8	More logs to read	5
2	IRC clients	7
3	XChat or hexchat	9
3.1	How to install?	9
3.2	Configurations Steps	9
4	FAQ	13
4.1	What are the topics covered?	13
4.2	How do I subscribe to the mailing list?	14
4.3	What are the requisites for the training?	14
4.4	How do I get the logs?	14
4.5	How long does a session run?	14
4.6	How long is the summer training?	14
4.7	Will the session details be sent to the mailing list?	14
4.8	In case I miss a class, how do I catch up?	14
4.9	Do we need to learn something beforehand for this training?	14
4.10	How many sessions happen in a day?	14
4.11	Will the time be the same for all the sessions?	15
4.12	If I have to leave class early, should I announce it mid class?	15
5	Shell commands	17
5.1	Gnome Terminal	17
5.2	pwd command	17
5.3	cd command	17
5.4	ls command	18
5.5	mkdir command	19
5.6	rm command	19

5.7	tree command	19
6	Filesystem Hierarchy Standard(FHS)	21
7	Mercurial	23
7.1	Reverting some uncommitted change	23
7.2	Removing a file	23
7.3	Difference between two revisions	24
7.4	Bitbucket	24
8	vi or vim editor	29
8.1	How to install?	29
8.2	Launch vim	29
8.3	Movement with in a file	31
8.4	Cut, Copy, Paste	31
8.5	Playing with multiple files	32
8.6	Find and Replace	32
8.7	File Splitting in vim	32
8.8	Word Completion	32
8.9	Where to go Next	33
9	Indices and tables	35

Contents:

We use IRC as our primary communication during summer training and it also a main communication medium for many FOSS projects.

Clients

- XChat
- HexChat
- webchat
- Waartaa

How to use IRC?

Read [how to register your nickname](#).

Register your nickname

Remember to register your nickname, you can follow [this guide](#).

Rules to follow

Be nice to others.

Always type full English words, no sms speak in any FOSS communications. That means no more 'u' or 'r', instead type 'you' or 'are'.

Though are few sort forms which are acceptable in IRC.

Short Form	Full Form
brb	Be right back
iirc	If I remember correctly

For more [Abbreviations Commonly Used on IRC](#).

Mailing list

Please join in our [mailing list](#). Remember not to do top post but only reply inline. To avoid top post use E-mail client (Thunderbird, Evolution).

- Top post reply:

```
Hello,  
  
Please refer to <http://dgplug.org/irclogs/2014/session01-welcomeandcommunication.  
→txt>  
for yesterday's training logs.  
  
The timing of the today's class is 06:30 P.M. (IST).  
  
Bar  
  
-- Foo<foo at gmail.com> wrote:  
> i have missed the yesterday's training class.  
> Where can i get the yesterday class's log?  
> What is the timing of the today's class?
```

- Inline reply:

```
Hello,  
  
-- Foo<foo at gmail.com> wrote:  
> i have missed the yesterday's training class.  
> Where can i get the yesterday class's log?  
  
Please refer to <http://dgplug.org/irclogs/2014/session01-welcomeandcommunication.  
→txt>  
for yesterday's training logs.  
  
> What is the timing of the today's class?  
  
The timing of the today's class is 06:30 P.M. (IST).  
  
Bar
```

Rules for the sessions

- Do not speak when the session is going on.
- If you have a question type ! and wait for your turn.
- Try to come online 5 minutes before the session starts.

- Address people by their IRC nick.
- Do not use sir and madam.

How to ask a question?

First read [this document](#). Remember to search in google and then only ask.

More logs to read

- [Log 1](#)
- [Log 2](#)

CHAPTER 2

IRC clients

There are various IRC clients, for our sessions you can use either xchat or hexchat client on your computer.

XChat or hexchat

XChat/hexchat is a popular Internet Relay Chat (IRC) client. It has a choice of a tabbed document interface or tree interface, support for multiple servers and is highly configurable.

- hexchat
- XChat

How to install?

For Fedora:

```
# dnf install xchat
```

For Ubuntu:

```
# apt-get install xchat
```

For windows please download hexchat from their [site](#).

Configurations Steps

Default xchat will open in network selection window where you have to select *FreeNode* and then connect.

After connect you will see connection complete window where you have option to join channel (provide *dgplug*)

After pressing OK, you are now in main window and joined *dgplug* channel. Now you have register your nickname to make sure someone else not using it.

You will get a verification mail for your registration and same you have to execute in XChat window.

Enjoy Xchating ...

XChat: Network List

User Information

Nick name:

Second choice:

Third choice:

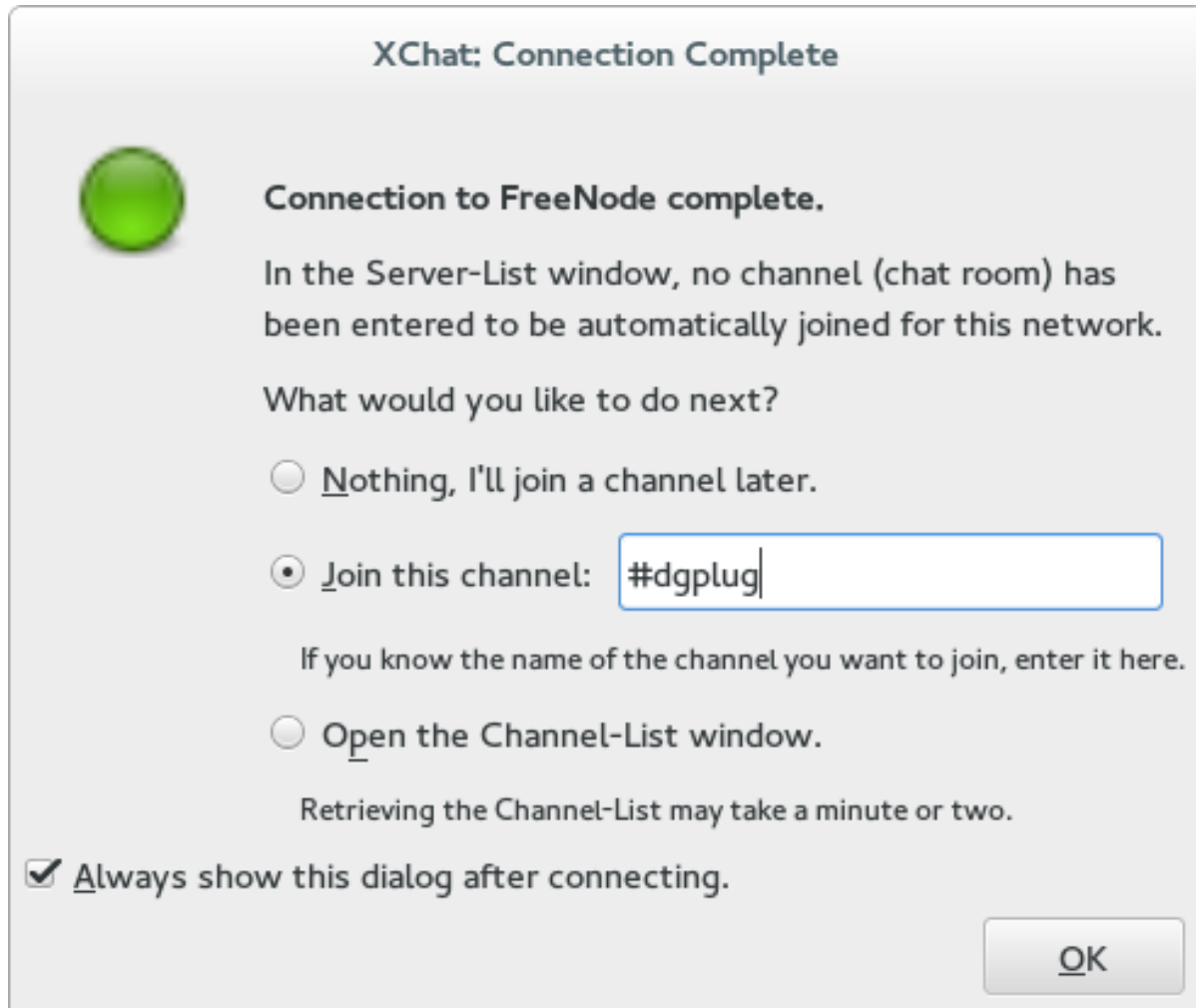
User name:

Real name:

Networks

FDFNet	<input type="button" value="Add"/> <input type="button" value="Remove"/> <input type="button" value="Edit..."/> <input type="button" value="Sort"/>
FEFNet	
FreeNode	
GalaxyNet	
GamesNET	

Skip network list on startup



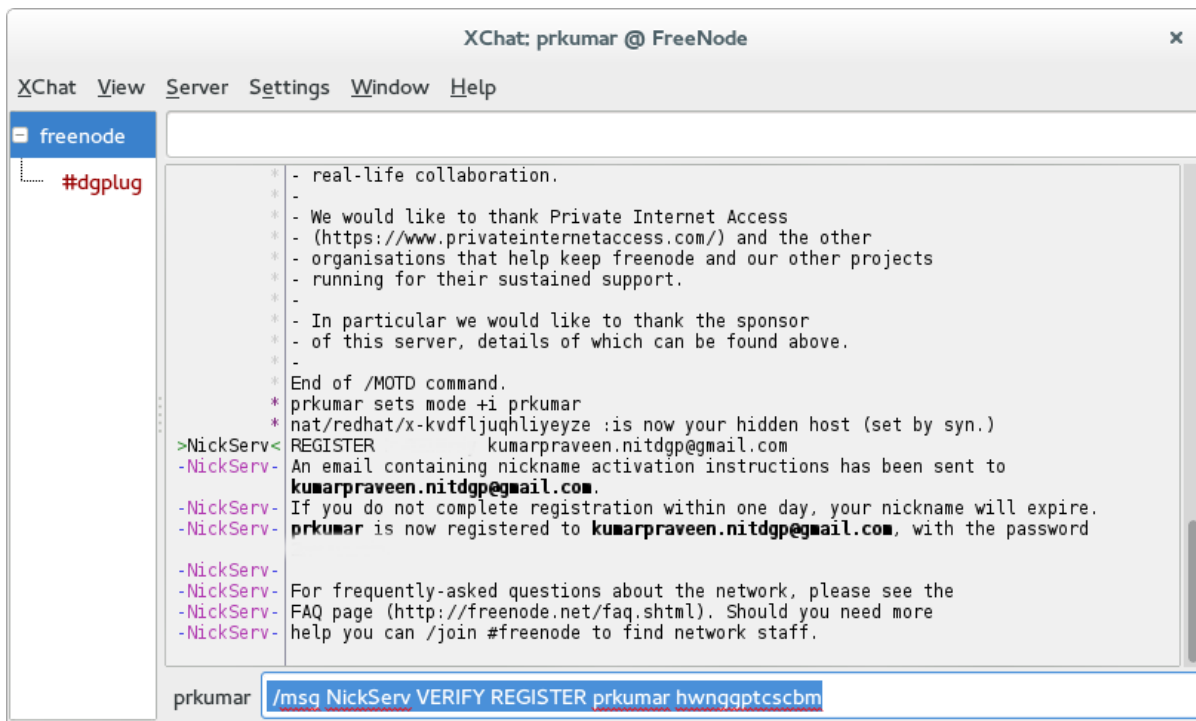
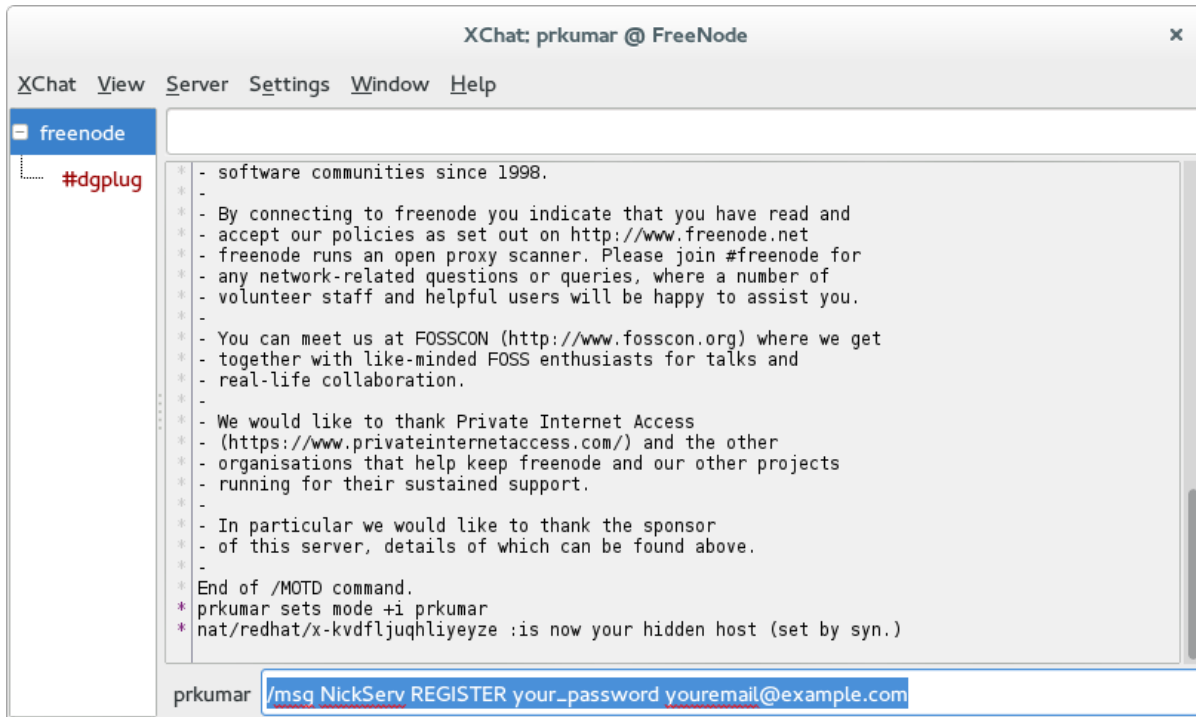


Table of Contents

- *FAQ*
 - *What are the topics covered?*
 - *How do I subscribe to the mailing list?*
 - *What are the requisites for the training?*
 - *How do I get the logs?*
 - *How long does a session run?*
 - *How long is the summer training?*
 - *Will the session details be sent to the mailing list?*
 - *In case I miss a class, how do I catch up?*
 - *Do we need to learn something beforehand for this training?*
 - *How many sessions happen in a day?*
 - *Will the time be the same for all the sessions?*
 - *If I have to leave class early, should I announce it mid class?*

The questions are not in any particular order.

What are the topics covered?

This page contains details about the topics covered in the training.

How do I subscribe to the mailing list?

Go to the [mailing list](#) page, put in your name and email id and click subscribe. Go to your inbox, open the email that you receive from the list, click on the link there and confirm.

What are the requisites for the training?

- A fast Internet connection
- Any Linux distribution. If you only have Windows, you can install it in a VM (say virtualbox).

How do I get the logs?

Find all the old logs at <http://dgplug.org/irclogs/>

How long does a session run?

It depends on that particular class. Generally 1-1.5 hours but some sessions went upto 3 hours. (Though long sessions happen rarely.)

How long is the summer training?

About three months.

Will the session details be sent to the mailing list?

Yes, we do send the details based on the sessions. Remember to keep an eye on the mailing list.

In case I miss a class, how do I catch up?

Read the logs if available, or ask for the logs from a friend. Most things taught, will have detailed docs available.

Do we need to learn something beforehand for this training?

Not really, if you follow the sessions properly you can learn while the training is going on.

How many sessions happen in a day?

Generally just one; rarely there might be more than one session in a day.

Will the time be the same for all the sessions?

Nope, it may change, depending on each session.

If I have to leave class early, should I announce it mid class?

Yes, just inform so and leave.

Shell commands

Linux shell or the terminal is the life line of the developers, and of any power user. Things which can be done on the GUI (by clicking on different buttons), can be done much efficiently on the terminal by using commands. One can not remember all the commands, but with regular usage one can easily remember the most useful ones.

The following guide will introduce you to some basic minimal commands required to use your Linux computer efficiently.

Gnome Terminal

The above is the screenshot of the Gnome terminal application. As you can see the command prompt contains these following information:

```
[username@hostname directoryname]
```

In our case the username is *babai*, hostname is *kdas-laptop*, and directory is mentioned as *~*. This *~* is a special character in our case. It means the home directory of the user. In our case the home directory path is */home/babai/*.

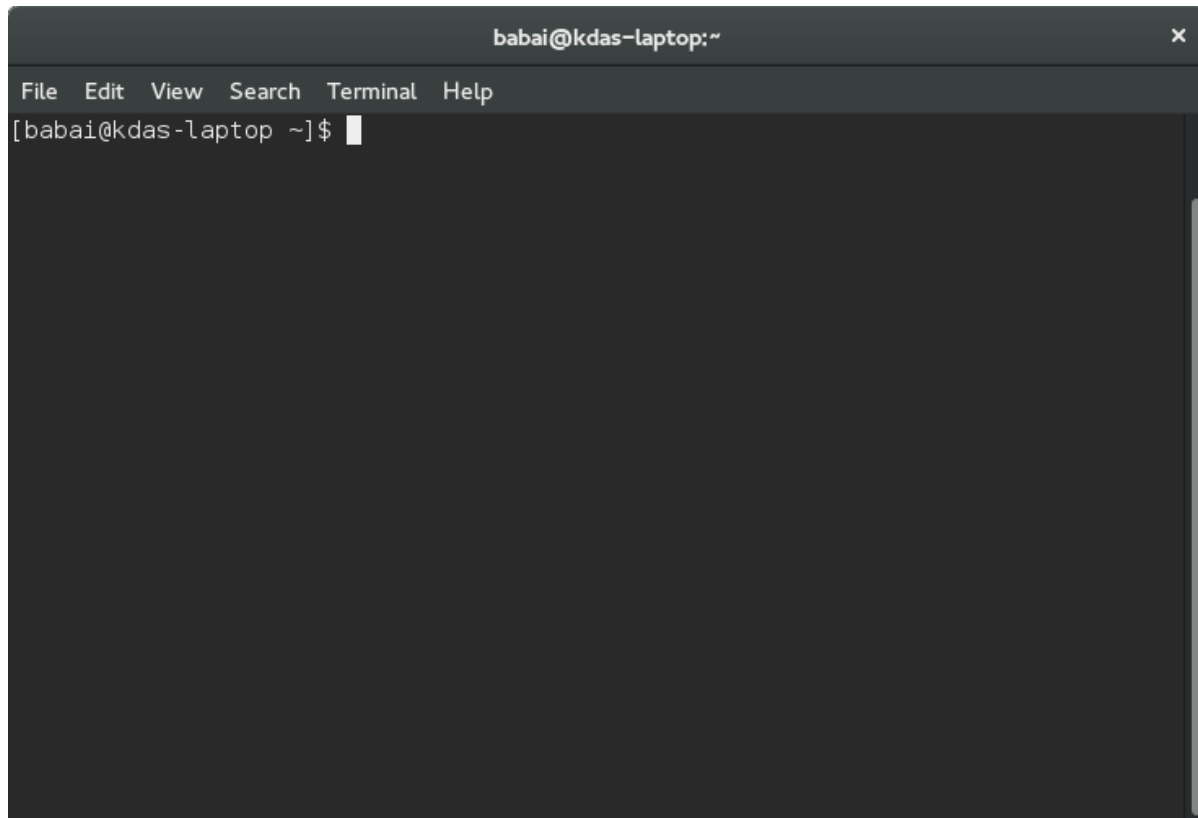
pwd command

pwd command will help you to find out the current directory. Let us see an example below:

```
[babai@kdas-laptop ~]$ pwd  
/home/babai
```

cd command

The next command we will learn is *cd*. This command will help you to change your current directory. We will move to */tmp* directory in our example.:

A screenshot of a terminal window titled 'babai@kdas-laptop:~'. The window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The terminal content shows a shell prompt '[babai@kdas-laptop ~]\$' followed by a cursor. The background is dark grey.

```
[babai@kdas-laptop ~]$ cd /tmp
[babai@kdas-laptop tmp]$ pwd
/tmp
[babai@kdas-laptop tmp]$ cd ~
[babai@kdas-laptop ~]$ pwd
/home/babai
```

Here you can see that first we moved to */tmp* directory, and then we moved back to the home directory by using *~* character.

ls command

We use *ls* command to view the files and directories inside any given directory. If you use *ls* command without any argument, then it will work on the current directory. We will see few examples of the command below.:

```
[babai@kdas-laptop ~]$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
[babai@kdas-laptop ~]$ ls /tmp/
cpython          systemd-private-759094c89c594c07a90156139ec4b969-colord.service-
↪hwUlhR
hogsuspend       systemd-private-759094c89c594c07a90156139ec4b969-rtkit-daemon.
↪service-AwylGa
hsperfdata_babai tracker-extract-files.1000
plugtmp          tracker-extract-files.1002
[babai@kdas-laptop ~]$ ls /
bin  cpython  etc  lib  lost+found  mnt  proc  run  srv  sysroot  usr
```

```
boot  dev      home  lib64  media      opt  root  sbin  sys  tmp      var
```

In the last two commands we provided a path as the argument to the `ls` command. `/` is a special directory, which represents root directory in Linux filesystem. You will know more in the next chapter.

mkdir command

We can create new directories using `mkdir` command. For our example we will create a `code` directory in our home directory.:

```
[babai@kdas-laptop ~]$ mkdir code
[babai@kdas-laptop ~]$ ls
code Desktop Documents Downloads Music Pictures Public Templates Videos
```

We can also create directories in a recursive way using `-p` option.:

```
[babai@kdas-laptop ~]$ mkdir -p dir1/dir2/dir3
[babai@kdas-laptop ~]$ ls dir1/ dir1/dir2/
dir1/:
dir2

dir1/dir2/:
dir3
```

rm command

`rm` command is used to remove a file, or directory. The `-rf` option is being used to remove in a recursive way.:

```
[babai@kdas-laptop ~]$ rm -rf dir1/dir2/dir3
[babai@kdas-laptop ~]$ ls dir1/ dir1/dir2/
dir1/:
dir2

dir1/dir2/:
```

tree command

`tree` command prints the directory structure in a nice visual tree design way.:

```
[babai@kdas-laptop ~]$ tree
.
- code
- Desktop
- dir1
|   - dir2
- Documents
- Downloads
- Music
- Pictures
|   - terminal1.png
```

- Public
- Templates
- Videos

CHAPTER 6

Filesystem Hierarchy Standard(FHS)

This guide has details about the basics of Filesystem Hierarchy Standard.

Just read the following [log](#) and keep reading the [Book](#).

A newer version of the FHS can be found [here](#).

Reverting some uncommitted change

Say now we delete the file **hello.txt** from the directory by mistake. *hg status* now shows that in output.

```
$ hg status
! hello.txt
```

We can revert back all the changes in our repo by using the following command.

```
$ hg revert --all
reverting hello.txt
```

In case you made some changes to `bye.txt` and want to revert only that file, you can do the following:

```
$ hg revert bye.txt
```

Removing a file

Let us add another text file named **deleteme.txt**, write anything inside of that file.:

```
$ hg add deleteme.txt
$ hg commit -m "Adds deleteme.txt file."
```

Now to remove it from the repository you can use `remove` command, remember to commit after removing.:

```
$ hg remove deleteme.txt
$ hg status
$ R deleteme.txt
$ hg commit -m "Removes deleteme.txt file"
```

We can see the commit log now.:

```
$ hg log
changeset: 4:5b1b400f84c3
tag: tip
user: Kushal Das <kushaldas@gmail.com>
date: Wed Jul 16 17:34:40 2014 +0530
summary: Removes deleteme.txt file

changeset: 3:5022651fd32b
user: Kushal Das <kushaldas@gmail.com>
date: Wed Jul 16 17:34:16 2014 +0530
summary: Adds deleteme.txt file.

changeset: 2:0e558dd2a91d
user: Kushal Das <kushaldas@gmail.com>
date: Mon Jul 14 20:53:32 2014 +0530
summary: Adds bye.txt file and fixes hello.txt.

changeset: 1:48bfb8a095cb
user: Kushal Das <kushaldas@gmail.com>
date: Mon Jul 14 20:29:59 2014 +0530
summary: Second commit.

changeset: 0:a4754244eb89
user: Kushal Das <kushaldas@gmail.com>
date: Mon Jul 14 20:00:14 2014 +0530
summary: First change.
```

Difference between two revisions

Using *hg diff* command we can actually see what all changed between two revisions.:

```
$ hg diff -r 0:1
diff -r a4754244eb89 -r 48bfb8a095cb hello.txt
--- a/hello.txt Mon Jul 14 20:00:14 2014 +0530
+++ b/hello.txt Mon Jul 14 20:29:59 2014 +0530
@@ -1,1 +1,1 @@
-Hello World.
+Hello World!
```

Now think how to do the same for one particular file! :)

Bitbucket

Visit [the site](#) and create an account there. After you login goto to manage account and add your ssh key there.

The screenshot shows the Bitbucket user interface. At the top, there is a navigation bar with 'Bitbucket', 'Dashboard', 'Teams', 'Repositories', and 'Create' buttons. The user's name 'Kushal das' is visible in the top right. A dropdown menu is open, showing options: 'View profile', 'Manage account', 'Inbox (0)', 'Invite a friend', and 'Log out'. The main content area is titled 'Manage' and shows 'SSH keys'. A blue 'Add key' button is present. Below it, a table lists the keys:

Key	Added
home	Not recorded Edit

The left sidebar contains various settings categories: GENERAL (Account settings, Email addresses, Notifications, Custom domain, Change username, Delete account), PLANS AND BILLING (Plan details), ACCESS MANAGEMENT (User groups, OAuth), and SECURITY (Change password, SSH keys, Connected accounts, Sessions, Audit log).

Now click on the Create button on the top menu to create a new repository. See the screenshot below.

The screenshot shows the 'Create a new repository' form. The title is 'Create a new repository' and there is a link 'You can also import a repository'. The form fields are:

- Name*: myproject
- Description: This is a test project.
- Access level: This is a private repository
- Repository type: Git, Mercurial
- Project management: Issue tracking, Wiki
- Language: Unspecified

At the bottom, there are 'Create repository' and 'Cancel' buttons. On the right side, there are two informational boxes: 'New to Bitbucket?' with a graduation cap icon and 'Working in a team?' with a group of people icon.

Now create the repository, which will bring you to the next screen. As we already have a repository in our computer, we will use the second link there.



Repository setup

Your repository is empty — let's put some bits in your bucket.

Get code into Bitbucket fast using the command line

Command line

> [I'm starting from scratch](#)

> [I have an existing project](#)

Need a complete walkthrough? Visit [Bitbucket 101](#) for a complete tutorial

This will show the command required to push it to the repository so that everyone can view it.



Repository setup

Your repository is empty — let's put some bits in your bucket.

Get code into Bitbucket fast using the command line

Command line

> [I'm starting from scratch](#)

▼ [I have an existing project](#)

Already have a Mercurial repository on your computer? Let's push it up to Bitbucket.

```
$ cd /path/to/my/repo
$ hg push ssh://hg@bitbucket.org/kushaldas/myproject
```

Edit this repository's **.hg/hgrc** to add your new repository

Want to grab a repo from another site? Try our [importer!](#)

I am running the command as given there.:

```
$ hg push ssh://hg@bitbucket.org/kushaldas/myproject
pushing to ssh://hg@bitbucket.org/kushaldas/myproject
searching for changes
remote: adding changesets
remote: adding manifests
remote: adding file changes
```

```
remote: added 5 changesets with 5 changes to 3 files
```

Now you can also create a file at **.hg/hgrc** so that you can just push/pull from you repository easily.:

```
[paths]
default = ssh://hg@bitbucket.org/kushaldas/myproject
```


We use editors to create or edit text files. For many of the dgplug's member it is the de-facto editor for programming.

How to install?

For Fedora:

```
$ sudo yum install vim
```

For Ubuntu

```
$ sudo apt-get install vim
```

Launch vim

To open vim editor, type

```
$ vim
```

To open a file in vim editor, type

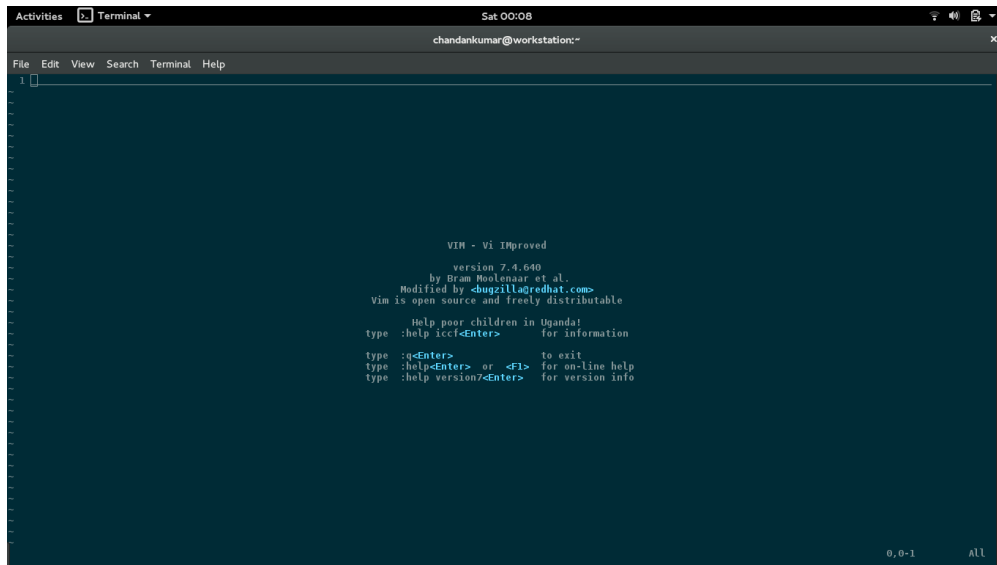
```
$ vim hello.txt
```

It will open a file 'hello.txt' in vim editor. It is in normal mode. If it is a new file, it will be empty. If it is an old file, then it will show some contents.

To write something in the file, Press *i* from the keyboard then type something in the file.

Now the vi editor is in INSERT mode.

To save the file, first press *Esc*, then type `:w` followed by *Enter* to save changes in the file. By pressing `:q`, we can close the file. Now, you are back to the terminal prompt. We can use `:x` to save the contents of the file and close the file.

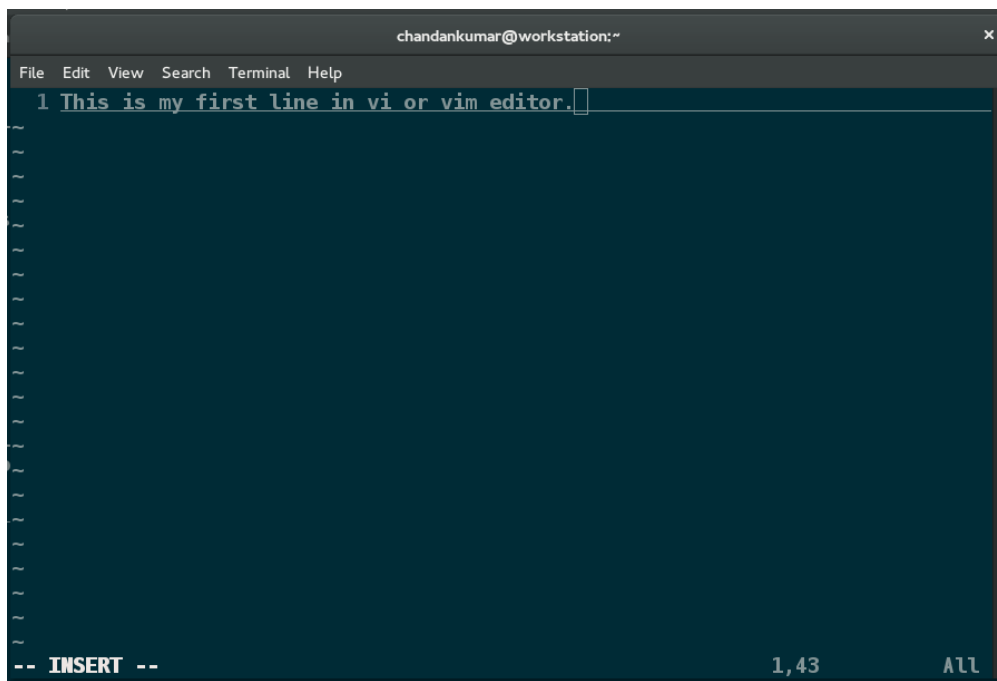


A terminal window titled "Terminal" with a menu bar (File, Edit, View, Search, Terminal, Help) and a status bar (Sat 00:08, chandankumar@workstation:~). The terminal displays the Vim startup screen with the following text:

```
VIM - VI IMproved
      version 7.4.640
      by Bram Moolenaar et al.
      Modified by <borgzillagredhat.com>
      Vim is open source and freely distributable

      Help poor children in Uganda!
      type :help icfc<Enter>      for information
      type :q<Enter>              to exit
      type :help<Enter> or <F1>   for on-line help
      type :help version7<Enter> for version info
```

The status bar at the bottom right shows "0,0-1" and "All".



A terminal window titled "Terminal" with a menu bar (File, Edit, View, Search, Terminal, Help) and a status bar (chandankumar@workstation:~). The terminal shows the Vim editor in INSERT mode. The first line of the file contains the text "1 This is my first line in vi or vim editor." followed by a cursor. The status bar at the bottom left shows "-- INSERT --" and the bottom right shows "1,43" and "All".

Let us again open the same file in vim editor, press *i*, then type random text. Now if you donot want to save, but want to go back to the last saved revision. just type *:e!* after pressing escape. If you donot want to save anything and also exit then type *':q!*'.

Movement with in a file

Let us reopen the file and do some basic movements with in the file.

Press *h* to move 1 char left

Press *j* to move 1 char down

Press *k* to move 1 char up

Press *l* to move 1 char right

To move word by word, press *w*

To come back word by word, press *b*

To go back to the beginning of the line, press *0*

To go back end of the line, press *\$*

Cut, Copy, Paste

e - To edit some text at the current cursor position

a - To append some text at the current cursor position

r - To replace to with some new character

x - To delete a character

To change a word - *cw*

To change from the cursor to the end of the line - *c\$*

To change from the cursor to the beginning of the line - *c0*

To change a whole line : *cc*

d - Too delete a character

dw - Too delete a word

d\$ - Too delete up to end of the line

dd - Too cut/delete a whole line

d0 - To delete at the beginning of the line

x - To cut a line

xw - To cut a word

xx - To cut a whole line

y - To copy

yy - To copy a whole line

p - To paste

To copy 2 lines - `2yy`

To delete 2 words - `2dw`

To see line number - `:set nu`

To go to particular line - `: <line number>`

To turn off line number - `:set nu!`

Playing with multiple files

To open multiple files

```
$ vi file1 file2 .... fileN
```

To move around the files

`ctrl + F` → Forward one screen

`ctrl + B` → Scroll backward

`ctrl + D` → Forward half screen

`ctrl + U` → Backward half screen

Find and Replace

to search a string: `/<searchstring>`

Press `n` to get to the next occurrences

To move search in opposite direction - `N`

To search specific text backward - `?searchtext`

`:s/oldstring/newstring` - will replace the first occurrence of the oldstring to the newstring

To replace all occurrences of the oldstring to new string in that line : `:s/oldstring/newstring/g`

To replace in all lines : `:1,$s/oldstring/newstring/g`

File Splitting in vim

To split the file, type

`:split` - for horizontal split

`:vsplit` - for vertical split

Word Completion

In insert mode press `Ctrl + p` or `Ctrl + n` to do word completion

Where to go Next

To read more about Vim editor, open the terminal and type

```
$ vimtutor
```


CHAPTER 9

Indices and tables

- `genindex`
- `modindex`
- `search`