

Project	Publishing Version Control System
Team	게임시스템
Document Title	게임 클라이언트, 서버 버전 관리 시스템
Document Class	

Revision	v 0.1	
Author	김동하	
Date	2008/09/04	
Revision History	v 0.1	First Release

Notes :

1 Introduction

기존 개발사와 퍼블리셔간 클라이언트 패치 파일 및 서버, 설정 파일들의 전달은 주로 FTP가 이용되어 왔다. FTP로 전달되는 파일들은 버전 관리가 되지 않기 때문에 커뮤니케이션 실수로 잘못된 파일이 전달되거나 문제 발생 시 어떤 파일을 누가 전달하였고 어디에서 문제가 되었는지 찾기도 어려웠으며 문제를 해결하기 위해 다시 파일을 전달하기까지의 시간이 오래 걸렸다.

이에 기존 FTP로 전달되던 방식을 대신 바이너리에 대한 버전 관리 시스템을 도입하여 클라이언트, 서버, 설정 파일들에 대한 신속한 전달, 버전 관리, 이력 관리뿐만 아니라 프로세스 또한 개선하여 업무 효율성 및 편의성, 정확성을 높이는 것을 퍼블리싱 게임 버전 관리 시스템의 목적으로 한다.

본 문서는 네오위즈 퍼블리싱 게임에서 사용되는 저장소(Repository)의 구조 및 사용 정책에 대해서 설명하며 네오위즈 퍼블리싱 게임 버전 관리 시스템인 서브버전(Subversion)에 대한 기본적인 이해 및 기본적인 툴의 사용법을 다룬다.

서브버전은 본래 프로그램 개발 시 소스 버전 관리 용도로 여러 매력적인 기능들(revision system, atomic commit, locking, branching, tagging, merging 등)을 지원하여 소프트웨어 개발에 널리 쓰이나 바이너리를 관리함에도 많은 장점이 있기에 네오위즈 게임 버전 관리 시스템으로 선택되었다.

Subversion에 대한 좀 더 자세한 이해를 원한다면 아래 문서를 먼저 읽어 보길 권장한다.

<http://svnbook.red-bean.com> (영문)

그리고 효율적인 버전 관리 시스템 사용을 위한 Windows 기반 Subversion Client인 Tortoisetsvn 을 사용해 보지 않았다면 설치해 보고 아래 문서도 꼭 읽어 보길 바란다. 물론 다른 Subversion Client들도 사용 가능하다.

<http://tortoisetsvn.net/downloads> (설치 파일, 한글 언어 팩 다운로드)

http://tortoisetsvn.net/docs/release/TortoiseSVN_en/index.html (영문)

2 클라이언트, 서버 전달, 개발 환경 QA 프로세스 변경 사항

2.1 기존 프로세스

- 1) 클라이언트, 서버 변경 파일 FTP로 전달, 테스트 환경 서버 준비 (개발사)
 - 2) 패치 생성 (시스템팀)
 - 3) 개발 QA 진행 (QA팀)
 - 4) 버그가 있을 경우 개발사 버그 수정 파일 전달 (개발사)
 - 5) QA 완료 (QA팀)
 - 6) 점검 시 클라이언트 서버 패치 적용 (시스템팀)
 - 7) 실서비스 전 환경 (38-1) 테스트
 - 8) 오픈
- ⇒ 업데이트 전달 시 커뮤니케이션이 복잡하고 개발사에서 변경 파일을 찾아 압축해 FTP에 업로드 해서 전달. 개발 환경에 변경 사항이 있을 때 마다 패치를 새로 만들어야 해서 개발 환경에 패치 적용 시간이 오래 걸리고 매우 번거로움

2.2 변경 후 프로세스

- 1) 클라이언트, 서버 변경 사항 커밋, 테스트 환경 서버 준비 (개발사)
 - 2) 커밋 로그 및 변경 사항이 QA팀에 메일로 자동 통보됨
 - 3) 패치 생성 필요 없이 SVN 체크아웃, 업데이트로 개발 환경 QA 진행 (QA팀)
 - 4) 버그가 있을 경우 수정 버전을 커밋 (개발사)
 - 5) QA 완료, 최종 리버전으로 패치 파일 생성 (QA팀)
 - 6) 점검 시 클라이언트 서버 패치 적용 (시스템팀)
 - 7) 실서비스 전 환경 (38-1) 테스트 (패치 테스트 포함)
 - 8) 오픈
- ⇒ 업데이트 전달 시 커뮤니케이션이 줄고 개발사에서는 ftp 파일 업로드 없이 클라이언트, 서버 변경 파일만 저장소에 커밋 하면 됨. 개발 환경에 패치를 생성할 필요 없으며 svn checkout, update 만으로 QA 준비 가능

2.3 개발사에서 해야 할 일

- 2.3.1 게임 클라이언트, 서버 담당자가 각 저장소 url에 변경 사항만 커밋 해 주면 된다.
(클라이언트, 서버 저장소 url은 게임별 저장소 구조를 참고)
- 2.3.2 커밋 로그 메시지가 메일로 QA 팀에 전달되기 때문에 상세한 커밋 로그를 남기는 것이 원활한 QA 진행에 도움이 된다.

3 저장소 정책 (Repository Policy)

- 3.1 게임(프로젝트)는 원칙적으로 하나의 저장소를 사용한다.
- 3.2 단 해외 진출 게임의 경우 국가(퍼블리셔)별로 저장소를 따로 할당한다.
- 3.3 저장소는 네오위즈에 위치하며 저장소 담당자에 의해 유지, 관리, 백업된다.
- 3.4 게임 별 저장소 구조 및 주소는 보안상 대외비이다. 필요한 사람에게만 전달되어야 한다.
- 3.5 허가된 IP Address 만이 저장소에 접근이 가능하다.
- 3.6 허가된 사용자만이 저장소로부터 체크아웃, 업데이트(read) 및 커밋(write)이 가능하다.
- 3.7 저장소에 접근하기 위한 권한 부여는 각 팀장의 승인을 받아 담당자에게 요청되어야 한다.
- 3.8 보안이 필요한 파일이 저장소에 저장되어야 할 경우 해당 파일은 암호화(encryption)되어 있어야 하고 문서의 경우 password로 잠금 설정 되어 있어야 하며 사전에 저장소 담당자에게 알려야 한다. 그렇지 않을 경우 해당 파일을 커밋한 사람에게 파일 보안에 대한 책임이 있을 수 있다.
- 3.9 저장소 디렉터리 구조의 변경이 필요할 경우 저장소 담당자와 협의해서 바꾸어야 한다.
- 3.10 네오위즈의 저장소는 기본적으로 바이너리(Binary)의 저장소이다. 소스 코드(Source Code)의 저장소가 아니다!!
- 3.11 저장소는 문서의 버전 관리 목적으로 사용 될 수 있으나 저장소 디스크 사용량 문제가 있으므로 문서의 종류는 버전 관리가 필요한 최소한의 것으로 제한한다. 문서를 저장소에 올릴 경우 저장소 담당자와 미리 협의하여야 한다.
- 3.12 일회성 파일들은 버전 관리 대상이 되어서는 절대 안 된다. (로그, 덤프 파일 등)
- 3.13 소스 코드가 아닌 바이너리를 저장하므로 소스 코드의 바뀐 부분만 저장되는 것이 아니라 바뀐 파일 전체가 저장된다. 따라서 저장소 사용 디스크 용량이 커질 우려가 있으니 큰 파일 커밋 시에는 신중을 기해야 한다.
- 3.14 커밋 시에는 누구나 변경 사항에 대해 알 수 있도록 상세한 로그 메시지를 적어 주어야만 커밋이 가능하다. 커밋 시 로그 메시지 및 변경 사항은 담당자들에게 메일이 발송되므로 누구나 어떤 내용이 업데이트 되었는지 알 수 있게 된다.
- 3.15 서버버전에서는 특정 리비전의 삭제가 불가능하다. 이는 시간이 경과함에 따라 저장소의 디스크 사용량이 늘어나 문제가 될 수 있음을 의미한다. 저장소의 용량은 최대한 할당해 두었지만 디스크는 유한하고 업데이트는 무한하다. 최악의 경우 용량 문제로 저장소를 새로 할당해야 하는 일이 있을 수 있다.
- 3.16 위 문제로 리비전 번호는 초기화 될 수 있다. 따라서 리비전 번호를 key로 잡는 일이 있으면 안된다.

4 저장소 사용 시 주의 사항

- 4.1 최초 개발사에 제공된 vpn (publvpn)을 연결하면 저장소 접근이 가능하다.
- 4.2 최초 개발사에 제공된 테스트 서버 서버 접속용 공용 Windows 계정에 저장소 read, write 권한이 부여되어 있다. (추후 사용자 별 계정으로 분리 예정)
- 4.3 체크아웃, 업데이트로 받는 클라이언트는 런처의 checksum 검사를 통화 할 수 없기 때문에 개발 환경 QA 시 담당자가 별도의 checksum 해제 작업이 필요하다. 추후 개발 환경에서의 checksum 검사를 우회 방법을 제공할 예정이다.
- 4.4 체크아웃, 업데이트로 받는 클라이언트는 풀클라이언트를 설치한 것과는 달리 설치 위치 등의 registry 정보가 입력되어 있지 않은 관계로 필요한 경우 registry 정보는 따로 설정해 주어야 한다.
- 4.5 커밋 시 변경 사항에 대한 자세한 로그 메시지를 남기지 않으면 커밋 되지 않도록 설정되어 있다. 커밋 로그 메시지를 남기지 않으면 아래와 같은 예러가 발생한다.
`Commit blocked by pre-commit hook (exit code 1) with no output.`
- 4.6 커밋 시 커밋 로그 메시지는 메일로 담당자들에게 전달 되므로 상세한 변경 사항을 커밋 로그 메시지로 남겨 두면 테스트(QA)시 많은 도움이 된다.
- 4.7 hook script 를 통해 커밋 시 변경 사항을 테스트 서버 환경에 적용하고 서버를 재시작 해주는 작업 등이 자동으로 가능하다. (네오위즈 시스템팀과 협의)

5 네오위즈 퍼블리싱 게임 저장소의 구조

게임 (프로젝트) root 저장소 주소: <http://gamesvn.neowiz.com/svn/> 저장소명

클라이언트 저장소 주소: <http://gamesvn.neowiz.com/svn/저장소명/trunk/client/> 설치폴더명

서버 저장소 주소: <http://gamesvn.neowiz.com/svn/저장소명/trunk/server/> 설치폴더명

예: FreeCell (저장소명, 프로젝트 이름)

```
|-branches
|-tags
|   |-patch
|       |-1 # 오픈 시 리비전에 대해 꼬리표(tag)를 달아 둔다
|       |-2 # 꼬리표는 네오위즈 패치 시스템 및 풀클라이언트 패킹 시 사용
|       |-3 # 이때 패치 시스템 버전 번호를 디렉터리 명으로 한다.
|       |-. ...
|-trunk
    |-client          # 클라이언트 개발 담당자가 관리
        |-FreeCell    # 게임 설치되는 폴더와 동일 구조
        |-. ...
    |-server          # 서버 개발 담당자가 관리
        |-게임 서버 type A
        |-게임 서버 type B
        |-. ...
    |-admin           # 시스템 담당자가 관리
        |-scripts     # 실 서비스 distribution(dist)을 위한 스크립트
        |-hosts       # 실 서비스 dist를 위한 호스트 파일
        |-dist_config # 실 서비스 dist를 위한 설정 파일
        |-. ...
    |-docs            # 버전 관리가 필요한 문서들을 저장
        |-서버구조도.vsd
        |-서버설치매뉴얼.doc
        |-서버장애처리매뉴얼.doc
        |-. ...
```

6 용어 정리 (1)

주. 이하 내용은 서브 버전을 사용해 보지 않은 사용자를 위한 내용입니다.

6.1 저장소(Repository): 모든 게임의 클라이언트, 서버 파일 및 설정 파일들은 이 저장소 안에 저장이 된다. 네트워크를 통해서 여러 사람이 접근 할 수 있으며 버전 관리 시스템마다 각각의 저장소 포맷을 가지고 있으나 Neowiz Publishing Repository 는 FSFS Format 을 사용한다.

6.2 체크아웃(Checkout): 저장소에서 클라이언트, 서버를 받아 오는 것. 체크아웃을 한 내용을 보면 클라이언트, 서버 이외의 다른 디렉터리와 파일들이 섞여 있는 것을 볼 수 있는데 이 디렉터리와 파일들은 버전 관리를 위한 파일들이다. 임의로 지우거나 변경하면 저장소와 연결이 되지 않으므로 주의해야 한다.

6.3 커밋(Commit): 체크아웃 한 파일을 수정, 추가, 삭제 등을 한 뒤 저장소에 저장하여 갱신하는 것. 커밋을 하면 CVS 의 경우 수정한 파일의 리비전이 증가하고 Subversion 의 경우 전체 리비전이 1 증가하게 된다. 커밋 시에는 변경 사항에 대한 상세한 Log Message 를 남겨야만 커밋이 가능하다.

6.4 업데이트(Update): 체크아웃을 해서 내용을 가져 왔더라도 다른 사람이 커밋을 하였다면 내용이 달라졌을 것이다. 이럴 경우 업데이트를 하여 저장소에 있는 최신 버전의 내용을 가져온다. 물론 바뀐 부분만 가져온다.

6.5 리비전(Revision): 파일 등을 수정하여 커밋하게 되면 일정한 규칙에 의해 넘버가 증가한다. 저장소에 저장된 각각의 파일 버전이라 할 수 있는데 Subversion 의 경우 파일 별로 리비전이 매겨지지 않고 한번 커밋 한 것(이를 체인지 셋 단위라 한다)으로 전체 리비전이 매겨 진다.

6.6 임포트(Import): 아무것도 들어 있지 않은 저장소에 다른 저장소의 디렉터리, 파일을 넣는 작업.

6.7 익스포트(Export): 체크아웃과는 달리 버전 관리 파일들을 뺀 순수한 내용을 받아 올 때 사용한다. 패치를 만들기 위한 변경 파일을 추출할 때, 풀클라이언트를 패키징하거나 서버, 설정 파일을 실서비스에 배포(distribution, aka. dist) 할 때 주로 사용된다.

7 용어 정리 (2)

7.1 트렁크(Trunk): 단어 자체의 뜻은 본체 부분, 나무줄기, 몸통 등이다. 게임(프로젝트)의 가장 중심이 되는 디렉터리이다. 모든 작업은 트렁크 디렉터리에서 이루어 진다. 네오위즈 퍼블리싱 게임 저장소 구조에서는 트렁크 디렉터리 아래에 클라이언트, 서버 디렉터리가 들어가게 된다.

7.2 태그(Tags): tag 는 꼬리표라는 뜻을 가지고 있다. 이 디렉터리는 오픈 시점의 클라이언트 리버전에 꼬리표(tag)를 붙여두는 공간으로 사용된다. 이는 네오위즈 differential 패치 시스템에서의 패치 파일 추출 하는 용도와 풀 클라이언트로 패킹할 때 사용한다. 네오위즈 패치 시스템에서의 무결한 패치파일 생성을 위해서는 꼬리표를 붙이는 것이 매우 중요하다. 꼬리표를 붙일 때 디렉터리의 이름은 클라이언트의 실제 version 값으로 만드는 것을 원칙으로 한다. 꼬리표는 패치 매니저에서 자동으로 붙일 수 있도록 한다. (지원 예정)

7.3 브랜치(Branches): 나무줄기(trunk)에서 뻗어져 나온 나무 가지를 뜻한다. 메인 프로젝트 이외에 프로젝트가 따로 진행되어야 할 경우 브랜치를 나눠 사용하고 후에 트렁크에 소스 코드를 머징(Merging)하는 것도 가능하다. 하지만 네오위즈 퍼블리싱 저장소는 바이너리 관리 용도로 저장소가 운영되기 때문에 특별히 사용되지는 않는다. 하지만 개발 및 QA 가 두 개 버전이 함께 진행되어야 하는 경우 등에 사용을 고려해 볼만 하다. 단, 사용이 필요할 경우 저장소 담당자와의 협의가 필요하다.

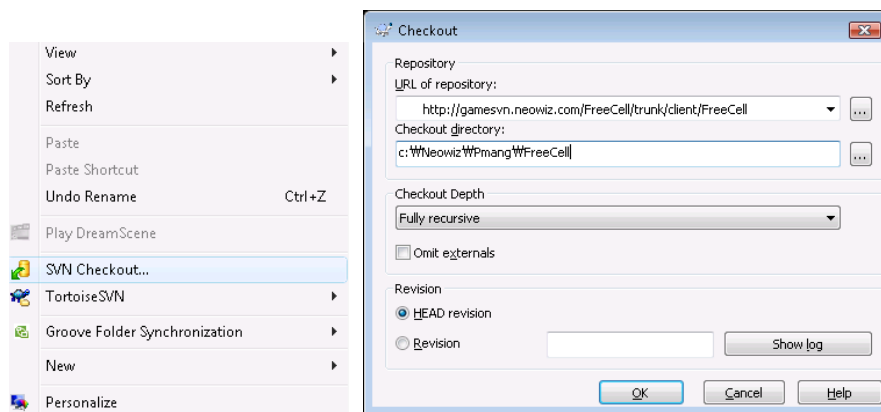
8 최초 체크 아웃 하기

8.1 서버버전을 이용해 작업하려면 저장소의 내용을 로컬 하드디스크의 작업 디렉터리 (working directory)에 다운 받아야 한다.

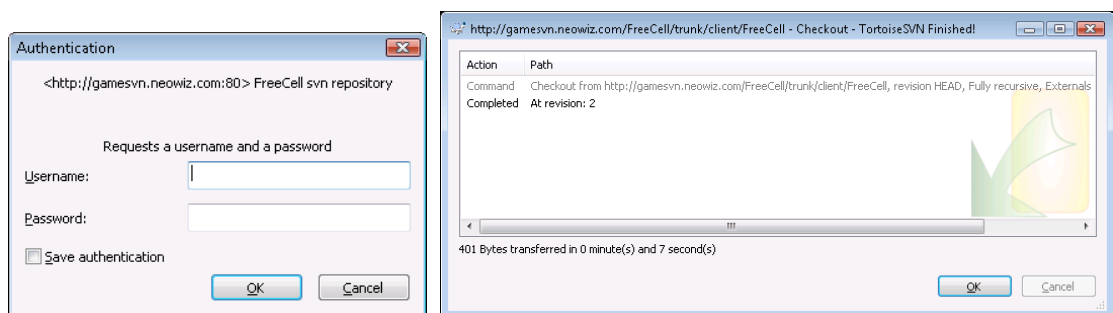
8.2 마우스 오른쪽 버튼을 클릭해 svn 체크아웃을 선택하고 저장소 주소와 작업 디렉터리를 아래와 같이 지정하고 확인을 누르면 된다. 최신 리비전을 받을 지 특정 리비전을 받을지도 선택이 가능하다,

저장소 URL: <http://gamesvn.neowiz.com/svn/게임저장소명/trunk/client/게임폴더명/>
 체크아웃 받을 작업 디렉터리(선택가능): c:\WNeowiz\WPmang\W게임폴더명

아래는 <http://gamesvn.neowiz.com/svn/FreeCell/trunk/client/FreeCell> 에서 로컬 컴퓨터 C:\WNeowiz\WPmang\WFreeCell 에 클라이언트를 체크아웃 하는 예제이다.



<그림 1, 2 체크아웃 명령 및 저장소 URL, 체크아웃 받을 디렉터리 설정>

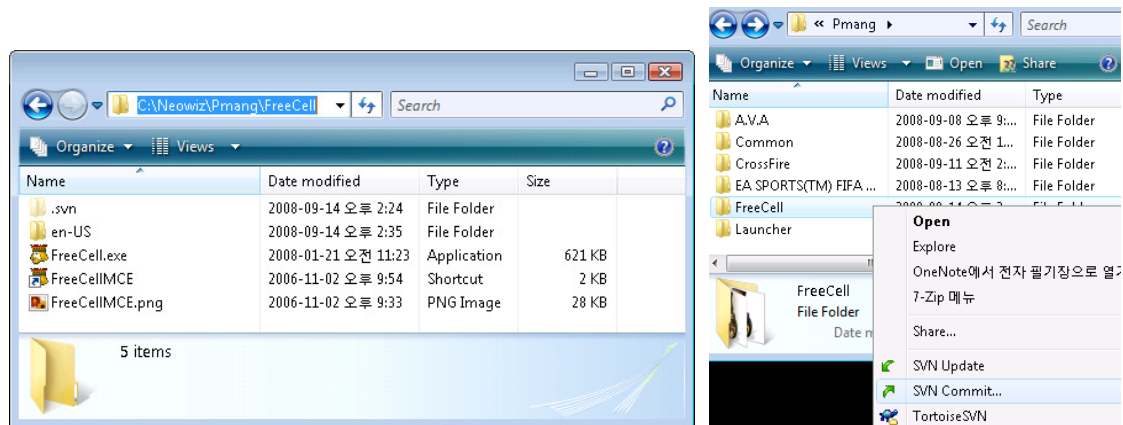


<그림 3, 4 인증 및 체크아웃 완료>

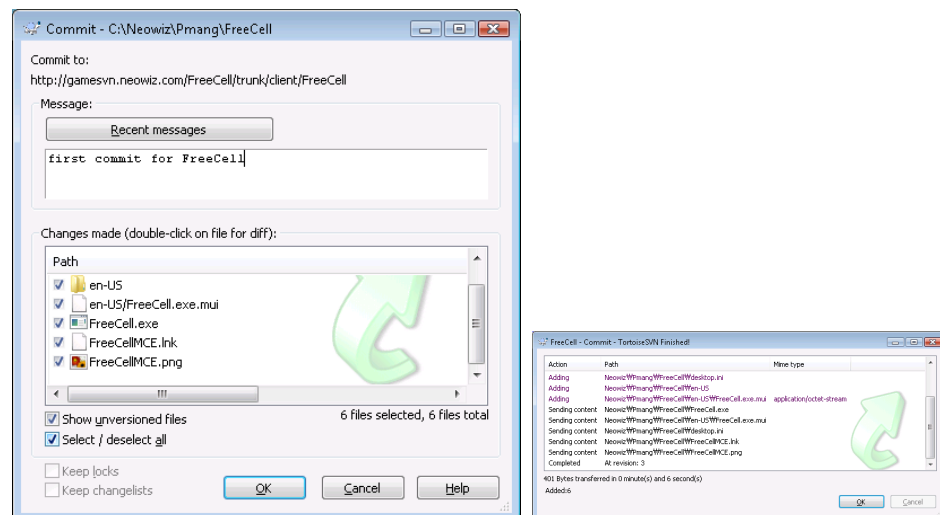
9 최초 파일 커밋 하기

9.1 최신 리비전으로 체크아웃을 완료하면 이제 클라이언트(서버) 파일을 올릴 수 있다.

클라이언트(서버) 설치 폴더 구조를 그대로 작업 디렉터리에 복사하고 상위 폴더에서 마우스 오른쪽 버튼을 눌러 SVN 커밋을 선택하면 작업 폴더로 복사한 파일들(버전 관리 대상 파일들)을 선택 가능하며 선택 후 커밋 로그 메시지를 입력하면 선택된 파일들이 저장소에 저장되고 리비전이 1 증가한다. (이때 커밋 로그를 남기지 않으면 커밋이 되지 않으니 주의할 것!) 이로써 클라이언트(서버) 파일을 최초로 저장소에 커밋 시키는 작업이 완료되었다.



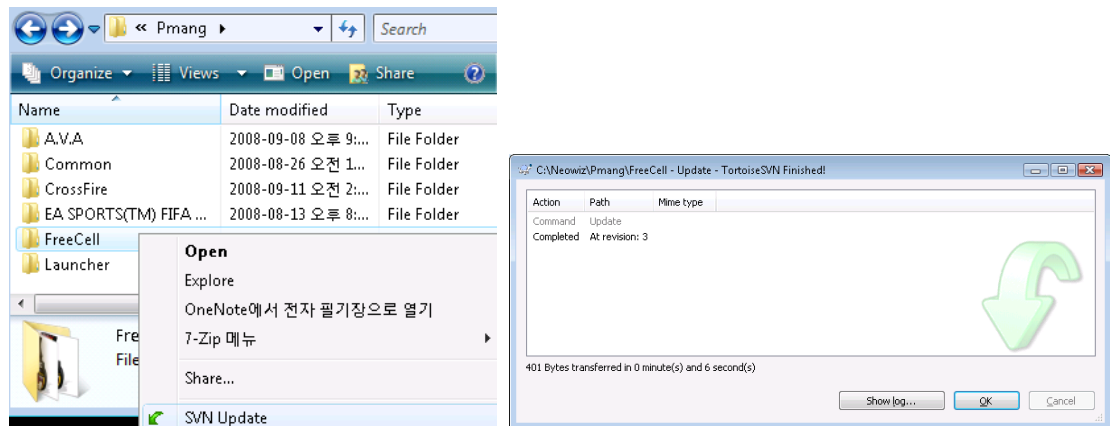
<그림 5, 6 작업 폴더에 파일을 복사, SVN 커밋>



<그림 7, 8 커밋 log message 입력 및 커밋 할 파일 선택, 최초 커밋 완료>

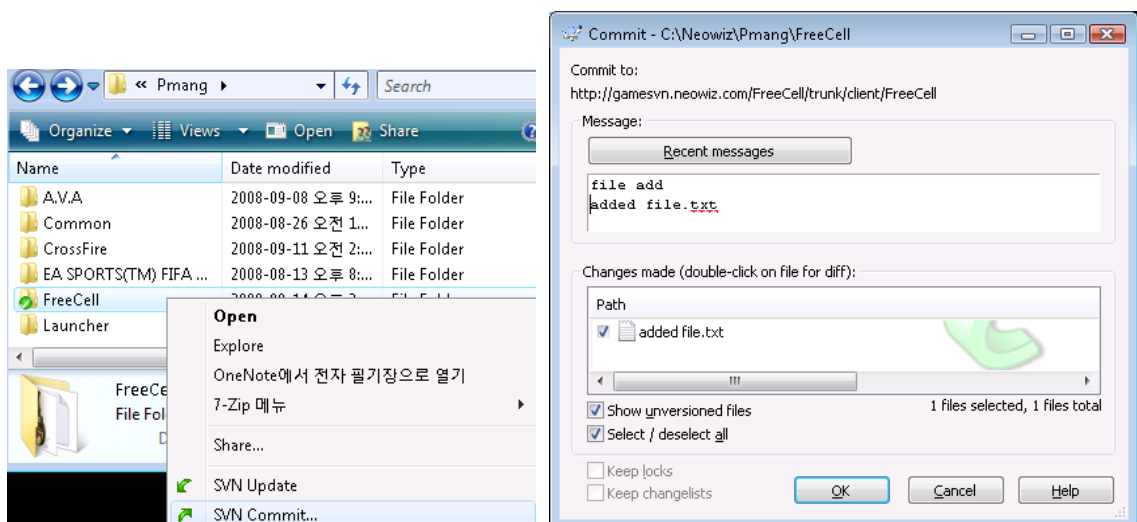
10 작업 폴더 업데이트, 변경된 파일 커밋 하기

10.1 저장소의 리비전 번호는 지속적으로 올라간다. 따라서 체크아웃 받은 작업 폴더에 클라이언트 파일 중 변경 사항이 있는 파일을 올리기 전에 항상 최신 리비전으로 업데이트 해 줘야 한다. SVN 업데이트 명령으로 최신 리비전(혹은 원하는 리비전)으로 업데이트가 가능하다.



<그림 9, 10 작업 디렉터리를 최신으로 업데이트 하기>

10.2 작업 디렉터리를 최신 리비전으로 업데이트 완료하면 이제 작업 디렉터리에 변경된 파일을 올리고 커밋이 가능하다. 작업 디렉터리 상위 폴더에서 마우스 오른쪽 버튼을 눌러 SVN 커밋을 선택하면 작업 디렉터리 내의 변경, 추가된 파일들(버전 관리 대상 파일들)을 선택이 가능하며 커밋 로그 메시지를 입력하고 확인을 누르면 선택된 파일들이 저장소에 저장되고 리비전이 1 증가한다.



<그림 11, 12 SVN 커밋, 변경/추가된 파일 선택 후 커밋>

11 저장소에 저장된 파일들을 익스포트 하기

11.1 프로젝트가 진행되면서 저장소에 파일들을 저장 하다가 버전 관리 용도 파일 없는 깨끗한 디렉터리, 파일을 꺼내 사용해야 할 경우에 익스포트 명령을 사용한다.
(예: 저장소에 저장된 서버 파일을 테스트 서버나 실제 서버에 배포 시, 풀클라이언트 패키징 시)

11.2 익스포트는 TortoiseSVN 에서도 가능하지만 커맨드 라인 도구인 svn.exe export 명령을 이용하면 자동화 등에 이용할 수 있기 때문에 더욱 편리하다. subversion command line client 는 아래 주소에서 최신 버전을 다운받을 수 있다.

<http://www.collab.net/downloads/subversion>

11.3 아래 예제는 svn.exe export 명령을 사용한 익스포트 방법이다. 좀 더 자세한 사용법은 svn.exe help export 를 참고하길 바란다.

저장소 url: <http://gamesvn.neowiz.com/svn/FreeCell/trunk/client/FreeCell>

익스포트할 타겟 디렉토리: c:\WNeowiz\Wpang\WFreeCell_Exported

usage: svn.exe export --username --password repository_url target_dir

```
c:\subversion> svn.exe export --username *** --password ***
http://gamesvn.neowiz.com/svn/FreeCell/trunk/client/FreeCell
c:\neowiz\pmang\FreeCell_Exported
```

```
A C:\Neowiz\Pmang\FreeCell_Exported
A C:\Neowiz\Pmang\FreeCell_Exported\added file.txt
A C:\Neowiz\Pmang\FreeCell_Exported\FreeCell.exe
A C:\Neowiz\Pmang\FreeCell_Exported\desktop.ini
A C:\Neowiz\Pmang\FreeCell_Exported\en-US
A C:\Neowiz\Pmang\FreeCell_Exported\en-US\FreeCell.exe.mui
A C:\Neowiz\Pmang\FreeCell_Exported\FreeCellMCE.lnk
A C:\Neowiz\Pmang\FreeCell_Exported\FreeCellMCE.png
Exported revision 4.
```

12 오픈 시점 리비전을 태깅하기(꼬리표 붙이기)

주. 아래 내용은 네오위즈 패치 시스템을 위한 참고를 위한 자료이다.

12.1 패치 관련 용어 정리

12.1.1 리비전(Revision): 서버버전의 버전 단위

12.1.2 버전(version): 네오위즈 패치 시스템의 버전

12.1.3 Differential patch: 클라이언트의 바뀐 부분만을 뽑아 패치 하는 네오위즈 패치 매니저의 패치 방식

12.2 네오위즈의 패치 방식인 Differential patch 는 사용자에게 오픈 된 시점의 게임

클라이언트 리비전을 tags\patch 폴더 아래에 패치 시스템 버전 값으로 꼬리표(tag)를 붙여야만 패치 시스템의 버전 관리가 되어 무결성을 보장할 수 있다.

12.3 꼬리표(tag)를 붙이는 작업은 수동으로 할 경우 문제가 될 소지가 있으므로 패치 매니저에서 자동으로 붙이는 것으로 한다. (지원 예정)

12.4 아래 예는 svn.exe copy 명령을 이용해 게임 오픈 시점의 리비전이 4 라고 했을 경우 프리셀을 패치 버전 1000 으로 꼬리표(tag)를 붙이는 방법이다. 좀 더 자세한 사용법은 svn.exe help copy 를 참고하길 바란다.

usage: svn.exe copy --username --password --message source_url@rev
target_url (참고. @rev 를 붙이지 않으면 최신 리비전을 의미한다)

```
c:\subversion>svn.exe copy --username *** --password *** --message  
tag_rev4_patch_1000  
http://gamesvn.neowiz.com/svn/FreeCell/trunk/client/FreeCell@4  
http://gamesvn.neowiz.com/svn/FreeCell/tags/patch/1000  
Committed revision 5. (참고. 꼬리표를 붙이면 리비전이 1 증가한다)
```

12.5 이때 꼬리표(tag)달아 둔 저장소 주소를 익스포트 하여 정확한 패치 시스템 버전의 풀클라이언트를 묶는 용도로 사용한다. 이때 패킹 시 불필요한 파일의 삭제와 version.dat 파일 등의 추가 작업을 해주고 풀클라이언트 패킹이 되어야 한다.

위 예제에서 풀클라이언트를 export 할 저장소 주소는 아래와 같고 version 은 1000 이다.

<http://gamesvn.neowiz.com/svn/FreeCell/tags/patch/1000>

마침.