

SVTI 持续集成环境 用户指南

目录

前言.....	1
1. 简介	1
1.1 持续集成是什么	1
1.2 工作流程	1
2. 使用方法	2
2.1 准备工作	2
2.2 提交代码	3
2.2.1 命令行界面提交代码	4
2.2.2 TortoiseGit 界面提交代码.....	5
2.3 结果通知	9
2.3.1 Web 界面	9
2.3.2 邮件通知	9
3. 常见问题与错误	10
3.1 有人正在上传	10
3.2 需要更新代码	11
3.3 难以消除的 CA 告警	11
3.4 上传分支错误	12
3.5 请运行 Git_AutoCA.exe	12
3.6 服务器当前分支错误	13
3.7 Web 页面出现错误调试信息	13
3.8 邮件内容有误或乱码	13

前言

为了能在团队搭建一个和公司相似的开发环境，更好的管理代码，保证主线代码永远是 OK 的。于是耗时大半个月，写了一些脚本和 C 程序，终于实现了与公司基本类似的持续集成开发环境，并不断地改进和添加一些新特性。欢迎大家反馈各种 BUG（最好把屏幕输出的调试信息截图保存），或者想添加一些实用小功能，可以向我提意见或者发到我的邮箱。

1. 简介

1.1 持续集成是什么

关于持续集成，给我的感觉就是开发人员要经常性地把代码合入主线，每次上传代码都会触发构建，然后服务器对代码进行自动化测试，测试通过并审核之后代码才能合入主线。

持续集成的好处很多文献资料都有说明。持续集成的关键是完全的自动化，读取源代码、编译、连接、测试，整个创建过程都应该自动完成。

1.2 工作流程

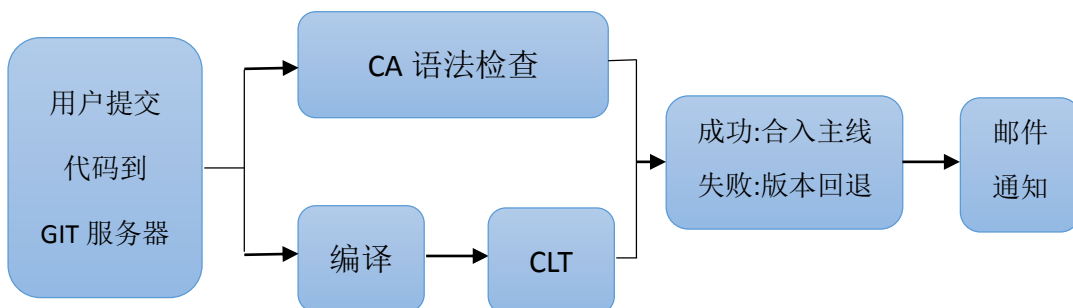


图 1.1 SVTI 持续集成环境工作流程

2. 使用方法

由于持续集成的环境主要是在服务器搭建，所以客户端的使用时比较简单的。
主要都是些 Git 的基本操作。

2.1 准备工作

(1) 安装 Git

(2) 公钥认证

需要认证才有权限下载及合入代码，目前公钥主要是人工管理。

a. 右击桌面空白处打开 Git Bash，输入 ssh-keygen，然后一直按 Enter 键。

```
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Administrator/.ssh/id_rsa):
Created directory '/c/Users/Administrator/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Administrator/.ssh/id_rsa.
Your public key has been saved in /c/Users/Administrator/.ssh/id_rsa.pub.
The key fingerprint is:
34:94:41:4b:98:c5:bc:b8:c7:b3:f2:75:5e:14:b7:8d Administrator@WIN-J5796U1CMV0
The key's randomart image is:
+--[ RSA 2048 ]-----+
|      B*o              |
|    oo+.               |
|   .+.                |
|  ....+o              |
| oS      E..          |
| . + .               |
|  . o . . .          |
| . . o . .           |
|  o . .               |
|-----+-----|
```

图 2.1 ssh-keygen 命令结果

b. 把文件 C:\用户\“你的用户名”\.ssh\id_rsa.pub 复制出来，重命名为自己的名字（如 liangjinchao.pub）然后发送并同时通知管理员。

(3) 设置邮箱

继续 Git Bash 里面输入以下命令(邮箱名和姓名写自己的)

```
git config --global user.email liangjinchao@svti.com
```

```
git config --global user.name liangjinchao
```

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

(4) 下载代码

如果以上步骤成功，那么就可以 Git clone 代码了：打开 git bash，切换到加密盘目录，然后输入命令：

```
git clone ssh://git@SVTI/SVTI/Code
```

或者（如果装了 Tortoise Git）：

在工作目录下右击 Git Clone，地址栏写上面的 ssh 地址(ssh://....)，然后确定。

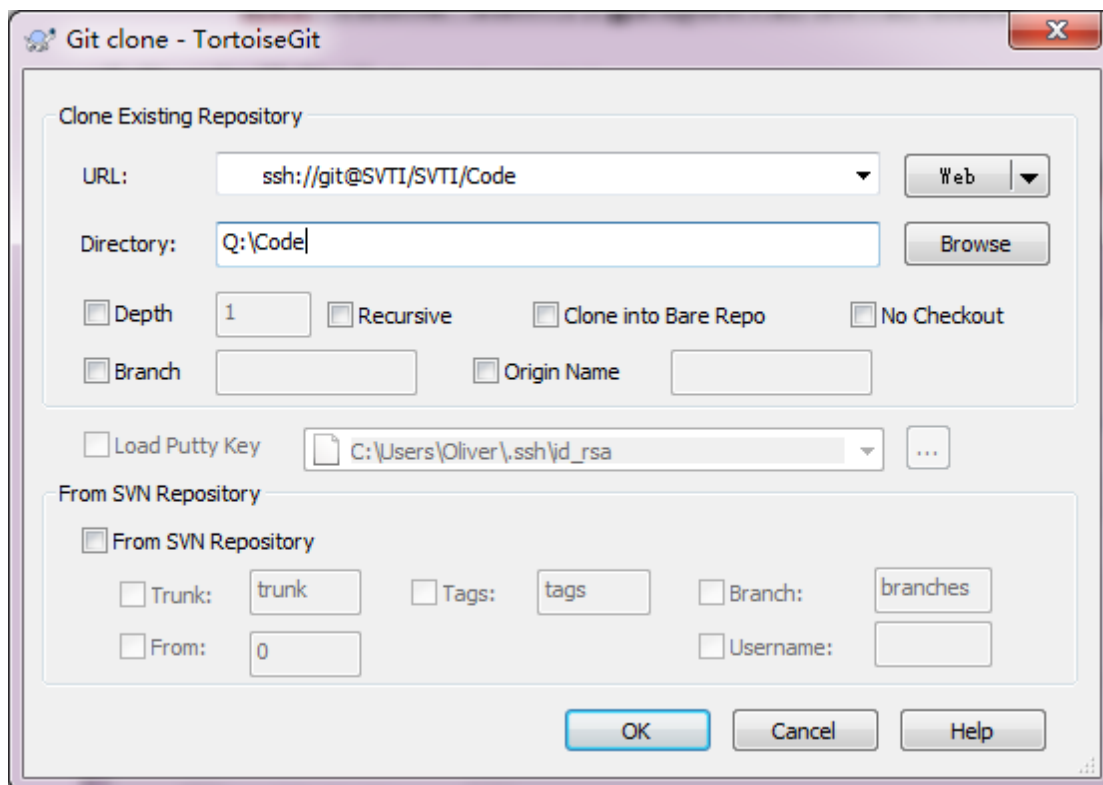


图 2.2 TortoiseGit 下载代码界面

如果下载成功，恭喜你，公钥认证成功。如果失败，则检查前面几步是不是没做好。

2.2 提交代码

提交代码的步骤和平常 Git 的操作一样，先 commit（提交到本地）然后更新代码最后 push（上传到服务器）。**唯一不同的是 push 的分支不是 master，而是远程分支 work。**当自动化测试通过后 work 分支会自动合入 master 分支。

2.2.1 命令行界面提交代码

命令行界面提交代码的相关命令如下：

- (1) commit 到本地：

```
git commit --all
```

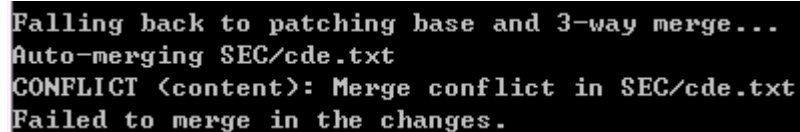
然后弹出来 vi 编辑器中，在文件开头填入提交信息，在将需要 commit 文件前面的#号去掉保存（ZZ）退出。

- (2) fetch+rebase 更新代码：

```
git fetch origin
```

```
git rebase origin
```

rebase 的过程中可能出现冲突导致失败。如果没有冲突则进行下一步。



```
Falling back to patching base and 3-way merge...
Auto-merging SEC/cde.txt
CONFLICT (content): Merge conflict in SEC/cde.txt
Failed to merge in the changes.
```

图 2.3 rebase 出现冲突的例子

这时打开产生冲突的文件，找到出现“<<<<<<<”、“=====”、“>>>>>>>”的地方，就是冲突产生的地方。修改这些地方的代码，其中“<<<<<<<”与“=====”之间的部分是服务器的主线代码，“=====”与“>>>>>>>”之间的部分是你提交的代码，两者产生冲突需要修改，且修改完不能含有原来的“<<<<<<<”、“=====”、“>>>>>>>”，然后再次 rebase：

```
git add -all
```

```
git rebase --continue
```

如果修改冲突时间过长，中途可能有人又合入了新的代码，则需再次进行步骤（2）。

- (3) push 代码到服务器：

```
git push origin master:work
```

上述都是合代码最基本的 Git 命令操作。git 命令还有很多高级的功能和用法，可以通过—help 参数来查看。

2.2.2 TortoiseGit 界面提交代码

如果实在是不习惯用命令行的方式来提交代码，可以使用图形化界面的 TortoiseGit，足以满足大多数基本操作。当然啦命令行的功能永远是最完整的。

使用 Tortoise Git 则提交代码步骤要经过三步：

- Commit:提交代码到本地，并写
- Fetch+Rebase:更新代码并与主线代码结合
- Push:上传代码到服务器

第一次使用 TortoiseGit 提交或下载代码前先检查下以下设置是否正确：右击桌面空白处 TortoiseGit----Settings，按图 2.4 所示设置。

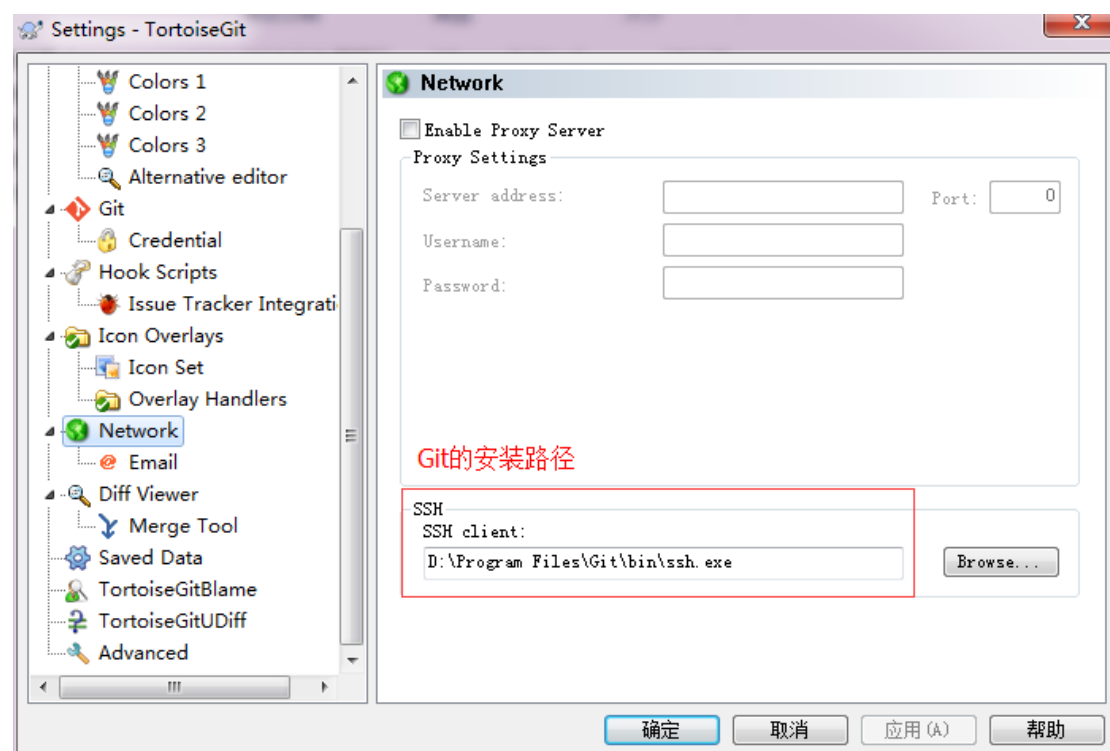


图 2.4 设置 SSH 客户端

TortoiseGit 提交代码具体操作如下：

(1) Commit 到本地：

右击仓库目录，选“Git Commit -> master”，然后输入提交信息，OK。

(2) Fetch+Rebase 更新代码：

右击仓库目录，选 Tortoise Git ---- Fetch 弹出窗口如图 2.5：

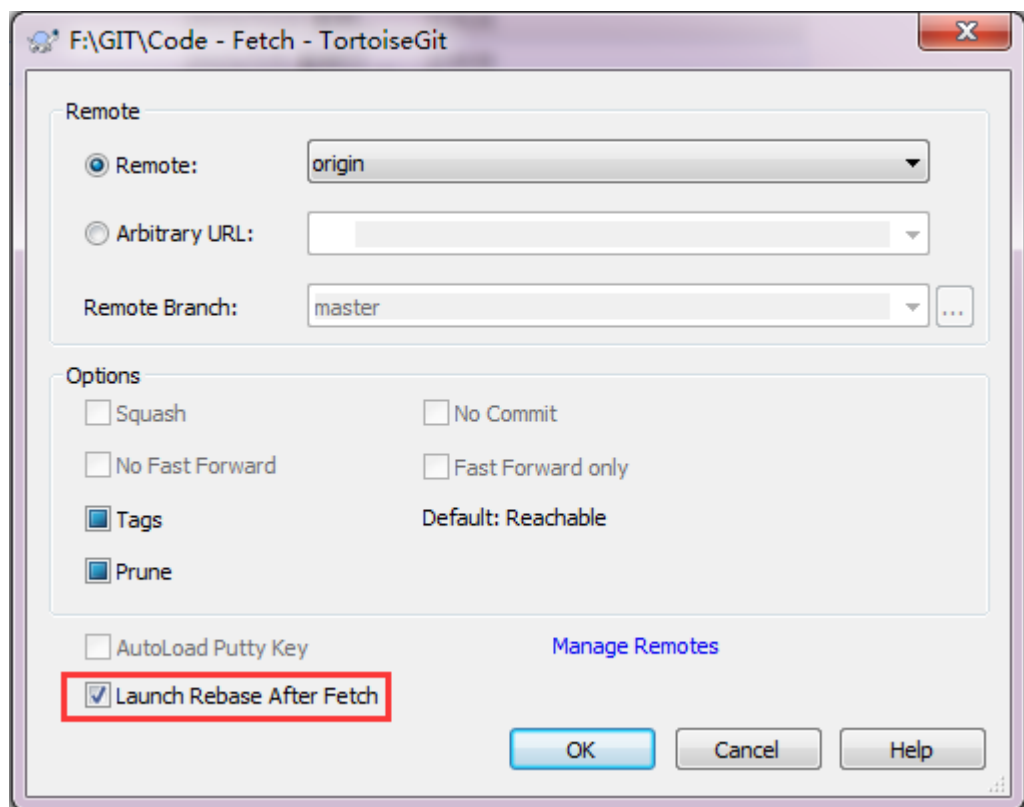


图 2.5 TortoiseGit 的 Fetch 更新代码界面

勾选“Launch Rebase After Fetch”，然后点击 OK。之后弹出的界面点 Rebase。Rebase 的过程可能产生冲突，比如你的代码和别人的改了同一个函数的同一行：

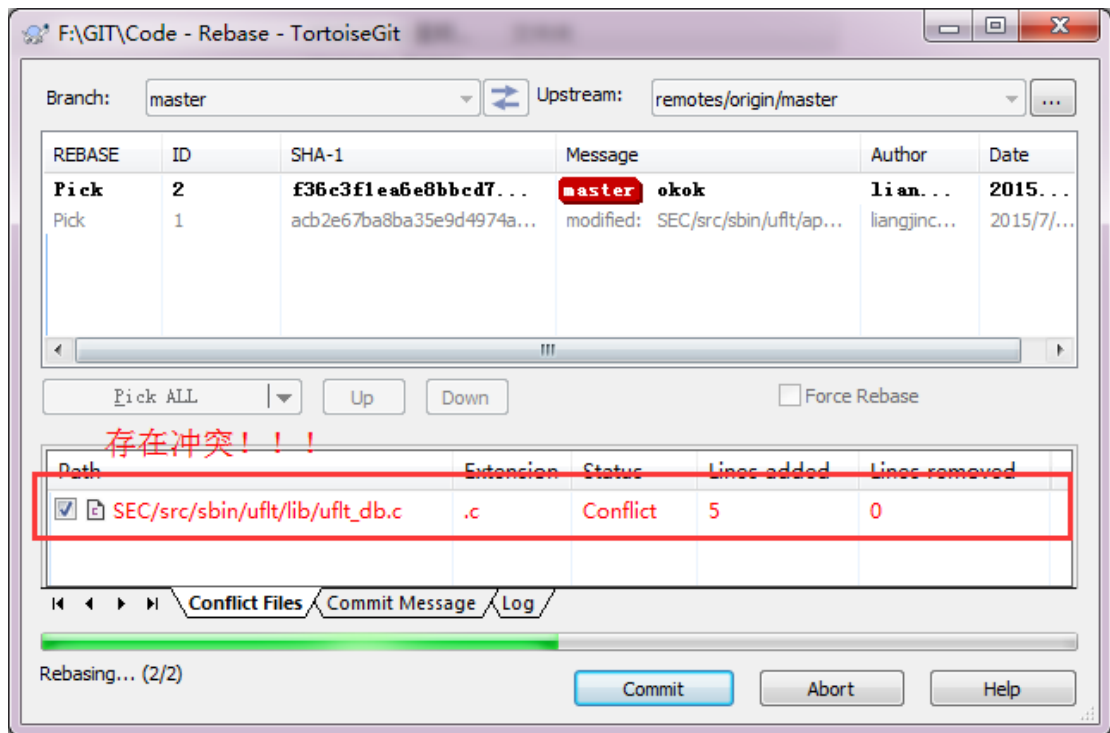


图 2.6 Rebase 时产生冲突

如果存在冲突则双击图 2.6 中红色框起来的部分编辑冲突（界面如图 2.7，注意不要有多余的换行符）。冲突修改期间如果有人又合入了新代码，则修改完冲突之后需要再次进行步骤（2）。如果不存在冲突则进行下一步。

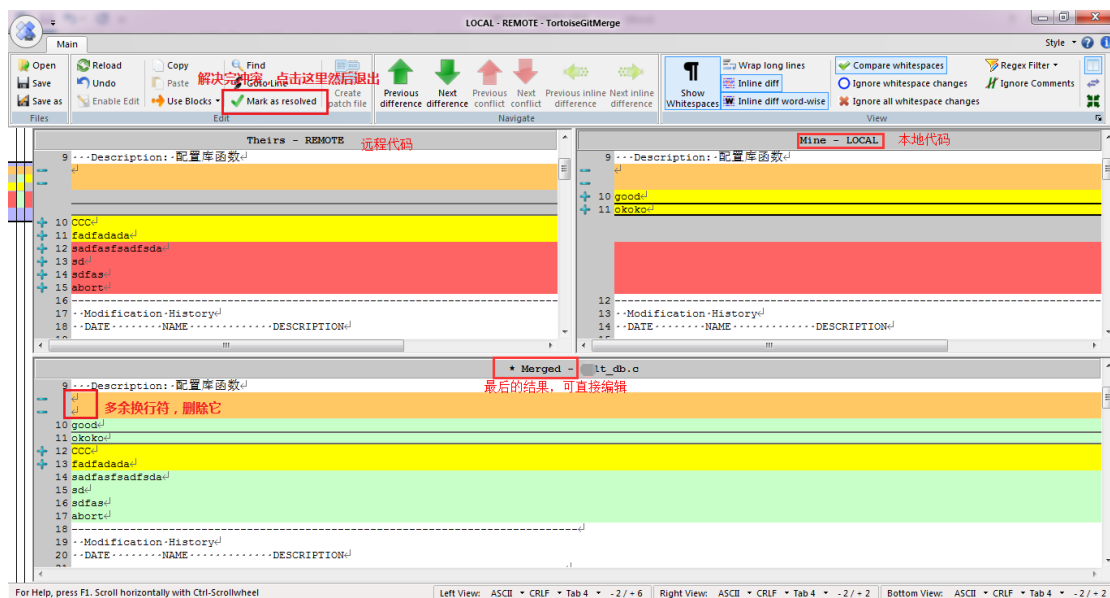


图 2.7 冲突编辑界面

（3）Push 代码到服务器：

右击仓库目录，选 Tortoise Git ---- Push 弹出窗口如图 2.8:

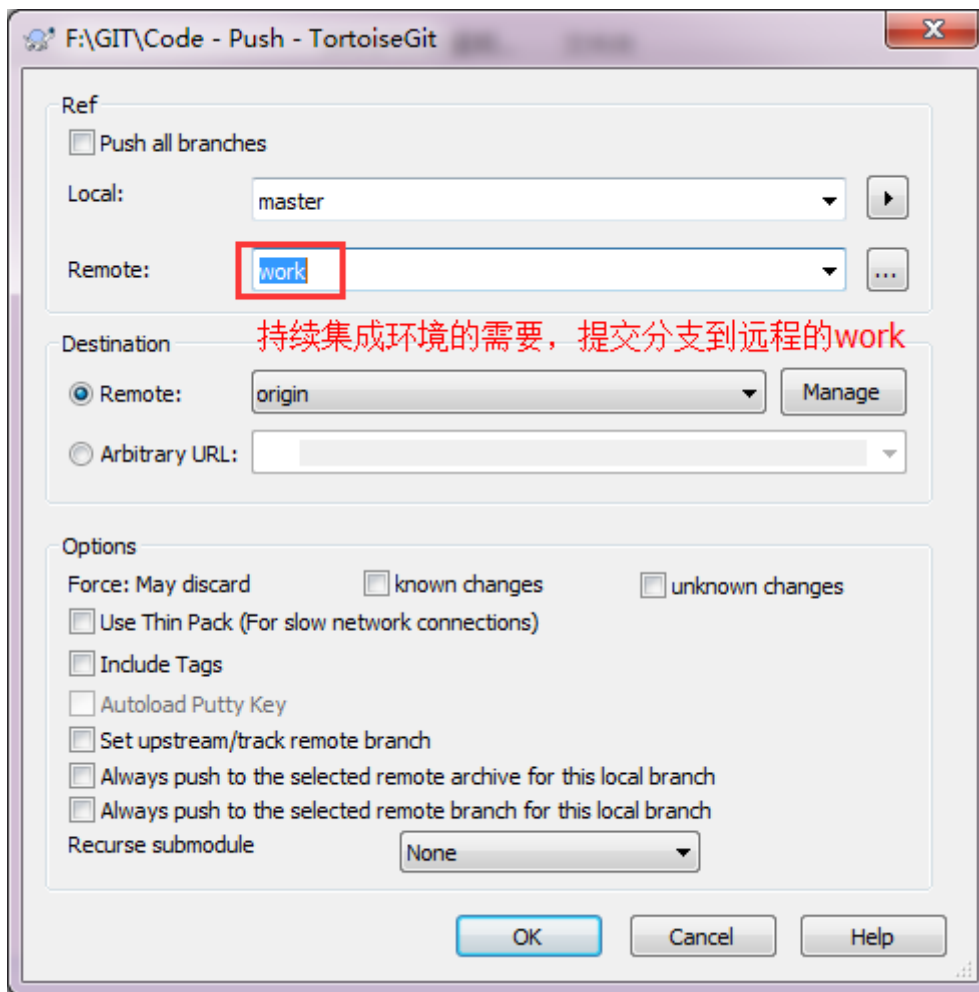


图 2.8 TortoiseGit 上传代码到服务器

如果弹出图 2.9 或图 2.10 的界面则表明代码上传成功啦！当然还需要通过自动化测试才能合入主线（除非出现图 2.10 的情况）。否则需要修改代码重新提交。

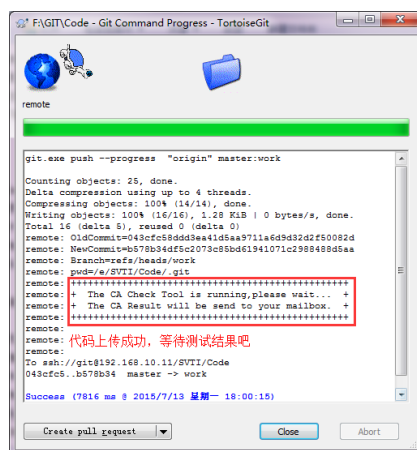


图 2.9 上传成功，需要等待测试结果

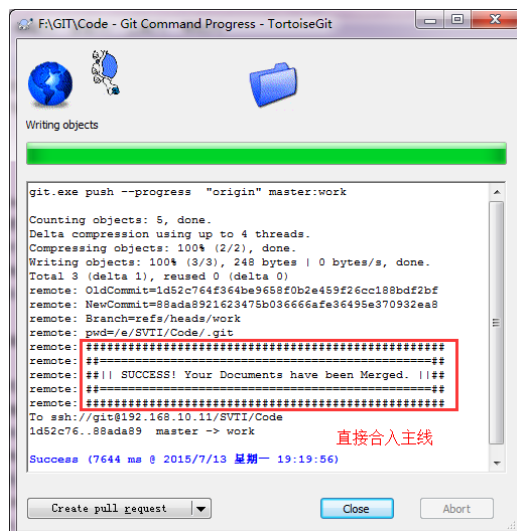


图 2.10 上传成功且直接合入主线

2.3 结果通知

2.3.1 Web 界面

打开以下链接实时地观察编译输出结果（有的任务可能还没开始）。

<http://svti:8080/>

2.3.2 邮件通知

当自动化测试结束后，会收到邮件通知。如果收不到邮件，请检查 2.1 节中的步骤(3)进行操作。

3. 常见问题与错误

当正常操作失败或屏幕打印奇怪的信息的时候，应及时截图保存错误信息，以备环境开发人员进行维护。下面列举了几种常见的问题。

3.1 有人正在上传

```
$ git push origin master:work
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 252 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: OldCommit=88ada8921623475b036666afe36495e370932ea8
remote: NewCommit=9cdf8481f98fa7c409dcb656c6a03fd2d6348bad
remote: Branch=refs/heads/work
remote: pwd=/e/SUTI/Code/.git
remote: =====
remote: !! Somebody is pushing, please try again later. !!
remote: =====
remote: Somebody = liangjinchao 正在构建的用户
remote: StartTime = Mon Jul 13 18:33:52 2015 开始构建的时间
remote: status = CAing 当前状态
To ssh://git@192.168.10.11/SUTI/Code
! [remote rejected] master -> work (pre-receive hook declined)
error: failed to push some refs to 'ssh://git@192.168.10.11/SUTI/Code'
```

图 3.1 有人正在合入或构建，请等待构建结束

这是最常见的问题。当两个用户同时上传，或者一个用户上传的代码仍在进行测试时，又有新的用户要上传代码时，就会出现这个问题。等待一段时间等持续集成测试完成然后再次上传即可。

遇到这个问题的时候要注意一下上传者和上传时间是否合理，如图 3.1 所示。如果上传者的上传时间（即图中的开始构建的时间）与现在时间相差超过 15 分钟（持续集成自动化测试的最长时间，具体大小跟编译时间、服务器性能等参数有关），说明出现了系统错误，截图并及时反馈问题给管理员。

3.2 需要更新代码

```
$ git push origin
To ssh://git@192.168.56.88/INF
 ! [rejected]      master -> master (fetch first)
error: failed to push some refs to 'ssh://git@192.168.56.88/INF'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

图 3.2 未更新主线代码

上传代码前需要更新代码（**fetch**）到本地，与本地的修改合并（**rebase**）才能 **push** 到服务器。具体参照 2.2.1 中的步骤（2）或 2.2.2 中的步骤（2）。

3.3 难以消除的 CA 告警

有时候可能有特殊原因需要及时把代码合入主线，但有的 CA 告警实在难以消除，可以向管理员申请增加 CA 告警的数量的限制。但前提是代码要能编译通过而且功能正常。

```
Q:\Code\SEC\src\kernel\uflt\uflt_krule_config.c L1148 Warning 429 [UFLT_KRULE_SetAction]
    Custodial pointer 'pstTmp' (line 1109) has not been freed or returned

*** Q:\Code\SEC\src\kernel\uflt\uflt_krule_config.c L825 Warning 429 UFLT_KRULE_SetAction(?)
    Q:\Code\SEC\src\kernel\uflt\uflt_krule_config.c L1174 Warning 429 [UFLT_KRULE_SetAction]
        Custodial pointer 'pstTmp' (line 1109) has not been freed or returned

Note 900
    Total 8 messages ( 1 Files), 2 Error, 5 Warning, 1 Info, 0 Note. [0/56]
```

Count	Num	Type
2	18	Error
3	429	Warning
2	515	Warning
1	774	Info

图 3.3 难以消除的 CA 告警

3.4 上传分支错误

```
$ git push origin
Counting objects: 15, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (7/7), done.
Writing objects: 100% (8/8), 651 bytes | 0 bytes/s, done.
Total 8 (delta 3), reused 0 (delta 0)
remote: OldCommit=75fcfe56294753f11e6037b6b92aab08a1a2eb6f
remote: NewCommit=acb2e67ba8ba35e9d4974a76826e9bc45fbdd455
remote: Branch=refs/heads/master
remote: pwd=/e/SVTI/Code/.git
remote:
remote: =====
remote: || You can not push to the branch ||
remote: || Please push to branch work    ||
remote: =====
To ssh://git@192.168.10.11/SVTI/Code
! [remote rejected] master -> master (pre-receive hook declined)
error: failed to push some refs to 'ssh://git@192.168.10.11/SVTI/Code'
```

图 3.4 无法上传到该分支

由于持续集成开发环境的需要，用户不能直接上传到主线分支，而应该上传到 work 分支。因此应使用命令 `git push origin master:work` 进行上传，而不是 `git push origin`。如果是 TortoiseGit 则参照图 2.8 的配置上传。当用户通过自动化测试之后，work 分支的内容会自动合入到主线分支。

3.5 请运行 Git_AutoCA.exe

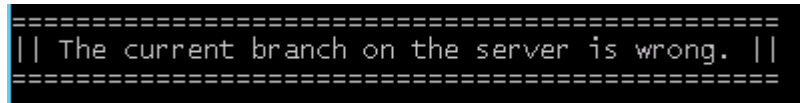
```
$ git push origin master:work
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 252 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: OldCommit=88ada8921623475b036666afe36495e370932ea8
remote: NewCommit=9cdf8481f98fa7c409dcb656c6a03fd2d6348bad
remote: Branch=refs/heads/work
remote: pwd=/e/SVTI/Code/.git
remote:
remote: =====
remote: !! Please run Git_AutoCA.exe on the git server. !!
remote: =====
To ssh://git@192.168.10.11/SVTI/Code
! [remote rejected] master -> work (pre-receive hook declined)
error: failed to push some refs to 'ssh://git@192.168.10.11/SVTI/Code'
```

图 3.5 请运行 Git_AutoCA.exe

自动 CA 程序未启动或遇到致命错误退出。出现这个问题，请及时将情况汇报给

管理员，以便及时修复 BUG。

3.6 服务器当前分支错误



```
=====
|| The current branch on the server is wrong. ||
=====
```

图 3.6 服务器当前分支错误

出现这个问题，表明服务器当前处在一个不合理的分支，将导致无法再 work 分支上传。出现这个问题，说明出现了系统错误，截图并及时反馈问题给管理员。

3.7 Web 页面出现错误调试信息

Jenkins 控制台的输出会打印很多程序的调试信息，反应了整个系统的工作状态。

当出现一些类似“Error”、“Failed”、“失败”字样的调试信息时，说明系统出现了 BUG。将问题及时汇报给管理员以维护持续集成环境。

3.8 邮件内容有误或乱码

系统错误，及时反馈问题给管理员，并提供当前的构建 ID 等信息。