

假设经过上一篇文章的学习，你已经在系统上安装了 Git，那么现在要做几件事来配置/定制 Git 环境。每台计算机上只需要配置一次，程序升级时会保留配置信息。可以在任何时候再次通过运行命令来修改它们。

注：如果还没有安装 Git，请参考：http://www.yiibai.com/git/git_environment.html

Git 自带一个 `git config` 的工具来帮助设置控制 Git 外观和行为的配置变量。这些变量存储在三个不同的位置：

`/etc/gitconfig` 文件：包含系统上每一个用户及他们仓库的通用配置。如果使用带有 `--system` 选项的 `git config` 时，它会从此文件读写配置变量。

`~/.gitconfig` 或 `~/.config/git/config` 文件：只针对当前用户。可以传递 `--global` 选项让 Git 读写此文件。

当前使用仓库的 Git 目录中的 `config` 文件(就是 `.git/config`)：针对该仓库。

每一个级别覆盖上一级别的配置，所以 `.git/config` 的配置变量会覆盖 `/etc/gitconfig` 中的配置变量。

在 Windows 系统中，Git 会查找 `$HOME` 目录下(一般情况下是 `C:\Users\%USER%`)的 `.gitconfig` 文件。Git 同样也会寻找 `/etc/gitconfig` 文件，但只限于 MSys 的根目录下，即安装 Git 时所选的目标位置。

用户信息

当安装完 Git 应该做的第一件事就是设置用户名称与邮件地址。这样做很重要，因为每一个 Git 的提交都会使用这些信息，并且它会写入到每一次提交中，不可更改：

```
$ git config --global user.name "maxsu"
$ git config --global user.email maxsu@yiibai.com
```

再次强调，如果使用了 `--global` 选项，那么该命令只需要运行一次，因为之后无论你在该系统上做任何事情，Git 都会使用这些信息。当你想针对特定项目使用不同的用户名称与邮件地址时，可以在那个项目目录下运行不使用 `--global` 选项的命令来配置。

很多 GUI 工具都会第一次运行时帮助你配置这些信息。

文本编辑器

既然用户信息已经设置完毕，你可以配置默认文本编辑器了，当 Git 需要您输入信息时会调用它。如果未配置，Git 会使用操作系统默认的文本编辑器，通常是 `Vim`。如果你想使用不同的文本编辑器，例如 `Emacs`，可以这样做：

```
$ git config --global core.editor emacs
```

注：Vim 和 Emacs 是像 Linux 与 Mac 等基于 Unix 的系统上开发者经常使用的流行的文本编辑器。如果你对 these 编辑器都不是很了解或者你使用的是 Windows 系统，那么可能需要搜索如何在 Git 中配

置你最常用的编辑器。如果你不设置编辑器并且不知道 Vim 或 Emacs 是什么，当它们运行起来后你可能会被弄糊涂、不知所措。

检查配置信息

如果想要检查你的配置，可以使用 `git config --list` 命令来列出所有 Git 当时能找到的配置。

```
yiibai@ubuntu:~$ git config --list
user.name=maxsu
user.email=maxsu@yiibai.com
color.status=auto
color.branch=auto
color.interactive=auto
color.diff=auto
...
```

上面的输出结果中，可能会看到重复的变量名，因为 Git 会从不同的文件中读取同一个配置 (例如：`/etc/gitconfig` 与 `~/.gitconfig`)。这种情况下，Git 会使用它找到的每一个变量的最后一个配置。

可以通过输入 `git config <key> :` 来检查 Git 的某一项配置。

```
yiibai@ubuntu:~$ git config user.name
maxsu
yiibai@ubuntu:~$ git config user.email
maxsu@yiibai.com
yiibai@ubuntu:~$
```

获取帮助

若在使用 Git 时需要获取帮助，有三种方法可以找到 Git 命令的使用手册：

```
$ git help <verb>
$ git <verb> --help
$ man git-<verb>
```

例如，要想获得 `config` 命令的手册，执行

```
$ git help config
```

这些命令很棒，因为随时随地可以使用而无需联网。如果你觉得手册或者本书的内容还不够用，你可以尝试在 Freenode IRC 服务器(`irc.freenode.net`)的 `#git` 或 `#github` 频道寻求帮助。这些频道经常有上百人在线，他们都精通 Git 并且乐于助人。