

Git 安装配置

在使用Git前我们需要先安装 Git。Git 目前支持 Linux/Unix、Solaris、Mac和 Windows 平台上运行。

Git 各平台安装包下载地址为：<http://git-scm.com/downloads>

Linux 平台上安装

Git 的工作需要调用 curl , zlib , openssl , expat , libiconv 等库的代码，所以需要先安装这些依赖工具。

在有 yum 的系统上（比如 Fedora）或者有 apt-get 的系统上（比如 Debian 体系），可以用下面的命令安装：

各 Linux 系统可以使用其安装包管理工具（apt-get、yum 等）进行安装：

Debian/Ubuntu

Debian/Ubuntu Git 安装命令为：

```
$ apt-get install libcurl4-gnutls-dev libexpat1-dev gettext \
libz-dev libssl-dev

$ apt-get install git

$ git --version
git version 1.8.1.2
```

Centos/RedHat

如果你使用的系统是 Centos/RedHat 安装命令为：

```
$ yum install curl-devel expat-devel gettext-devel \
openssl-devel zlib-devel

$ yum -y install git-core

$ git --version
git version 1.7.1
```

源码安装

我们也可以在官网下载源码包来安装，最新源码包下载地址：<https://git-scm.com/download>

安装指定系统的依赖包：

```
##### Centos/RedHat #####

$ yum install curl-devel expat-devel gettext-devel \
openssl-devel zlib-devel
```

```
##### Debian/Ubuntu #####  
$ apt-get install libcurl4-gnutls-dev libexpat1-dev gettext \  
libz-dev libssl-dev
```

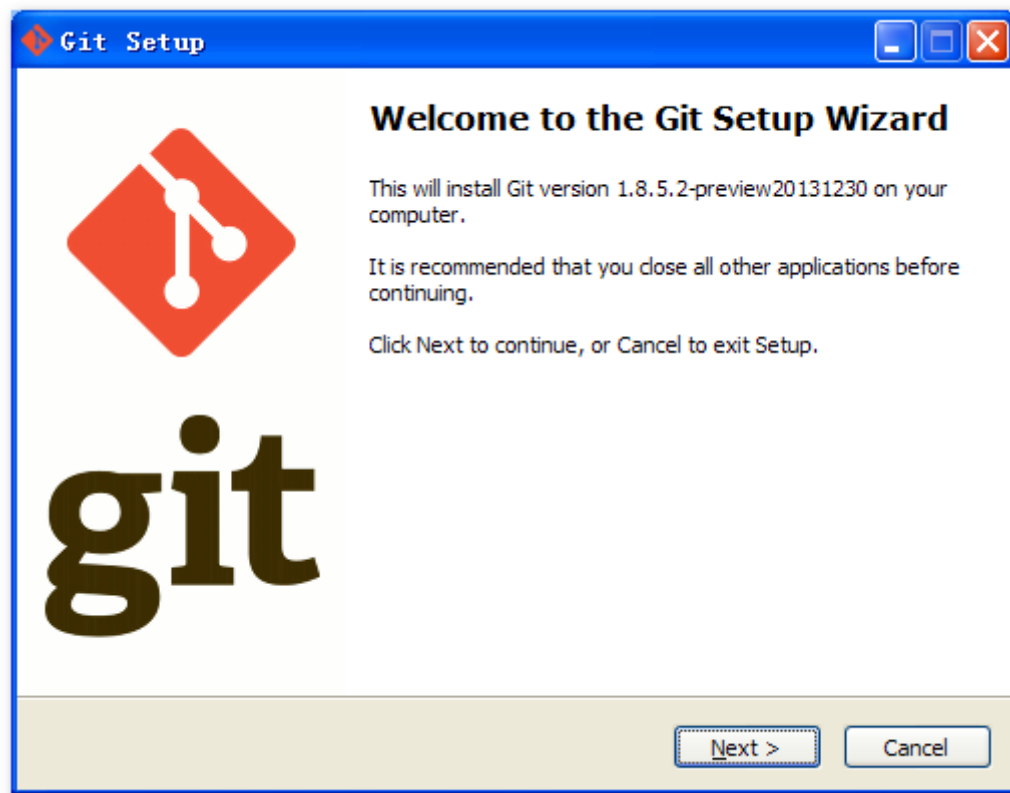
解压安装下载的源码包：

```
$ tar -zxf git-1.7.2.2.tar.gz  
$ cd git-1.7.2.2  
$ make prefix=/usr/local all  
$ sudo make prefix=/usr/local install
```

Windows 平台上安装

在 Windows 平台上安装 Git 同样轻松，有个叫做 msysGit 的项目提供了安装包，可以到 GitHub 的页面上下载 exe 安装文件并运行：

安装包下载地址：<https://gitforwindows.org/>



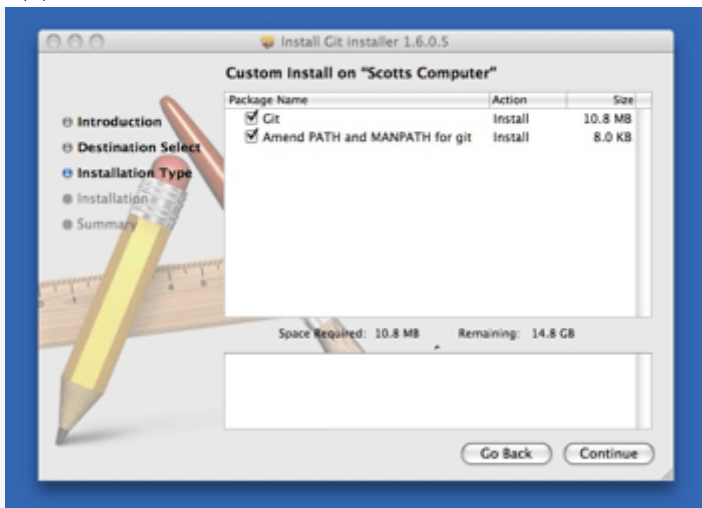
完成安装之后，就可以使用命令行的 git 工具（已经自带了 ssh 客户端）了，另外还有一个图形界面的 Git 项目管理工具。在开始菜单里找到"Git"->"Git Bash"，会弹出 Git 命令窗口，你可以在该窗口进行 Git 操作。

Mac 平台上安装

在 Mac 平台上安装 Git 最容易的当属使用图形化的 Git 安装工具，下载地址为：

<http://sourceforge.net/projects/git-osx-installer/>

安装界面如下所示：



Git 配置

Git 提供了一个叫做 `git config` 的工具，专门用来配置或读取相应的工作环境变量。

这些环境变量，决定了 Git 在各个环节的具体工作方式和行为。这些变量可以存放在以下三个不同的地方：

- `/etc/gitconfig` 文件：系统中对所有用户都普遍适用的配置。若使用 `git config` 时用 `--system` 选项，读写的就是这个文件。
- `~/.gitconfig` 文件：用户目录下的配置文件只适用于该用户。若使用 `git config` 时用 `--global` 选项，读写的就是这个文件。
- 当前项目的 Git 目录中的配置文件（也就是工作目录中的 `.git/config` 文件）：这里的配置仅仅针对当前项目有效。每一个级别的配置都会覆盖上层的相同配置，所以 `.git/config` 里的配置会覆盖 `/etc/gitconfig` 中的同名变量。

在 Windows 系统上，Git 会找寻用户主目录下的 `.gitconfig` 文件。主目录即 `$HOME` 变量指定的目录，一般都是 `C:\Documents and Settings\%USER%`。

此外，Git 还会尝试找寻 `/etc/gitconfig` 文件，只不过看当初 Git 装在什么目录，就以此作为根目录来定位。

用户信息

配置个人的用户名称和电子邮件地址：

```
$ git config --global user.name "runoob"
$ git config --global user.email test@runoob.com
```

如果用了 `--global` 选项，那么更改的配置文件就是位于你用户主目录下的那个，以后你所有的项目都会默认使用这里配置的用户信息。

如果要在某个特定的项目中使用其他名字或者电邮，只要去掉 `--global` 选项重新配置即可，新的设定保存在当前项目的 `.git/config` 文件里。

文本编辑器

设置 Git 默认使用的文本编辑器，一般可能会是 Vi 或者 Vim。如果你有其他偏好，比如 Emacs 的话，可以重新设置：

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

差异分析工具

还有一个比较常用的是，在解决合并冲突时使用哪种差异分析工具。比如要改用 vimdiff 的话：

```
$ git config --global merge.tool vimdiff
```

Git 可以理解 kdiff3，tkdiff，meld，xxdiff，emerge，vimdiff，gvimdiff，ecmerge，和 opendiff 等合并工具的输出信息。当然，你也可以指定使用自己开发的工具，具体怎么做可以参阅第七章。

查看配置信息

要检查已有的配置信息，可以使用 git config --list 命令：

```
$ git config --list
http.postbuffer=2M
user.name=runoob
user.email=test@runoob.com
```

有时候会看到重复的变量名，那就说明它们来自不同的配置文件（比如 /etc/gitconfig 和 ~/.gitconfig），不过最终 Git 实际采用的是最后一个。

这些配置我们也可以在 ~/.gitconfig 或 /etc/gitconfig 看到，如下所示：

```
vim ~/.gitconfig
```

显示内容如下所示：

```
[http]
  postBuffer = 2M
[user]
  name = runoob
  email = test@runoob.com
```

也可以直接查阅某个环境变量的设定，只要把特定的名字跟在后面即可，像这样：

```
$ git config user.name
runoob
```

← Git 教程

Git 工作流程 →



1 篇笔记

写笔记



1、最新git源码下载地址：

<https://github.com/git/git/releases>

<https://www.kernel.org/pub/software/scm/git/>

可以手动下载下来在上传到服务器上面

2 移除旧版本git

centos自带Git，7.x版本自带git 1.8.3.1（应该是，也可能不是），安装新版本之前需要使用yum remove git卸载（安装后卸载也可以）。

```
[root@Git ~]# git --version    ## 查看自带的版本
git version 1.8.3.1
[root@Git ~]# yum remove git    ## 移除原来的版本
```

3 安装所需软件包

```
[root@Git ~]# yum install curl-devel expat-devel gettext-devel openssl-devel zlib-devel
[root@Git ~]# yum install gcc-c++ perl-ExtUtils-MakeMaker
```

下载&安装

```
[root@Git ~]# cd /usr/src
[root@Git ~]# wget https://www.kernel.org/pub/software/scm/git/git-2.7.3.tar.gz
```

5 解压

```
[root@Git ~]# tar xf git-2.7.3.tar.gz
```

6 配置编译安装

```
[root@Git ~]# cd git-2.7.3
[root@Git ~]# make configure
[root@Git ~]# ./configure --prefix=/usr/git ##配置目录
[root@Git ~]# make prefix=/usr/git
[root@Git ~]# make install
```

7 加入环境变量

```
[root@Git ~]# echo "export PATH=$PATH:/usr/git/bin" >> /etc/profile
[root@Git ~]# source /etc/profile
```

8 检查版本

```
[root@Git git-2.7.3]# git --version
git version 2.7.3
```

宋某人C 2年前 (2017-03-23)