

# jamesmf的专栏

目录视图 摘要视图 RSS 订阅

个人资料



jamesmf

访问： 97346次  
积分： 1287  
等级： BLOC > 4  
排名： 千里之外  
  
原创： 23篇    转载： 56篇  
译文： 0篇    评论： 17条

文章搜索

文章分类

[C/C++](#) (23)  
[好文转载](#) (26)  
[编译相关](#) (7)

文章存档

[2013年12月](#) (3)  
[2013年09月](#) (3)

【免费公开课】[C语言指针与汇编内存地址](#)    [有奖试读—漫话程序员面试求职、升职加薪、创业与生活](#)    [chinapub读书会第8期：程序员代码面试揭秘](#)

## 15分钟学会使用Git和远程代码库

2013-12-22 12:59    1797人阅读    [评论\(0\)](#)    [收藏](#)    [举报](#)

分类： [好文转载](#) (25)

[目录\(?\)](#) [\[+\]](#)

本文由 [伯乐在线](#) - [cjpan](#) 翻译自 [Nico](#)。欢迎加入[技术翻译小组](#)。转载请参见文章末尾处的要求。

Git是个了不起但却复杂的源代码管理系统。它能支持复杂的任务，却因此经常被认为太过复杂而不适用于简单的日常工作。让我们诚实一记吧：**Git**是复杂的，我们不要装作它不是。但我仍然会试图教会你用（我的）基本的**Git**和远程代码库干活的工作步骤，在**15分钟**内。

### 工作步骤

我会展示以下的步骤，通常能帮我独自在一台或多台机器上做项目。

1. 创建一个远程的空代码库（在[BitBucket](#)上）
2. 在本地代码库添加一个项目
3. 在分支上开发新功能
4. **a)** 保留新功能 或者 **b)** 丢弃它们
5. 也许，回到某个早先的时间点
6. 将本地代码库推送到远程代码库
7. 在另一台机器上取得远程代码库

### 安装Git

在大多数\*nix系统（Linux、OS X）上，Git已经被安装了。你通过发送下面的命令，可以通过Git自身，把它更新到

2013年06月 (1)

2012年07月 (5)

2012年05月 (1)

展开

阅读排行

问题 error C2065: '\_begi' (8346)

VTD-XML (8115)

XML解析技术之VTD-XML (8106)

WinPcap编程错误: erro (7908)

C/C++笔试编程题集 (4784)

LeakDiag的使用 (3270)

持续集成反模式 (2466)

C语言中的多级指针 (2458)

如何处理构造函数的失败 (2023)

15分钟学会使用Git和远和 (1796)

评论排行

Lex和Yacc从入门到精通 (4)

WinPcap编程错误: erro (2)

注册表中保存的VC安装跟 (2)

问题 error C2065: '\_begi' (1)

C C++语言—入门必做的 (1)

LeakDiag的使用 (1)

各种计算机语言的经典书 (1)

如何在VS2005下编译安 (1)

变参函数的实现原理 (1)

如何处理构造函数的失败 (1)

推荐文章

\*Android官方开发文档Training系列课程中文版: 高效显示位图之在非UI线程中处理图片

\*Binder工作机制

\* Java Web基础知识之Filter: 过滤一切你不想看到的事情/a>

\*Untiy Native Render Plugin在VR中的绘制(二): 透明排序

最新的开发版本（不推荐）。

```
git
clone https://github.com/git/git
```

在Windows上，你可以在[这里](#)下载Git的安装程序。如果你真的需要其他系统的安装程序，Mac OS X安装文件在[这里](#)，Linux的操作指导在[这里](#)。

## 创建一个远程代码库

很多人喜欢用[Github](#)。我个人更喜欢[BitBucket](#)，因为它提供了不限制的私有代码库，那是我最需要的。你可以将下列指令转换到Github上，这些过程是相同的。

那么，去到[www.bitbucket.org](#)并注册一个账号。一旦完成，登录后点击最上方的“create（创建）”按钮。照着填写表格，勾选私有代码库。你可不想让其他人来偷窥你的Facebook的杀手级应用的源代码，对吧。

你现在可以离开BitBucket了，我们在已经有了所有那里需要的东西了。

## 设置Git

在我们能用Git工作之前，我们需要做个一次性的配置。为了Git能跟踪到谁做了修改，我们需要设置你的用户名。我强烈建议你使用与注册BitBucket账号相同的用户名和电子邮箱地址。发送这些命令，相应地替换掉其中的“your\_username”和“your\_email@domain.com”（注意引号）：

```
git
config --global user.name "your_username"

git
config --global user.email your_email@domain.com
```

我们也会设定推送（push）的默认值为‘simple’。要了解这是什么意思，快速阅读我之前发布的[关于推送的默认值](#)（非必须）。发送这条命令：

```
git
config --global push.default simple
```

我们都设好了。你无需在你的机器上再重复这些配置，但如果你在另一台机器上工作的话，不要忘记这些配置。如果你忘记做初始的配置，Git不会允许你提交任何东西，这会让你困扰。

## 创建一个本地代码库

作为例子，我们会假装我们有一个网站（无所谓技术）存在于我们机器上的‘workspace’文件夹下的‘my\_site’文件夹内。在命令行中，去到你的站点的根文件夹。在OS X和Linux上：

d

\*随机过程--Metropolis-Hastings 算法

\*Fresco 图片库研读分析

最新评论

XML解析技术之VTD-XML 简介及 david0624: 给力,既有理论,又有实践.详细地解析了各个技术在处理xml的时候的优缺点,让大家更理智地选择.

WinPcap编程错误: error C2075: 145435475: 多谢楼主,看了好长时间了,终于知道是什么原因了.

VTD-XML banrisun: 赞一个

如何处理构造函数的失败? aaronlin999: good

变参函数的实现原理 慌张凉凉: 你好! 我这样实现的变参数算真的变参数嘛? 求指教 #include long mul2(int i,.....

注册表中保存的VC安装路径 good\_jianhong: 如您所说,他的路径是D:\Microsoft Visual Studio\VC98这样子的,但是安装...

WinPcap编程错误: error C2075: kongzhi\_ren: 说的是实话, VC6.0 的库函数比较老, VS2010的是比较新的

Lex和Yacc从入门到精通(6)-解析 wenbranger: 问题已解决!

Lex和Yacc从入门到精通(6)-解析 wenbranger: 我看了相关的解决办法: http://www.cs.man.ac.uk/~pjj/cs21

Lex和Yacc从入门到精通(6)-解析 wenbranger: 另外的两条错误信息: "frame.l", line 44: unrecognized..

```
./workspace/my_site/
```

在Windows上:

d

```
: \workspace\my_site
```

我们首先需要告诉Git这个文件夹是我们需要跟踪的项目。所以我们发送这个命令来初始化一个新的本地Git代码库

```
git
init
```

Git会在my\_site文件夹内创建一个名为.git的隐藏文件夹，那就是你的本地代码库。

## 加载（Stage）文件

我们现在需要命令Git我们需要加载（stage）所有项目文件。发送：

```
git
add .
```

最后的“.”符号的意思是“所有文件、文件夹和子文件夹”。假如我们只想要把特定文件添加到源代码控制中去，我们可以指定它们：

```
git
add my_file, my_other_file
```

## 提交文件

现在，我们想要提交已加载（staged）的文件。阅读“添加一个时间点，在这里你的文件处在一个可还原的状态”。我们提交我们的文件时，总是附着着有意义的注释，描述了它们现在的状态。我一直用“initial commit”来作为第一个提交的注释。

```
git
commit -m "initial
commit"
```

就这样。现在你随时都可以回滚到这个提交状态。如果你有需要检查你现在的已加载（staged）和未加载（unstaged）文件的状态、提交等，你可以询问git的状态：

```
git
status
```

# 创建分支

建立分支是你创建代码的独立版本的动作，独立于你的主干分支。默认地，每次你提交到**Git**的文件都会被储存在“**master**（主干）”分支。

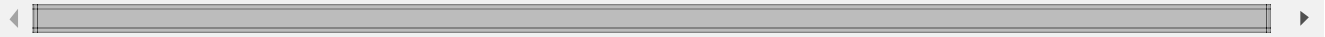
现在我们来说说，你想要向项目里添加一个功能，但你想要能够回滚到现在版本，以防出现差错，或者你决定要放弃这个功能。这就是你创建分支的时候了。创建并同时切换到你新建的分支，发送：

```
.it
checkout -b new_feature
```



或者，你可以先创建一个分支然后手动切换，就像这样：

```
.it
branch new_featuregit checkout new_feature
```



要看你现在项目下所有的分支，发送这个：

```
.it
branch
```



现在你可以在你的项目上无所顾忌地做任何你想做的：任何时候，你都可以回到你创建分支前的状态。注意，你同时可以有多个分支，甚至可以从一个分支上再创建一个分支。

# 合并分支

当你对你的新功能满意了的时候，你想要把它加到主干分支上。当你在你的新功能分支上时，你首先需要加载（**stage**）并且提交你的文件：

```
.it
add .git commit -m "adds
my new feature"
```



然后你移到你的主干分支：

```
.it
checkout master
```



像这样合并：

```
.it
merge new_feature
```

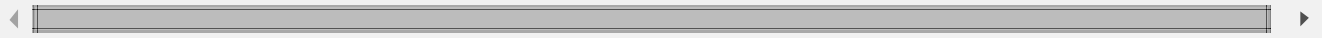


此时，你的主干分支和你的新功能分支会变成一样的了。

# 丢弃分支

相反，如果你打算丢弃你在分支里做的修改，你首先需要加载（stage）你的文件并且在分支里提交：

```
.it
add .git commit -m "feature
to be discarded"
```



然后，你移到主干分支：

```
.it
checkout master
```



现在，你的代码处于你创建分支之前的状态了。

# 删除一个分支

如果你要把你的分支合并到主干分支，从主干（master）分支上发送：

```
.it
branch -d new_feature
```



假如修改已经合并了，它只会删除分支。假如分支没有合并，你会得到一个错误信息。删除一个未合并的分支（通常你不想保留的修改），你需要发送一样的命令附带一个大写D。意思是“强制删除分支，无论如何我不想要它了。”：

```
.it
branch -D new_feature
```



# 回滚到之前的提交状态

在某些时候，你可能想要回到之前的代码版本。首先，你需要找到你想回到哪个版本。要看所有的完成了的提交，发送：

```
.it
log
```



这会输出你的提交的历史记录，像这样：

```
ommit
ca82a6dff817ec66f44342007202690a93763949Author: your_username your_email@domain.comDate:
```

```
ommit
085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7Author: your_username your_email@domain.comDate:
```

```
ommit
a11bef06a3f659402fe7563abf99ad00de2209e6Author: your_username your_email@domain.comDate:
```

如果你想回到“adds my new feature”这个提交，简单地用提交的ID做签出（**checkout**）（我通常只用到ID开头的9个字符）

```
:it
checkout 085bb3bcb
```

你也可以签出到一个新的分支，像这样：

```
:it
checkout -b my_previous_version 085bb3bcb
```

只是别太疯狂了！你的分支越复杂，就越难确定你真正在做什么。

## 推送到远程代码库

在第一次你想推送一个本地代码库到远程代码库时，你需要把它添加到你的项目配置里。像这样做：

```
gin https://your_username@bitbucket.org/your_username/name_of_remote_repository.git
```

注意这里的“**origin**”只是一个习惯。它是你的远程代码库的别名，但是你可以用其他任何你喜欢的词。你甚至可以有多个远程代码库，你只需要给它们起不同的别名。

之后，你想要推送你的本地代码库的主干分支到你的远程代码库：

```
:it
push origin master
```

如果你使用**Bitbucket**，在这时，你会被请求输入你的密码。照做，你的本地代码库会被推送到你的远程代码库上。

## 取得远程代码库的一份本地拷贝

如果你还没有一份远程代码库的本地版本（例如，如果你在另一台机器上开始工作，这 [可能](#) 有用过这个项目），你首先需要拷贝（**clone**）它。去到你的代码库想要拷贝到的文件夹下，并发送：

```
:it
clone https://your_username@bitbucket.org/your_username/name_of_remote_repository.git
```

另一方面，如果你已经在本地的项目上工作了，只是想从远程代码库上取得它最新的版本，移动到项目的根目录下，并发送：

```
it
pull origin master
```

## 别名

Git允许你为你常用的命令创建快捷方式（别名）。例如，如果你不想每次都输入`git commit -m “some comment”`，而是输入`git c “some comment”`，你可以向你的git全局配置里添加一个别名来实现，像这样：

关闭

```
it
config --global alias.c
commit
-m'
```

这是我使用的别名列表：

```
it
config --global alias.c
commit
-m'

it
config --global alias.co
checkout'

it
config --global alias.cob
checkout
-b'

it
config --global alias.br
branch'

it
config --global alias.m
merge'

it
config --global alias.a
add
.'
```

```
it
config --global alias.s
status'

it
config --global alias.dbr
branch
-d'
```

## 进一步

当然，还有比这些更多的Git内容。如果你想要更了解Git，我推荐官方文档和教程，你可以在<http://git-scm.com/documentation>找到。

有任何建议、技巧和问题？在下面留言！

原文链接：[Nico](#) 翻译：[伯乐在线](#) - [cjpan](#)

译文链接：<http://blog.jobbole.com/53573/>

[ 转载必须在正文中标注并保留原文链接、译文链接和译者等信息。]

顶

0

踩

0

上一篇 [你应该从 SVN 迁移到 Git 吗？](#)

下一篇 [Git 分支管理是一门艺术](#)

我的同类文章

好文转载（25）

• [介绍一个成功的 Git 分支模型](#)

2013-12-22 阅读 982

• [你应该从 SVN 迁移到 Git 吗？](#)

2013-09-30 阅读 365

• [我痛恨 Git 的 10 个理由](#)

2013-09-30 阅读 380

• [持续集成反模式](#)

2012-05-19 阅读 2467

• [IncrediBuild 加速原理](#)

2011-11-05 阅读 535

• [Git 分支管理是一门艺术](#)

2013-12-22 阅读 292

• [迁移到 Git 的八个理由](#)

2013-09-30 阅读 350

• [典型分布式计算技术的分析...](#)

2013-06-30 阅读 562

• [用GDB调试程序](#)

2011-11-24 阅读 275

[更多文章](#)

参考知识库

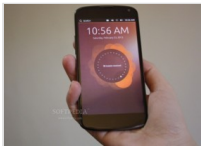


**Git**知识库  
[222](#) 关注 | [298](#) 收录

猜你在找

- [iOS8-Swift开发教程](#)
- [拥抱开源：Github使用指南](#)
- [Windows CE车载应用的实现与相关技术点](#)
- [Linux高薪入门实战精品（上）](#)





ubuntu手机



赢话费扑克



青鸟北大



捕鱼1000炮



云服务器免费



免费云服务器



电脑编程

查看评论

暂无评论

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

\* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题   Hadoop   AWS   移动游戏   Java   Android   iOS   Swift   智能硬件   Docker   OpenStack  
VPN   Spark   ERP   IE10   Eclipse   CRM   JavaScript   数据库   Ubuntu   NFC   WAP   jQuery  
BI   HTML5   Spring   Apache   .NET   API   HTML   SDK   IIS   Fedora   XML   LBS   Unity  
Splashtop   UML   components   Windows Mobile   Rails   QEMU   KDE   Cassandra   CloudStack  
FTC   coremail   OPhone   CouchBase   云计算   iOS6   Rackspace   Web App   SpringSide   Maemo  
Compuware   大数据   aptech   Perl   Tornado   Ruby   Hibernate   ThinkPHP   HBase   Pure   Solr  
Angular   Cloud Foundry   Redis   Scala   Django   Bootstrap

