

# CVS 入門操作

學校：台北科技大學

編寫者：呂宗螢

指導教授：梁文耀 老師

## CVS 指令快速查表

指令	作用
cvcs login	登入 CVS
cvcs import	放個 module 進 CVS 管理
cvcs checkout (co)	拿出一個 module
cvcs add	增加檔案
cvcs remove	移除檔案
cvcs commit (ci)	確認改變，每次都要輸入改變的資訊，叫 log message。(-m “message”)
cvcs update (up)	別人有新的送交行為時，更新自己的版本，同步櫃子的版本。
cvcs status (st)	看目前 repository 裡面有那些檔，檔案版本等等資訊 -l 只看 local
cvcs diff	看兩個版本的差異
cvcs log	看一下 log 歷史紀錄
cvcs annotate	查看誰對這個檔案有修改
cvcs logout	登出 CVS

## 設定 CVS 環境

首先

在 .bash\_profile 加入 export CVSROOT=:pserver:帳號@140.124.183.91:2401/cluster/2/home/cvs/後，重新登入

這是爲了方便以後下 cvs 指令

## Windows CVS 軟體

TortoiseCVS : <http://www.tortoise cvs.org/>

# CVS 操作

## 一、 登入 CVS

```
cv$ login
```

或

```
cv$ -d ":pserver:帳號@140.124.183.91:2401/cluster/2/home/cvs/" login
```

正確回答密碼之後，cv\$ 會將認證資訊，儲存在 `~/.cvspass` 中，之後，就不必再做登入的動作了。  
(除非刪除了 `.cvspass`)

## 二、 登出 CVS

```
cv$ logout
```

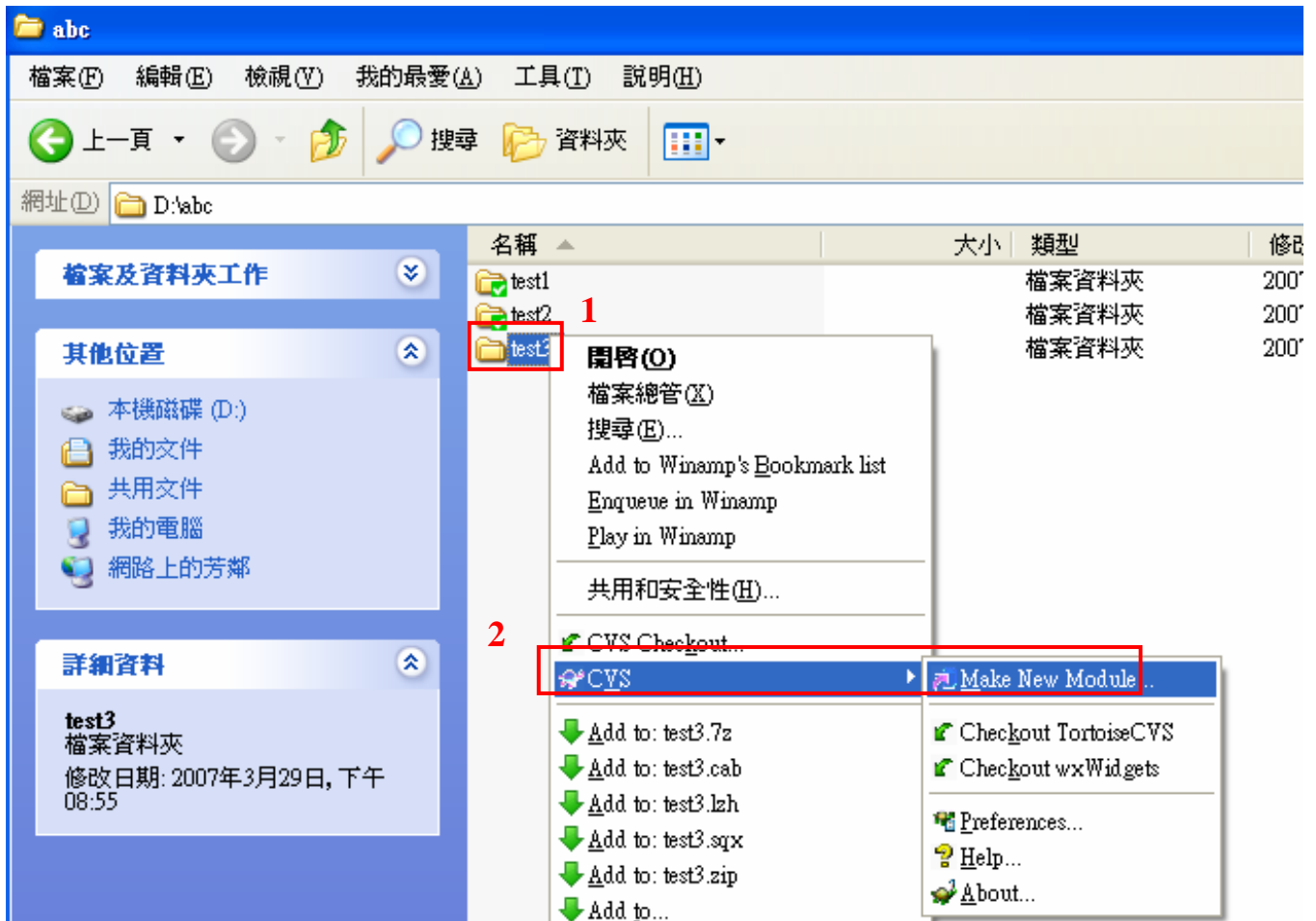
## 三、 新增一個 module 於 cvs 維護

```
cv$ import module\_name ntut_eps start
```

會將目前目錄下的所有檔案上傳至 cvs 管理，`module_name` 為上傳至 cvs 顯示名稱。`Ntut_eps` 是一個廠商標籤(vendor tag)，`start` 是版本標籤。這時會跳出 vi 來叫你輸入對這次 import 模組的 log 資訊。

### TortoiseCVS 操作

1. 點選資料夾
2. 右鍵->CVS->Make New Module



### 3. 設定

protocol : Password server(:pserver:)

server : 140.124.183.91

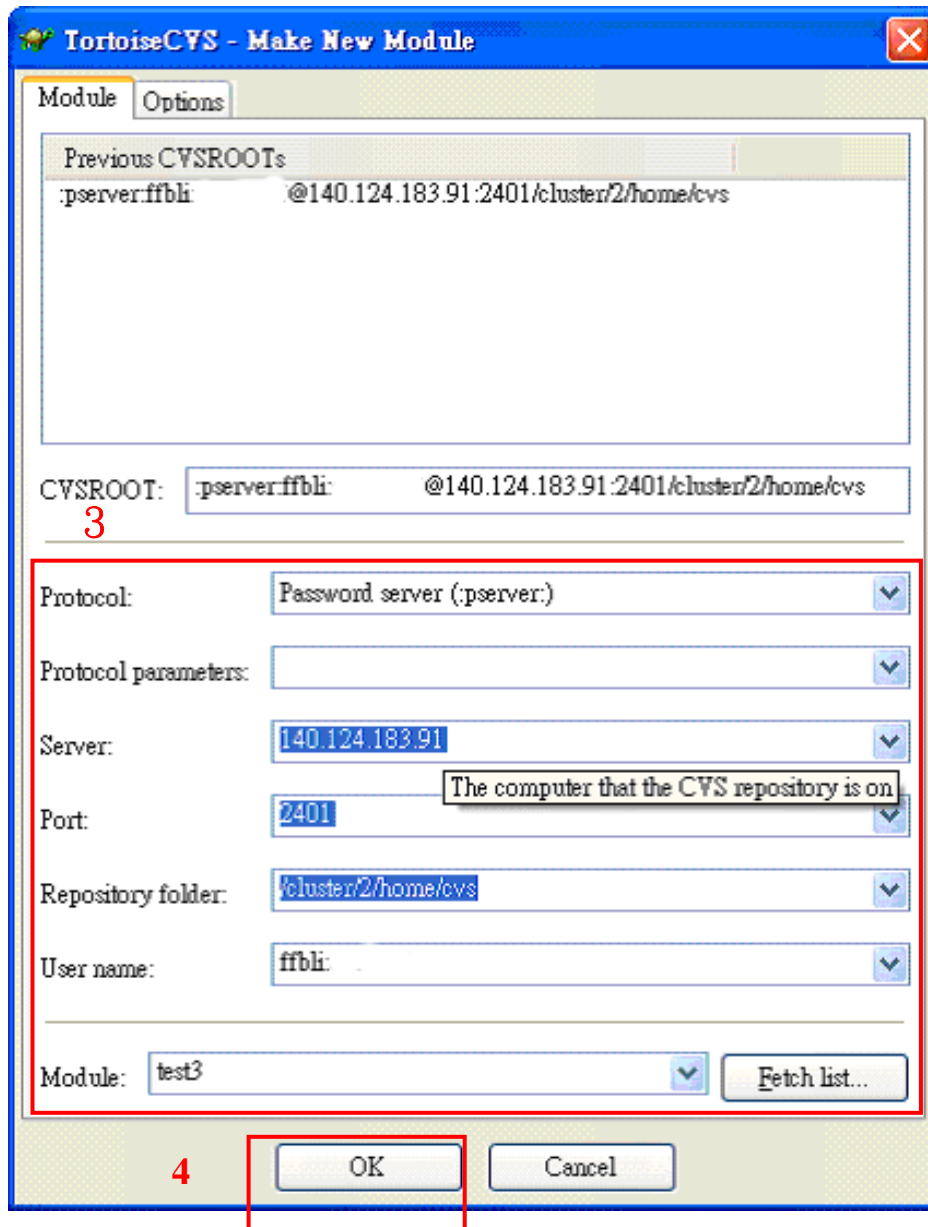
port : 2401

Repository folder : /cluster/2/home/cvs

User name : 帳號:密碼(以冒號隔開)

Module : module 名稱(如果需要更改的話)

### 4. 按 ok 完成



#### 四、 取出 module

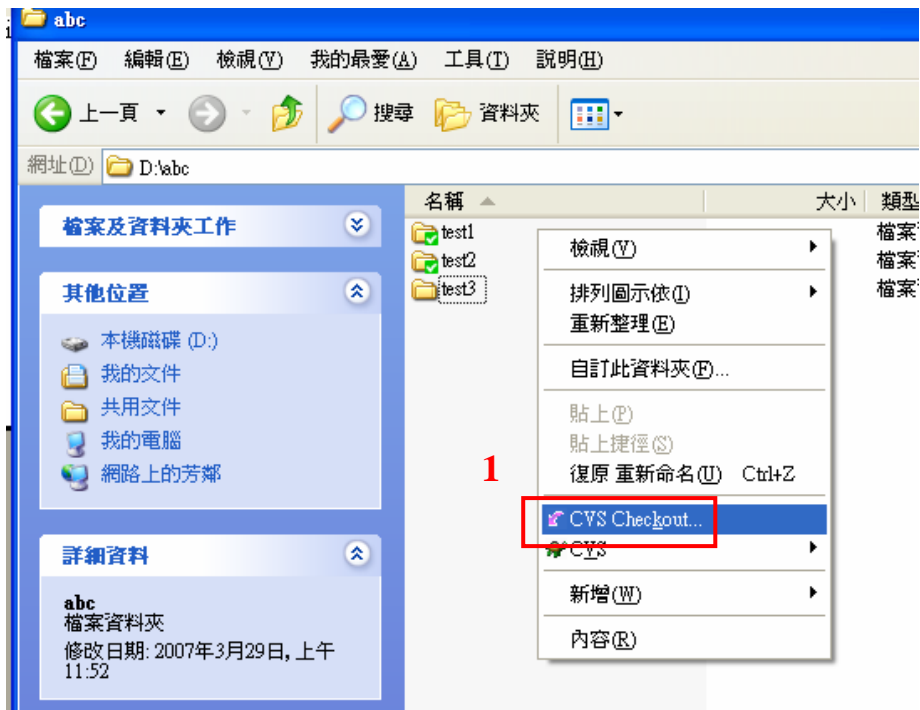
cvsc checkout [module\\_name](#)

或

cvsc co [module\\_name](#)

#### TortoiseCVS 操作

1. 右鍵點選 checkout



## 2. 設定

protocol : Password server(:pserver:)

server : 140.124.183.91

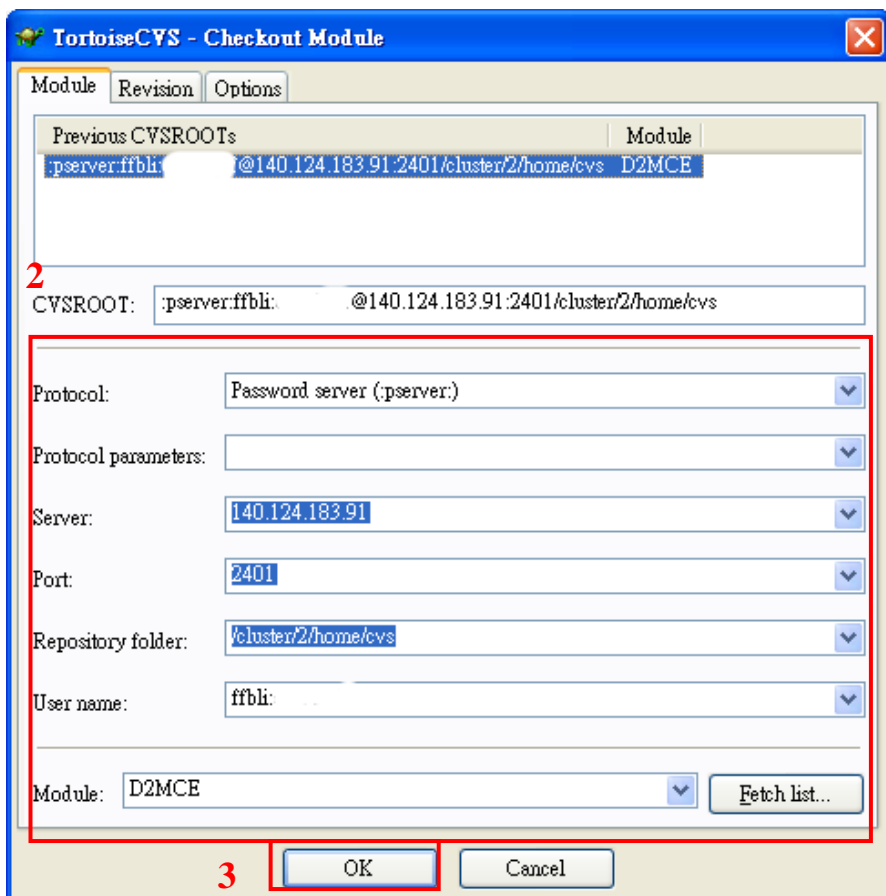
port : 2401

Repository folder : /cluster/2/home/cvs

User name : 帳號:密碼(以冒號隔開)

Module : 想要取出的 module 名稱(如果需要更改的話)

## 3. 按 ok 完成



## 五、 曾經 checkout 過，想取得最新檔案

cv<sup>s</sup> update //會更新所有的檔案及子目錄內容

或

cv<sup>s</sup> up

或

cv<sup>s</sup> up -d //加上-d 則會將 cv<sup>s</sup> 檔案庫新增的目錄也下載

或

cv<sup>s</sup> update file\_name

### TortoiseCVS 操作

#### 1. 右鍵 CVS update



## 六、 新增檔案及目錄

1. 建立檔案 or 目錄
- 2.

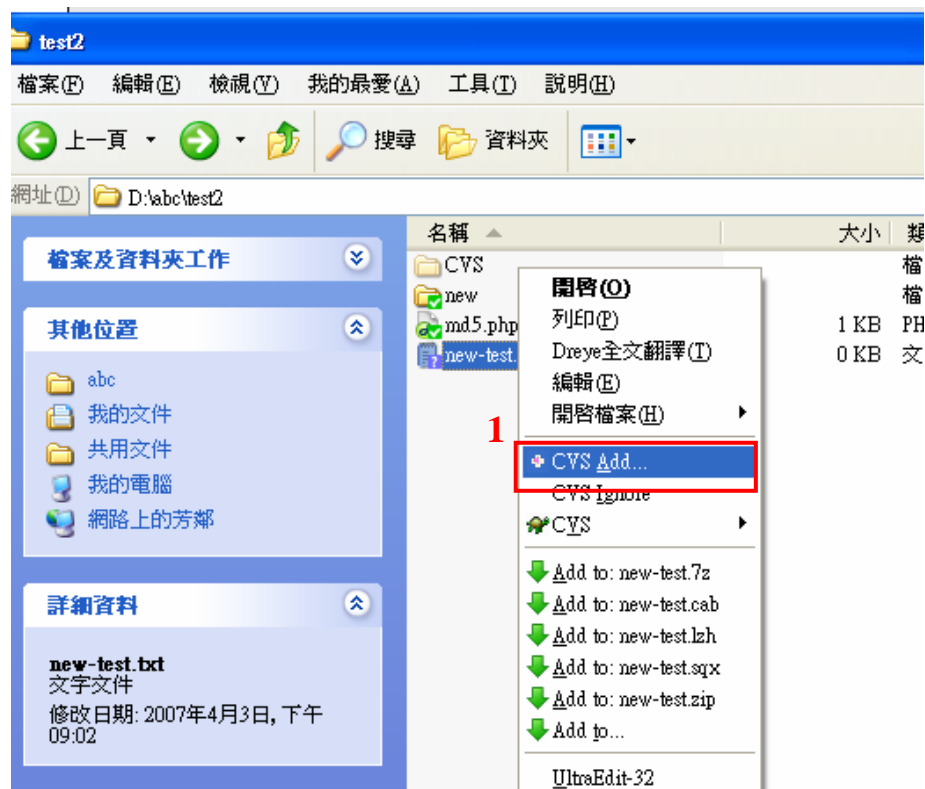
```
cv$ add file_name/directory
```

3. 新增檔案時才需要(目錄可以不用)

```
cv$ commit -m "add file" file_name
```

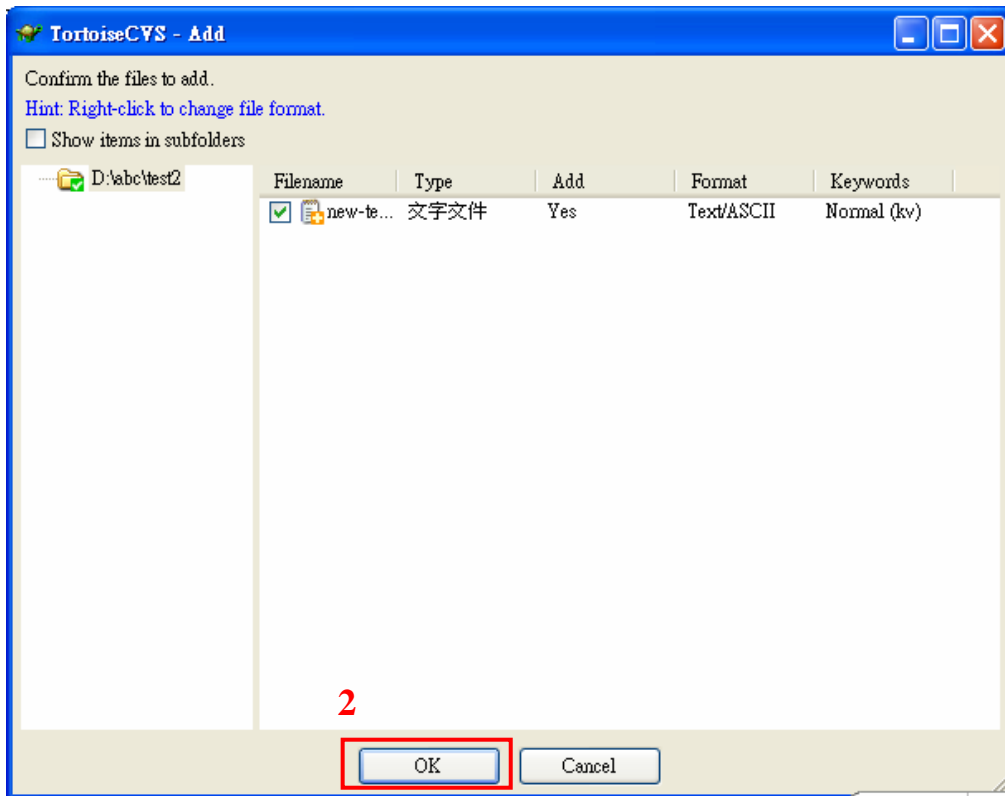
### TortoiseCVS 操作

1. 點選要加入檔案/目錄, 右鍵 CVS Add



2. ok 確認





## 七、 修改程式，並且更新 **CVS** 檔案庫

1. vi 編寫已有檔案(file\_name))
2. 下指令

```
cvcs commit file_name
```

或

```
cvcs ci file_name
```

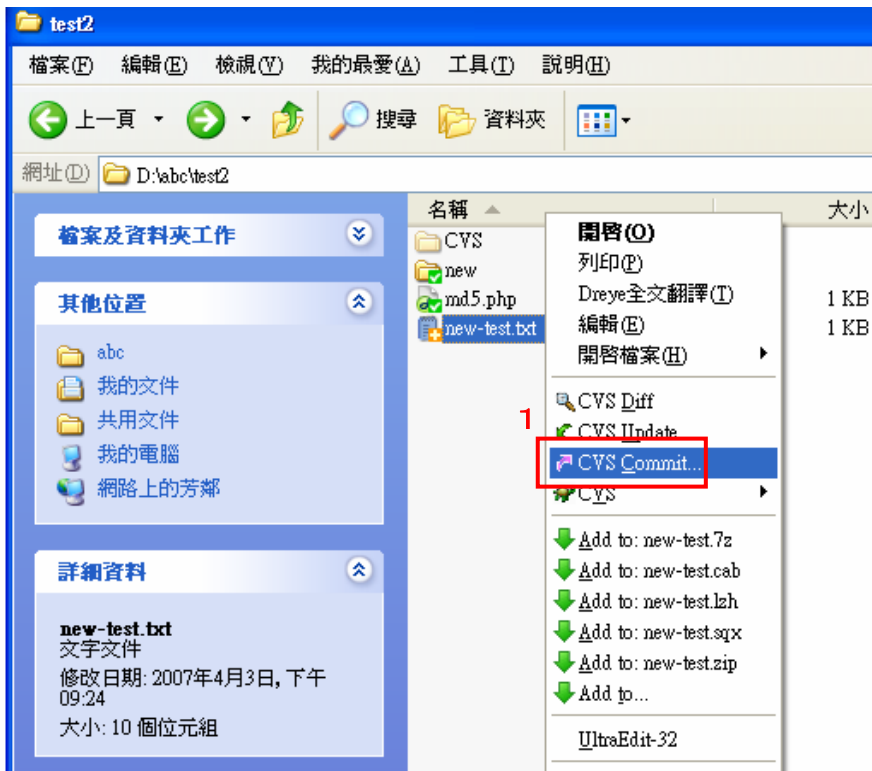
或

```
cvcs commit -m "just test" file_name
```

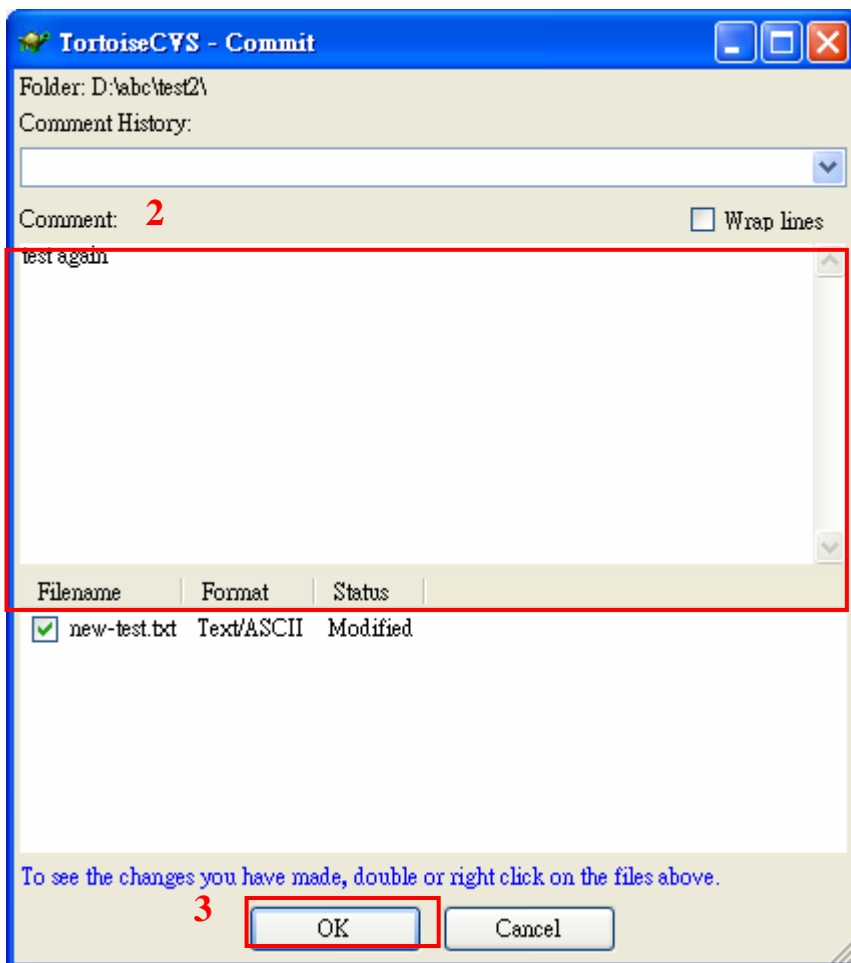
如果不加 file\_name 的話，則為目前目錄下所有所有檔案及子目錄全部存入。

### TortoiseCVS 操作

1. 右鍵 CVS Commit



2. 輸入 comment 做爲該檔更新的 log(可做可不做)
3. ok 確認



## 八、 刪除檔案

1. rm 檔案

2.

```
cvsc remove file_name
```

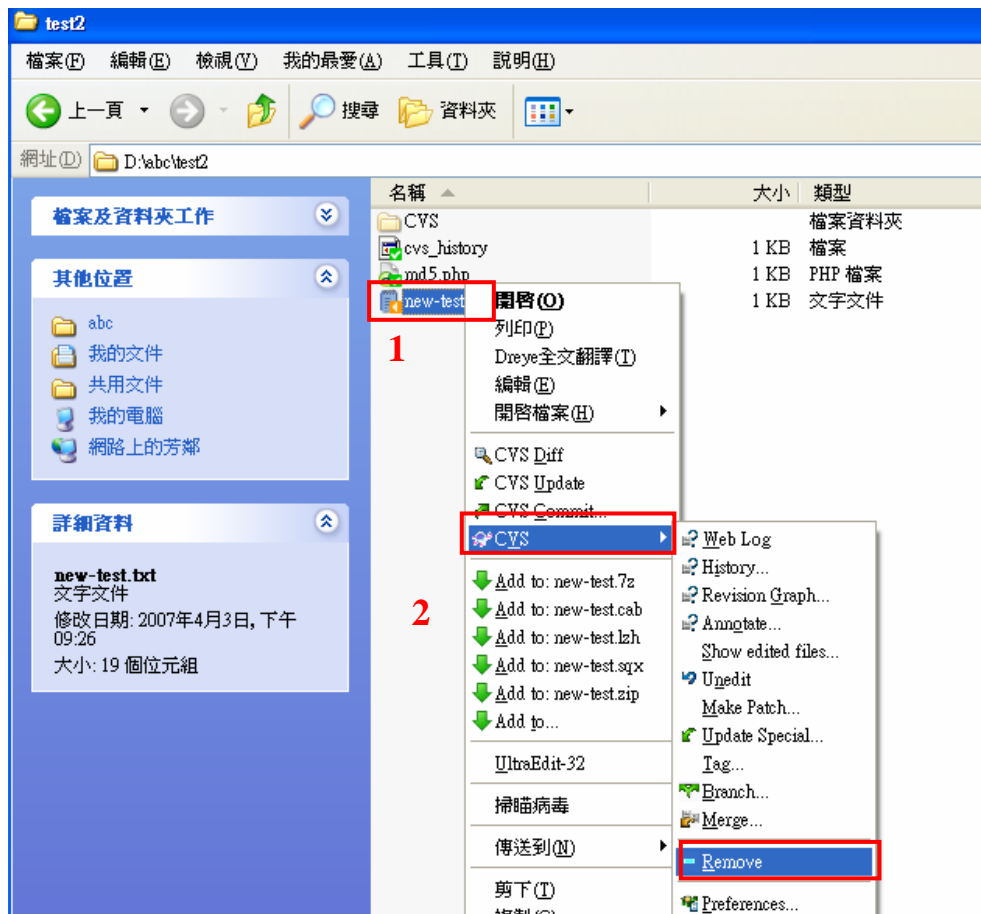
3.

```
cvsc commit -m "remove file" file_name
```

## TortoiseCVS 操作

1. 點選要刪除檔案,

2. 右鍵 CVS-> Remove



## 九、 刪除目錄

1. cd 目錄

2. rm 目錄下所有檔案(ex : rm file1 file2 file3 or rm \*)

3.

```
cvsc remove 目錄下所有檔案 //ex : cvsc remove file1 file2 file3
```

4.

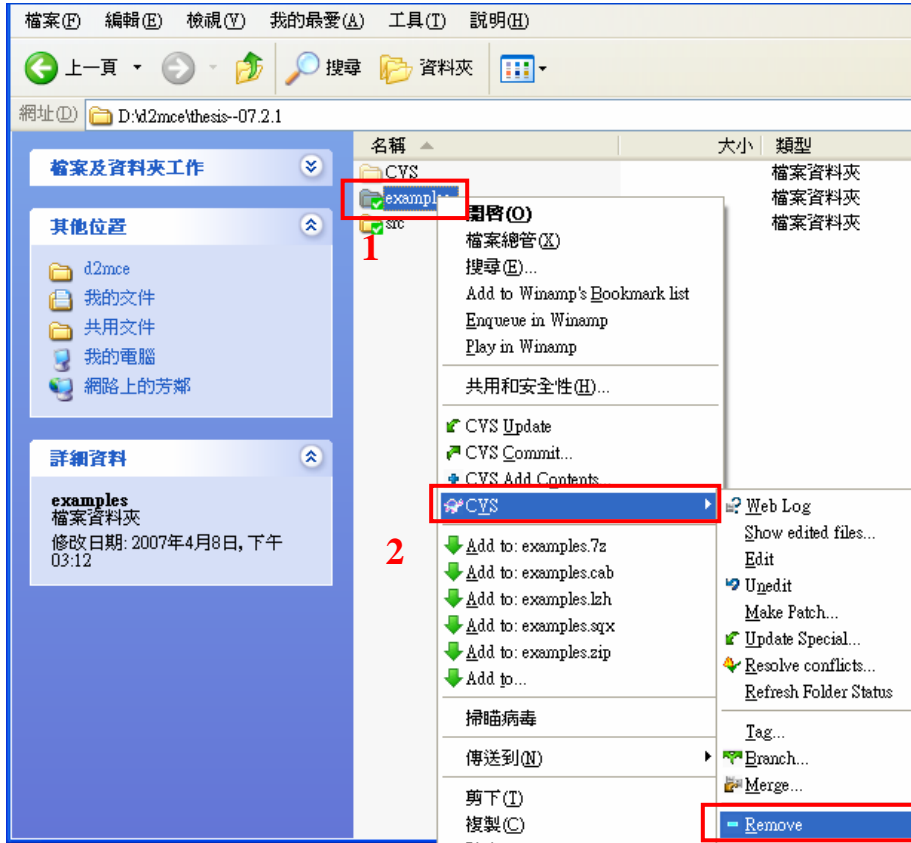
```
cvsc commit -m "remove directory" 目錄下所有檔案 //ex : cvsc commit file1 file2 file3
```

5.

`cvsexample -P // -P` 會將空目錄由 cvs 資料庫中刪除

## TortoiseCVS 操作

1. 點選要刪除目錄
2. 右鍵 CVS-> Remove



## 十、 更改檔名

1. `mv 舊檔名 新檔名`
- 2.

`cvsexample remove old_name`

3.

`cvsexample add new_name`

4.

`cvsexample commit -m "old name change to new name" old_name new_name`

## 十一、 更改目錄名

1. `mkdir new_directory //即要改的新名稱`
- 2.

```
cv$ add new_directory
```

3. mv old\_directory/\* new\_directory //將所有舊目錄下檔案移到新目錄

4.

```
cd old_directory ; cv$ remove files_name //在舊目錄下 cv$ remove 所有檔案
```

5.

```
cd new_directory ; cv$ add files_name //在新目錄下 cv$ add 所有檔案
```

6.

```
cd.. ; cv$ commit -m "old directory change to new directory" //回到上一層目錄 commit
```

7.

```
cv$ update -P //更新
```

## 十二、依時間點 取出/更新 過去的 module

依時間點取出 module

```
cv$ -q checkout -D "2007-04-09 00:00:00 GMT" module_name
```

-q 減少一些訊息

-D 依時間點 格式為”yyyy-mm-dd hh-mm-ss GMT”

依時間點更新 module

```
cv$ -q update -D "2007-04-09 00:00:00 GMT" module_name
```

注意：因為取出舊版本，會使得目前目錄下的版本為舊的，任何修改將不能直接存入 cvs，因為 cvs 不允許修改過去歷史。只能以分支(branch)的方式，在新分支修改。

取消依時間點更新

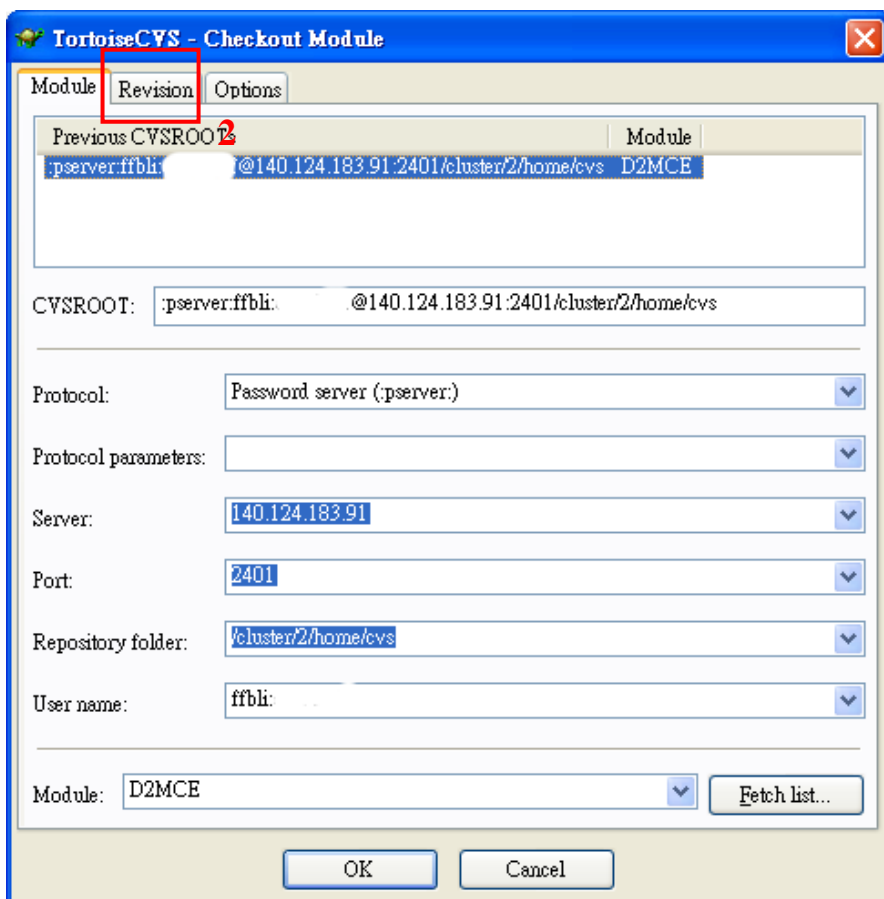
```
cv$ -q update -A
```

## TortoiseCVS 操作

1. 右鍵點選 checkout



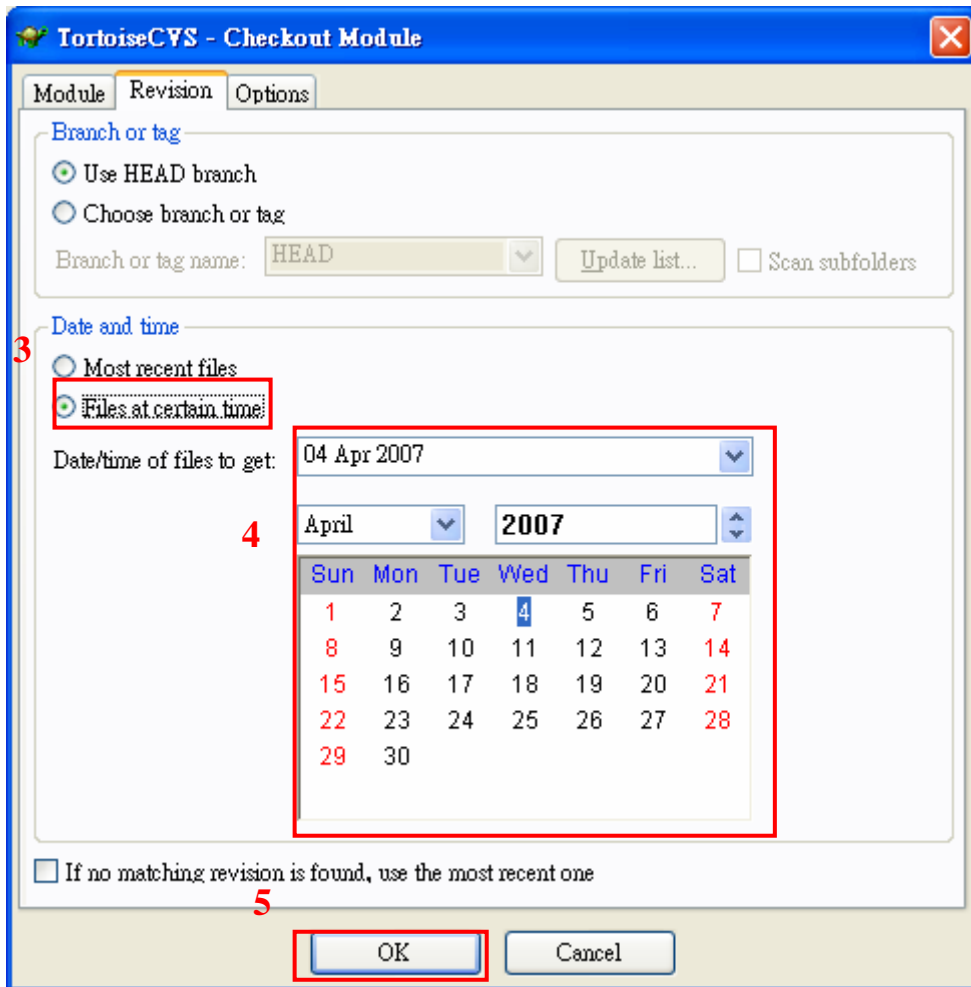
## 2. 點選 Revision



## 3. 選取 Date and time 內的 files at certain time

## 4. 選取時間

## 5. 確認 ok 完成



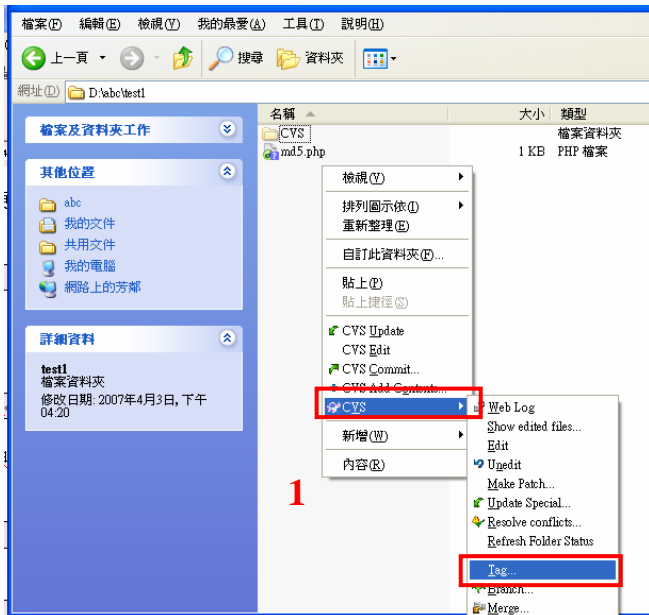
### 十三、依標籤(tag) 取出/更新 module

標記 tag

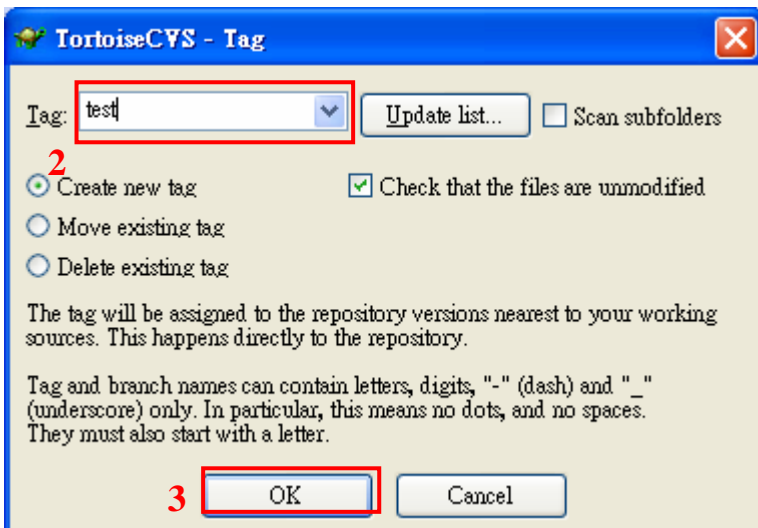
```
cvsv -q tag tag_name
```

#### TortoiseCVS 操作

1. 右鍵點選 CVS->tag



2. 輸入 tag name
3. 確認 ok 完成



依 tag 取出 module

```
cvsv -q checkout -r tag_name module_name
```

-q 減少一些訊息

-r 依 tag\_name or branch name 取出

依 tag 更新 module

```
cvsv -q update -r tag_name
```

取消依 tag 更新

```
cvsv -q update -A
```

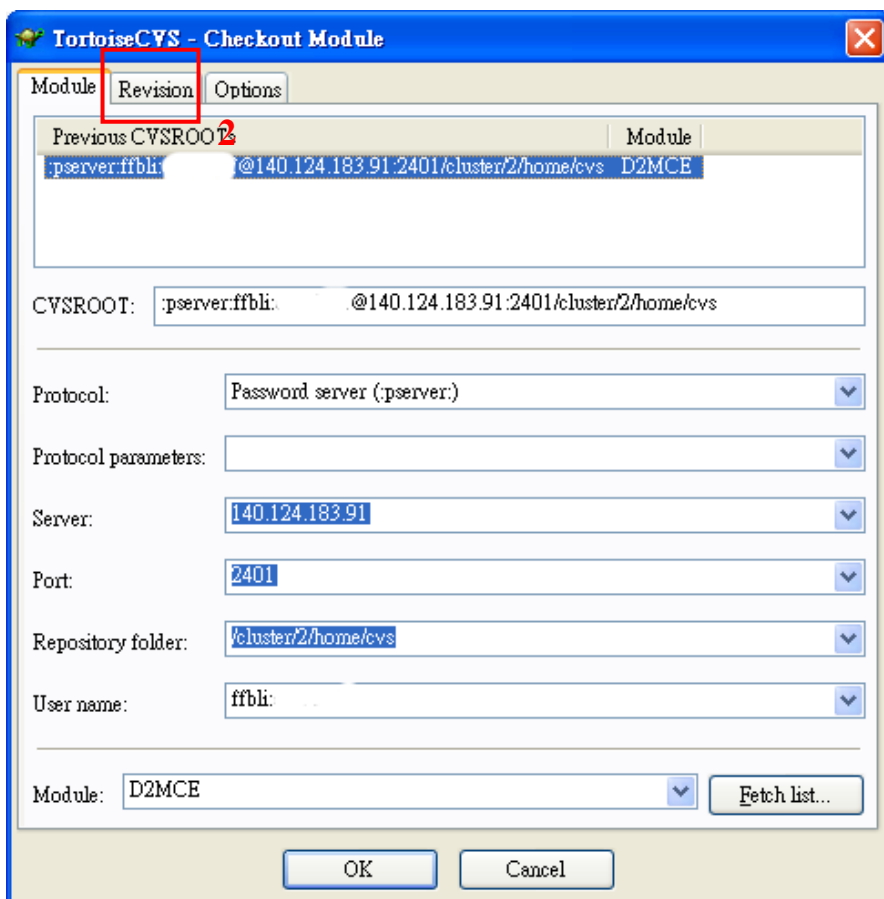
**TortoiseCVS 操作**

1. 右鍵點選 checkout





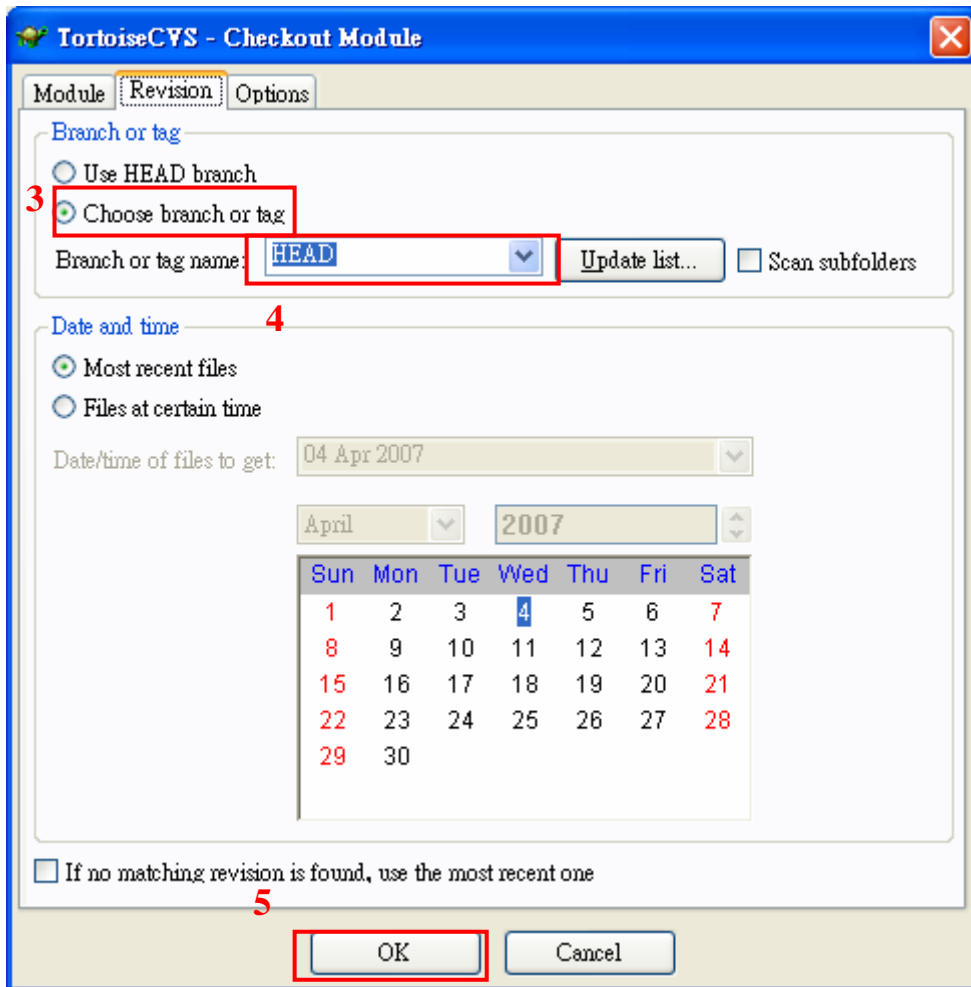
## 2. 點選 Revision



## 3. 選取 Branch or tag 內的 choose branch or tag

## 4. 輸入 tag name

## 5. 確認 ok 完成



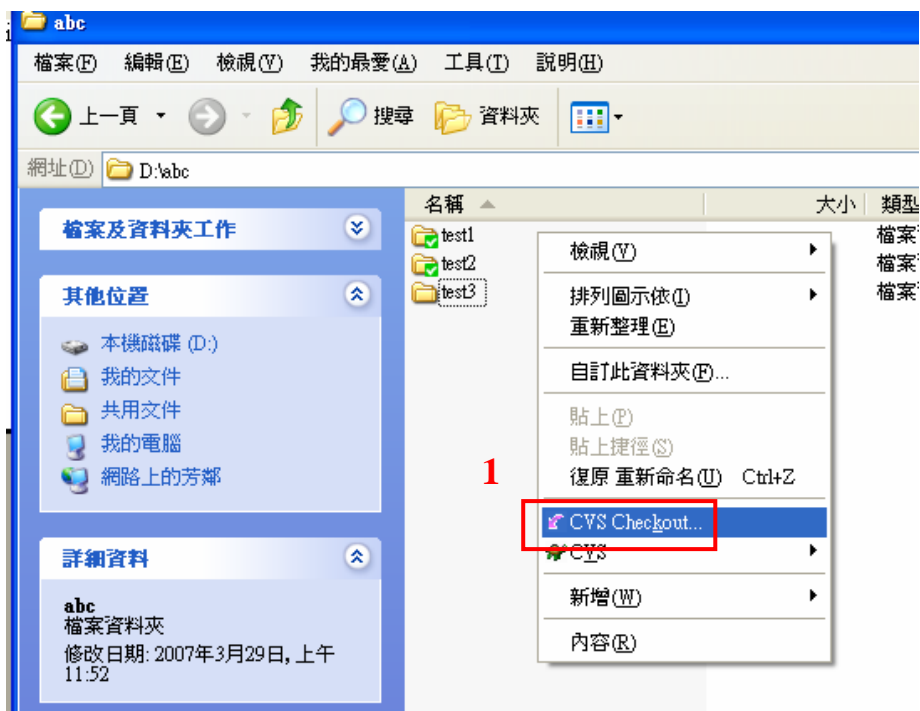
#### 十四、取出 release 版本的 module

```
cvsexport -r 標記 -d 取名存放的目錄名稱 module 名稱
```

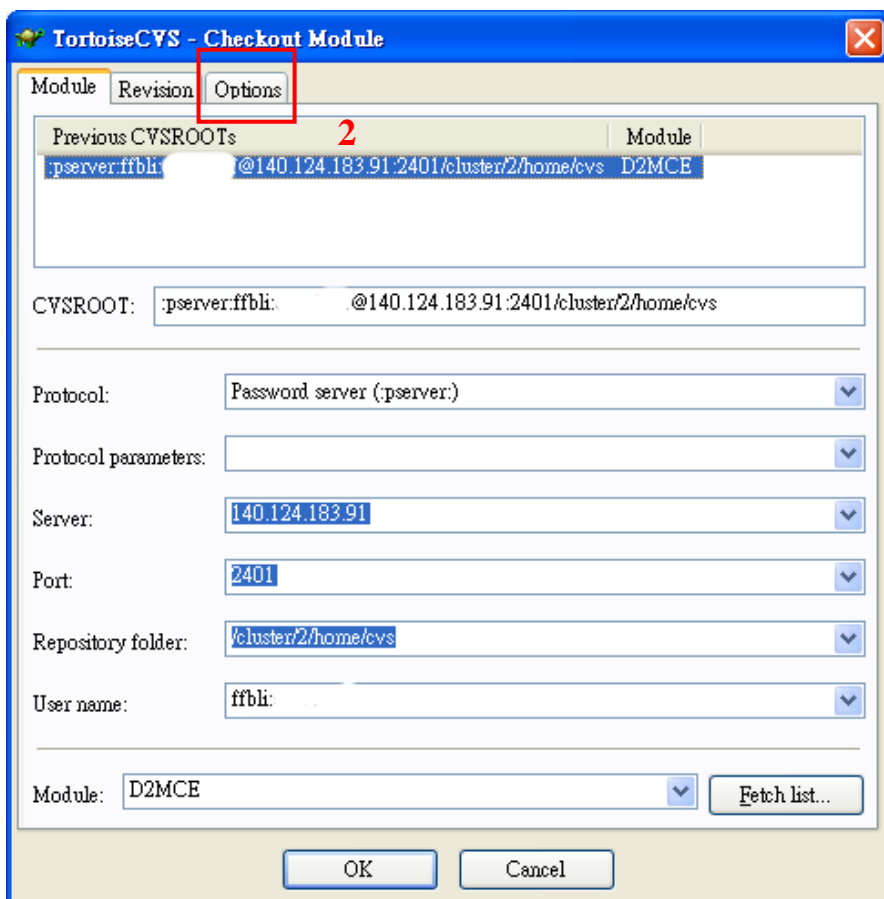
可以取出不包含 CVS 資訊目錄的 module

#### TortoiseCVS 操作

1. 右鍵點選 checkout

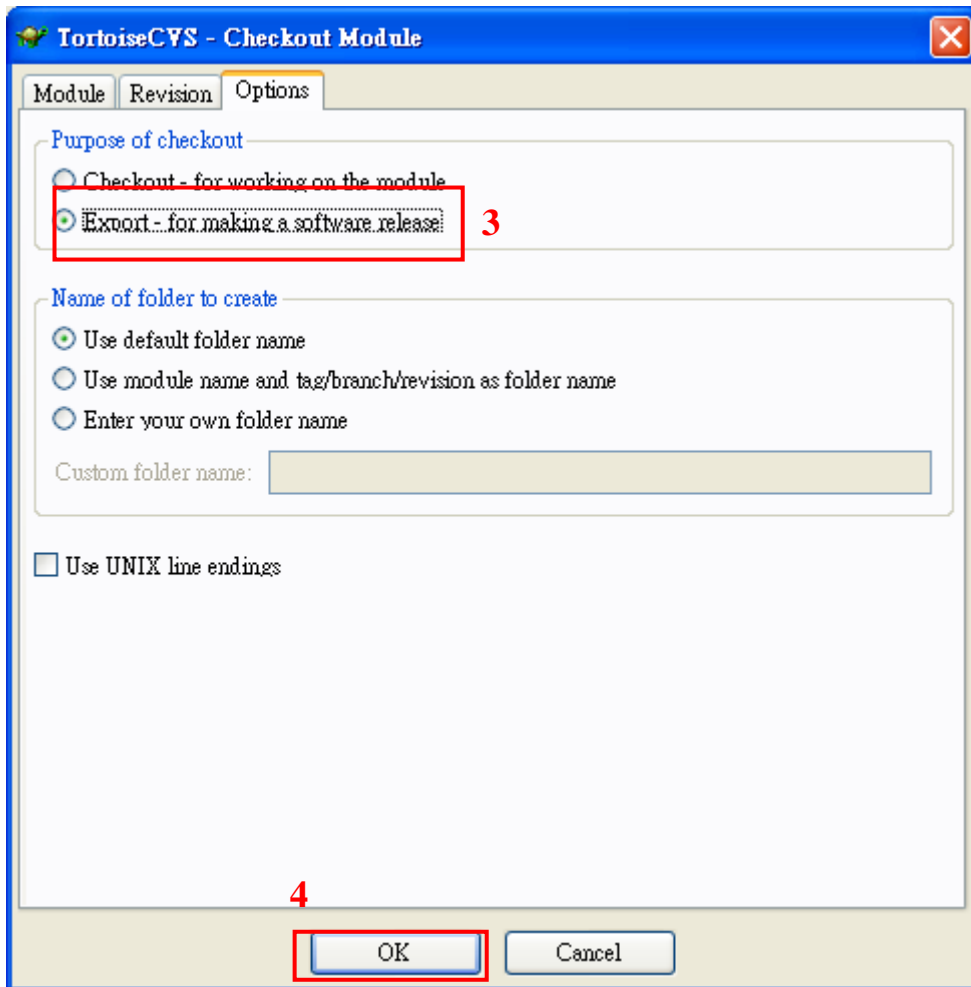


## 2. 點選 Options



## 3. 選取 Export

## 4. 確認 ok 完成



## 十五、分支(branch)

1. 依 tag 取出過去一個 module 版本

```
cvsexport checkout -d new_branch_module -r branch_tag
```

-d 會產生一個目錄 new\_branch\_module

2. cd new\_branch\_module

3.

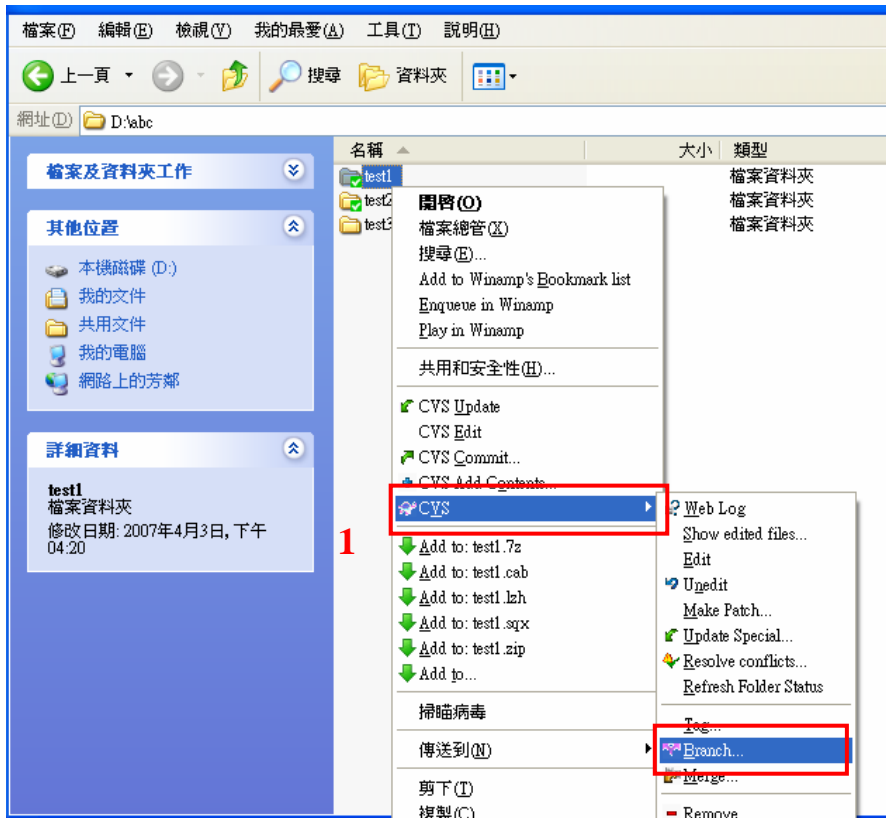
```
cvsexport -q tag -b newbranch
```

4. 更新目前工作版本(因為 cvs tag 只有更新 cvs 端，而沒有改變目前工作版本)

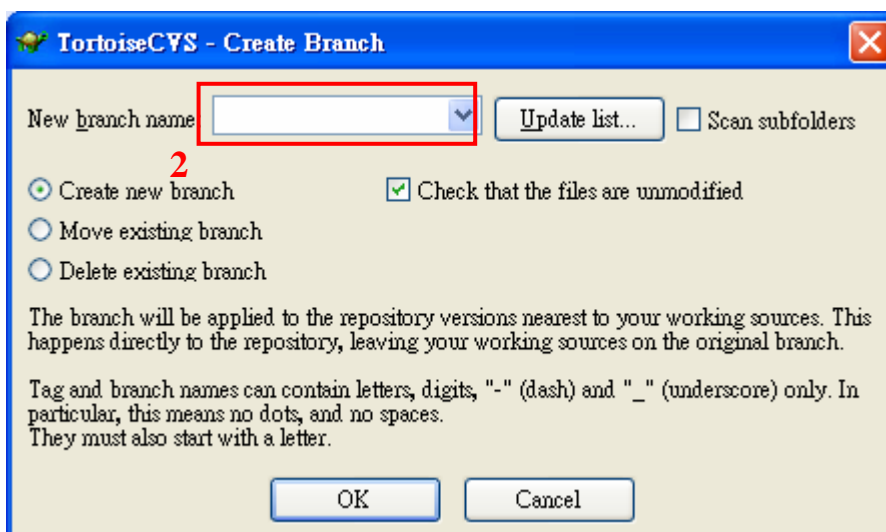
```
cvsexport -q update -r newbranch
```

### TortoiseCVS 操作

1. 右鍵點選 checkout



2. 輸入 tag name
3. 確認 ok 完成



## CVS 指令快速查表

指令	作用
cvcs login	登入 CVS
cvcs import	放個 module 進 CVS 管理
cvcs checkout (co)	拿出一個 module
cvcs add	增加檔案
cvcs remove	移除檔案
cvcs commit (ci)	確認改變，每次都要輸入改變的資訊，叫 log message。(-m “message”)
cvcs update (up)	別人有新的送交行為時，更新自己的版本，同步櫃子的版本。
cvcs status (st)	看目前 repository 裡面有那些檔，檔案版本等等資訊 -l 只看 local
cvcs diff	看兩個版本的差異
cvcs log	看一下 log 歷史紀錄
cvcs logout	登出 CVS