**Git开发代码管理方案**

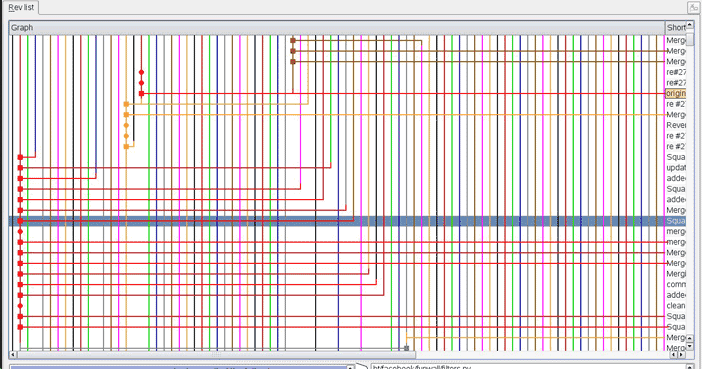
# 1、Git介绍

Git是**分布式**版本控制系统，与SVN类似的集中化版本控制系统相比，集中化版本控制系统虽然能够令多个团队成员一起协作开发，但有时如果中央服务器宕机的话，谁也无法在宕机期间提交更新和协同开发。甚至有时，中央服务器磁盘故障，恰巧又没有做备份或备份没及时，那就可能有丢失数据的风险。

但Git是分布式的版本控制系统，客户端不只是提取最新版本的快照，而且将整个代码仓库镜像复制下来。如果任何协同工作用的服务器发生故障了，也可 以用任何一个代码仓库来恢复。而且在协作服务器宕机期间，你也可以提交代码到本地仓库，当协作服务器正常工作后，你再将本地仓库同步到远程仓库。

Git众多优点中最显著的一点，就是版本的分支（branch）和合并（merge）十分方便。有些传统的版本管理软件，分支操作实际上会生成一份现有代码的物理拷贝，而Git只生成一个指向当前版本（又称”快照”）的指针，因此非常快捷易用。

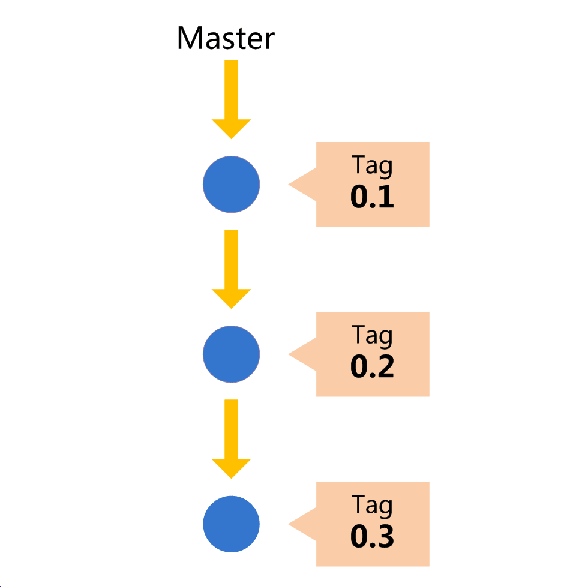
但是，太方便了也会产生副作用。如果你不加注意，很可能会留下一个枝节蔓生、四处开放的版本库，到处都是分支，完全看不出主干发展的脉络。如下图：



# 2、Git分支管理策略

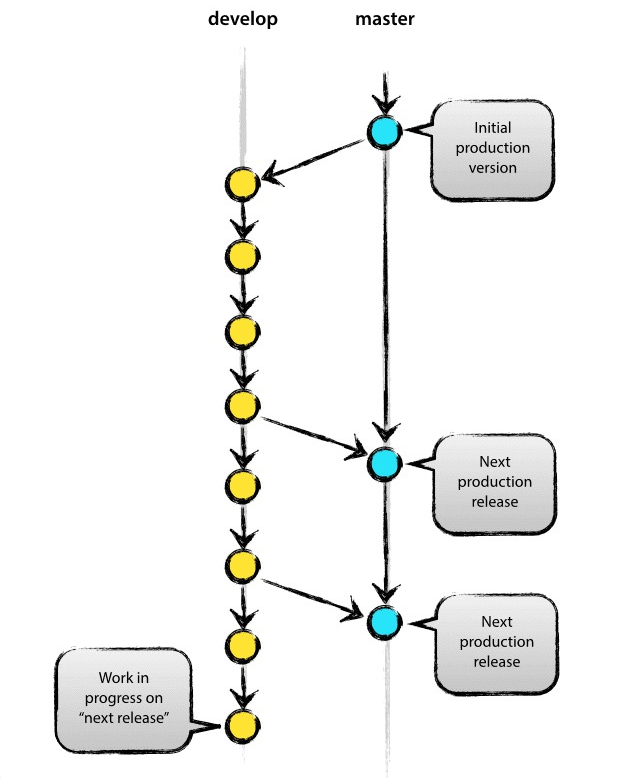
## 2.1 主分支

主分支（master）是git仓库建立起来的默认分支。代码库应该有一个、且仅有一个主分支。所有提供给用户使用的正式版本，都在这个主分支上发布。



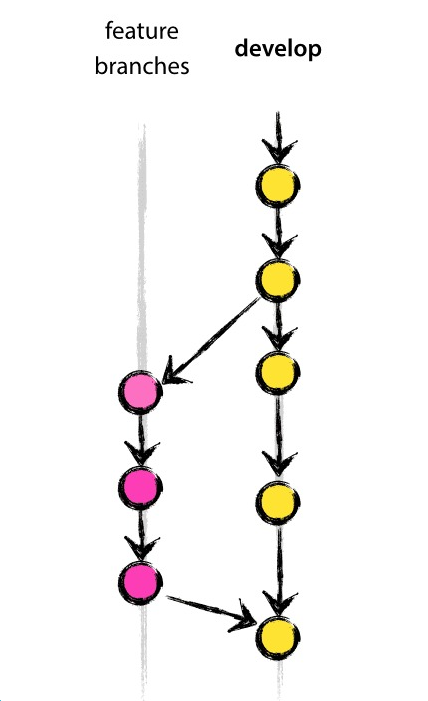
## 2.2 开发分支

日常的开发应该都在开发（Develop）分支上进行，需要从master分支上拉去建立，当develop分支上的代码达到了一个稳定状态，并且准备发布时，所有的代码变更都应该合并到master分支，然后打上发布版本号的tag。



## 2.3 特性分支

特性分支用于开发某个**新的特性或者功能**。特性分支的命名一般可以命名为：**feature-xxx**。开发人员在准备开发某个新功能时候，需要从develop分支上拉取建立特性分支。最终这些特性分支会被合并到develop分支(目的是在新版本中添加新的功能)或者被丢弃(它只是一个令人失望的试验)。特性分支只存在开发者本地版本库，不在远程版本库。



## 2.4 发布分支

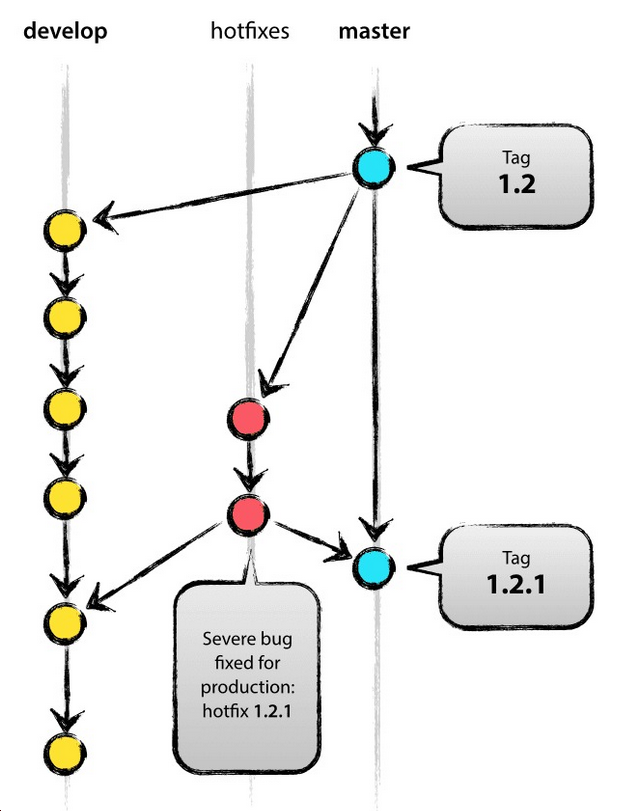
发布分支（release）用于支持一个新版本的发布。当develop分支功能开发达到要对外发布新版本的时候，建议建立发布分支，进行最终的测试，允许在最后时刻进行一些小修小改。甚至允许进行一些小bug的修改，为新版本的发布准要一些元数据(版本号，构建时间等)。通过在release分支完成这些工作，develop分支将会合并这些特性以备下一个大版本的发布，另外也会合并到master版本上，对外进行发布。其命名为：release-xxx。

## 2.5 内部转测分支

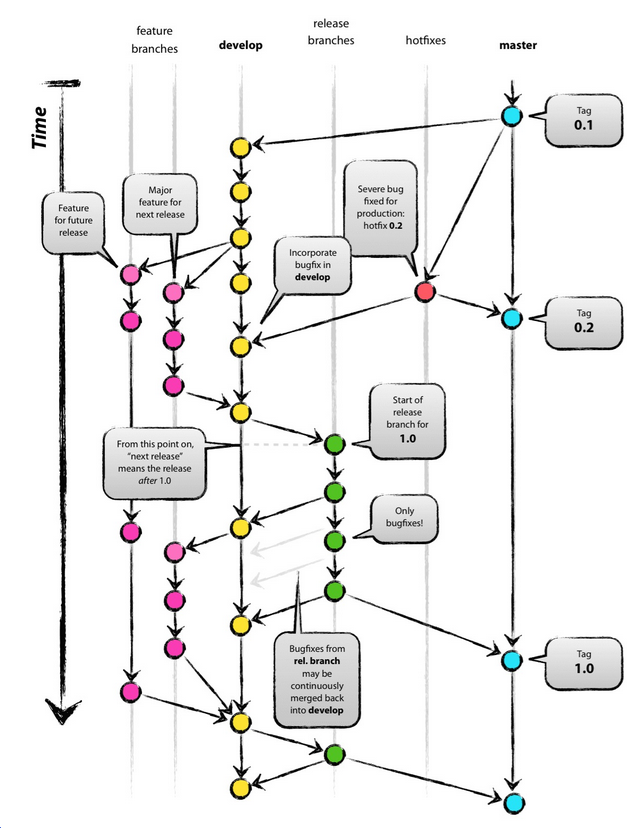
内部转测分支（inner-release）用于对开发完成的功能进行发布转测，其命名为：inner-release-xxx。从develop上拉去建立，发布工质量部进行测试，对测试的bug在该分支上进行修复，最终合并到develop版本上，该内部转测分支也就可以删除掉了。

## 2.6 紧急修复分支

紧急修复分支（hotfix）是基于生产环境的产品的一个bug急需解决而必须创建的。当客户现场环境发现产品存在bug需要紧急修复的时候需要建立hotfix分支,即从master分支上该版本对应的tag上进行拉去建立，bug修改完毕后，必须合并回develop分支和master分支。其命名为：hotfix-xxx。



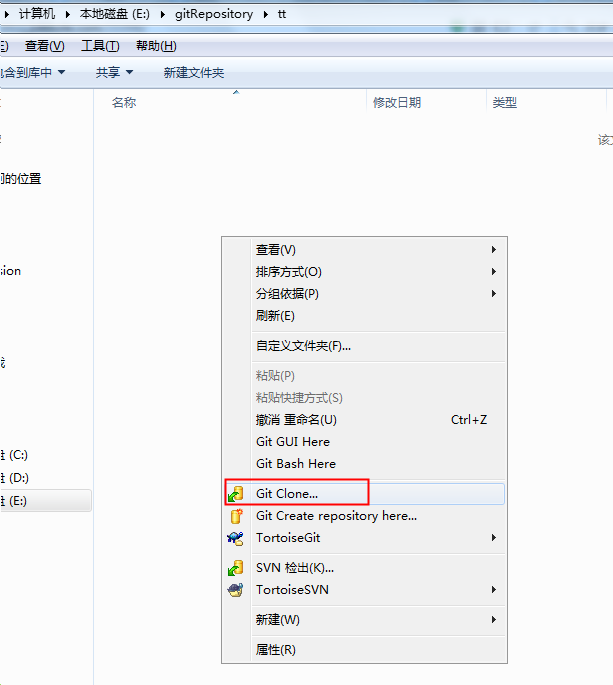
综上所述：最终我们分支管理如下图所示：

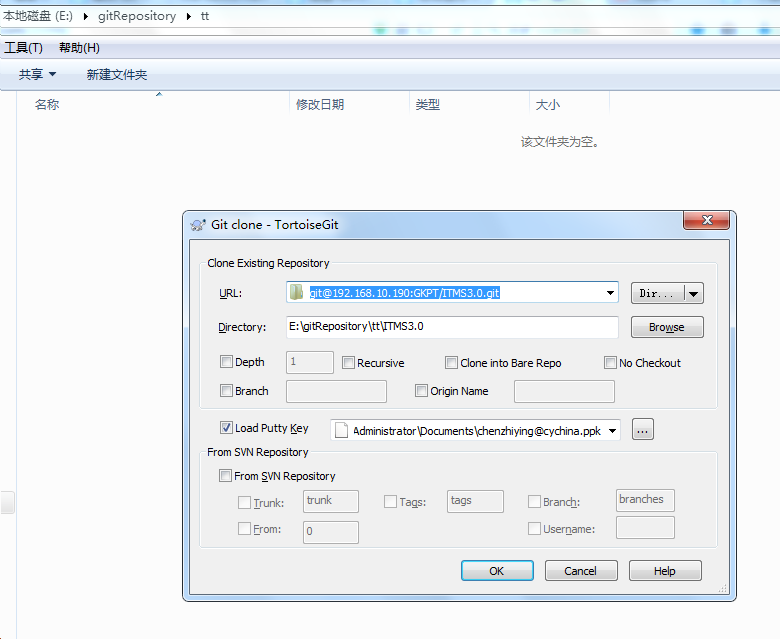


# 3 具体操作

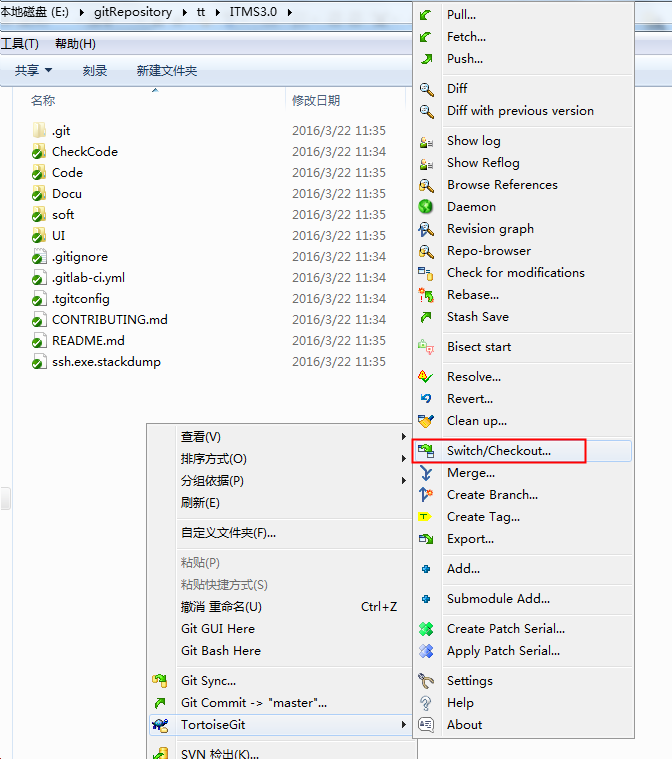
开发人员日常工作基本步骤：

## 1）clone远程服务器代码

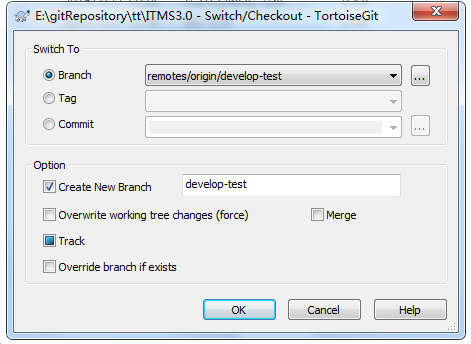
命令方式：**$ git clone** git@192.168.10.190:GKPT/ITMS3.0.git 



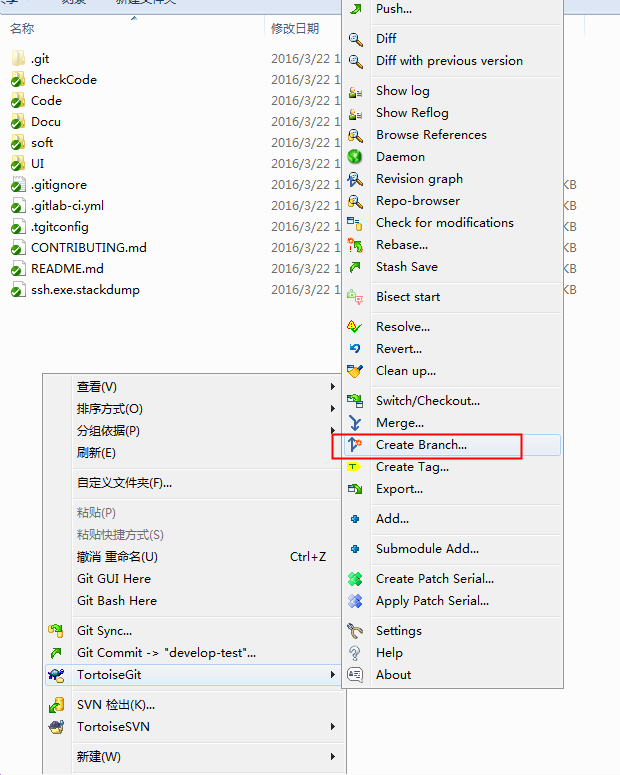
## 2）切换到develop分支

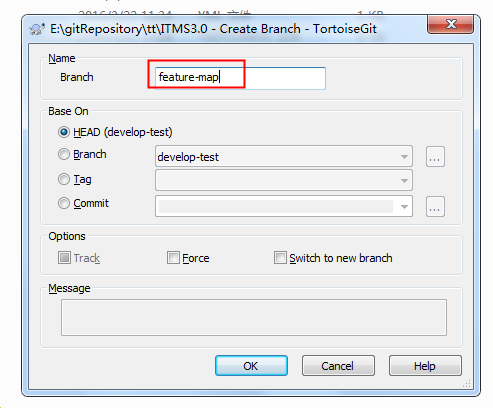


选择develop分支

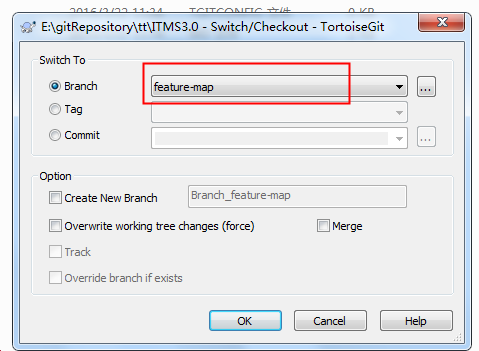


## 3）创建自己的开发分支

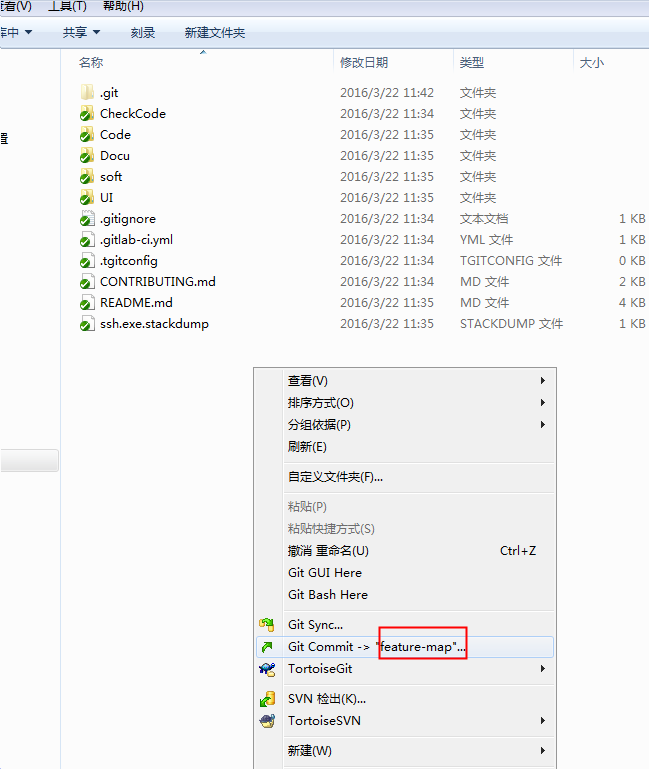




## 4）切换回刚刚步骤3创建的开发特性分支



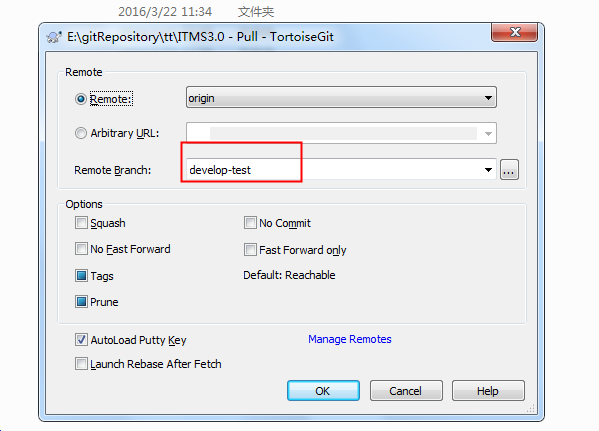
这样我们就处于了刚刚新建的开发特性分支下了，现在可以在本地进行你的代码开发工作，你可以随时进行commit。（这里的commit只是提交到你本地开发分支库，不会影响到远程服务器上的库）等该功能开发完成后，你就可以push到develop分支上。



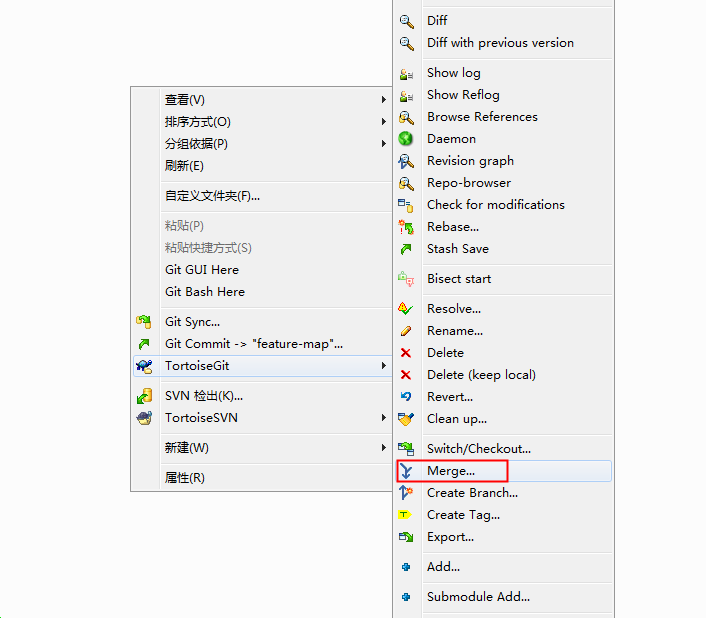
## 5）Merge develop分支上代码

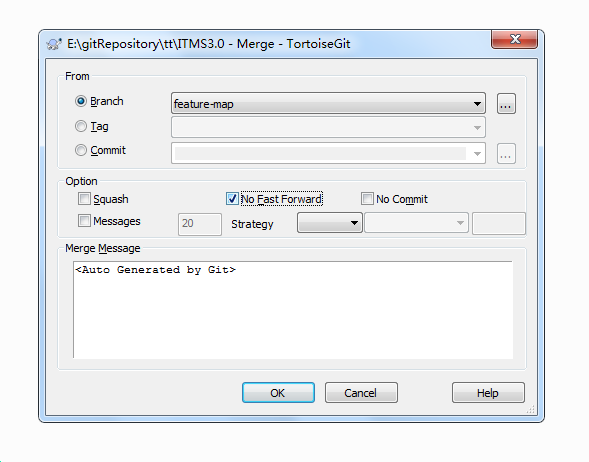
在merge到develop分支前，可能你本地版本与服务器上的版本不一致（也就是不是最新的那个版本了），需要先pull代码到你本地（先要切换到develop分支上，进行pull）。然后执行merge操作。

先切换到develop分支上，然后pull最新代码



接下来merge代码





## 6）push到develop分支上

