

Git environment settings

Before you can use Git, you must install and make some configuration changes. Here are the steps to install Git client on Ubuntu and Centos Linux.

Before you can use Git, you must install and make some configuration changes. Here are the steps to install Git client on Ubuntu and Centos Linux.

Install Git client

If you are using Debian distributed by GNU / Linux, the following apt-get command is required.

```
root @ ubuntu: ~ # sudo apt-get install git-core
[sudo] password for root:

root @ ubuntu: ~ # git --version
git version 1.9.1
```

And if you are using RPM distributed by GNU / Linux, you use the yum command as below:

```
[CentOS ~] $
su -
Password:

[CentOS ~] # yum -y install git-core

[CentOS ~] # git --version
git version 1.7.1
```

Customize Git environment

Git provides a git configuration tool that allows you to set up various configurations. Git saves all common configurations in the .gitconfig file located in your home directory. To set up configuration values like a global, you add the --global option, and if you want to remove this option, your configurations are specifically defined for the current Git repository.

You can also set up empty configuration for the system. Git stores the values in / etc / gitconfig, which contains configuration for all users and repositories on the system. To set these values, you must have the right root and use the --system option.

When the above code is compiled and executed, it produces the following result:

Set user name

This information is used by Git for every commit.

```
root @ ubuntu: ~ # git config --global user.name "QTM"
```

Set up email id

This information is used by Git for every commit.

```
root @ ubuntu: ~ # git config --global user.email "QTMteam@gmail.com"
```

Avoid merging commits when pulling

You pull the latest changes from the remote repository and if these changes are different or divergent, then by default Git creates merged commits. We can avoid this according to the following settings:

```
root @ ubuntu: ~ # git config --global branch.autosetuprebase always
```

Floating color

The following commands create color in the Git control panel.

```
[root @ ubuntu: ~ # git config --global color.ui true
```

```
[root @ ubuntu: ~ # git config --global color.status auto
```

```
[root @ ubuntu: ~ # git config --global color.branch auto
```

Set the default editor

By default, Git uses the default editor, which is taken from VISUAL or EDITOR. You can specify another editor using config.

```
[root @ ubuntu: ~ # git config --global core.editor vim
```

Set the default merging tool

Git does not provide a default merging tool for interacting with opposing changes in the working tree. You can set up this tool according to the following settings:

```
[root @ ubuntu: ~ # git config --global merge.tool vimdiff
```

List Git settings

To verify the Git settings in the local repository, use the git config -list command as shown below.

```
root @ ubuntu: ~ # git config --list
```

The above command will produce the following result:

```
user.name = Viet Jack
user.email=QTMteam@gmail.com
push.default = nothing
branch.autosetuprebase = always
color.ui = true
color.status = auto
color.branch = auto
core.editor = vim
merge.tool = vimdiff
```

According to Tutorialspoint

Previous article: [Basic about Git](#)

Next lesson: [Git life cycle](#)

You finished reading the article "**Git environment settings**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.