

目 录

基础篇	2
1 CVSNT 安装配置	2
1.1 CVSNT 安装:	2
1.2. 建立仓库	2
1.3. 建立用户	3
2 Wincvs 安装配置	4
2.1 安装	4
2.2 用户登陆	4
2.3 管理 CVS 用户	5
2.4 导入模块	7
2.5 导出模块	10
进阶篇	14
3 使用脚本语言	14
3.1 安装脚本	14
3.2 指定语言库路径	14
3.3 使用脚本功能	15
4 用户权限控制	17
4.1 设定管理员	17
4.2 建立用户组	17
4.3 赋予权限	17
4.4 权限规则	20
结束语	20

CVSNT+Wincvs 安装配置

作者: oyoung 完成日期: 2006-8-16 邮件: oyoung6@hotmail.com

出处: <http://blog.csdn.net/oyoung>

版权声明: 本文档可任意转载, 但请勿修改内容, 转载时请务必标明作者和出处信息。

用 CVS 做版本控制已经有很长一段时间, 有一些经验教训, 很早之前就想下来, 不过到现在才有时间把这些东西整理出来和大家一起分享, 希望对大家有所帮助。

我们在网上看到很多 CVS 的文章都是基于用 Linux 做 CVS 服务器的, 介绍 windows 的很少, 不过用 windows 做服务器还是有的。下面我就介绍下 Windows XP 下的 CVS 服务和客户端的安装配置。

基础篇

1 CVSNT 安装配置

1.1 CVSNT 安装:

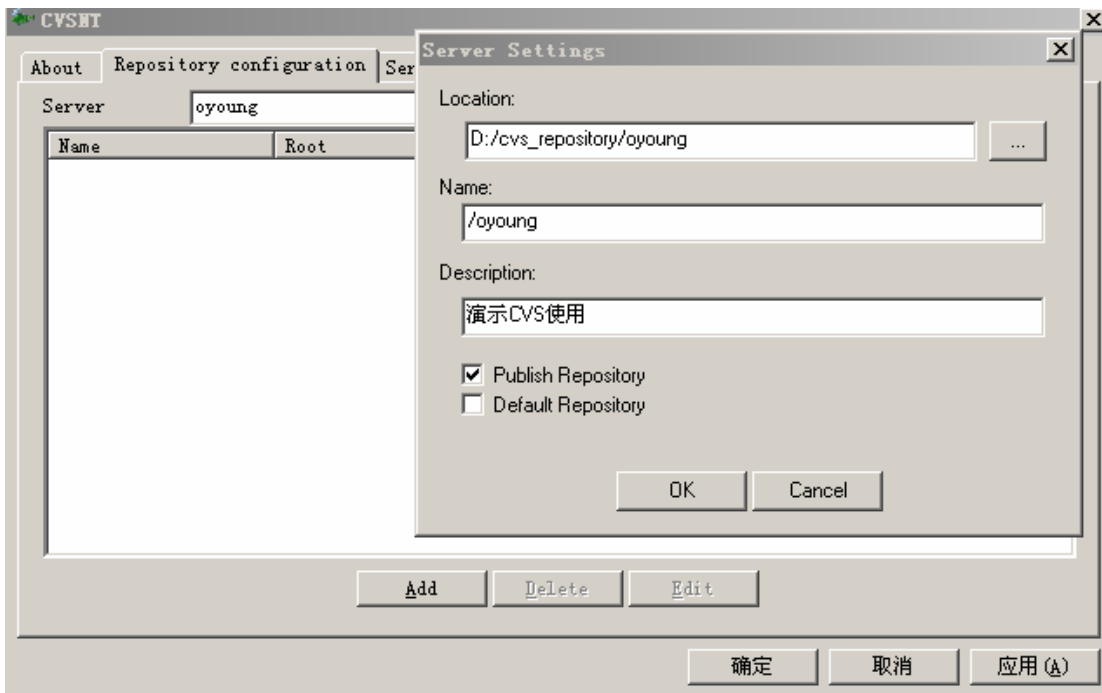
版本: cvsnt-2.5.02.2115

下载: <http://www.march-hare.com/cvsnt/features/cvsnt/>

CVSNT 是 windows 下的 CVS 服务器, 安装在服务器端。我们只需默认安装即可, 然后重启。值得一提的是要注意版本, 开源项目有些版本就是不稳定。我曾经试过安装过一个版本(具体不记得了, 也是 2.5 的), 用到 rtag 命令打标签, 和用 python 产生 baseline report 时就报错, 后来换成了现在这个版本就没问题了。这个版本用到现在还没发现有什么问题, 个人认为应该还是比较稳定的。

1.2. 建立仓库

先建立需要放仓库的目录, 比如 D:\cvs_repository\oyoung。然后程序菜单里打开 CVSNT->CVS NT Control Panel, 在 Repository Configuration 点 add, 添加你的仓库路径和名字, 如下图:

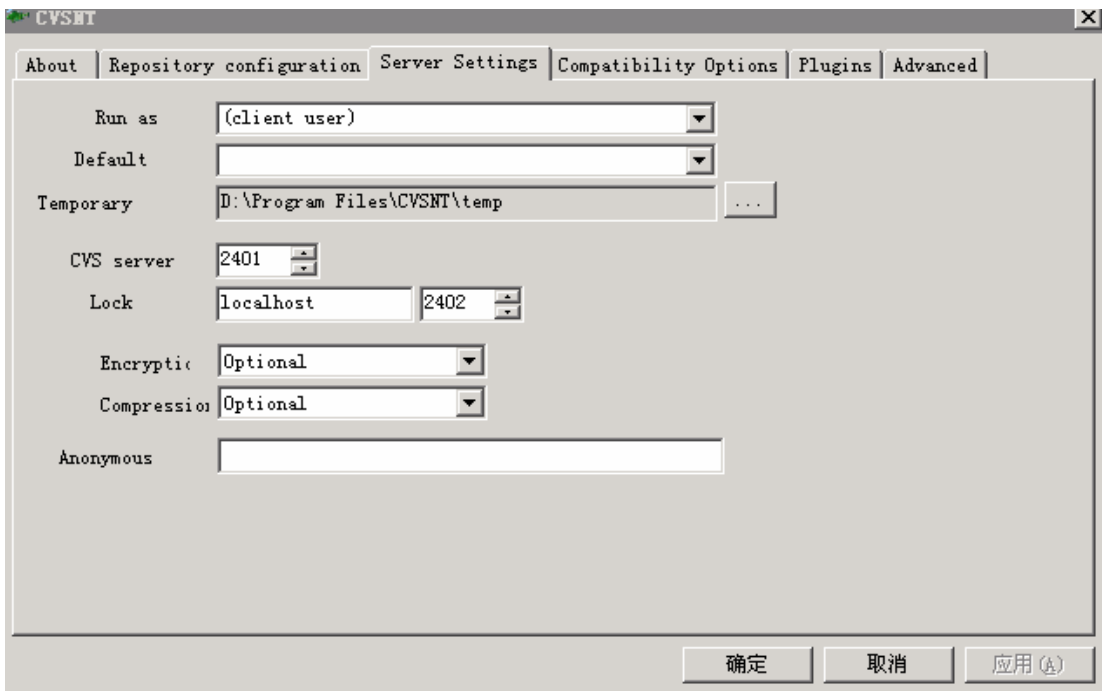


确认后，会提示要初始化仓库目录，确认即可。现在看下你的仓库目录 `D:\cvs_repository\oyoung`，是不是多了个文件夹 `CVSROOT` 呢？如果有的话，就恭喜你了，这个 `CVSROOT` 就是这个仓库的配置管理文件库。

1.3. 建立用户

在你的 windows 建立 administrators 角色的用户 `cvsadmin` 和 `cvsuser`,用于映射后面的 cvs 管理员和普通用户。

1.4. 确保上面建立的用户 `cvsadmin` 和 `cvsuser` 有权限访问你的 CVS Temporary 目录，如下图：



2 Wincvs 安装配置

Wincvs 是 cvs 的一个客户端图形工具，不管服务器是 windows 还是 Linux,都可使用。Wincvs 提供了大部分的图形操作界面，避免了直接用 cvs 命令来控制。如果你要用到高级的功能，也可以在它的命令窗口里输入执行。下面我们来介绍下 wincvs 的安装和配置。

2.1 安装

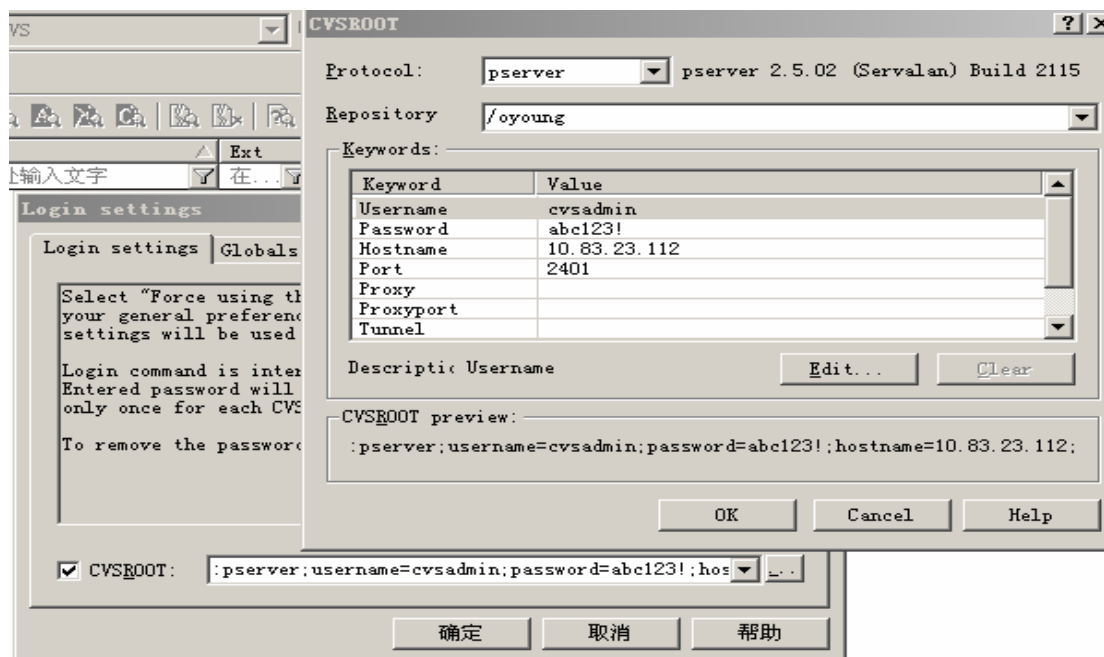
版本: wincvs 2.0.2.4 (build 4)

下载: <http://www.wincvs.org/download.html>

直接点下一步即可。使用 wincvs 前请务必先安装 cvsnt。因为上面我们说过，wincvs 只是提供了图形的操作界面，图形的操作也是要转成相应的 cvs 命令来执行。但我们不是装服务器，所以不必配置 cvsnt，只要安装就好了。

2.2 用户登陆

现在我们打开 wincvs,在菜单栏里选择“admin->login”,勾选 CVSROOT,点后面的“...”按钮，添加 CVSROOT 信息，如下图：



- `protocol`: 传输协议, 默认用 `pserver`, 其他的没去研究。
- `repository`: 表示你的 `cv`s 仓库, 刚才我们已经建立了仓库 `oyoung`, 前面加正反斜杠都可以
- `username`: 现在就是我们的操作系统帐号
- `password`: 操作系统帐号密码
- `hostname`: 服务器的主机名或 IP
- `port`: `cv`snt 的端口号, 默认为 2401。
- 其余可不填 (没去研究☺)

配置完后, 点确定, 如果 `wincvs` 输出框提示如下图信息, 则表明登陆成功。注意一点, “`CVS exited normally with code 0`” 表示操作成功, 而 “`CVS exited normally with code 1`” 表示操作失败, 这个我们和经常用到的 Boolean 值刚好相反!

```

cvs -d :pserver;username=cvsadmin;password=abc123!;hostname=10.83.23.112;port=2401:/oyoung login
Logging in to :pserver:cvsadmin@10.83.23.112:2401:/oyoung

***** CVS exited normally with code 0 *****

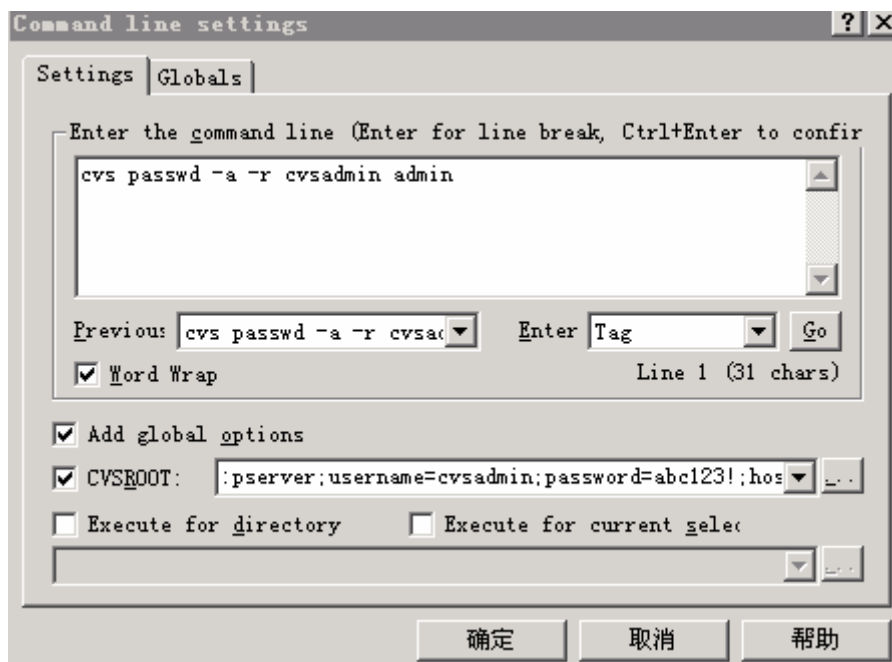
```

2.3 管理 CVS 用户

上面我们建立的是操作系统的用户, 可以用来登陆 `CVS`, 但是这样就依赖于操作系统, 而且我们不可能为每一个用户都建立一个操作系统帐号, 所以我们应该用 `CVS` 的帐户管理。

2.3.1 添加用户:

“`wincvs->admin->command line`”, 勾选 “`CVSROOT`”, 选择刚才输入的 `CVSROOT` 信息, 在命令框里输入 “`cvs passwd -a -r cvsadmin admin`”, 建立 `CVS` 帐号 “`admin`”, 并映射到系统帐号 “`cvsadmin`”, 然后输入密码, 如下图 (详细可参考 `passwd` 命令):



这样我们已经为 cvs 添加了用户 admin 现在我们看下服务器仓库 oyoung 下的 CVSROOT,就会发现多了个文件 passwd。里面一行代表一个用户，记录了映射的操作系统帐号和 cvs 帐号和密码，如下：

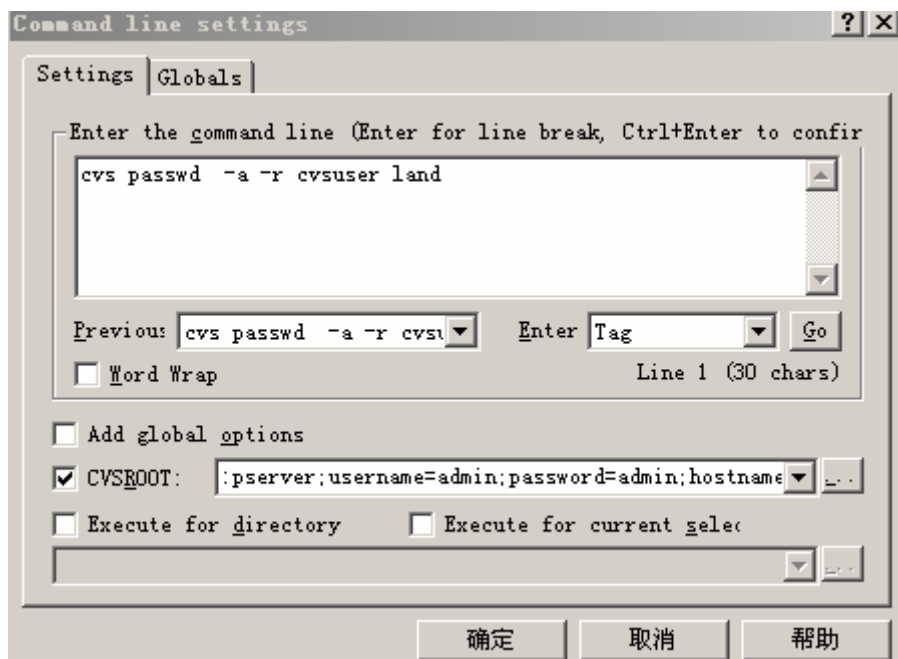
```
admin:A.f0Kvdiyy8q2:cvsadmin
```

2.3.2 赋予管理员权限：

上面我们已经添加了 cvs 帐号 admin,但只是一般的用户，我们需要指定哪些帐号为管理员，以有权限进行配置管理。为此我们进入到服务器上仓库 oyoung 下的文件夹 CVSROOT,新建一个文件 admin(注意无后缀名)，里面添加 admin 为管理员。可以添加多个管理员，一个帐号一行，如下：

```
admin
admin2
```

现在 admin 已经是管理员，可以使用管理员的功能了，如添加用户，管理用户权限等。现在我们就以 admin 身份了再添加一个普通用户 land，如下图：



```
cvscvs -d :pserver;username=admin;password=admin;hostname=10.83.23.112;port=2401:/oyoung passwd -a -r cvsuser land
Adding user land@10.83.23.112
***** CVS exited normally with code 0 *****
```

添加用户成功，以后我们还会谈到用户权限控制问题。

2.3.3 停用操作系统帐号

映射完毕后，我们就可以完全使用 cvs 帐号，而禁止操作系统帐号登陆。我们只需修改仓库 oyoung 下的 CVSROOT/config 文件（这个 CVSROOT 也可 checkout 到本地修改，下面 2.5 节会有介绍），将第 2 行修改，如下：

```
# SystemAuth=YES
```

修改为：

```
SystemAuth=NO
```

现在我们再用刚才的 cvsadmin 操作系统帐号登陆就会失败了！

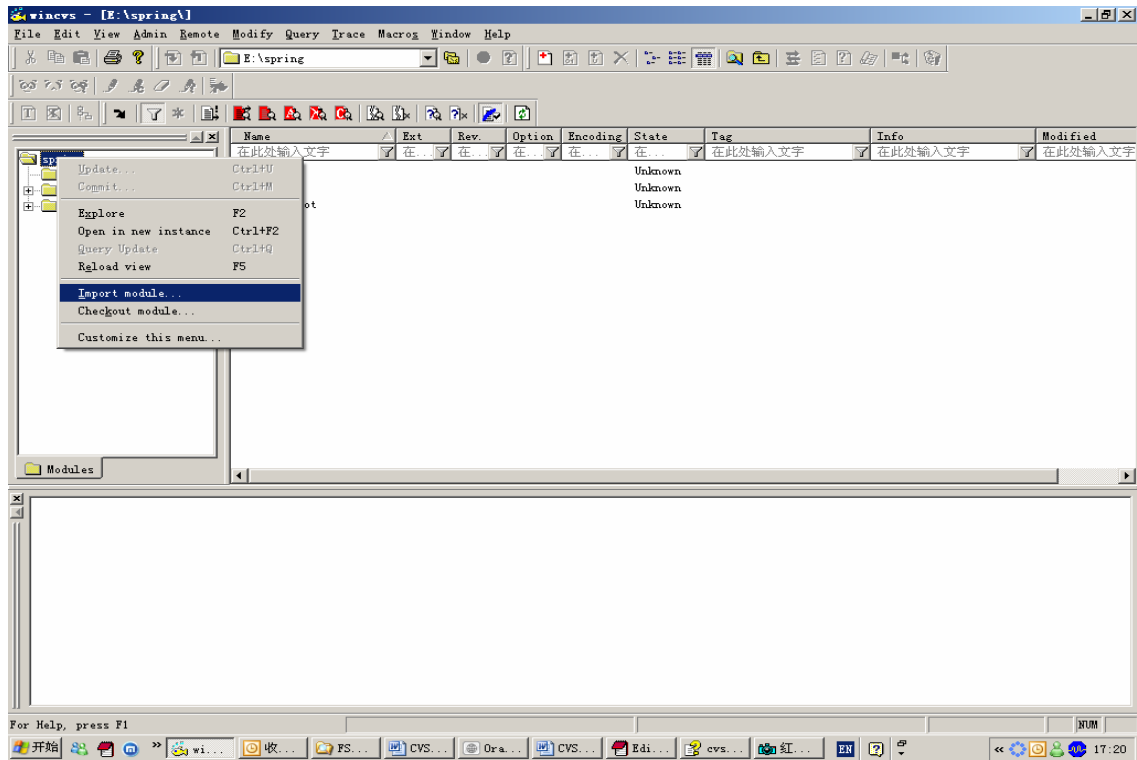
```
cvscvs -d :pserver;username=cvsadmin;password=abc123!;hostname=10.83.23.112;port=2401:/oyo... login
Logging in to :pserver:cvsadmin@10.83.23.112:2401:/oyoung
cvs [login aborted]: no such user cvsadmin in CVSROOT/passwd
***** CVS exited normally with code 1 *****
```

2.4 导入模块

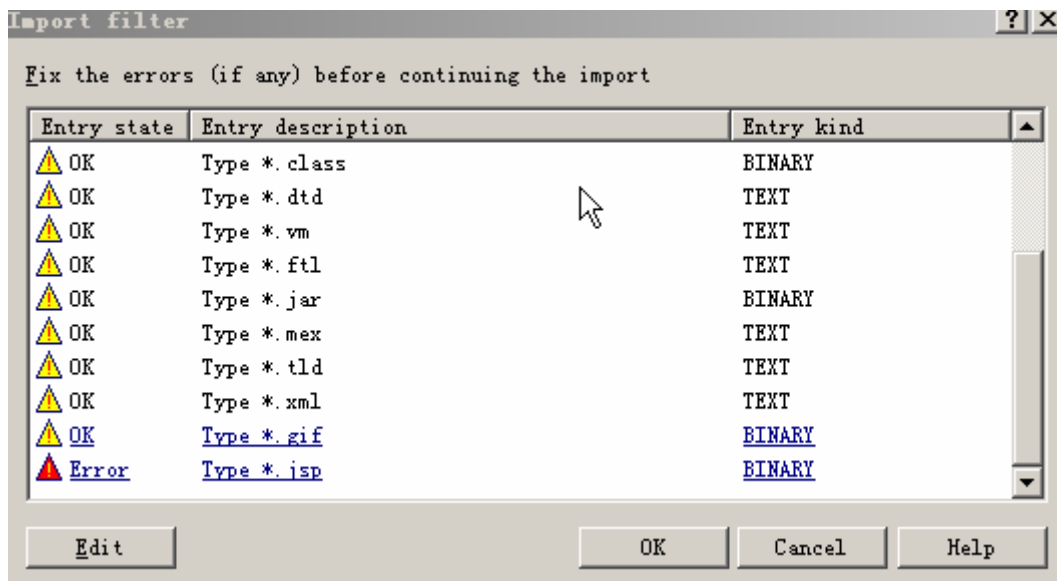
一切都准备好以后，我们就可以把需要进行版本控制的文件导入到我们刚建立好的 oyoung 仓库了。比如我要把 spring 文件夹里的全部东西导入到仓库 oyoung 里。这里有两种办法：

2.4.1 利用 wincvs 图形导入

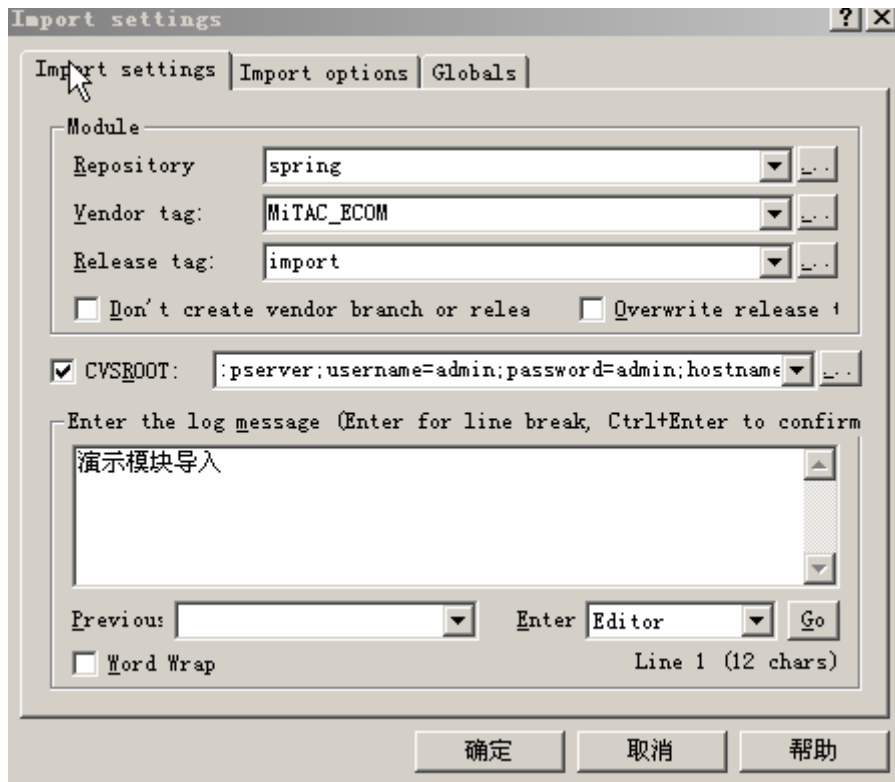
在 wincvs 下工具栏的浏览框里找到你要导入的目录 spring,然后在左边树形菜单里右键 spring,选择”import module...”, 如下图：



Wincvs 会对要导入的文件过滤，提示一些信息（如下图），但我们一般可以不理它，如果文件很多的话，这样的过滤会消耗很多时间，所以我们可以用下面的第 2 种方法（命令行导入），这样就可以避免这些过滤。



点 ok 后，弹出对话框填写 Repository(服务器上的模块名), Vendor tag(厂商标签), Release(发布标签), 和选择 CVSROOT. 标签里不要使用空格，最好用下划线“_”连接。如下图：



确定后，如果提示如下信息，那么恭喜你！导入成功了！

```
N spring/src/com/crackjee/example/spring/LogAdvisor.java
N spring/src/com/crackjee/example/spring/PasswordAdvisor.java
N spring/src/com/crackjee/example/spring/ExceptionAdvisor.java
cvs server: Importing D:/cvs_repository/oyoung/spring/empty
N spring/empty/readme.txt

No conflicts created by this import

***** CVS exited normally with code 0 *****
```

2.4.2 命令行导入

上面提到过，可以用命令行导入来避免文件的过滤。实际上，我们在 cvs 上用到的一切操作，都是要转成 CVS 相应命令执行的。如上面的图形导入，我们在输入框里就可以看到它转成了下列的命令执行。如下图：

```
cvs -r -d :pserver;username=admin;password=admin;hostname=10.83.23.112;port=2401:oyoung import -I ! -I CVS -W "*.java -k 'b'" -W "*.class -k 'b'" -W "*.jar -k 'b'" -W "*.gif -k 'b'" -W "*.jsp -k 'b'" -m 演示模块导入 -spring MiTAC_ECOM import (in directory E:\spring)
```

现在我们打开“admin->command line”，为了避免和刚才导入的模块名 spring 重复，我们把模块名改成 spring_cmd,最后命令如下：

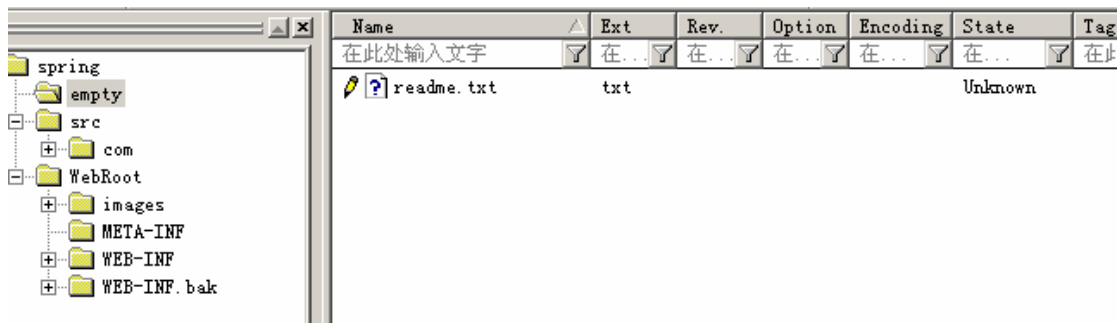
```
cvs import -m "演示命令行导入" spring_cmd MiTAC_ECOM import
```

然后选择 CVSROOT,如下图：



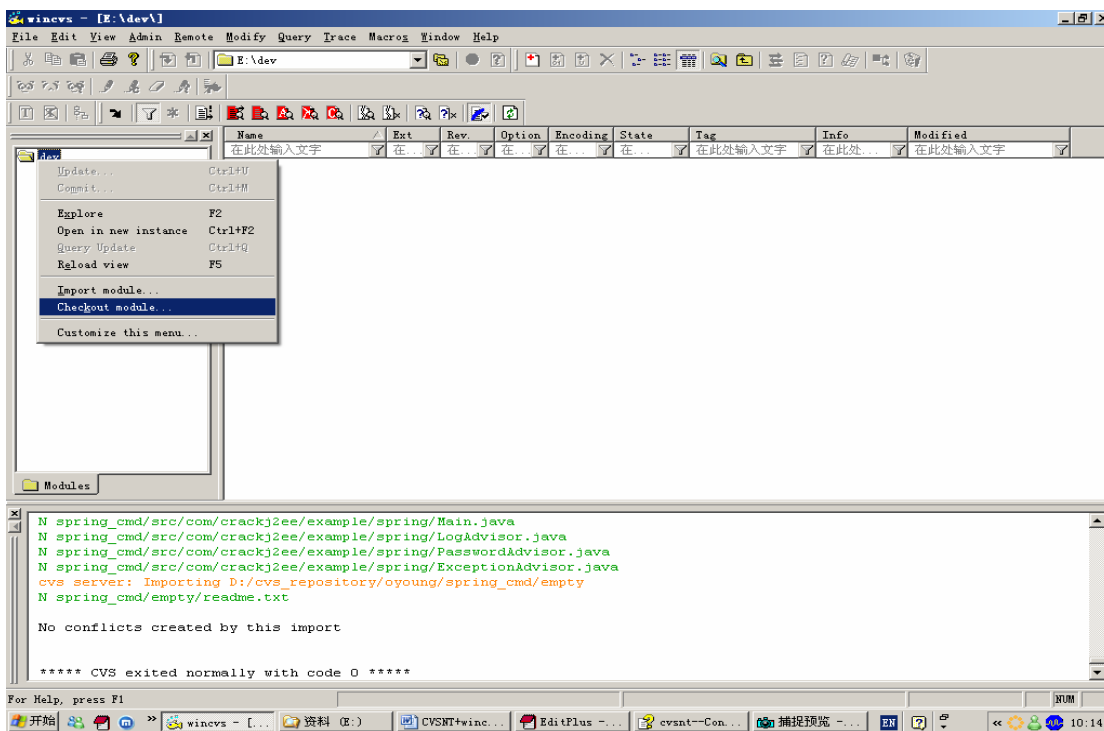
确定后，我们同样可以看到导入成功，并在服务器上多了一个 spring_cmd 模块。

最后要注意的是，CVS 默认认为空目录不存在，所以如果 spring 里有空文件夹，这个空文件夹将不会被导入。但是有时我们需要导入一个空目录，比较笨的方法就是随便建立一个文件，空文件也可以。如上面的 spring/empty,我就建了一个空文件 readme.txt.如下图：

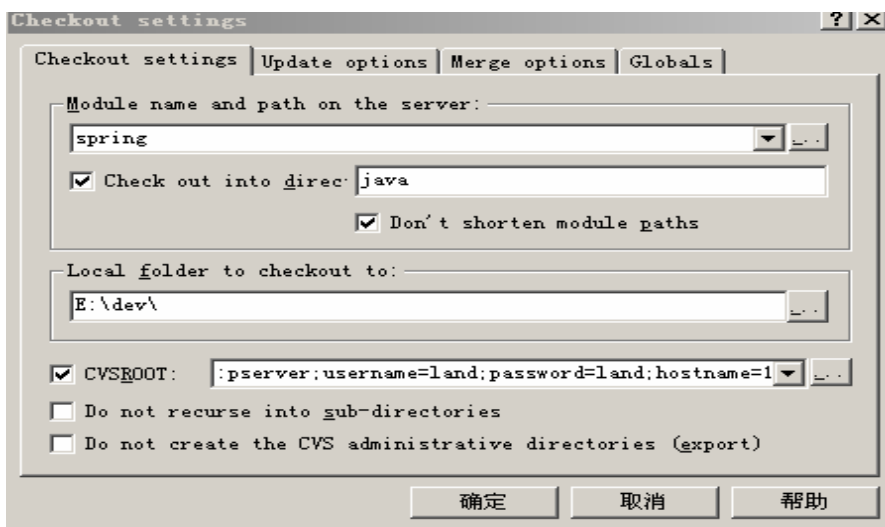


2.5 导出模块

在导入模块后，我们就可以把模块导出来进行控制了。首先我们在 cvs 浏览框里选择你要把模块导出到哪里，比如我的是 E:\dev，然后在菜单里右键 dev.选择“checkout module”，如下图：

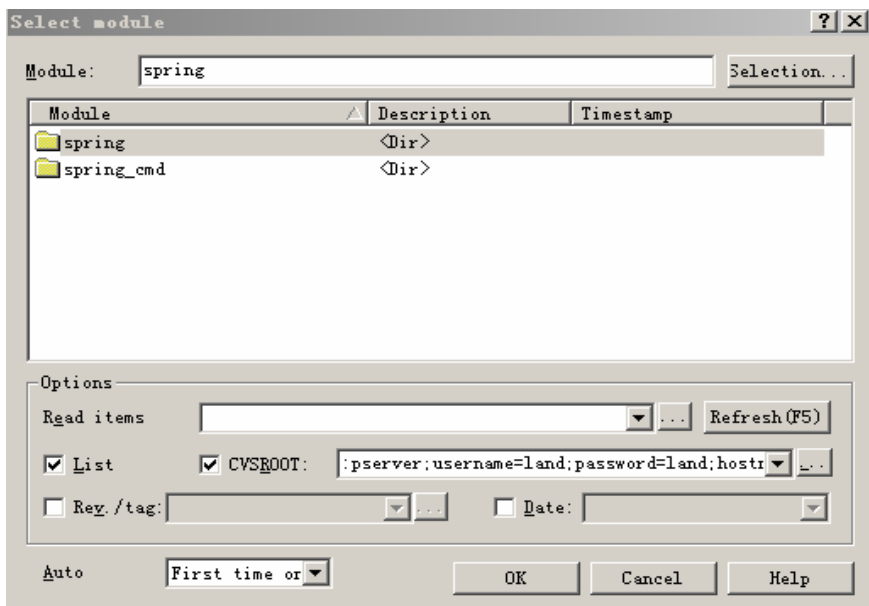


在弹出的对话框里填入相关信息，如下图：



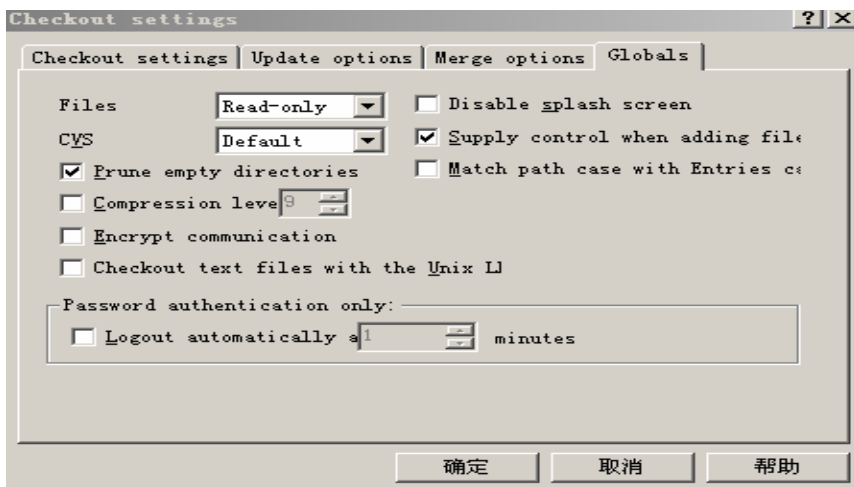
- Module name path on the server: 要导出的模块名或模块下的某个目录
- Check out into directory: 导出到本地的目录结构
- Don't shorten module paths: 截短模块路径，是 Check out into directory 的子项。勾选则把模块（包括 WebRoot, src）导出到 e:\dev\java\spring.; 否则导出到 e:\dev\java
- CVSROOT: 这里我们用普通用户 land 身份导出模块。
- Do not recurse into sub-directory: 不导出子目录的内容
- Do not create the CVS administrative directory: 不导出 cvs 管理文件。这在我们项目发布时相当有用。

如果我们不清楚模块名，或者只想导出模块下的某个目录，那么我们可以点“Module name path on the server”后面的浏览按钮。在对话框里选择 CVSROOT 和勾选 List, 再点“Refresh(F5)”，就可以看到刚才我们导入的两个模块，如下图：

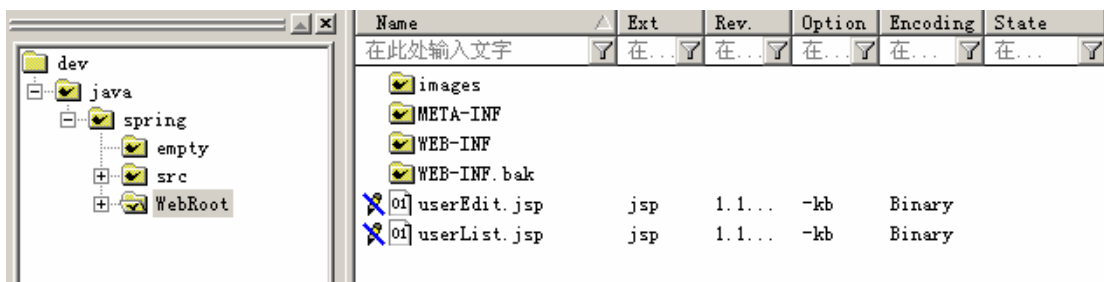


这样我们就可以自由选择要导出的目录了。如果你是管理员身份，那么你还可以看到 CVSROOT 仓库管理模块，你也可以导出到本地来修改文件。

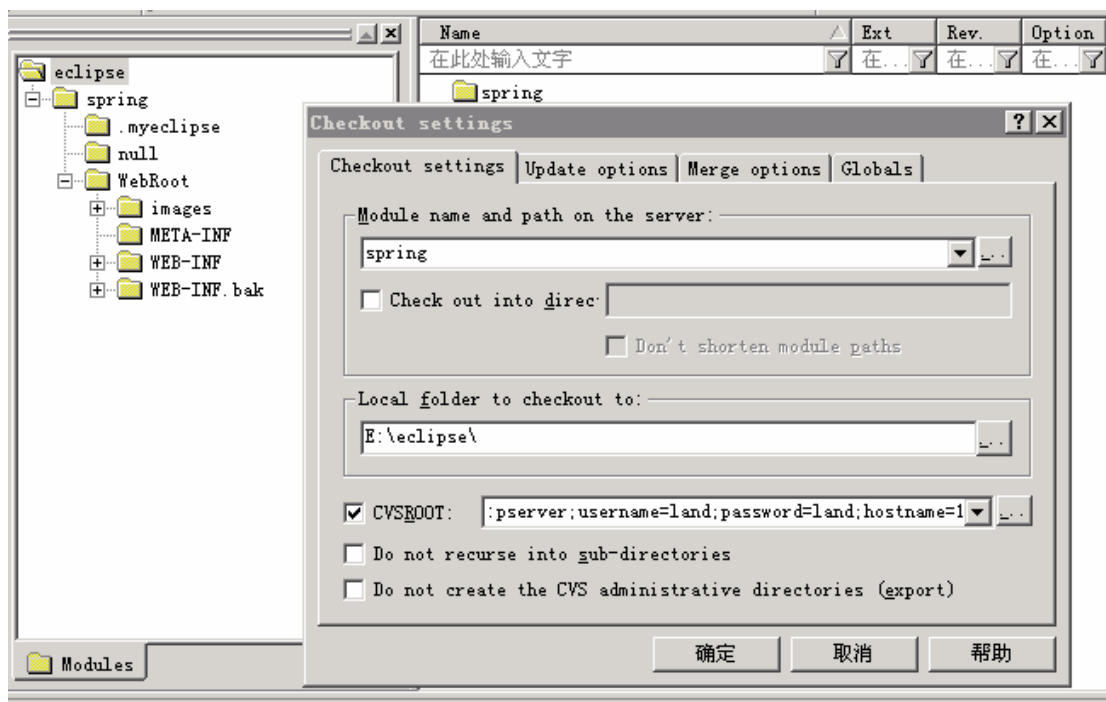
另外我们还需要在 Globals 面板里设置 Files 为“Read-only”，这样导出的文件为只读。以后我们 commit 一个文件后，文件也会变成只读，这对文件冲突管理相当有用。”Prune empty directory”，勾选则不导出空目录，这个视需要而定。其余选项就不多说了。如下图：



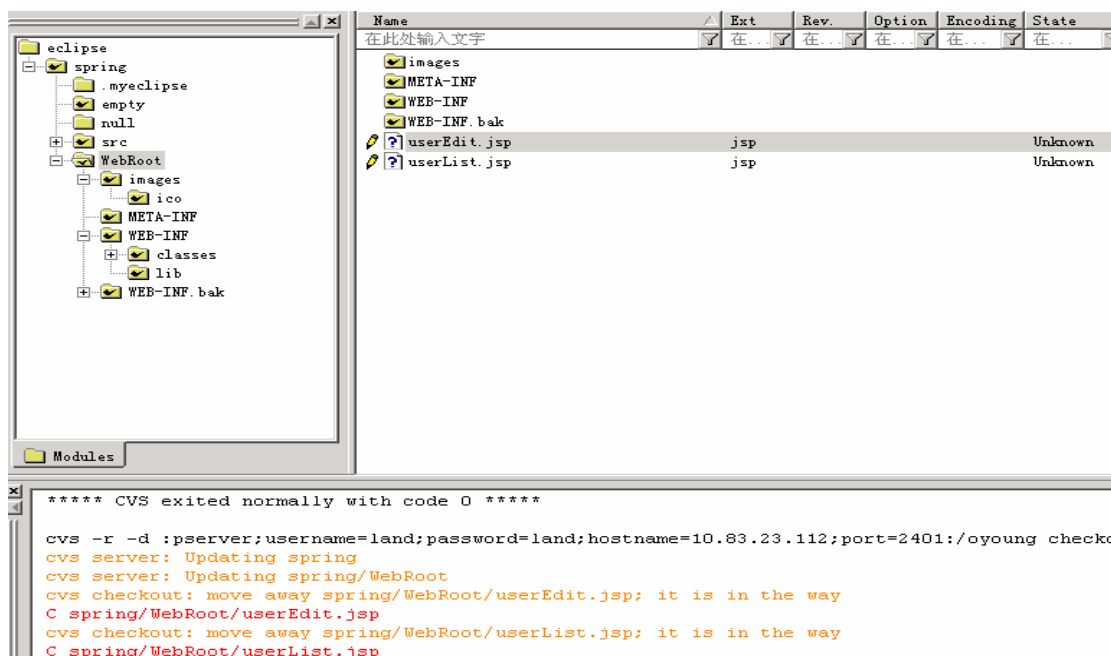
确定导出成功后，我们在 e:\dev\java 下看到了 spring 这个模块。如下图：



注意：导出的目录里不能有和 cvs 上的目录，文件名相同，否则会提示 “it is in the way “ 错误。比较常见的情况是，可能我们项目中途才开始用 CVS 来管理，那么开发人员本地已经有了模块的某些部分（如已有了 e:\eclipse\spring\WebRoot），只需没有那部分就行了(如 src)。那么我们想只需要在 e:\eclipse\导出 spring 模块，WebRoot 会被 cvs 上的覆盖，而 src 也导出来了。如下图：



确定后，我们发现 src 模块已经被成功导出，但 WebRoot 模块却出现了问题。如下图：



因为 WebRoot 模块里已经存在了 userEdit.jsp CVS 就无法再将这个文件导出覆盖本地文件，所以 userEdit.jsp 状态还是 Unkown,没有和 cvs 关联。解决办法就是在本地把 WebRoot 文件夹全部删除，然后再从 CVS 全部导出，实际上不算办法^_^（不知道 CVS 为什么不提供覆盖☹）。

至此，CVS 的安装和基础配置我们就已经做好，包括经建立 cvs 服务器，建立仓库，导进模块，导出模块。现在我们就可以使用 cvs 进行版本控制，至于 wincvs 的使用，很多文章都有介绍，我就不再废话了。下面讲下一些使用脚本语言和用户权限控制。

进阶篇

3 使用脚本语言

3.1 安装脚本

3.1.1 安装 python

版本：2.4.3

下载地址：<http://www.python.org/ftp/python/2.4.3/python-2.4.3.msi>

安装比较简单，直接点下一步即可。

3.1.2 安装 perl

版本：5.8.8.817

地址：<http://downloads.activestate.com/ActivePerl/Windows/5.8>

/ActivePerl-5.8.8.817-MSWin32-x86-257965.msi

安装比较简单，直接点下一步即可。

3.1.3 安装 tcl

版本：8.4.13

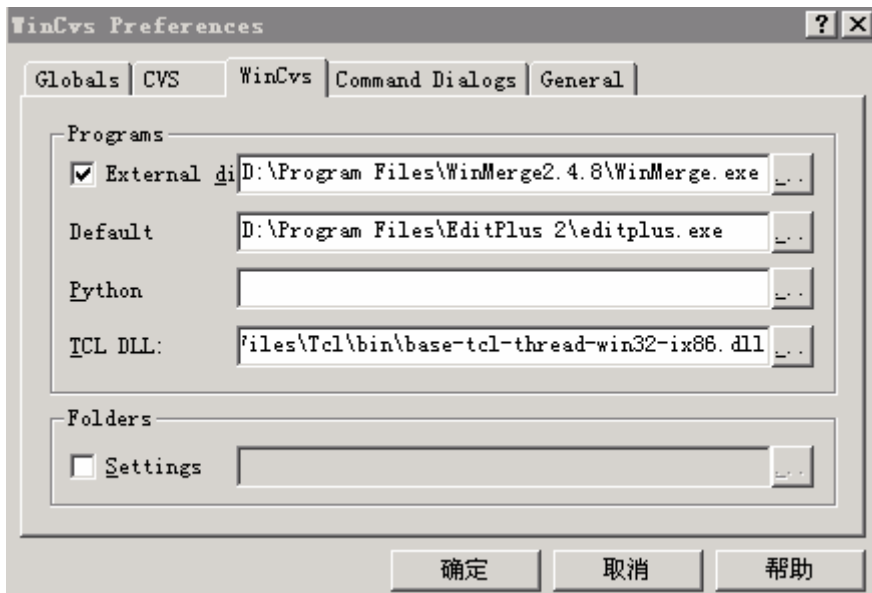
地址：<http://downloads.activestate.com/ActiveTcl/Windows/8.4.13>

/ActiveTcl8.4.13.0.261555-win32-ix86-threaded.exe

安装比较简单，直接点下一步即可。

3.2 指定语言库路径

安装完 3 个脚本语言之后。打开 wincvs->Admin->Preference->Wincvs.指定 TCL DLL 的路径到你安装的 tcl 库。注意不用指定 Python，如下图：



3.3 使用脚本功能

完成上述的步骤后，打开 wincvs->Macros.我们就可以看到多了很多的功能。这些都是用上面 3 种语言写的，代码在你安装的 Macros 目录下，如 D:\Program Files\GNU\WinCvs 2.0\Macros。当然，如果你也可以用这些脚本语言自己写一些功能。我们也可以下载脚本放到该目录下，相关下载见：

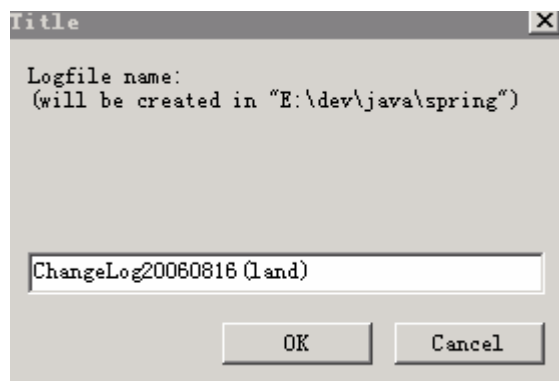
http://ximbiot.com/cvs/wiki/index.php?title=CVS_Clients。

在这里，我们只介绍其中的一些功能，其他的自己可以试一下。

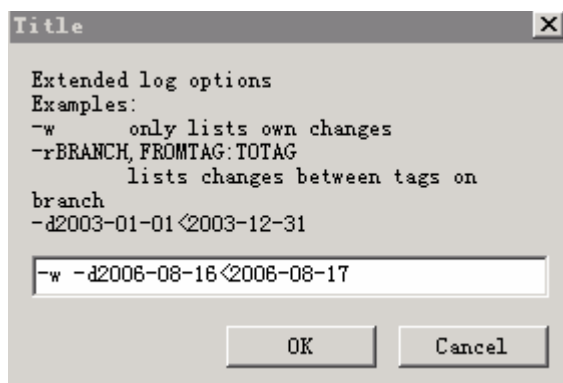
3.3.1 产生修改日志

这个功能可以根据条件统计所有 commit 到服务器的文件。

首先进入 Macros->cvs->Build changeLog (advanced)，给文件命名：ChangeLog20060816(land)，如下图：

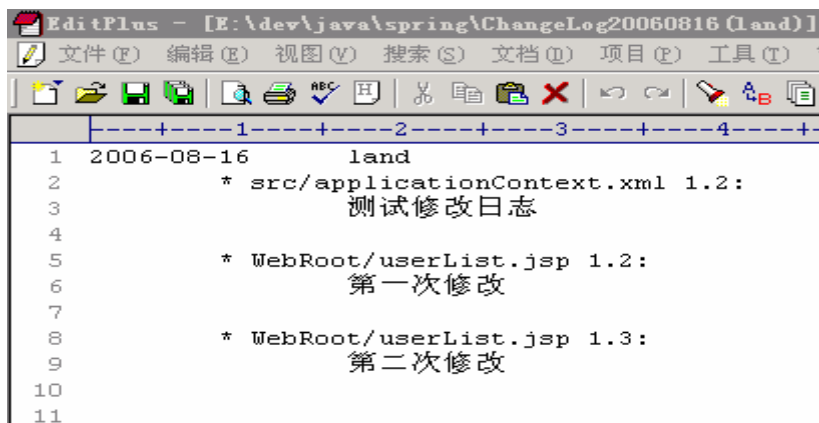


点 ok 后，再填写选项，如下图：



- `-w` 只列出登陆人的修改文件。否则列出全部用户
- `-r` 列出某个分支，从开始到结束标签之间的
- `-d` 列出日期之间的。这里”<”表示大于等于开始日期，小于结束日期，所以上面命令 `d2006-08-16<2006-08-17` 只列出 2006-08-16 这一天的修改文件。

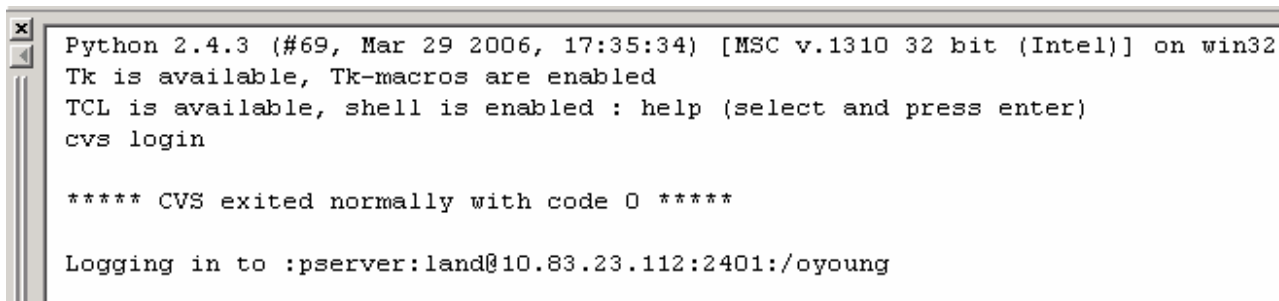
OK 后就会在当前目录产生文件 `ChangeLog20060816(land)`，内容如下图：



我们可以看到，它列出了修改人 `land` 在 2006-08-16，把修改后版本是 1.3 文件的 `userList.jsp` Commit 进了服务器，注释是：第二次修改。

3.3.2 利用输出窗口当命令窗口

现在我们可以 `wincvs` 的输出窗口里直接输入 `cvs` 命令，然后回车执行即可，非常方便。比如我们输入登录命令，然后回车，它就会以当前路径的 `CVSROOT` 登录，如下图：



4 用户权限控制

4.1 设定管理员

首先我们要指定管理员。添加一个帐号 `admin`,然后赋予它管理员的权限, 具体操作见上面 2.3.

4.2 建立用户组

我们可以把用户权限我们都分配到组, 这样方便于管理。在仓库 `oyoung` 的 `CVSROOT` 里建立文件 `group`, 并把相应帐号归到各个组下, 帐号之间用空格间隔。如下:

```
PMGrp:land
RDGrp:simon oyoung
TestGrp:test
```

上面建立了 3 个组, `PMGrp`, `RDGrp`, `TestGrp`, 然后我们把用户 `land` 归到 `PMGrp`, `simon` 和 `oyoung` 归到 `RDGrp`, `test` 归到 `TestGrp`。

4.3 赋予权限

CVS 有 5 种权限, `read,write,create,tag,control`, 各种权限对应的操作如下:

- Read:可以更新, 读取
- Write:可以编辑,commit,删除
- Create:可以建新的文件
- Tag: 可以打标签
- Control: 可有权限修改文件的权限,

.如果我们没有限制, 那么全部用户都可以拥有上述前 4 种权限。而 `control` 这个权限会分配给文件创建者和管理员。

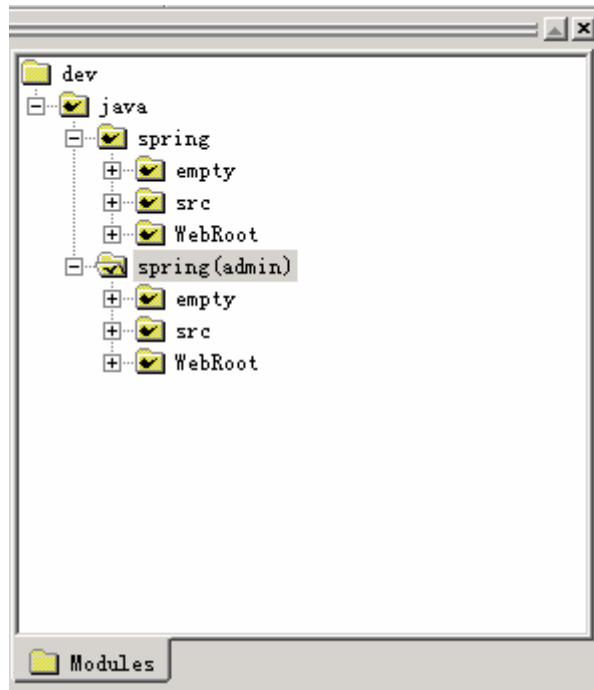
4.3.1 显示权限列表

命令: `cvs lsacl`(详细可参考 `Wincvs` 自带的 `CVS` 帮助文档)

选项:

- `-d` 只列出目录而非文件
- `-R` 递归

首先我们要以 `admin` 身份导出 `spring` 模块, 为方便比较, 模块还是导出到 `e:/dev/java`,不过模块名改成 `spring(admin)`,以区分 `land` 身份导出的 `spring`.如下图:



我们选中 spring(admin)/WebRoot/WEB-INF,然后在 Wincvs 的输出框里,输入命令”cvs lsacl”,然后回车,结果如下图:

```

cvs lsacl

***** CVS exited normally with code 0 *****

Directory: .
Owner: admin

```

列出了 WEB-INF 的 owner 是 admin.

4.3.2 添加删除用户权限

命令: cvs chacl(详细可参考 Wincvs 自带的 CVS 帮助文档)

选项:

- -a 添加权限
- -d 删除权限
- -u 指定用户或组
- -m 指定错误提示信息
- -p 指定优先级

4.3.2.1 添加用户权限

比如,我们禁止组 PMGrp 对 spring(admin)/WebRoot/WEB-INF 的写权限,这样组成员 land 就不能 update 和 edit WEB-IN 文件夹里的所有文件。首先选中 spring(admin)/WebRoot/WEB-INF,然后输入命令,如下图:

```

cvs chacl -a nowrite -u PMGrp -m "不好意思,您没有权限执行该操作,请联系oyoung"

***** CVS exited normally with code 0 *****

setting ACL for directory .

```

现在我们用 cvs lsacl 看下权限分配情况,结果如下图:

```

cvs lsacl

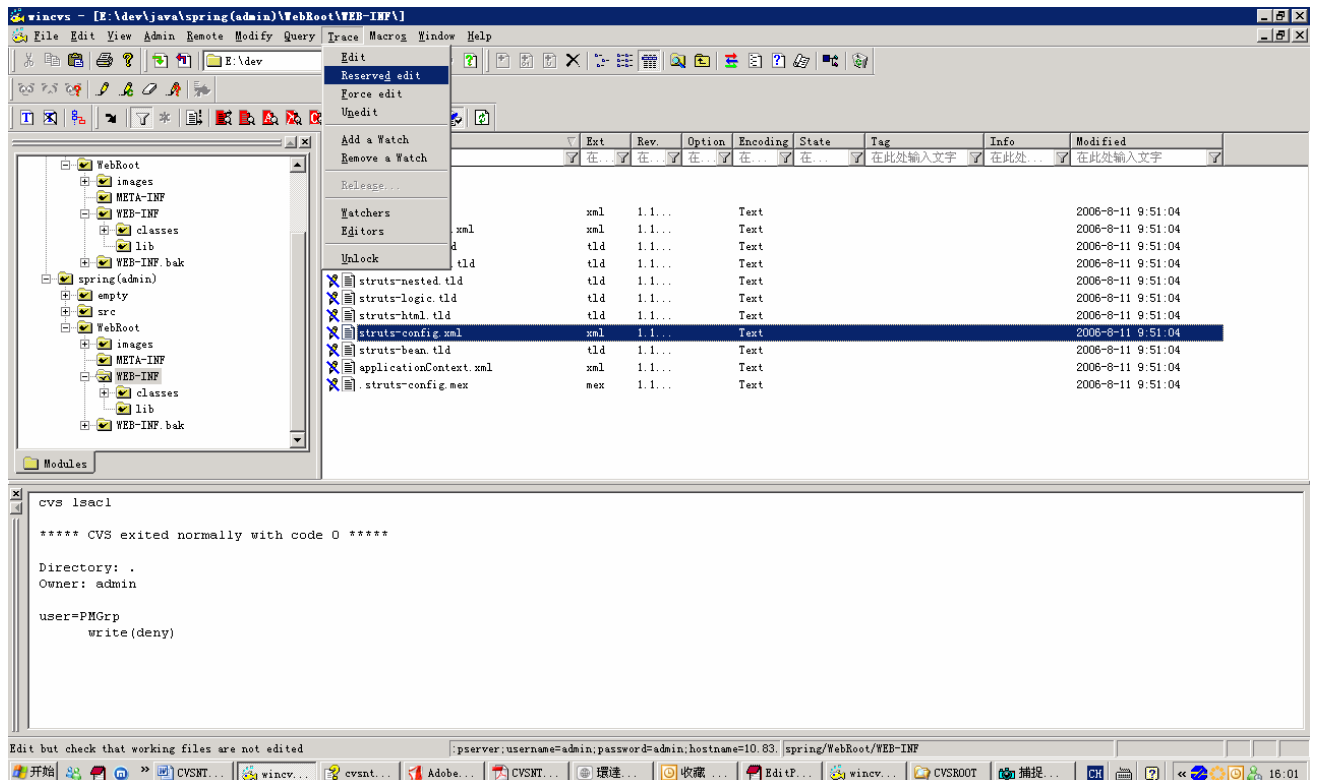
***** CVS exited normally with code 0 *****

Directory: .
Owner: admin

user=PMGrp,message=不好意思,您没有权限执行该操作,请联系oyoung
write(deny)

```

已经显示添加了禁止写权限，那么我们再看下实际情况，我们选中 spring/WebRoot/WEB-INF/struts-config.xml（这个是以 land 身份登录的），然后选中菜单栏的 Trace->Reserved Edit 编辑它，如下图：



结果提示，编辑错误，如下图：

```

cvs -r edit -c -- struts-config.xml (in directory E:\dev\java\spring\WebRoot\WEB-INF\)
cvs server: Edit on file 'struts-config.xml' refused by server
cvs edit: invalid data from cvs server

***** CVS exited normally with code 1 *****

```

如果我们强行修改，然后 commit，也会提示错误，如下图：

```

cvs -r commit -m "test acl" -- struts-config.xml (in directory E:\dev\java\spring\WebRoot\WEB-INF\)
cvs server: User land is unable to write to ./struts-config.xml - ignoring
cvs server: 不好意思,您没有权限执行该操作,请联系oyoung

```

4.3.2.2 删除用户权限

我们删除刚才添加的权限，修改命令，如下图即可：

```
cvsc> cvs chacl -d -u PMGrp

***** CVS exited normally with code 0 *****

deleting ACL for directory .
```

4.4 权限规则

有时我们对目录和其下的文件，或者对用户和所在的组设定不同的权限，这样就会带来权限的冲突。CVS 会遵循一定的规则，经过实际测试，总结了以下几条规则：

- 1 目录权限>文件权限
当对目录和其下的文件赋予不同的权限给同一个用户时，则目录权限覆盖文件权限。
- 2 用户权限>组权限
当对同一文件赋予了不同的权限给用户和其所在组时，用户权限覆盖组权限。
- 3 顺序规则
当用户属于多个组，而各个组发生权限冲突时，则以 `cvs lsacl` 列出的顺序靠前的为准。而排序的参数是以我们设定权限时用 `-p` 指定的参数，排序规则按字符串升序排。

结束语

花了 3 天时间终于把整个文档整理了出来，都有点累不想写了，所以后面权限规则我都没做实际演示，希望各位能够原谅。其实我还想写下 Wincvs 的日常使用，不过网上这方面的资料很多，我就懒得再去大费口舌了。需要提醒一点就是，关于文件冲突管理部分：我们编辑文件时最好用 **Reserved Edit**（保留编辑），这样第 2 个人再用 **Reserved Edit** 编辑这个文件时，CVS 就会告诉他错误，因为某某在哪什么时候正在编辑这个文件。好了，再说就不叫结束语了，希望这个文档对大家有所帮助！也欢迎来信交流指正！