

1. CVS(Concurrent Versions System)란 무엇인가?

평소 버전 관리나 리소스 관리 등에 관심을 많이 갖고 있었다면 CVS나 Visual Source Safe 에 대해 이미 알고 있거나 많이 들어보았을 것이다. 여러분이 어느 정도 짐작하고 있듯 CVS는 소스의 버전을 관리해주는 시스템이다. 프로젝트를 진행하다 보면 각종 파일들을 수정하게 되는데 수정되는 과정을 버전으로 관리하여 보 관할 수 있다. 주로 프로그램 소스의 관리를 주 목적으로 하지만 그 외의 이미지나 사운드 파일까지도 가능하다.

현재 CVS는 운영체제 별로 여러 종류가 나와 있다. 사실 기존의 CVS는 커맨드 라인 기반의 프로그램이었으나 지금은 누구나 쉽게 사용 가능한 프로그램들이 많이 나와있다. 여기서는 Windows에서 쉽게 사용 가능한 WinCvs 와 totoise CVS를 간 단히 다루도록 하겠다.

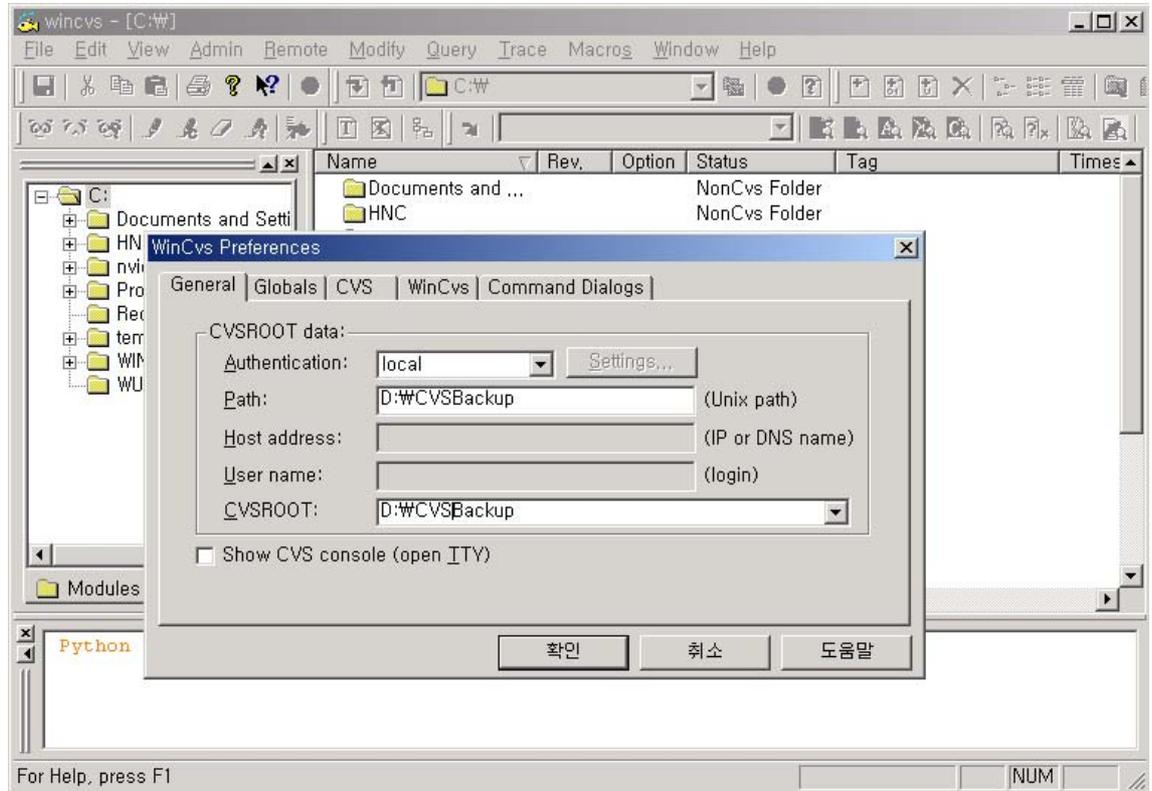
2. 기본 용어 해설

Repository	등록한 리소스 등을 저장하는 저장소를 말한다.
Checkout	보통 저장소로부터 전체 모듈을 받는 작업을 말함.
Commit	저장소로 수정사항을 전송하는 것.
Export	Refers to extraction of an entire module from the repository, without any CVS administrative files: export 된 모듈은 cv s 제어를 받지 않는다.
Import	전체 디렉토리 구조를 전송함으로써 저장소에 새로운 모듈을 생성하는 것.
Module	디렉토리 계층. 프로젝트는 일반적으로 저장소내에 단 하나의 모듈로서 존재하는 것이 보통이다.
Release	전체 프로젝트의 버전.
Revision	한개의 화일에 대한 버전.
Tag	개발상의 특정 시점을 나타내는 화일들의 집합에 주어지는 문자열(명칭).
Update	저장소로부터 다른 개발자들의 수정사항을 얻는 것. 단지 개발자 개인 PC 로의 지역복사만을 갱신한다.

3. CVS의 설치

먼저 WinCvs를 설치 해 보도록 하자. WinCvs는 www.wincvs.org 에서 구할 수 있다. 기본적인 사용은 totoise CVS를 이용하여 사용할테지만 totoise CVS는 저장 소(Repository)를 만들어주는 기능이 없다.

우선 WinCvs를 리소스 들을 저장해 둘 컴퓨터(서버)를 하나 정해서 설치한다.
 일단 기본 설치는 실행 파일을 실행한 후 Next만 누르면 쉽게 설치가 가능하다.

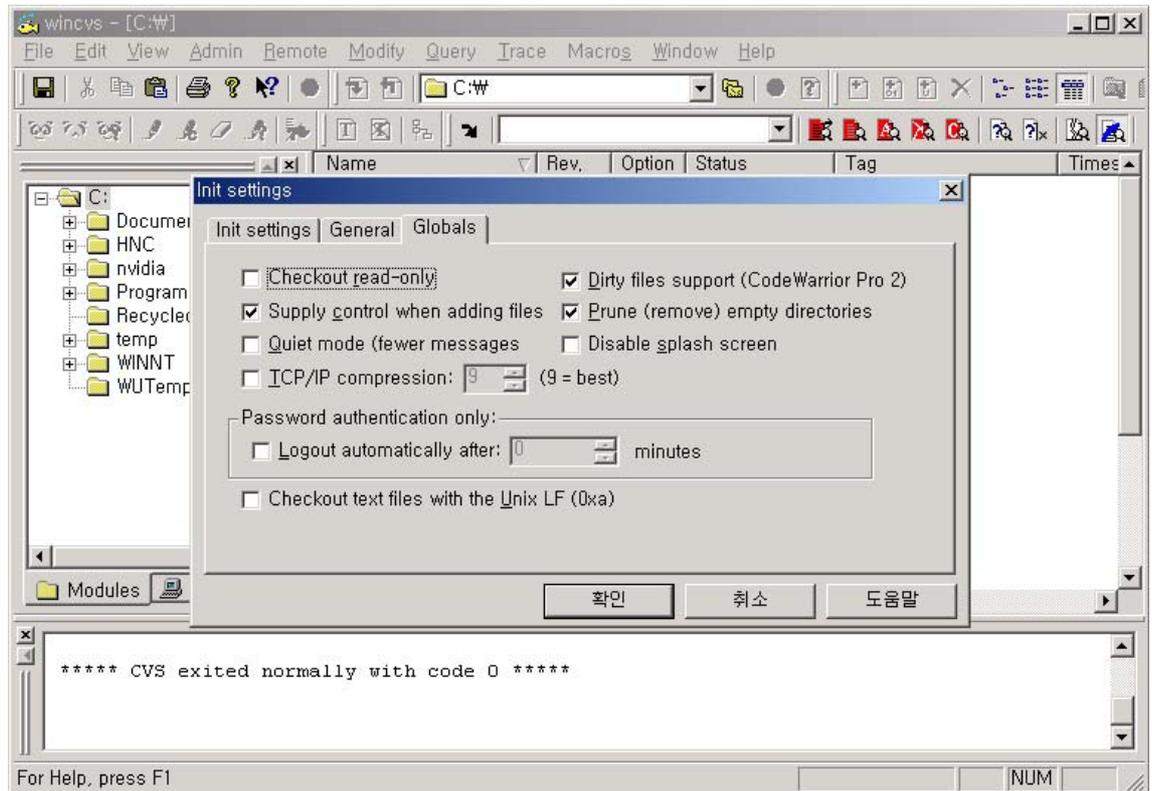


< 초기 실행 화면 >

프로그램을 설치 후 실행하면 바로 위 화면이 나타난다. 먼저 CVS 관련 데이터를 저장할 곳을 지정해야 한다. CVSROOT에 위치를 지정한다. 폴더는 자동으로 생기지 않으므로 미리 생성해둬야만 한다. Authentication 부분은 내부 네트워크를 사용할 것이기 때문에 local로 지정해둔다. 다른 탭의 옵션은 일단 신경 쓰지 않아도 된다. 확인 버튼을 누른다.

다음으로는 저장소(Repository)를 생성해야 한다. 메뉴에서 Remote->Create a new repository를 선택 한다. General 부분은 처음에 지정한 곳으로 되어 있을 것이다. 다른 곳으로 지정할 수도 있다. 단 미리 생성되어 있는 폴더를 지정해야 한다. Global 탭에 가면 여러 옵션이 있다. Checkout read-only 부분에만 언체크 하고 나머지는 그대로 두고 확인 버튼을 누른다. 그러면 지정한 폴더 안에 CVSROOT라는 폴더가 생긴다. 이 폴더는 절대로 삭제해서는 안된다. 생성된 저장소 폴더는 다른 내부 네트워크에서 액세스가 가능하도록 공유 설정을 해준다. CVSROOT 폴더가 아니라 CVSROOT 폴더가 있는 상위폴더를 지정해야 한다.

이것으로 WinCvs의 역할은 끝났다. 허무할지도 모르겠지만 WinCvs로는 저장소 생성 작업만 하였다. WinCvs만으로도 모든 작업이 가능하지만 훨씬 편리한 기능의 totoise CVS가 있기 때문에 일단 WinCvs는 여기까지로 설명하고 마무리 하기로 한다.

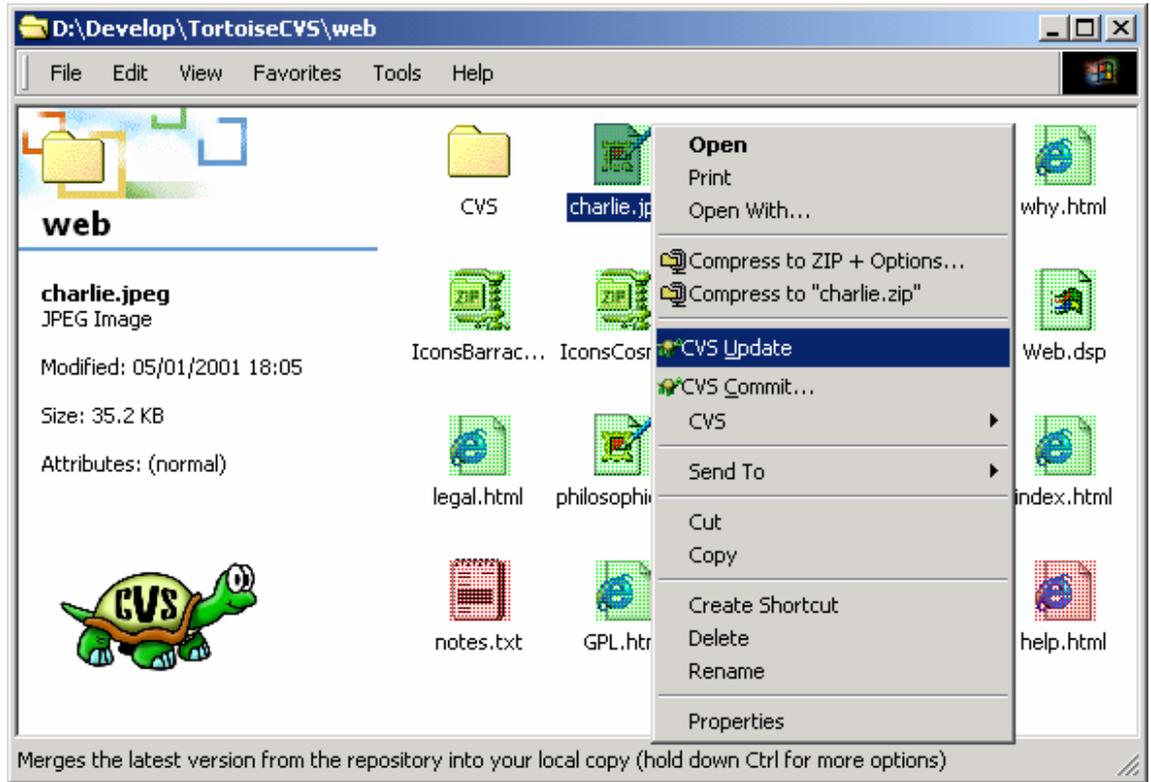


< Create a new repository >

4. CVS의 활용

다음은 totoise CVS를 설치하도록 하겠다. WinCvs는 저장소를 생성할 컴퓨터에만 설치하고 totoise CVS는 CVS를 사용할 모든 클라이언트에 설치하도록 한다. WinCvs는 설치하지 않아도 된다.

Totoise Cvs는 <http://www.tortoise cvs.org/> 에서 구할 수 있다.



< totoise Cvs를 이용한 화면 >

Totoise Cvs는 탐색기 상에서 CVS의 기능을 편리하게 사용 할 수 있는 툴이다. 탐색기나 폴더에서 바로 마우스 오른쪽 버튼만으로 CVS의 기본적인 기능이 사용 가능하며 저장소에 저장된 버전과 비교하여 차이를 색으로 표시하여 한눈에 알아볼 수 있도록 한다.

	<i>Unmodified</i>	The file or folder is up-to-date with the CVS repository version.
	<i>Modified</i>	The file or folder has been modified from the current CVS repository version.
	<i>Conflict</i>	The file or folder has a conflict with the current CVS repository version.
	<i>Not In CVS</i>	The file or folder is not in the CVS repository.
	<i>CVS Watch Icon Overlay</i>	The file or folder is controlled under CVS Watch.
	<i>Ignore Icon Overlay</i>	The file or folder being ignored by the CVS repository.

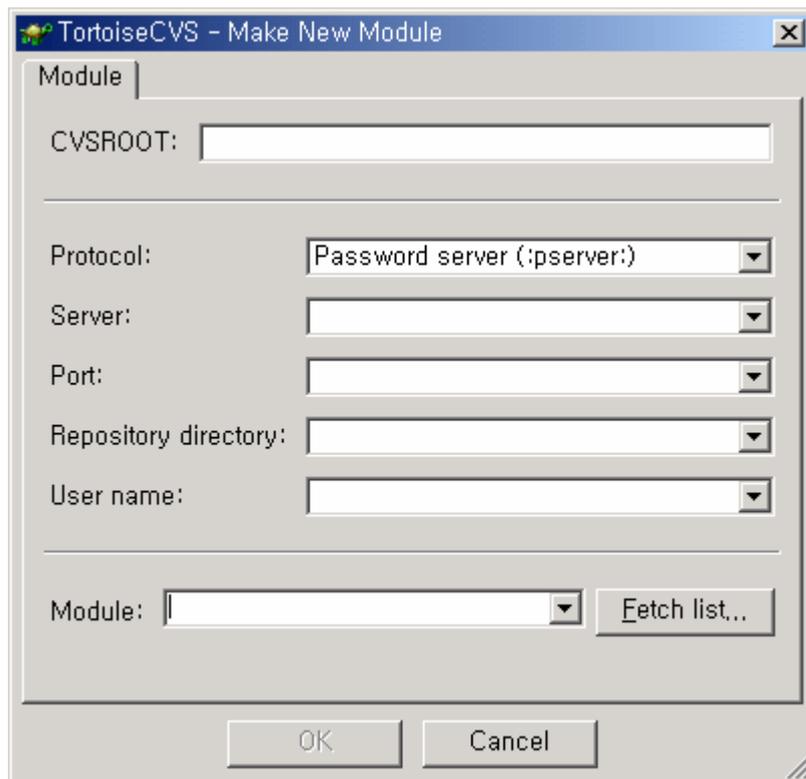
연두색 : CVS에 저장된 최신 버전의 파일을 의미한다.

빨간색 : CVS에 저장된 버전의 파일 내용에서 내용이 변경된 파일을 의미한다.

물음표 : 저장소에 등록된 파일이 아님을 의미한다.

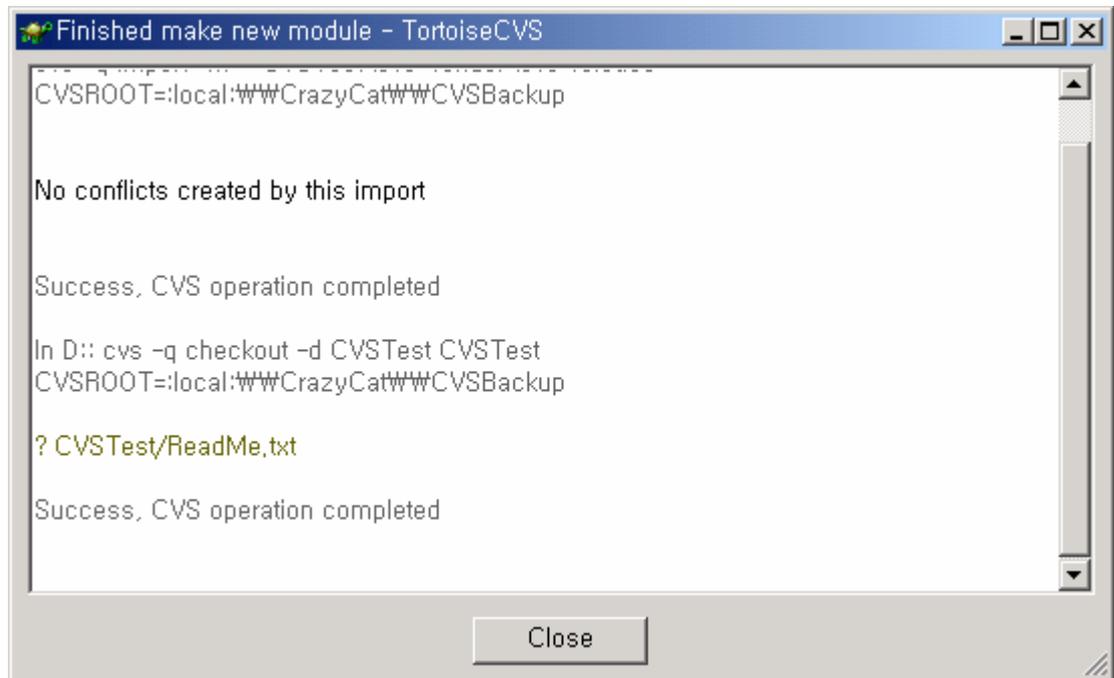
흰색 : 저장소에 저장하지 않도록 지정해놓은 파일을 의미한다

설치는 역시 간단하다. Next버튼만 누른 후 재부팅이 필요하다. 재부팅을 꼭 하도록 한다



재부팅이 끝나고 나면 이제부터 totoise Cvs의 사용이 가능하다. 먼저 탐색기를 열고 CVS 저장소에 등록하고 싶은 폴더를 선택 후(폴더 안에는 아무 파일이라도 들어있어야 한다) 마우스 오른쪽 버튼을 누른다. 그러면 Cvs Checkout 메뉴와 CVS 메뉴가 있을 것이다. CVS 메뉴로 들어가 Make New Module.. 이라는 메뉴를 선택하면 다음과 같은 메뉴가 나타날 것이다. 먼저 Protocol 을 Locally mounted directory (:local:)로 선택한다. 그러면 Server 부분과 Port부분은 비활성화가 될 것이다. 이제 CVSROOT를 지정한다. 아까 저장소를 생성한 폴더의 주소를 적어줘야 한다. :local:WWCrazyCatWWCVSBackUp 이런 식이 될 것이다. 저장소는 네트워크 공유가 되어 있어야 한다. 마지막으로 등록할 모듈 이름을 지정해줘야 한다. 기본적으로는 폴더 이름으로 되어 있다. 모든 것을 완료 후 확인을 누른다.

그럼 다음과 같은 화면이 뜰 것이다. Success, CVS operation completed 란 메시지가 나타났는지 꼭 확인해보도록 하자. Success 가 뜨지 않으면 무언가 이상이 있는 것이므로 다시 한번 설정과 저장소의 공유여부 등을 확인 해 보자.



이제 등록된 폴더로 들어가보자. CVS라는 폴더가 생겼을 것이다. 이 곳에 각종 CVS관련 정보가 저장되는 듯 하다. 등록된 폴더 안에 파일들의 아이콘 가운데에 ? 표시가 되어 있을 것이다. 아직 폴더만 등록된 상태이고 내부 파일들은 아직 등록이 되지 않은 상태이다. 이제 파일들을 등록해보자. 등록된 폴더에 마우스 오른쪽 버튼으로 메뉴를 부르면 Add Contents 라는 메뉴가 있을 것이다. 그것을 실행한다. 폴더 안에 파일들 리스트가 나타나고 체크가 되어 있다. 등록을 원하지 않는 파일들은 언체크 해준다.(미리 원하지 않는 파일들은 삭제해두거나 옮겨놓는 것도 좋다. Debug 폴더 등.)

OK를 누르면 파일들이 등록된다. 이제 마지막으로 Commit 을 할 차례이다. ?로 나오던 파일들이 빨강게 나타날 것이다. Commit 을 통해 지금의 리소스들을 최신의 것이라는 것으로 등록해 준다. Commit을 할 때 수정되는 내용을 같이 적어 둘 수도 있다. Commit 은 자신이 가지고 있는 리소스를 저장소에 최신의 버전으로 저장해준다. Success 메시지가 뜨고 나면 이제 연두색의 파일로 변해 있을 것이다. 이제 저장소에 등록된 리소스들은 자신을 비롯해서 어느 클라이언트에서든 Check Out 할 수 있다. 마우스 오른쪽 버튼을 눌렀을 때 나오는 Check Out 메뉴를 통해 등록된 모듈을 통째로 받아 올 수 있다. CVSROOT 나 기타 설정은 전의 설정을 참

고하고 모듈 이름만 정확하게 입력하면 된다.

5. 실무에서의 활용 예

- 1) A군과 B군은 같은 게임의 다른 분야를 맡아서 프로그래밍을 하고 있다. 기존에는 둘 중 한명이 다른 한명의 작업물을 가지고 직접 소스를 합쳐야 했지만 CVS를 사용하면 걱정이 없다. 서로 최신의 작업을 check out하고 수정된 사항을 commit 하면 끝이다. 다만 주의할 점은 작업에 들어가기전에 저장소에서 update를 받고 작업 후에는 문제가 발생하지 않는지 세세히 테스트 한 후 commit 해야 하는 점 뿐이다.
- 2) A군은 프로그램을 짜다가 무언가 문제가 발생했음을 깨달았다. 다시 원활하게 돌아가던 시절로 복구 하고 싶다. 그럴 때에는 totoise cvs에 update special 메뉴를 사용하면 원하던 시기로 복구할 수 있다. 단 commit 은 자주 해주어야 할 것이다.
- 3) A군은 변경된 그래픽 오브젝트를 최신의 것으로 유지하고 싶다. 그러나 언제 무엇이 수정되었는지 알기도 힘들고 알아도 가져오는데 불편하다. 그러나 그래픽 리소스도 CVS로 관리된다면 항상 UPDATE만 하면 된다. 텍스트가 아닌 데이터는 저장하는데 용량이 좀 크겠지만 그렇다고 해도 프로젝트 관리하는데 부담될 정도는 아닐 것이다.

6. 그 외의 참고사항

1) 소규모 팀에서 같은 소스를 가지고 작업할 경우에 충돌을 하는 경우도 생긴다. 자신이 수정한 것을 commit 했으나 제대로 되지 않고 보라색으로 파일이 바뀌는 경우가 있다. 이미 다른 쪽에서 같은 부분을 가지고 작업해서 수정을 했고 commit 을 했을 경우에 이런 현상이 발생한다. 당황하지 말고 보라색으로 바뀐 소스를 확인해보자. 충돌 부분은 이렇게 나타난다.

```
<<<<<<< filename
    현재 개발자가 고친 부분
=====
    저장소로부터 합쳐진 부분
>>>>>>> revision
```

아마 대부분 별로 대단하지 않은 부분일 것이다. 적절히 수정을 가한 후 commit을 하면 완료된다.

2) 저장소로 지정된 폴더를 일주일에 한번 정도라도 CDRW등을 이용하여 백업해두도록 하자. 폴더만 백업해도 totoise CVS등을 통해 나중에 그대로 사용이 가능하다. 불의의 사고에 항상 대비하는 자세를 갖도록 하자. 별로 어려운 작업도 아니다.

7. 끝맺음

본인이 CVS에 관한 정보를 처음 구하고자 했을 때 많은 어려움이 있었다. 대부분이 콘솔 기반의 설명에다가 너무 어려웠기 때문이다. 여기저기 흩어져있는 정보를 모으고 스스로 이것 저것 시도 끝에 얻은 정보를 조금이라도 도움이 되었으면 하는 마음에 이렇게 문서로 만들게 되었다. 비록 기본적인 설치와 기능 설명 그리고 간단한 예를 들었을 뿐이지만 많은 분들이 CVS를 접하고 유용하게 활용하였으면 하는 마음이다. 본인도 아직 모르는 부분이 많지만 위의 설명 중 이해가 되지 않거나 제대로 되지 않으면 wonderv@hanmail.net 으로 문의 메일을 보내주시길 바란다.