TortoiseSVN

1. 도구 개요
2. 설치 및 실행
3. 주요 기능
4. 활용 예제

1. 도구 개요

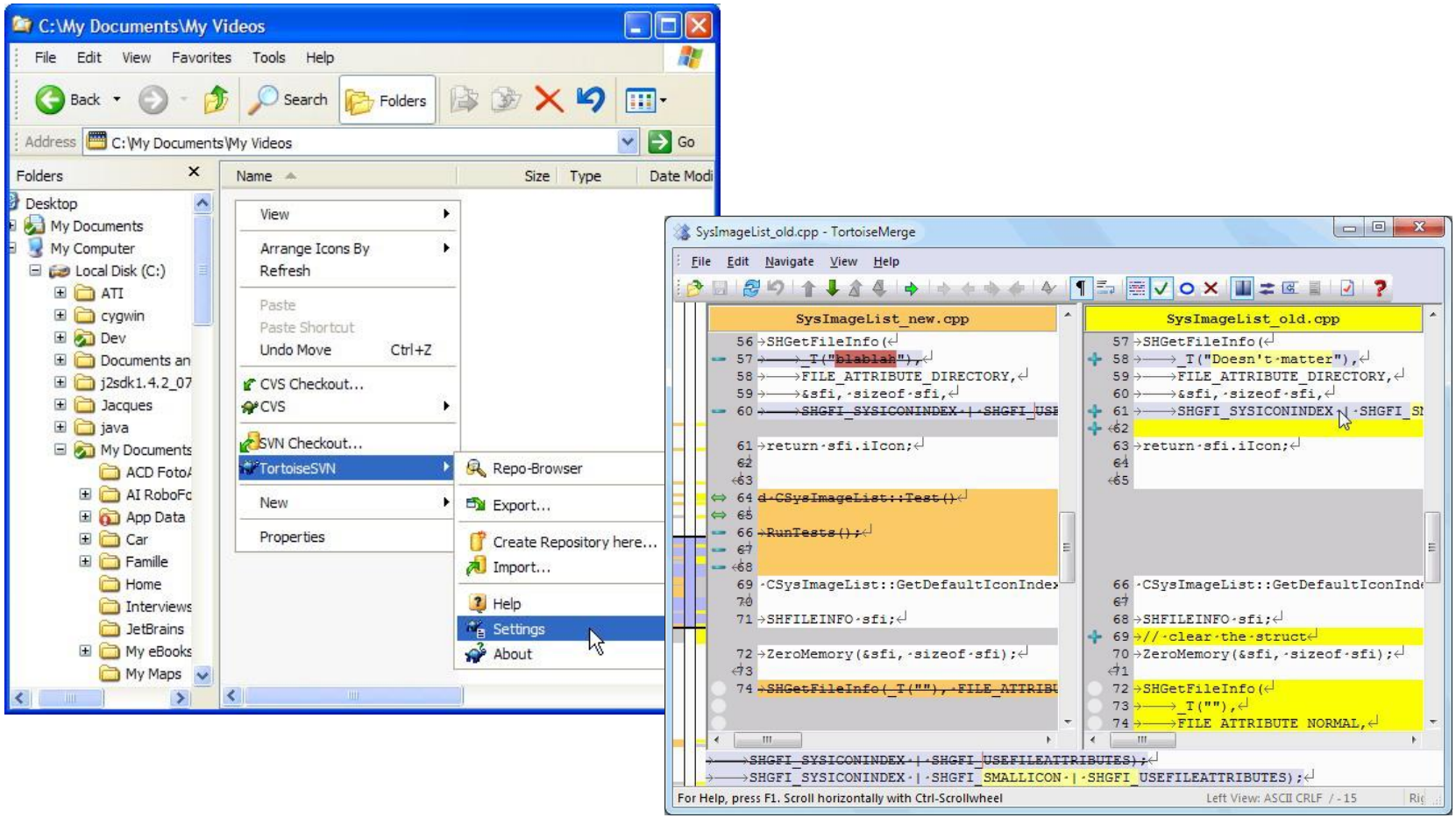
1.1 도구 정보 요약

도구명	Tortoise SVN (http://tortoisesvn.net)	라이선스	GNU GPL v2.0
소개	• Subversion를 통해서 소스 버전 관리를 할 수 있게 하는 클라이언트 도구		
특징	• Windows Explorer에서 곧바로 Subversion를 사용하여 버전 컨트롤 가능 • Importing, Branching, Merging기능과 더불어 특정 파일의 변경로그를 살펴 볼 수 있음		
주요기능	• Checkout / Commit / Update / Revert • Diff / Show log • Import/Branch/Tag		
실행환경	• Windows	사전설치도구	• Subversion
카테고리	• 형상 관리	최신버전	• v1.8.8 (2014.10)
관련도구	• Subversion, Subclipse, EasySVN		

1. 도구 개요

1.2 스크린 캡쳐 및 주요 기능

- 윈도우 탐색기와 통합되어 편리한 형상 관리 환경 제공
- Diff를 통한 파일 버전 별 내용 비교
- Checkout/Checkin/Commit/Update/Import 등 SVN의 형상 관리 기능 제공



2. 설치 및 실행

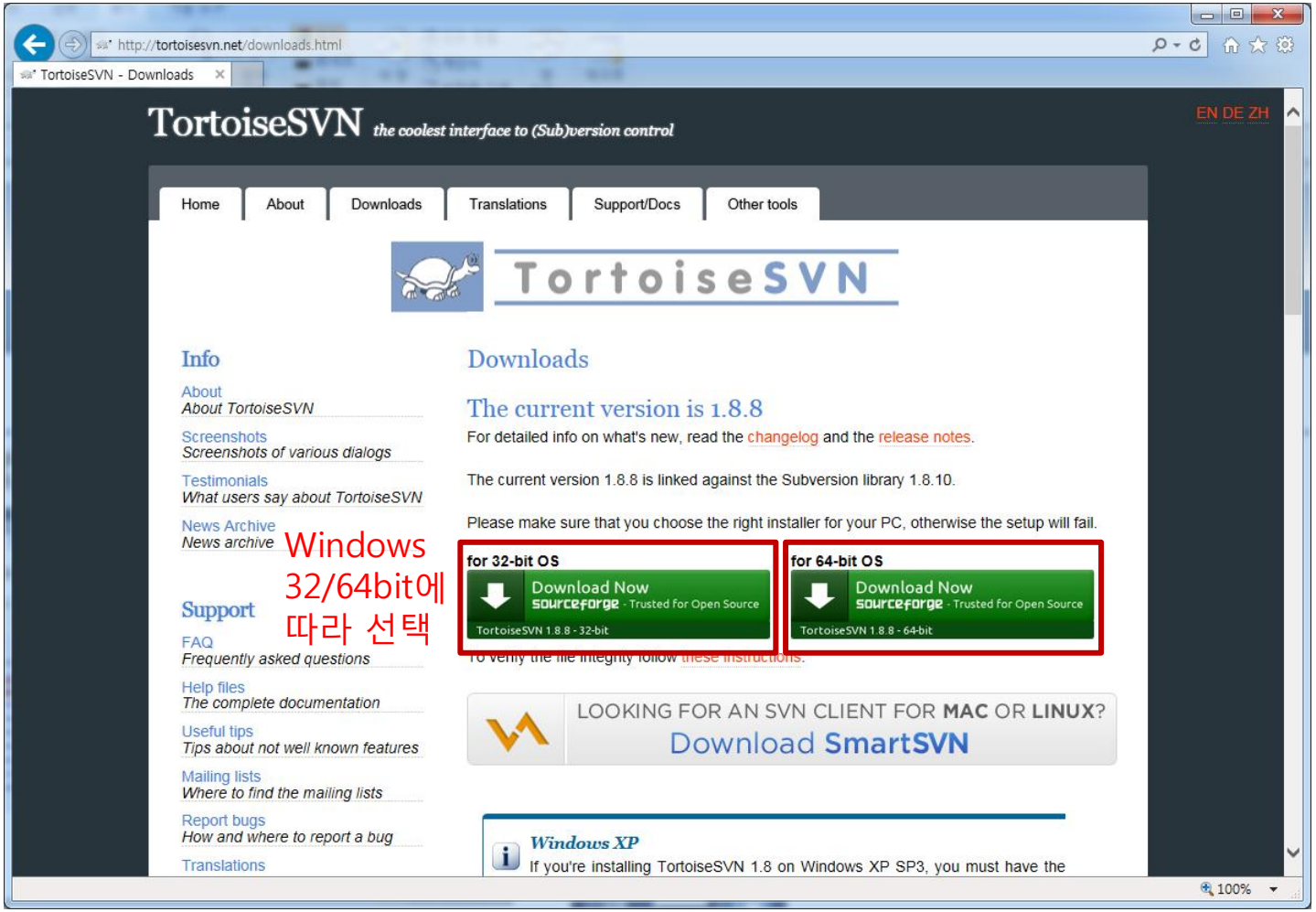
세부 목차

2.1 다운로드
2.2 설치
2.3 설치 확인

2. 설치 및 실행

2.1 다운로드

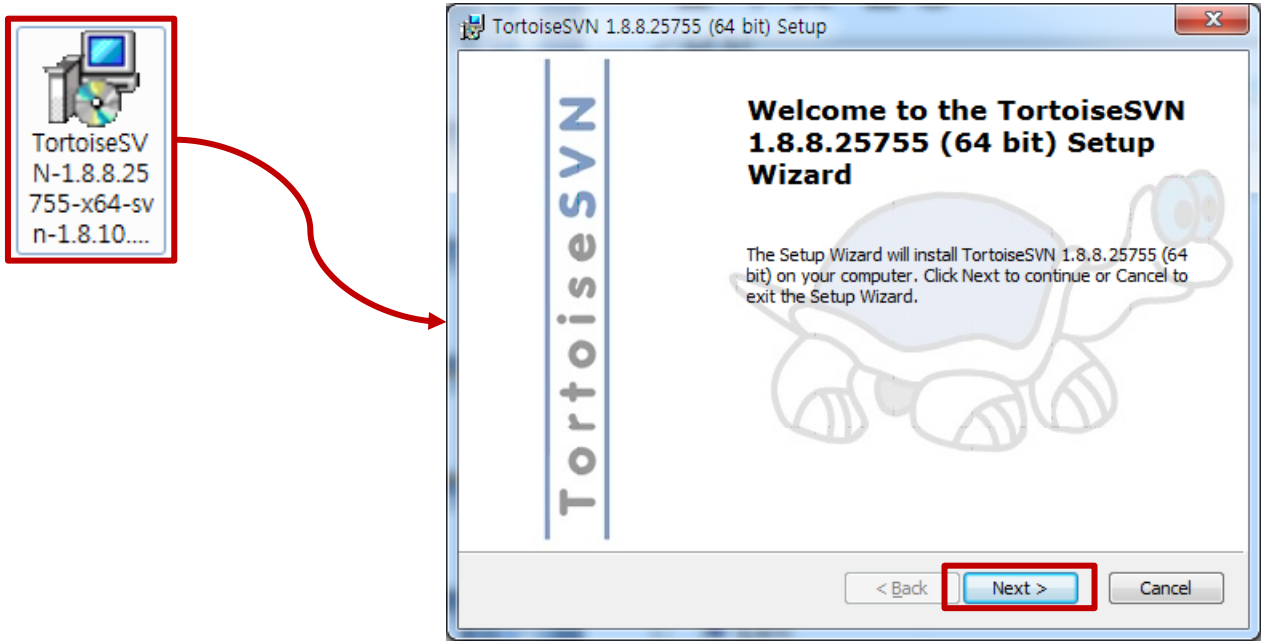
- http://tortoisesvn.net/downloads.html 에서 사용하는 OS에 해당하는 파일을 다운 받는다.



2. 설치 및 실행

2.2 설치 (1/4)

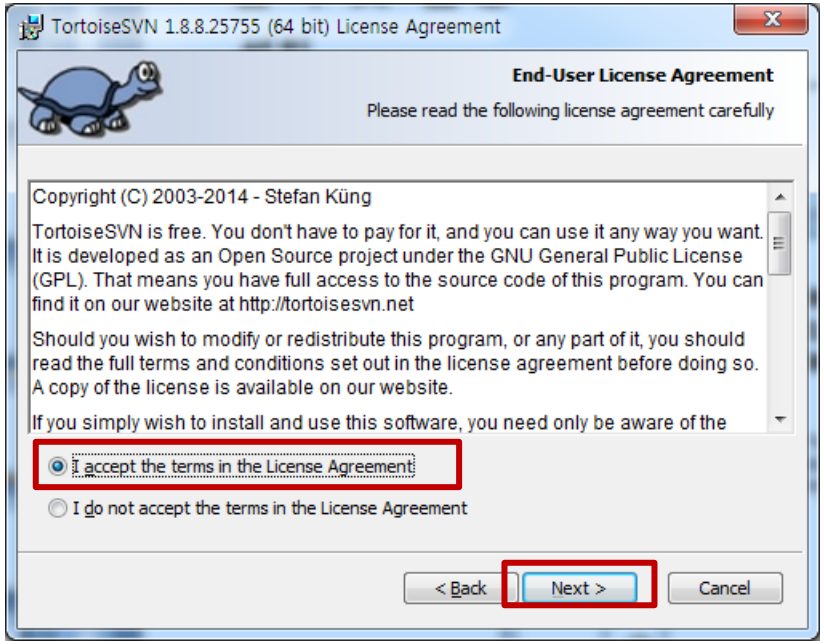
- TortoiseSVN-1.8.8.25755-x64-svn-1.8.10.msi 를 실행한다.
- 설치 창이 나오면 Next 를 클릭한다.

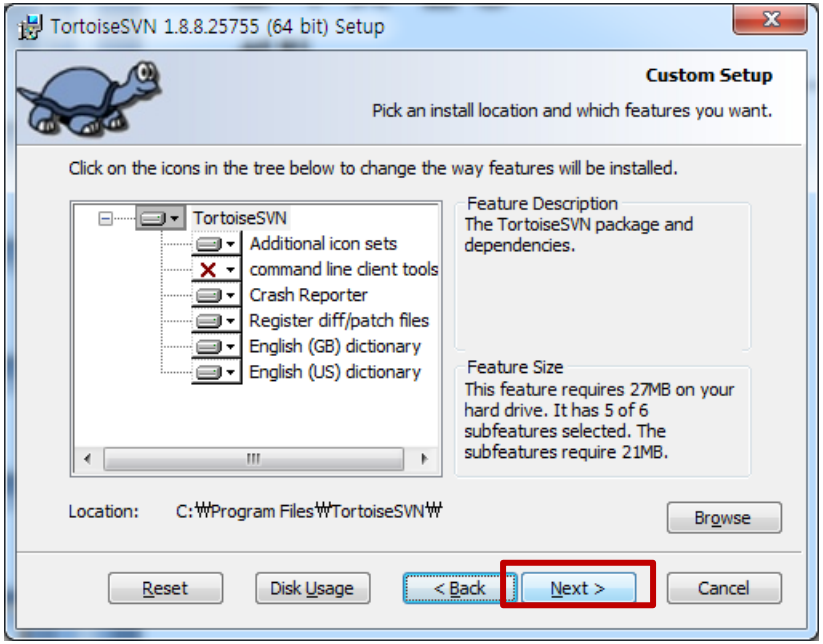


2. 설치 및 실행

2.2 설치 (2/4)

- 사용자 라이선스 화면에서 [I accept ...]를 체크한 후 Next 를 클릭한다.
- 다음 화면에서도 Next 를 클릭한다.

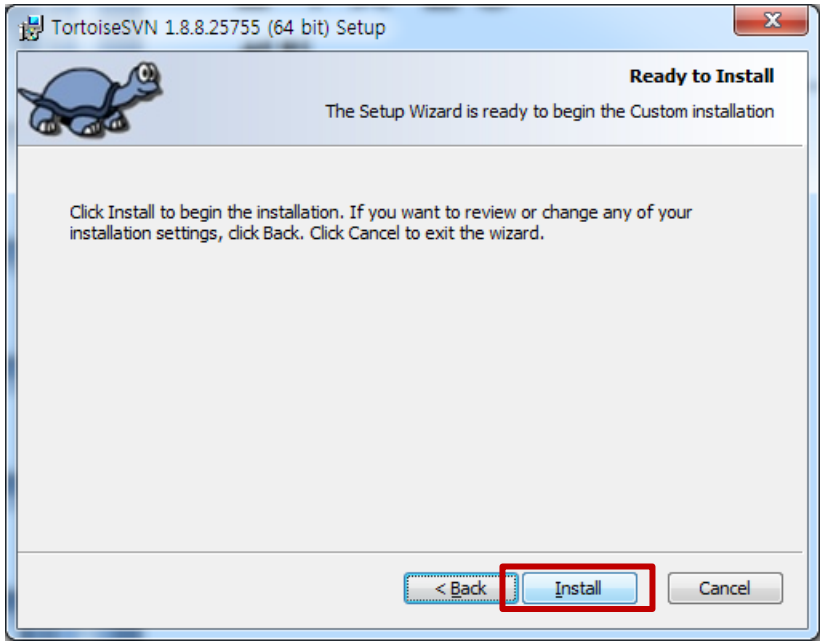


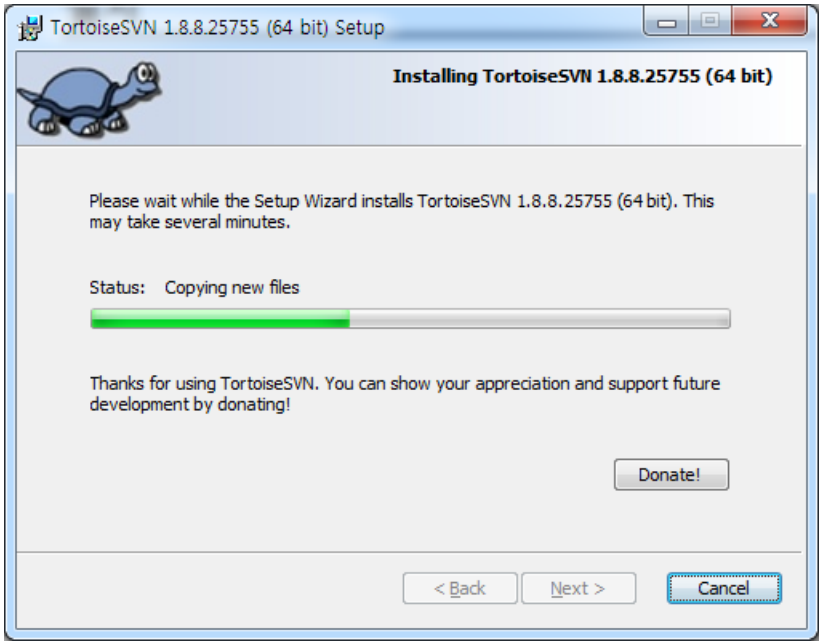


2. 설치 및 실행

2.2 설치 (3/4)

- Install 을 클릭하면 설치가 진행된다.

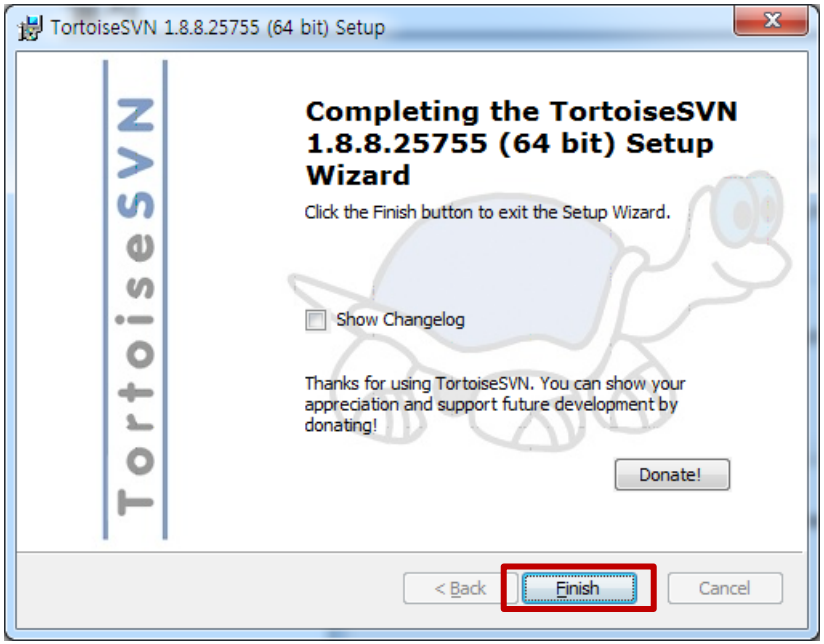




2. 설치 및 실행

2.2 설치 (4/4)

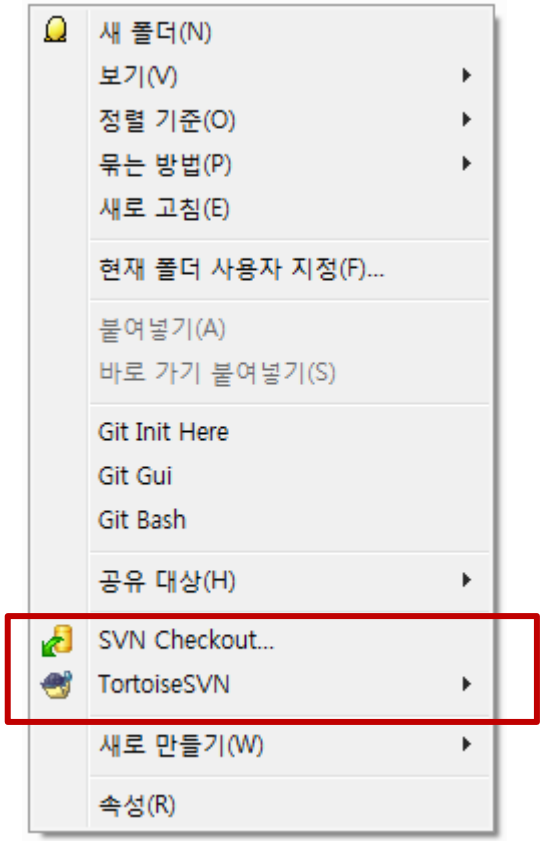
- 설치 완료 확인 화면에서 Finish 를 클릭한다.



2. 설치 및 실행

2.3 설치 확인

- 윈도우탐색기에서 마우스 오른쪽 버튼을 클릭한다.
- 정상적으로 설치되었다면 SVN 관련 기능이 표시된다.



3. 주요 기능

세부 목차

3.1 기본 개념 설명
3.2 Checkout
3.3 Add
3.4 Commit
3.5 Update
3.6 Delete
3.7 Diff
3.8 Show log
3.9 Revert

3. 주요 기능

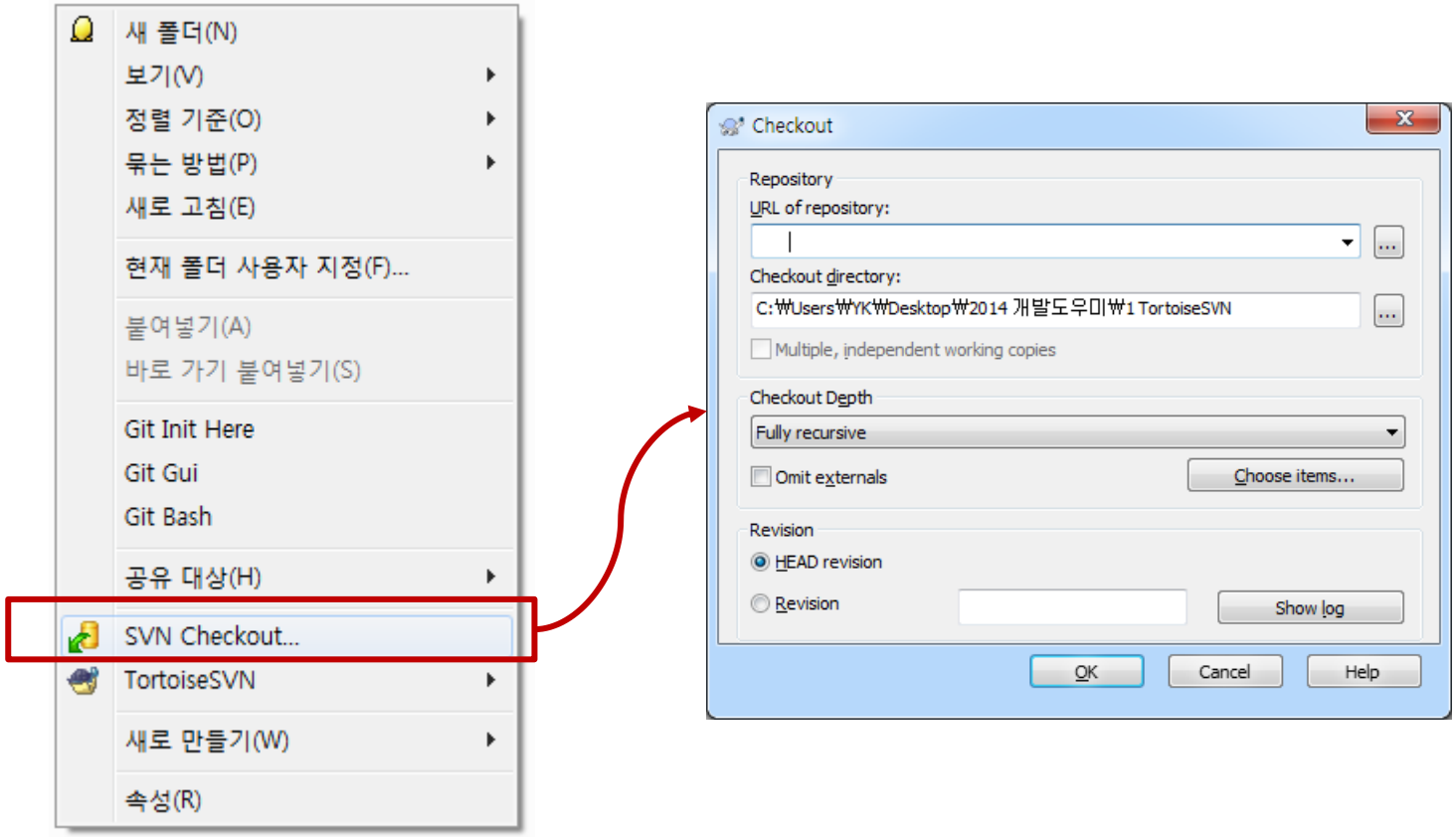
3.1 기본 개념 설명

- SVN(Subversion)은 모든 형상관리할 파일들을 히스토리 정보를 포함하여 중앙의 데이터베이스에서 관리한다.
- Repository는 SVN 서버에서 실행되고 있는 파일서버의 형태이며 최종적인 원본 데이터를 관리한다.
- TortoiseSVN는 Repository로부터 최종 버전을 제공받아 로컬 PC에 작업 데이터를 저장한다.
- 로컬 PC의 작업 데이터에 대해 수정 작업을 수행한 후 최종 변경된 내용을 Repository에 반영한다.

3. 주요 기능

3.2 Checkout (1/4)

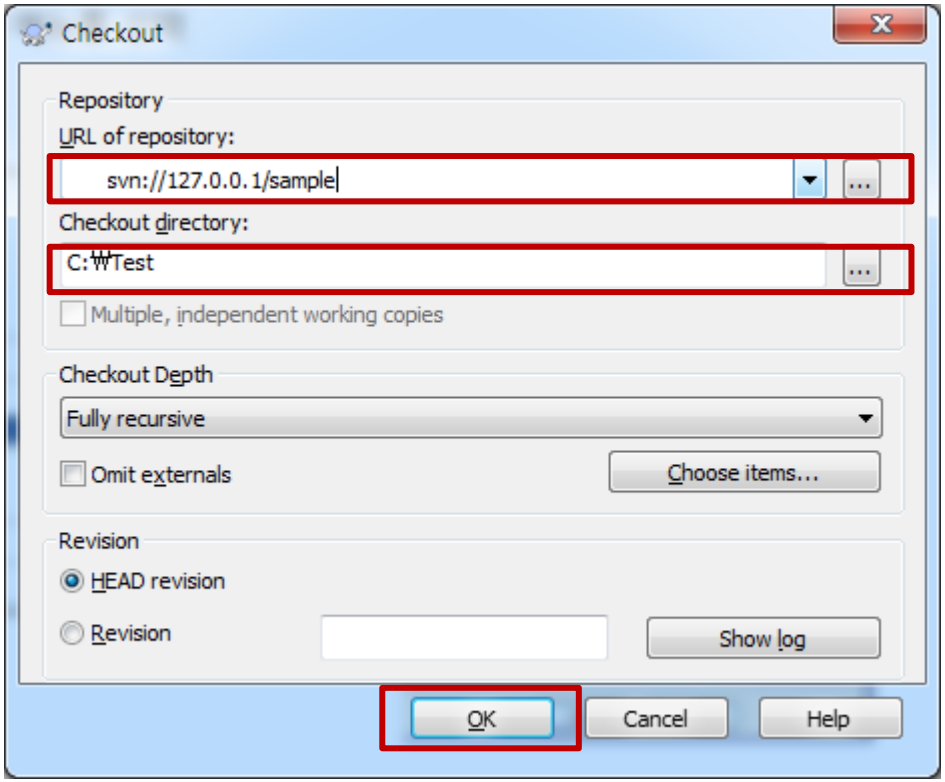
- SVN서버의 Repository와 로컬 PC의 폴더를 연결하여 형상관리 환경을 설정하기 위해 Checkout 을 수행한다.
- 윈도우탐색기에서 마우스 오른쪽 버튼을 클릭한다.
- SVN Checkout 을 클릭하면 Checkout 창이 열린다.



3. 주요 기능

3.2 Checkout (2/4)

- URL of repository 에 SVN 서버의 Repository 주소를 입력한다.
- Checkout directory 에 Checkout 대상이 되는 로컬 폴더를 입력한다.
- OK 를 클릭한다.



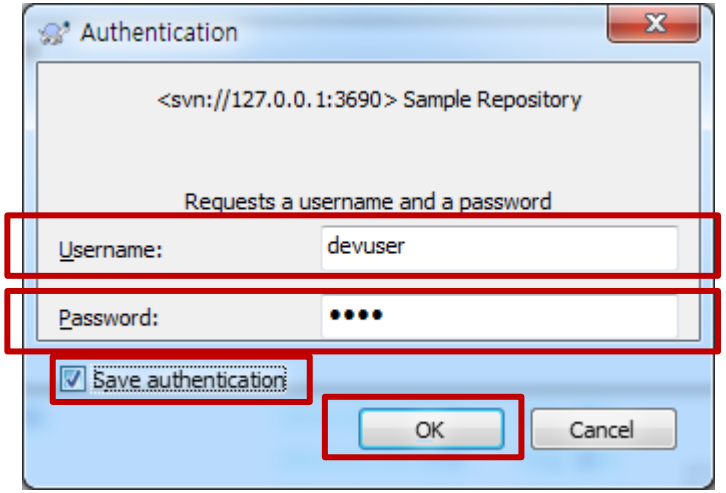
SVN 서버의 Repository 주소

Checkout 대상 로컬 폴더

3. 주요 기능

3.2 Checkout (3/4)

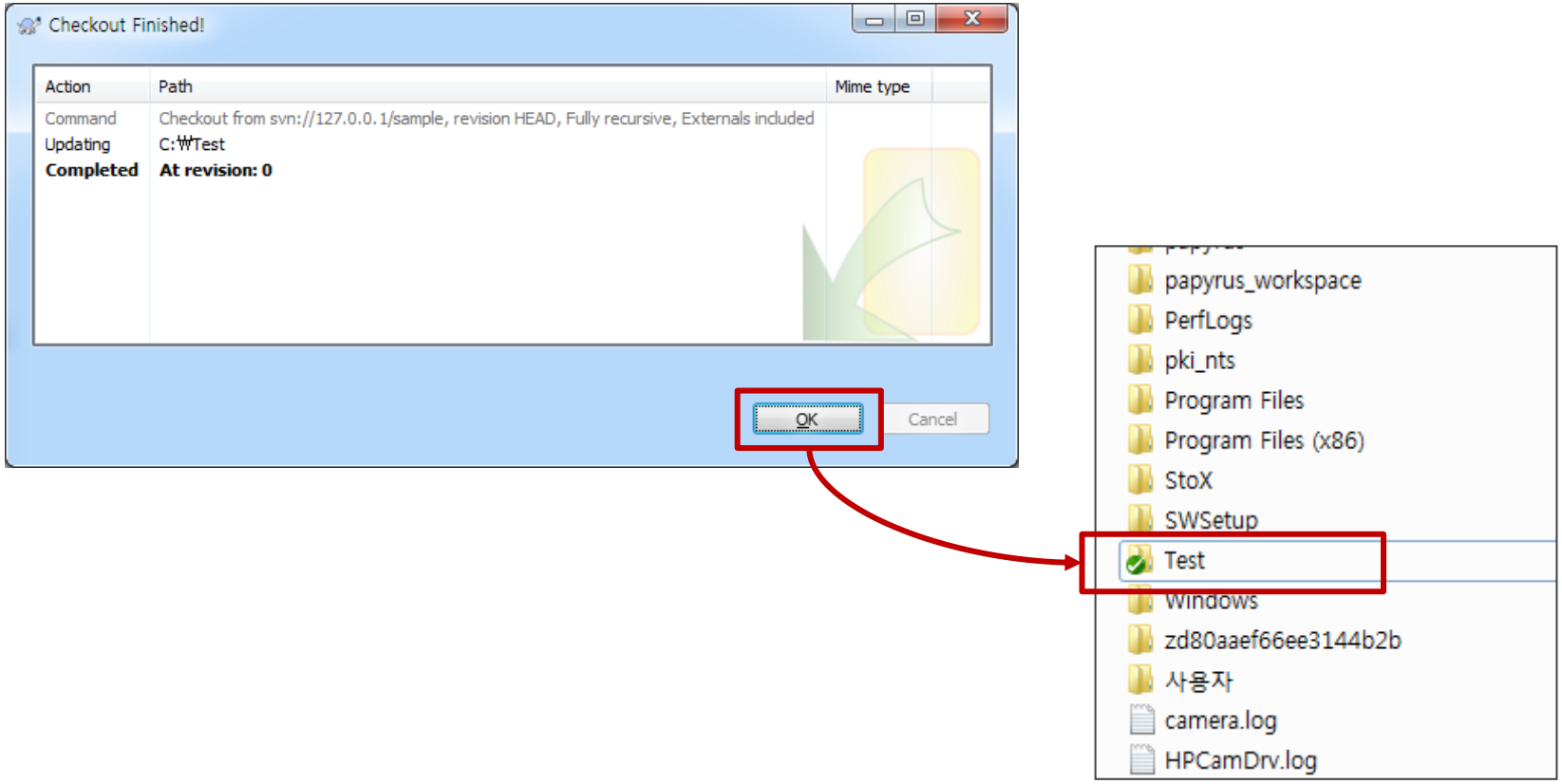
- Username 에 SVN 사용자 ID를 입력한다.
- Password 에 사용자 암호를 입력한다.
- Save authentication 을 체크하고 OK 를 클릭한다.



3. 주요 기능

3.2 Checkout (4/4)

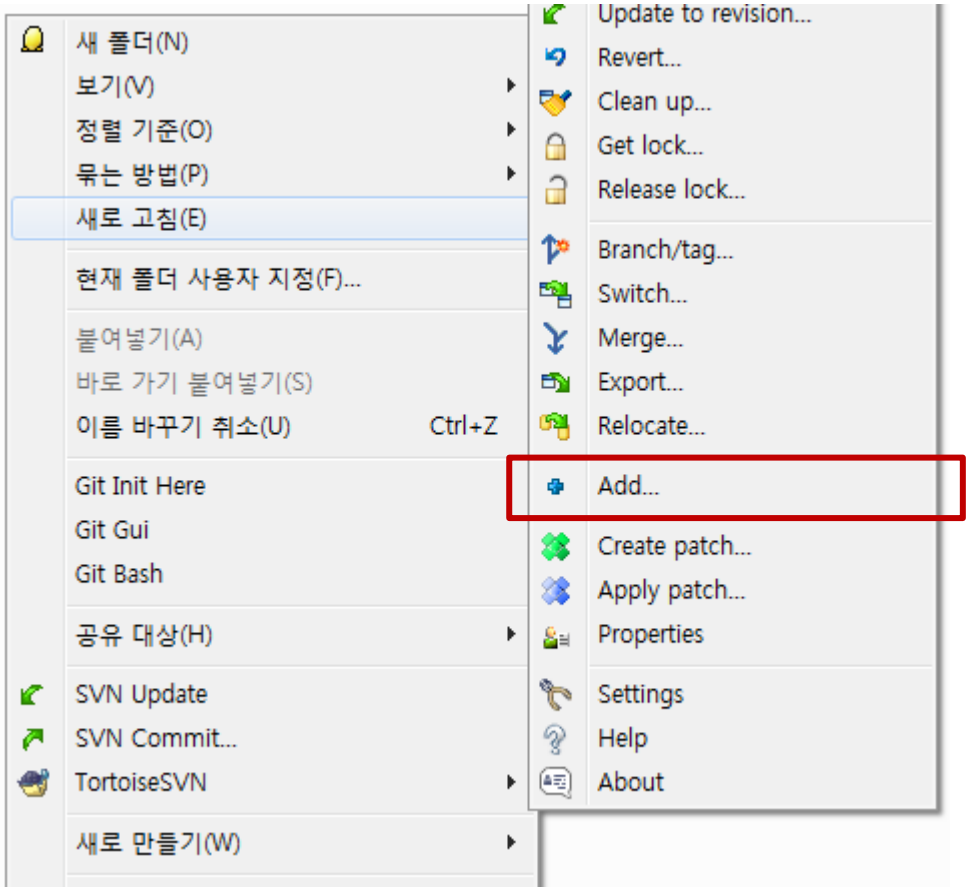
- Checkout 작업 확인 내용 창이 열린다.
- OK 를 클릭한다.
- 탐색기에서 Checkout 대상 폴더의 아이콘 모양이 모양이 바뀐 것을 확인 한다.



3. 주요 기능

3.3 Add (1/2)

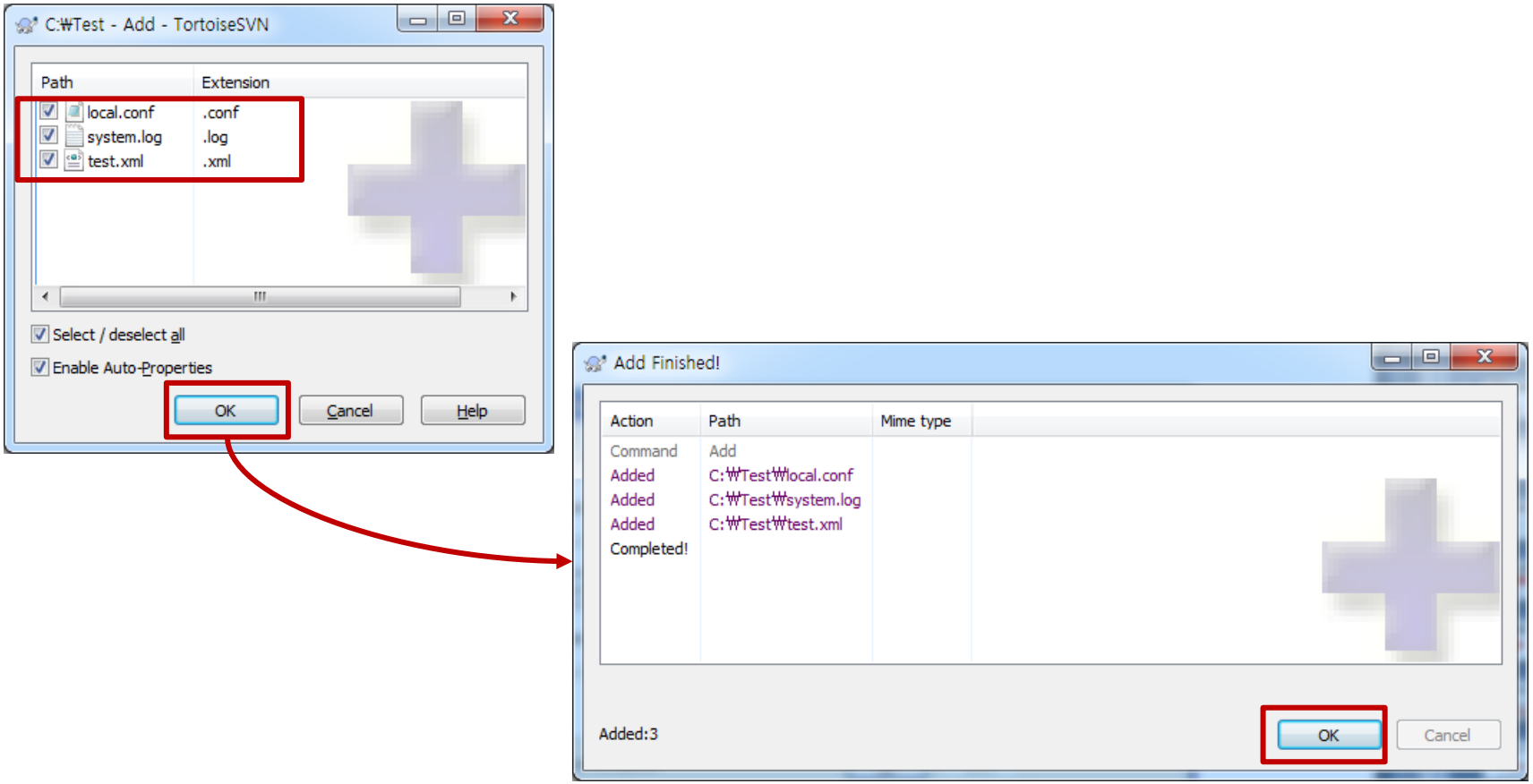
- 신규로 만들어진 파일을 Repository에 등록하기 위해 Add를 수행한다. (최종 파일 등록은 Add 후 Commit 까지 해야 완료됨)
- 탐색기에서 마우스 오른쪽 버튼을 클릭한다.
- TortoiseSVN -> Add 를 클릭한다.



3. 주요 기능

3.3 Add (2/2)

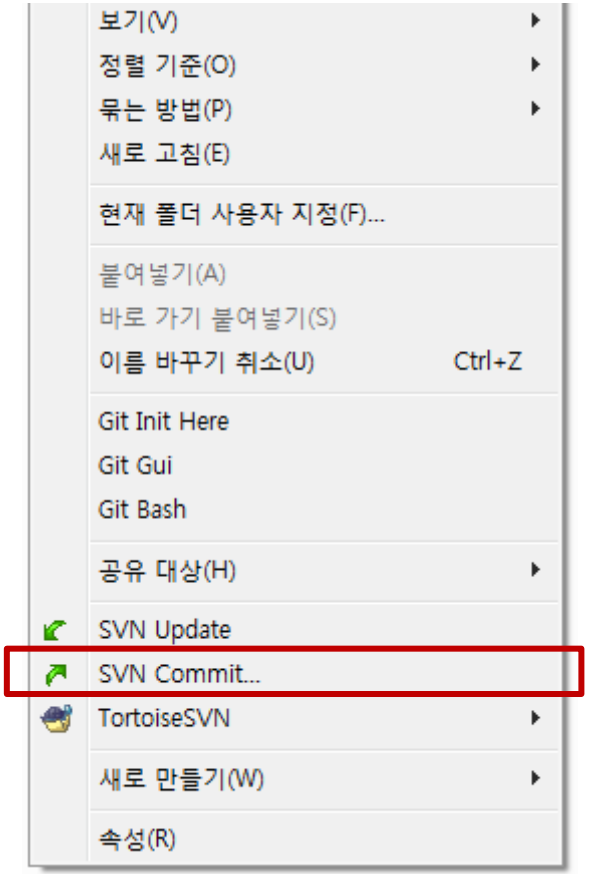
- 추가할 파일이 체크되었는지 확인하고 OK 를 클릭한다.
- Add 결과 확인창이 열리면 확인 후 OK 를 클릭한다.



3. 주요 기능

3.4 Commit (1/3)

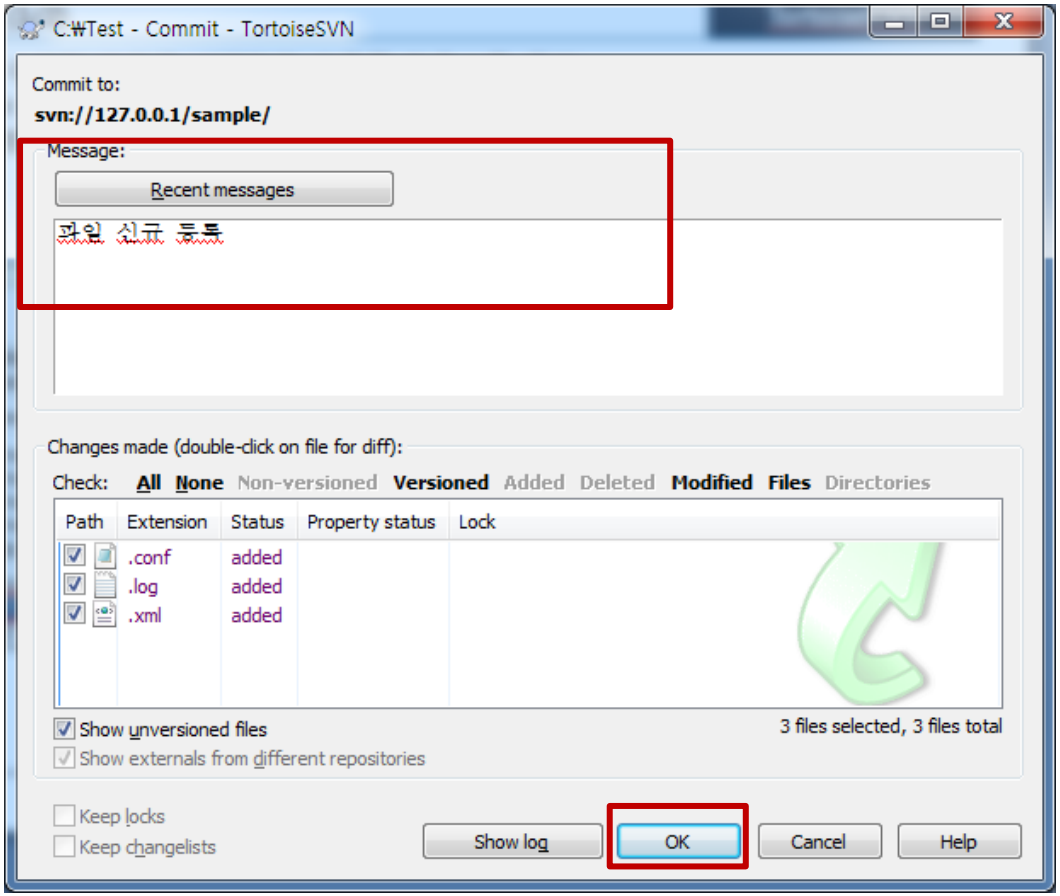
- 작업 데이터의 추가/수정/삭제 등 변경사항을 Repository에 최종 반영하기 위해 Commit 을 수행한다.
- 탐색기에서 마우스 오른쪽 버튼을 클릭한다.
- SVN Commit 을 클릭한다.



3. 주요 기능

3.4 Commit (2/3)

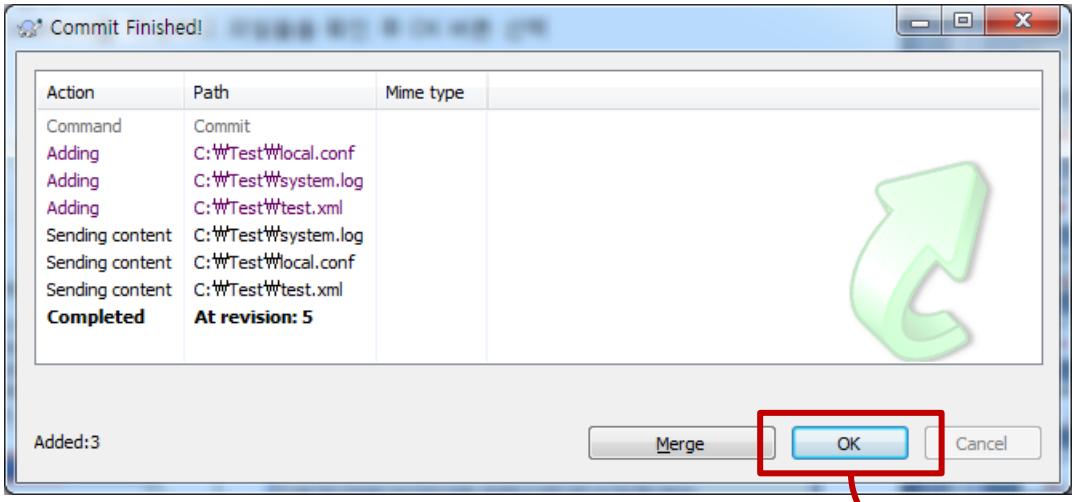
- Commit 확인창이 열리면 Message 를 입력한다.
- 변경사항을 확인하고 OK 를 클릭한다.

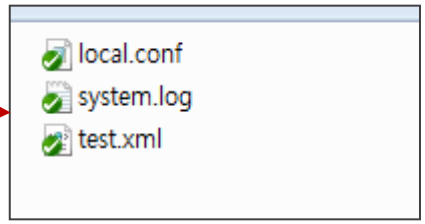


3. 주요 기능

3.4 Commit (3/3)

- Commit 결과창이 열리면 내용을 확인하고 OK 를 클릭한다.
- Commit 결과에 따라 변화사항을 확인한다



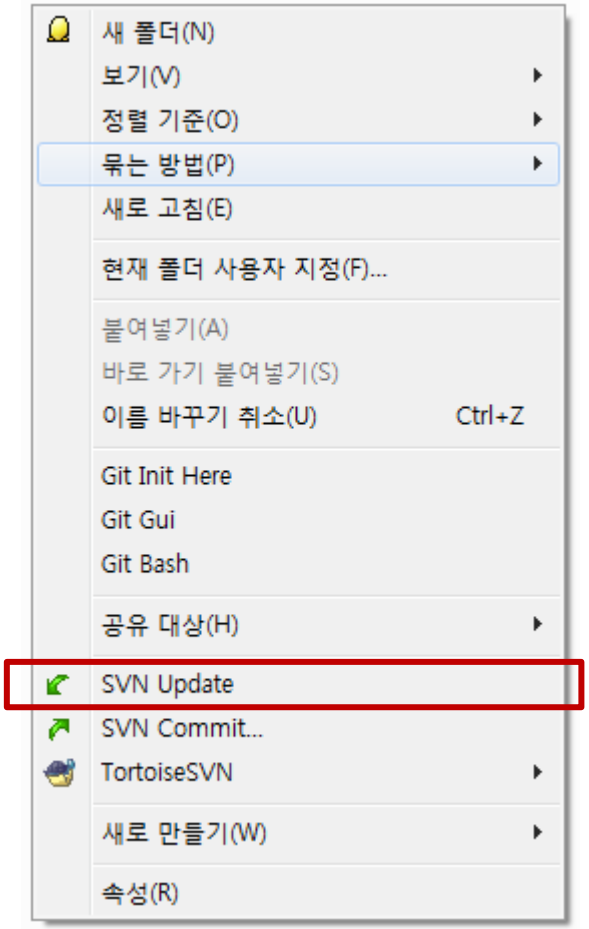


신규로 Add 된 파일들이 Commit되어 아이콘이 변화한 예임

3. 주요 기능

3.5 Update (1/2)

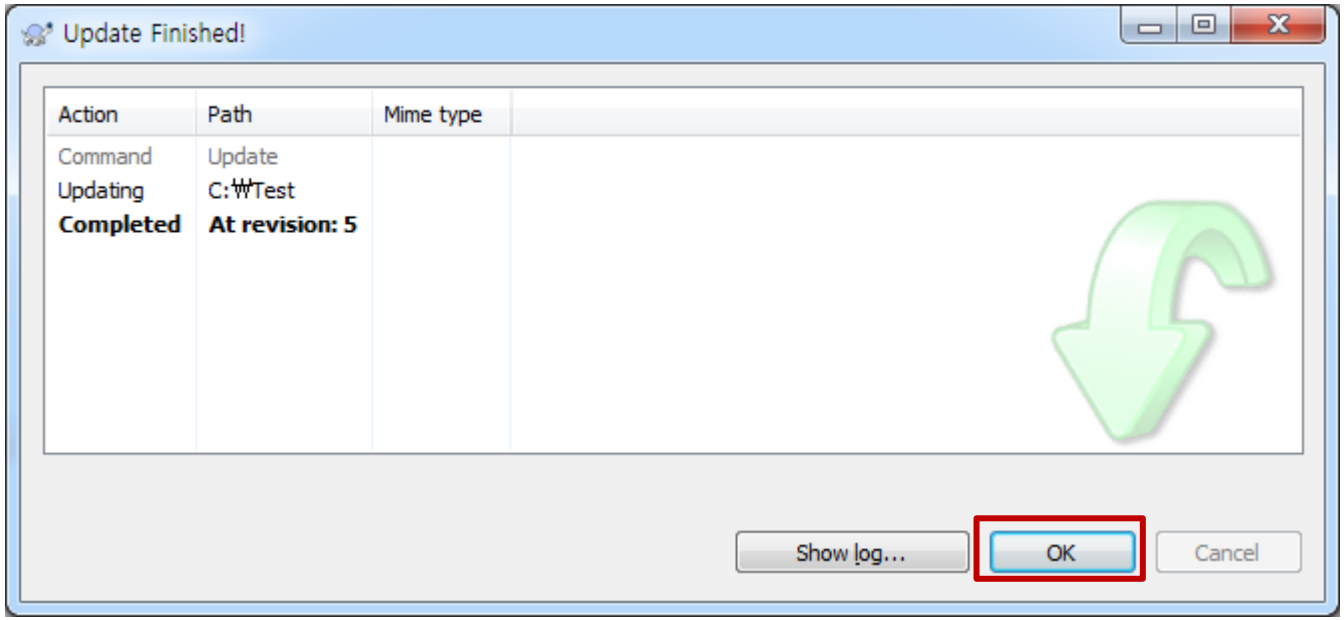
- Repository의 원본 데이터를 로컬 PC로 내려받기 위해 Update를 수행한다.
- 탐색기에서 마우스 오른쪽 버튼을 클릭한다.
- SVN Update 를 클릭한다.



3. 주요 기능

3.5 Update (2/2)

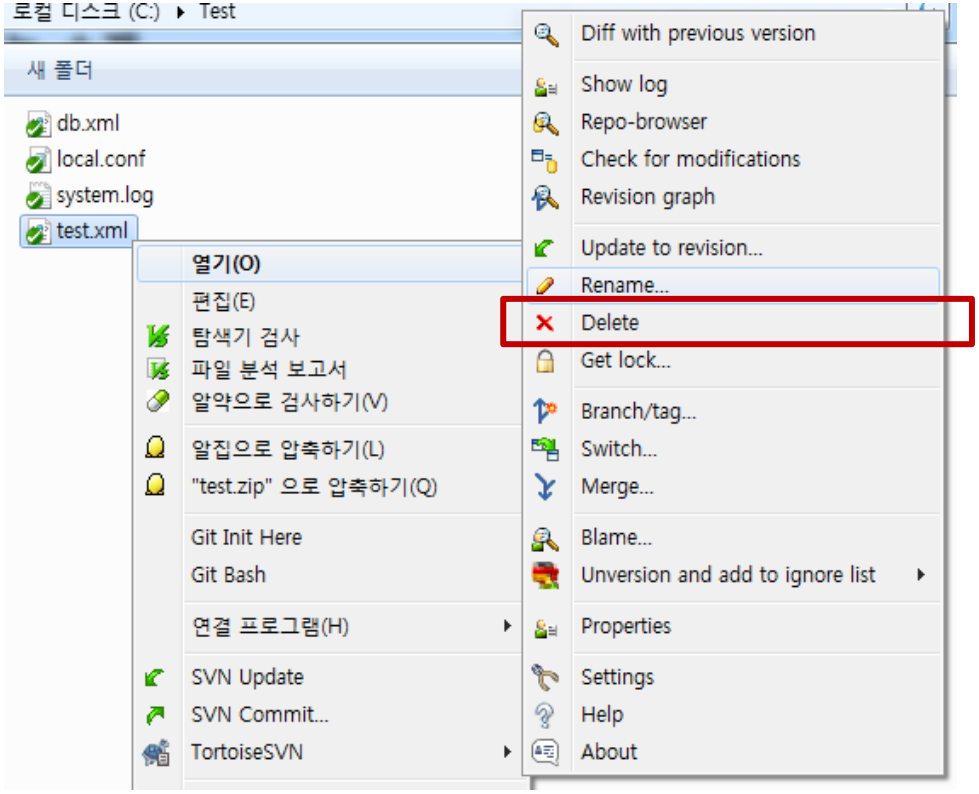
- Update 결과창이 열리면 내용을 확인하고 OK 를 클릭한다.



3. 주요 기능

3.6 Delete (1/2)

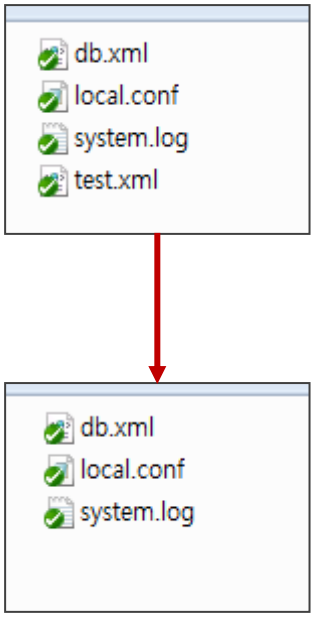
- 로컬PC 및 Repository에서 파일을 삭제하기 위해 Delete를 수행한다.
  (최종 파일 삭제는 Delete 후 Commit 까지 해야 완료됨)
- 삭제할 파일을 선택하고 마우스 오른쪽 버튼을 클릭한다.
- TortoiseSVN -> Delete 를 클릭한다.

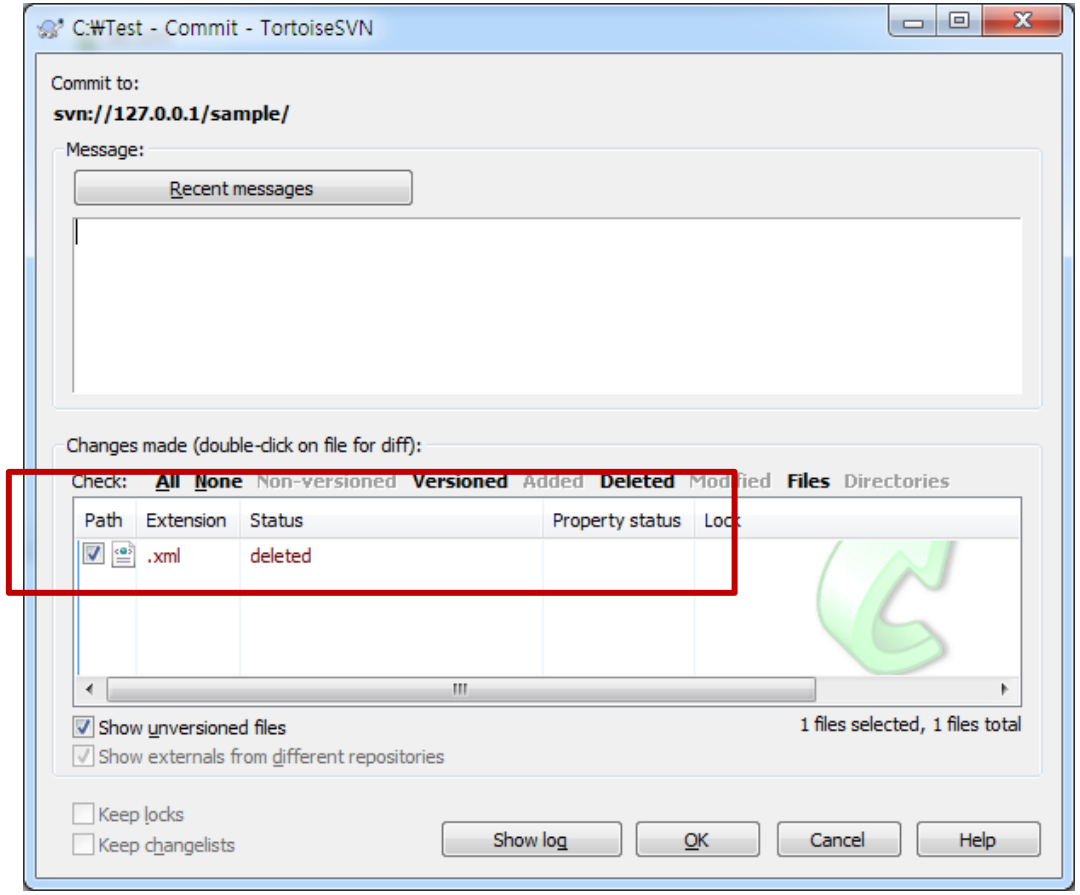


3. 주요 기능

3.6 Delete (2/2)

- 탐색기에서 delete 한 파일이 사라지게 되고
- Commit 을 실행하면 변경내역에 delete 항목을 확인할 수 있다.

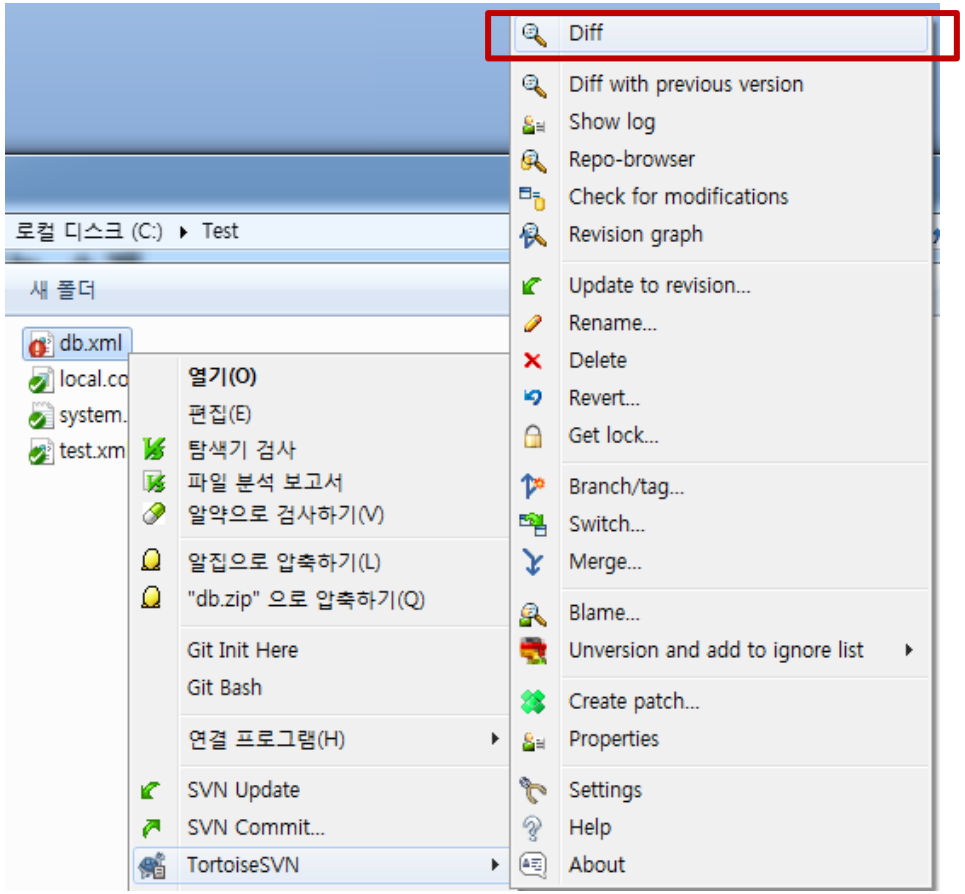




3. 주요 기능

3.7 Diff (1/4)

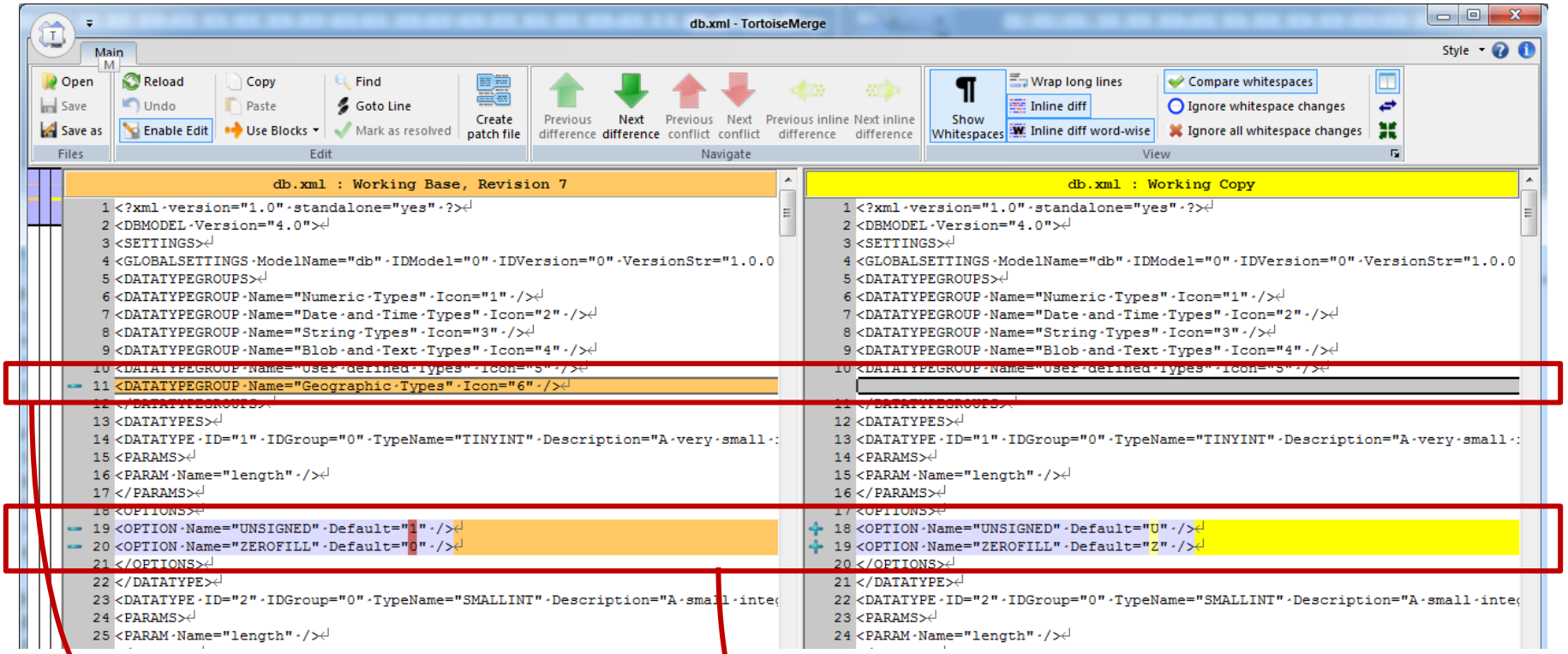
- 로컬 PC의 작업 데이터와 Repository의 원본 데이터의 내용을 비교하기 위해 Diff 를 수행한다.
- 비교할 파일을 선택하고 마우스 오른쪽 버튼을 클릭한다.
- TortoiseSVN -> Diff 를 클릭한다.



3. 주요 기능

3.7 Diff (2/4)

- Diff 창을 통해 차이점을 보여준다.
- 왼쪽에 Repository의 원본 데이터, 오른쪽에 로컬PC의 작업 데이터 표시
- 오른쪽의 작업 데이터는 편집/저장이 가능하다.



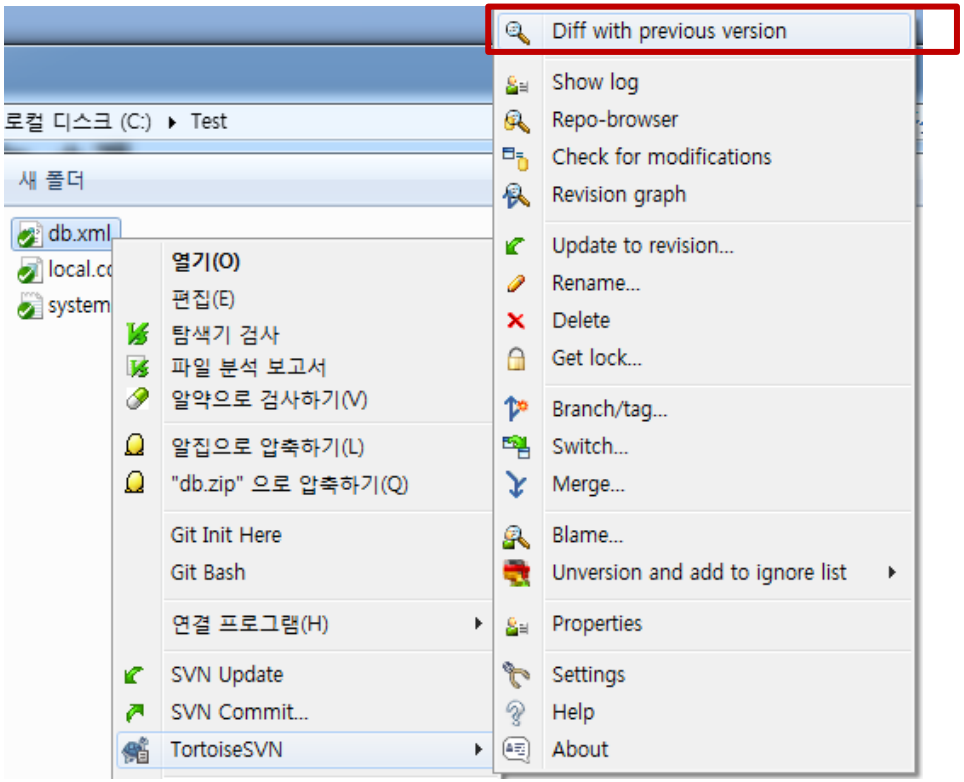
로컬 파일에서 내용이 삭제된 부분 표시

로컬 파일의 내용이 수정되어 일치하지 않는 부분 표시

3. 주요 기능

3.7 Diff (3/4)

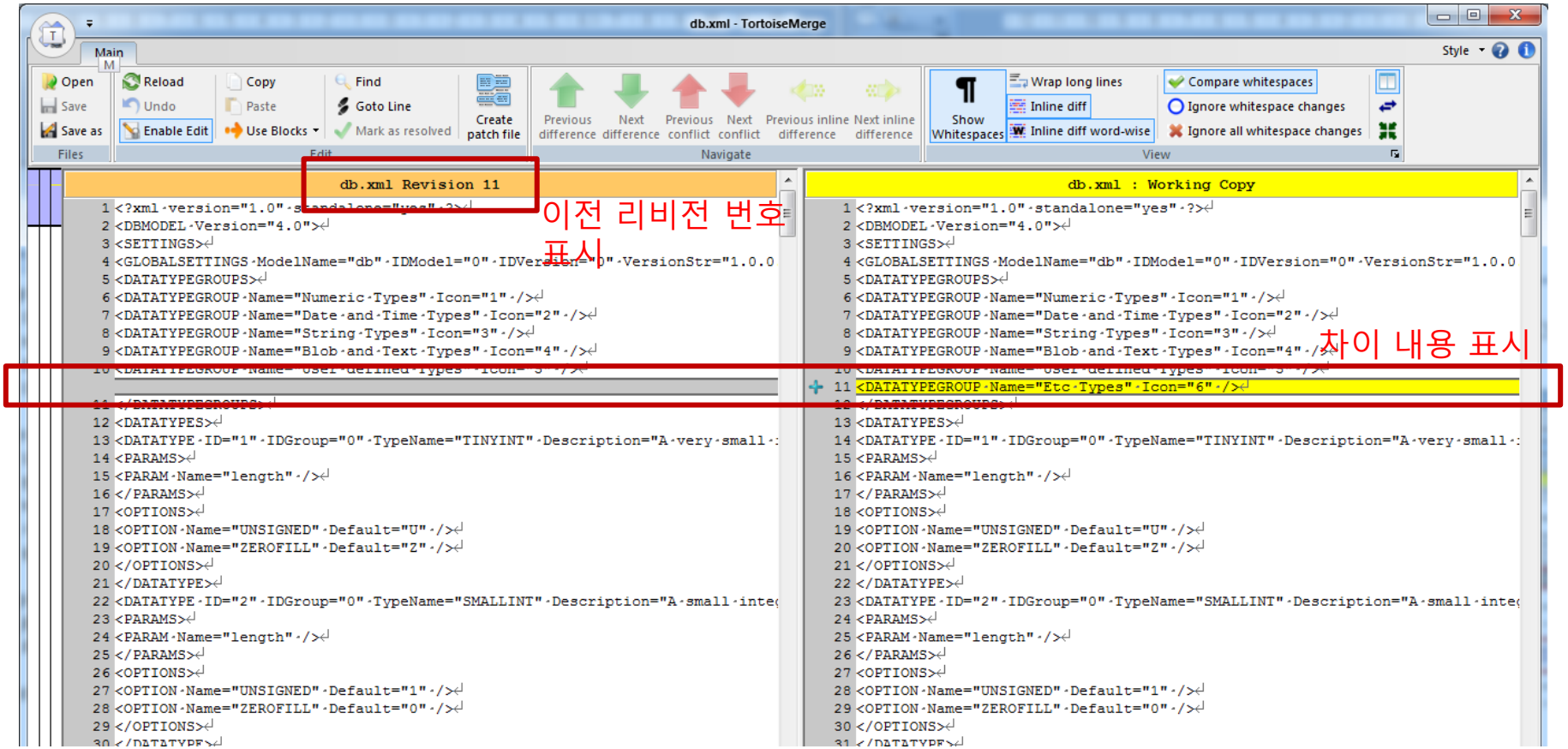
- 현재 파일과 과거 파일의 차이을 비교하기 위해 Diff with previous vsersion 을 수행한다.
- 비교할 파일을 선택하고 마우스 오른쪽 버튼을 클릭한다.
- TortoiseSVN -> Diff with previous vsersion 을 클릭한다.



3. 주요 기능

3.7 Diff (4/4)

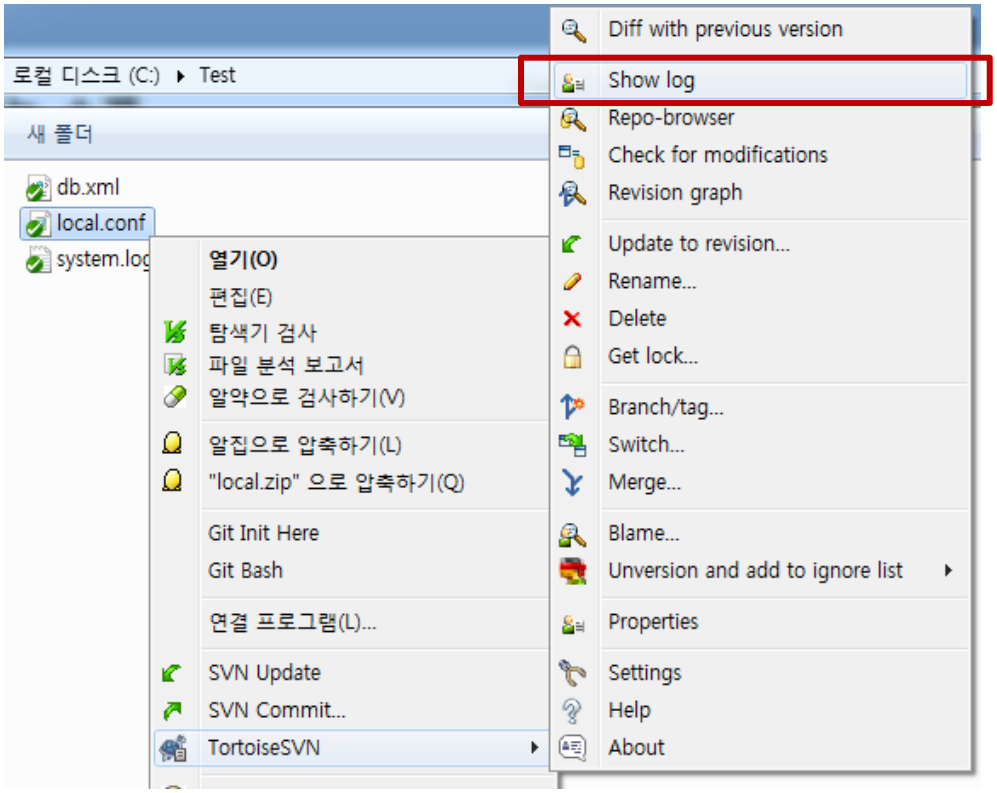
- Diff 창을 통해 차이점을 보여준다.
- 왼쪽에 바로 직전 버전의 데이터, 오른쪽에 현재 데이터 표시
- 오른쪽의 작업 데이터는 편집/저장이 가능하다.



3. 주요 기능

3.8 Show log (1/2)

- 파일에 대한 변경 히스토리와 메시지를 확인하기 위해 Show log 를 수행한다.
- 파일을 선택하고 마우스 오른쪽 버튼을 클릭한다.
- TortoiseSVN -> Show log 를 클릭한다.



3. 주요 기능

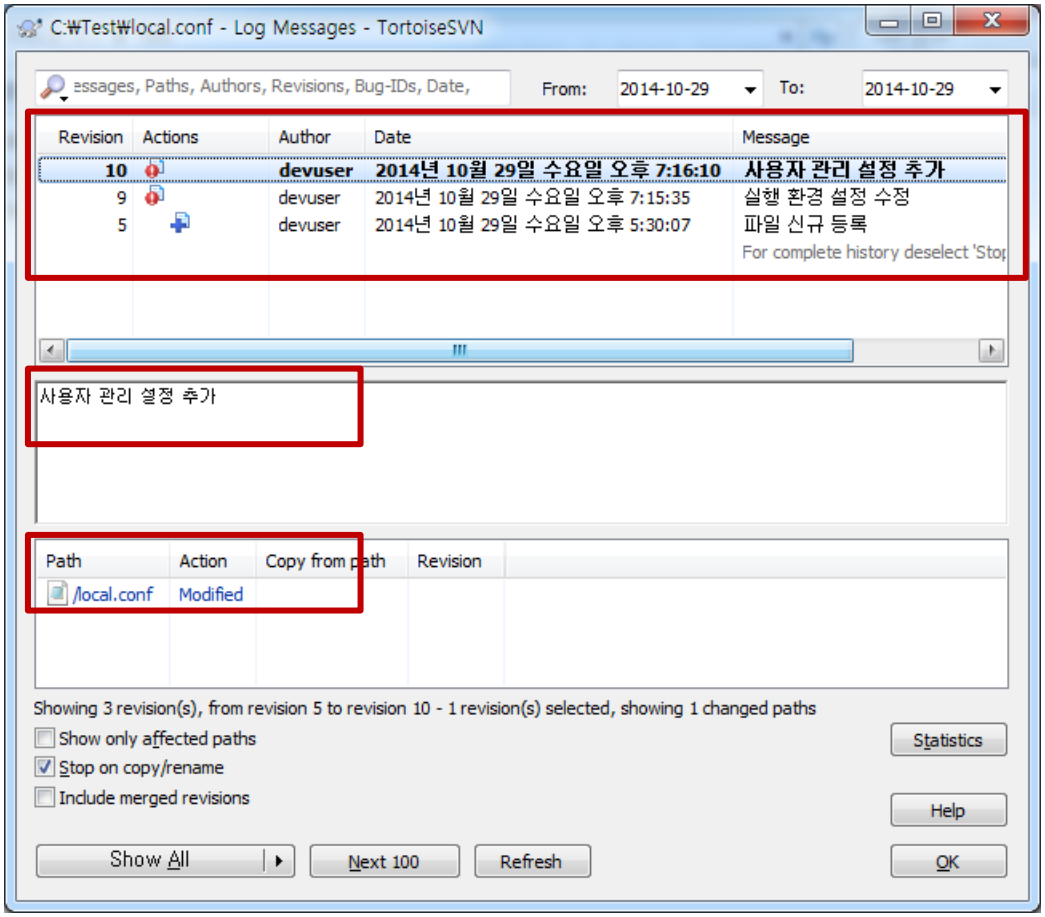
3.8 Show log (2/2)

- 파일에 대한 변경 히스토리를 확인하기 위해 Show log 를 수행한다.
- 해당 파일의 변경 내역, Revision별 Action, Message 내역 등을 확인할 수 있다.

변경 히스토리 내역 확인

Revision별 Message 확인

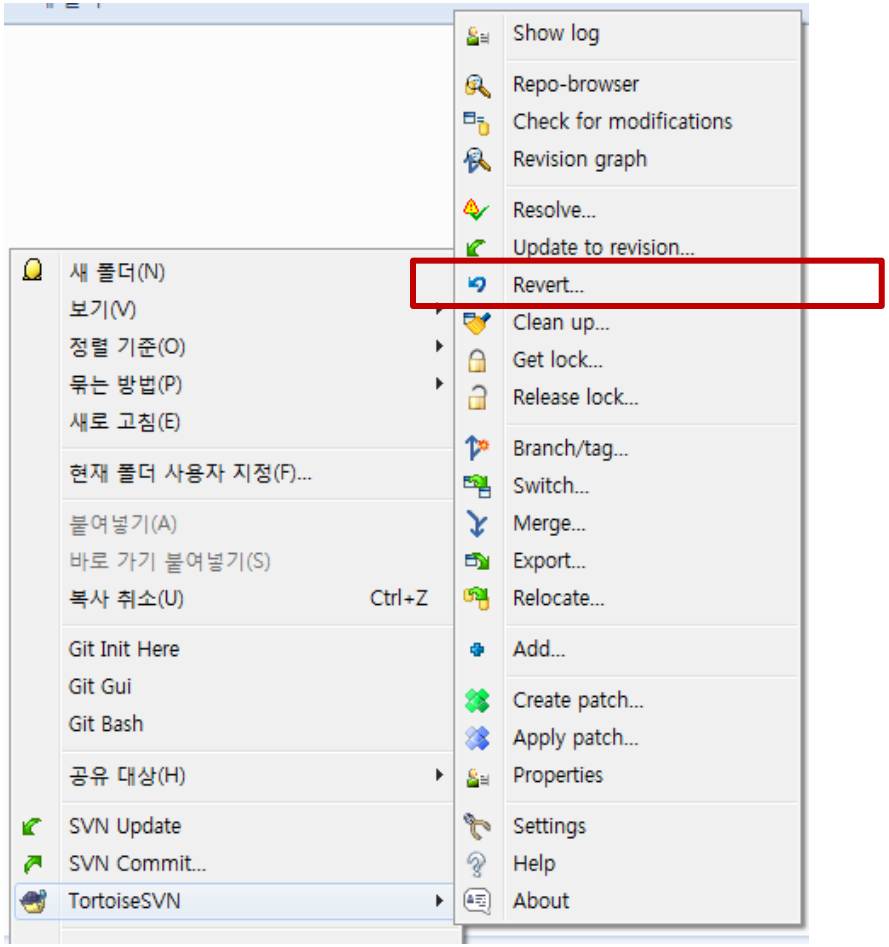
Revision별 Action 확인



3. 주요 기능

3.9 Revert (1/3)

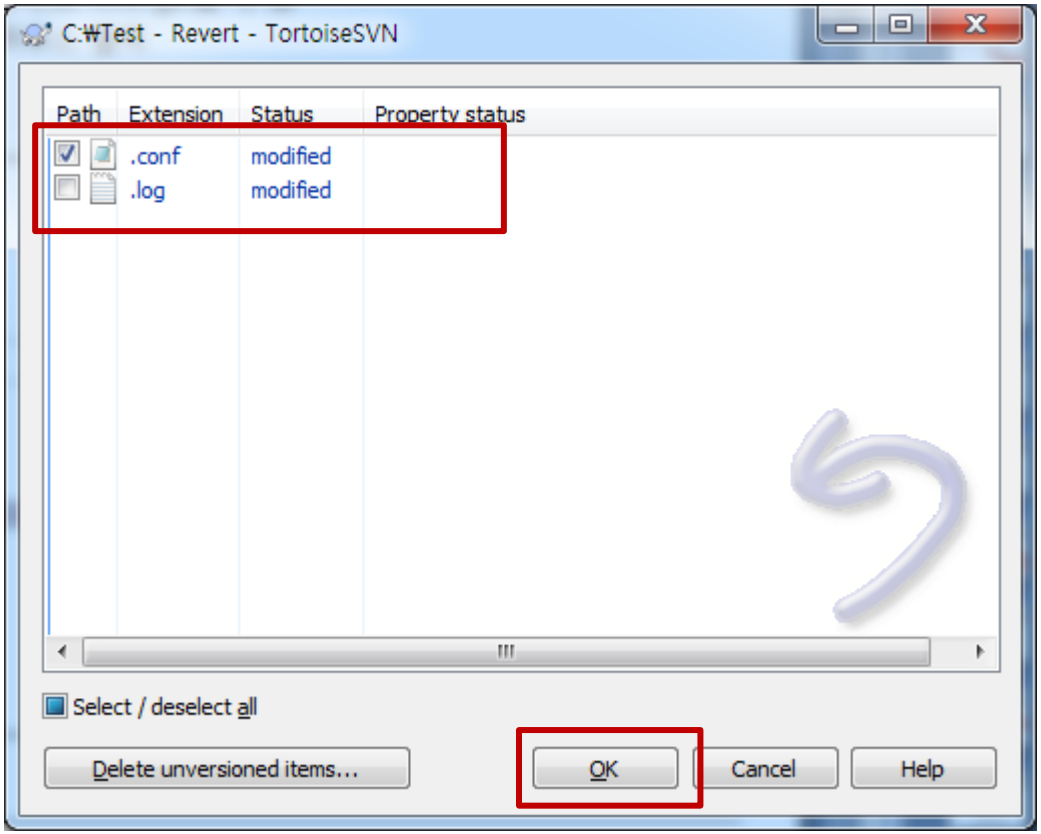
- 작업 데이터의 변경된 내용을 반영하지 않고 원본 데이터의 내용으로 돌아가기 위해 Revert 를 수행한다.
- 탐색기에서 마우스 오른쪽 버튼을 클릭하여 TortoiseSVN -> Revert 를 선택한다.



3. 주요 기능

3.9 Revert (2/3)

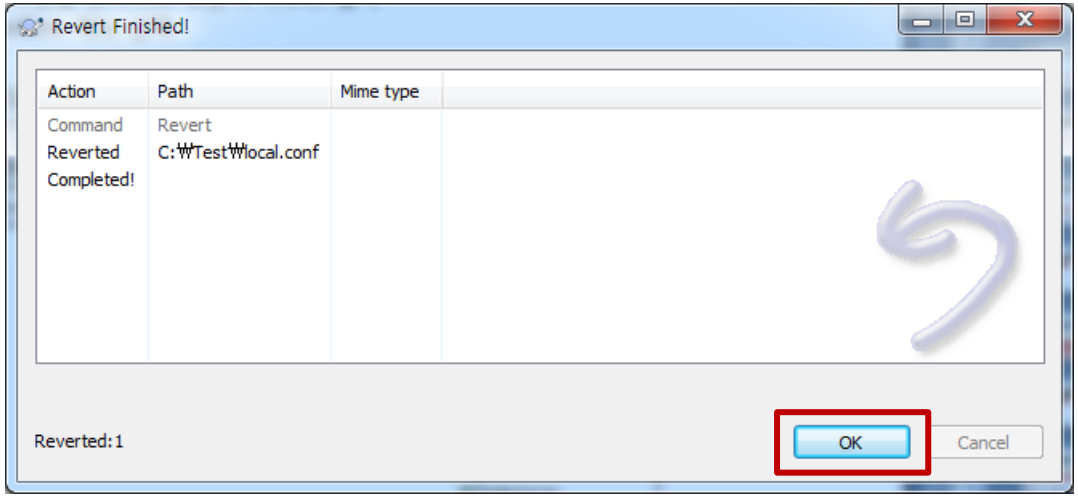
- 취소할 변경 내역을 확인하여 선택한다.
- OK 를 클릭한다.

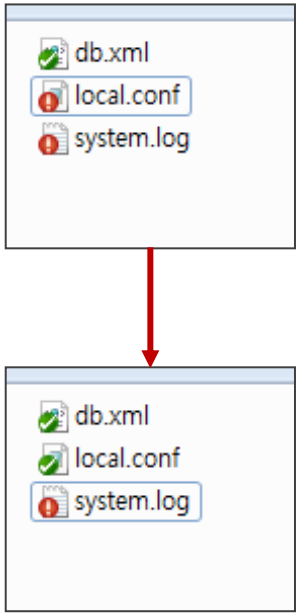


3. 주요 기능

3.9 Revert (3/3)

- Revert 확인창에서 변경 취소 내용을 확인하고 OK 를 클릭한다.
- 탐색기에서 수정된 내용이 취소되어 원본 데이터로 돌아간 것을 아이콘 모양 변화로 확인할 수 있다.





4. 활용 예제

세부 목차

4.1 예제 소개
4.2 Project 생성
4.3 Branch 생성
4.4 Branch별 형상관리

4. 활용 예제

세부 목차

- 기존 개발된 SW에서 기능이 추가된 신규 SW를 새로 개발하는 상황에서 기존소스는 별도로 유지 관리하면서 신규소스를 별도로 개발하기 위한 형상관리 환경을 만든다.

  1. Project를 생성하여 기존소스의 형상관리를 설정한다.

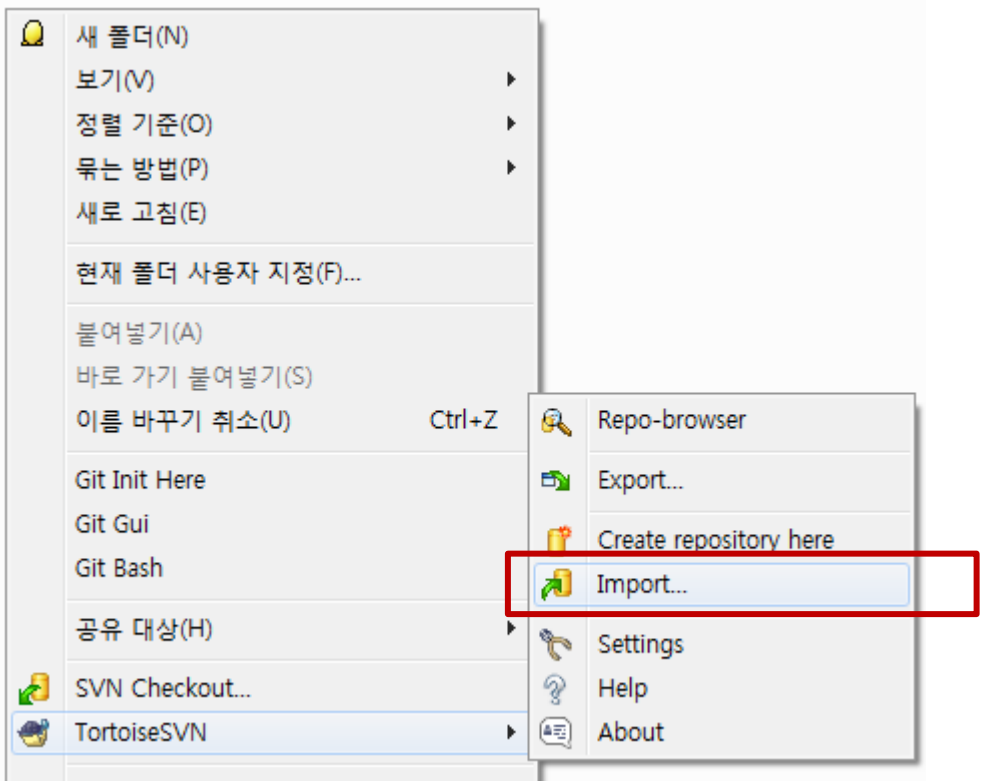
  2. Branch 생성하여 신규소스의 형상관리를 설정한다.

  3. Branch별 형상관리를 통한 소스를 수정한다.

4. 활용 예제

4.2 Project 생성 (1/8) – 기존소스 등록

- 기존소스를 SVN 서버에 등록하기 위해 Import Project를 수행한다.
- 탐색기에서 기존소스가 위치한 폴더로 이동한다.
- 해당 폴더 내에서 마우스 오른쪽 버튼을 클릭한다.
- TortoiseSVN -> Import... 를 클릭한다.

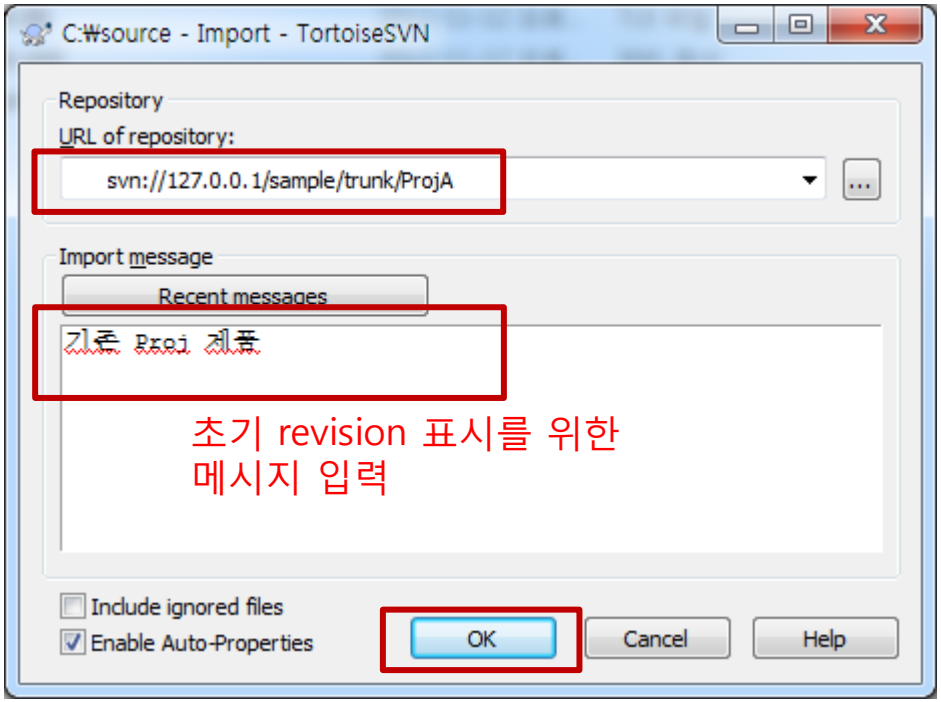


4. 활용 예제

4.2 Project 생성 (2/8) – 기존소스 등록

- 프로젝트를 생성할 Repository URL 을 입력한다.
- Import message 를 입력하고 OK 를 클릭한다.

SVN 서버 URL에 /trunk/ProjA(예시)를 추가하여 Repository URL을 입력

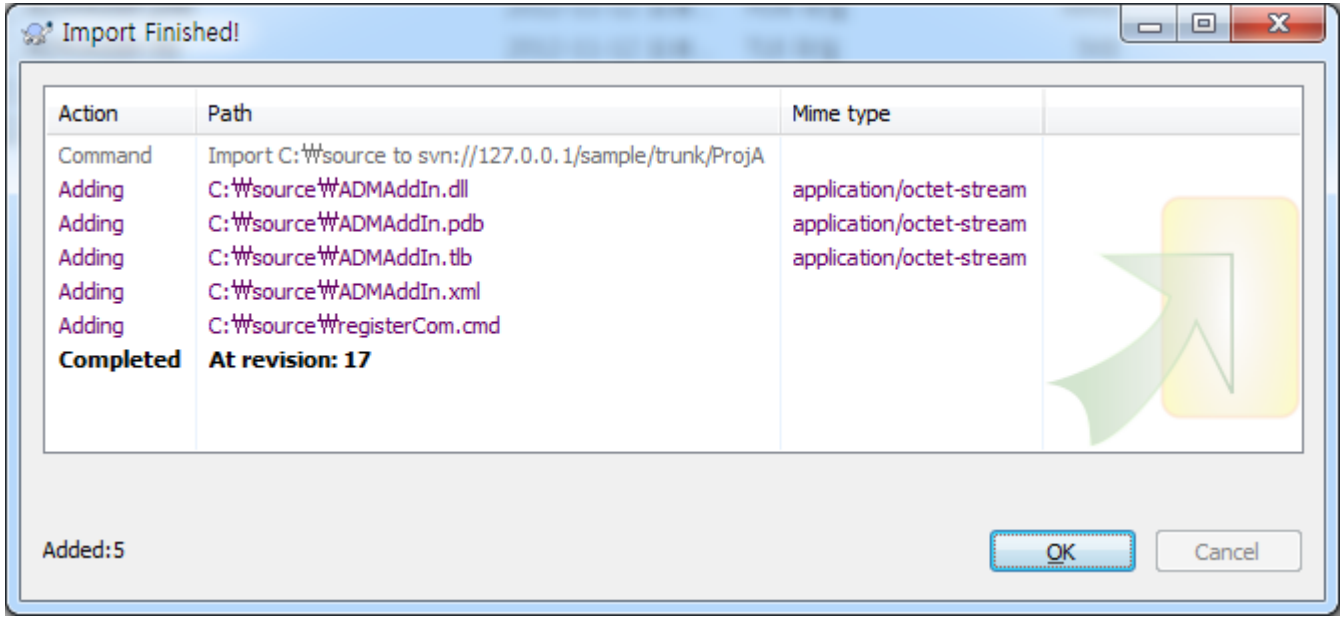


초기 revision 표시를 위한 메시지 입력

4. 활용 예제

4.2 Project 생성 (3/8) – 기존소스 등록

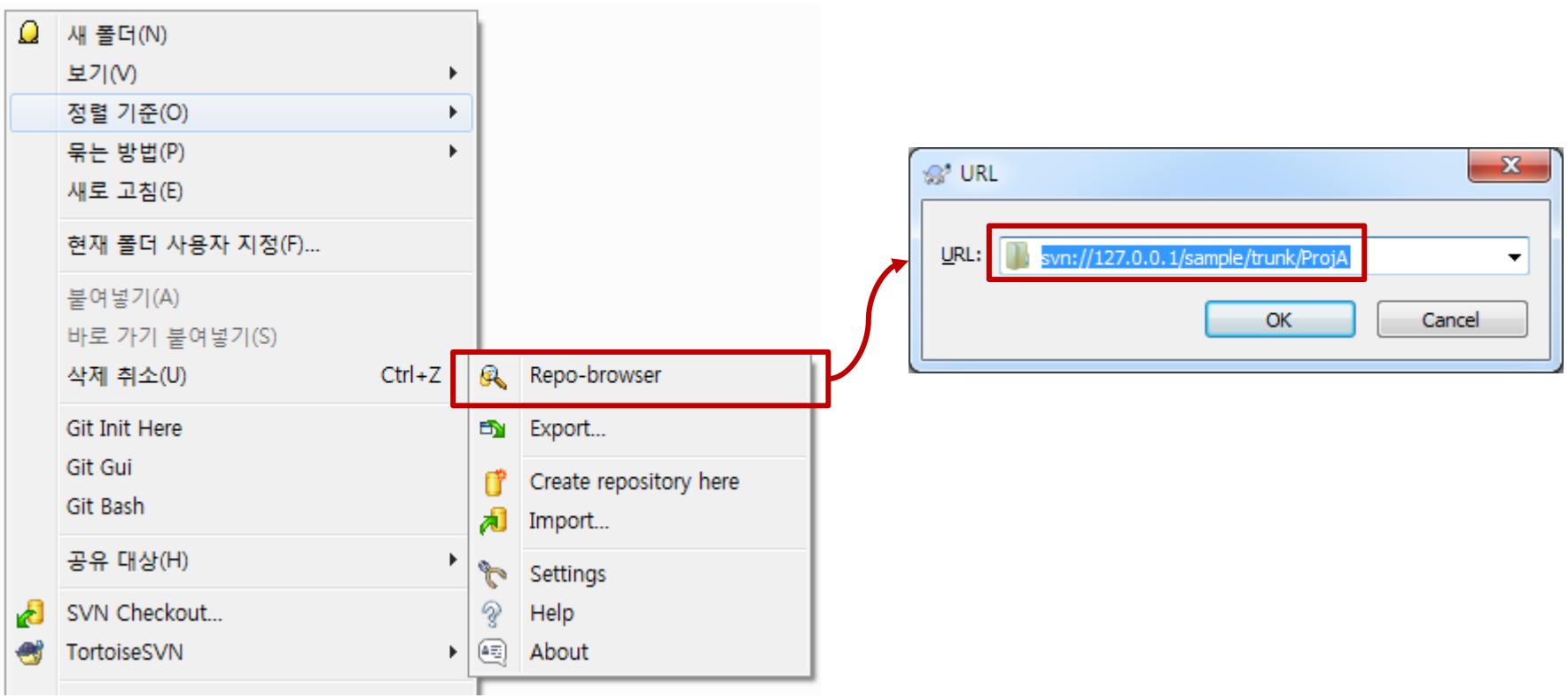
- Import 완료창의 Action 내용을 확인하고 OK 를 클릭한다.



4. 활용 예제

4.2 Project 생성 (4/8) – 생성 확인

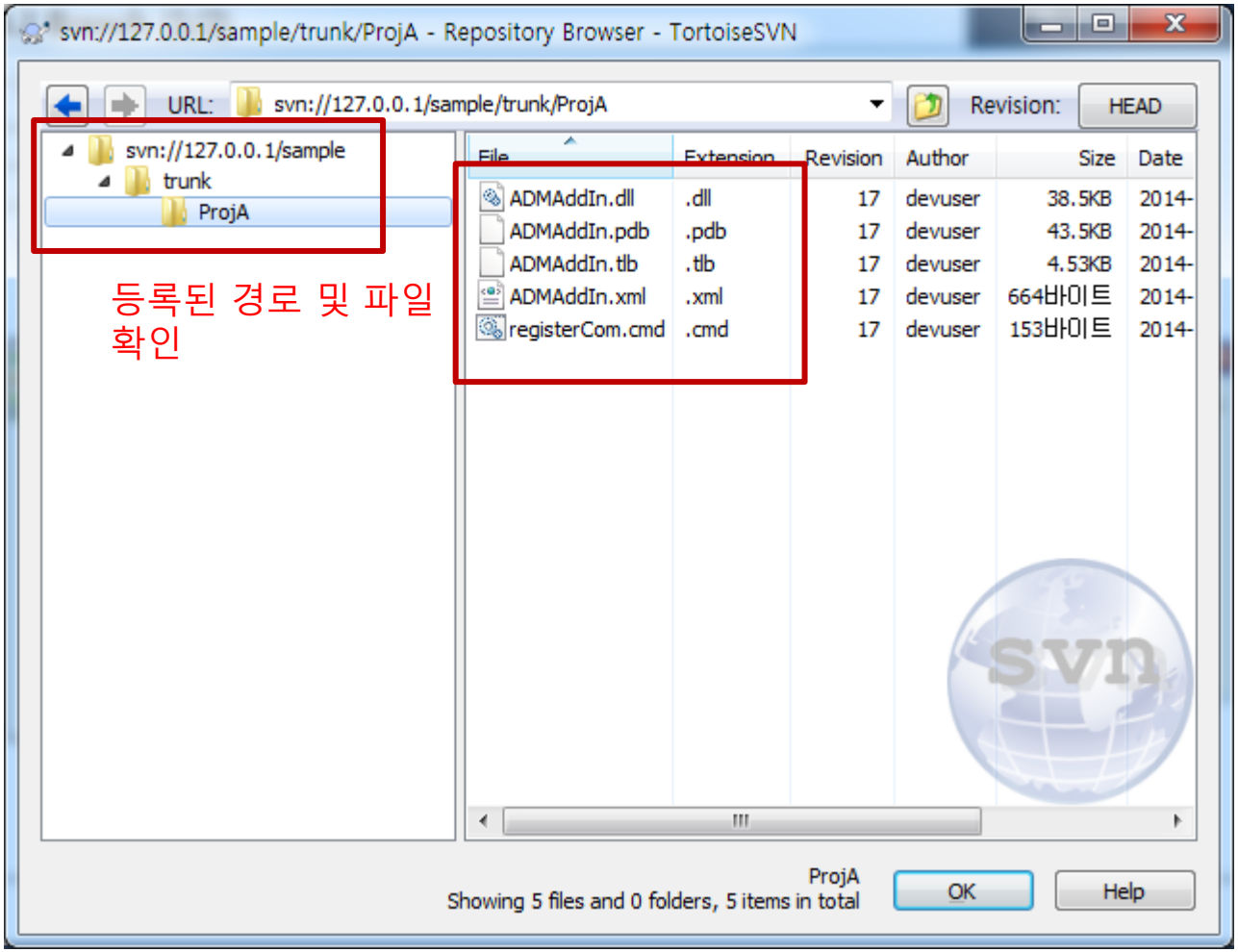
- Importing 결과 확인을 위해 Repo-browser를 수행한다.
- 탐색기에서 마우스 오른쪽 버튼을 클릭한다.
- TortoiseSVN -> Repo-Browser 를 클릭한다.
- URL 입력창에서 앞서 생성한 프로젝트의 Repository URL을 입력 후 OK 를 클릭한다.



4. 활용 예제

4.2 Project 생성 (5/8) – 생성 확인

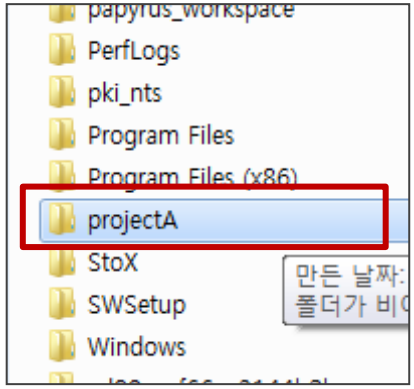
- SVN 서버에 해당 Repository 경로 및 파일이 등록된 것을 확인한다.

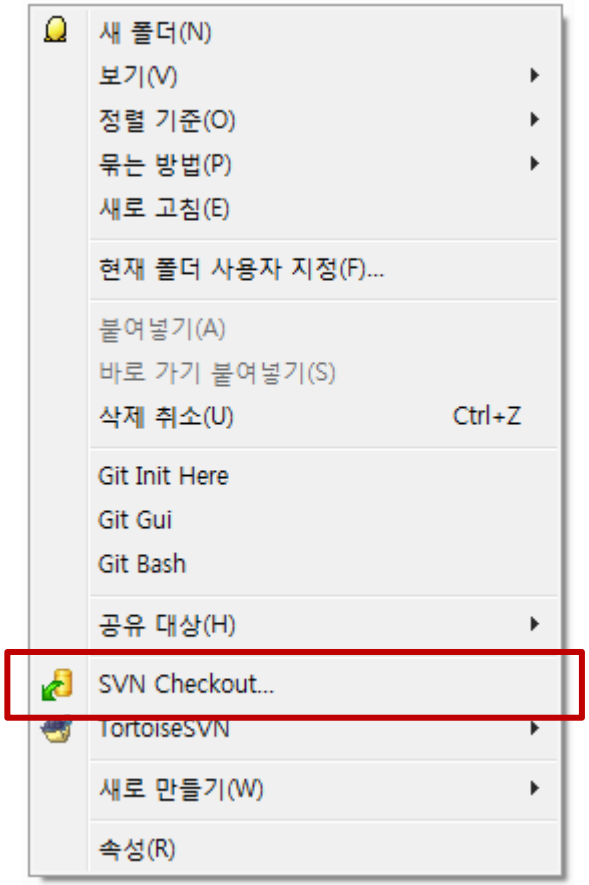


4. 활용 예제

4.2 Project 생성 (6/8) – 기존소스 Checkout

- 기존소스를 형상관리하기 위해 checkout을 수행한다.
- 기존소스를 관리할 폴더를 생성한 후 만들어진 폴더로 이동한다.
- 마우스 오른쪽 버튼을 클릭하여 SVN Checkout을 클릭한다.





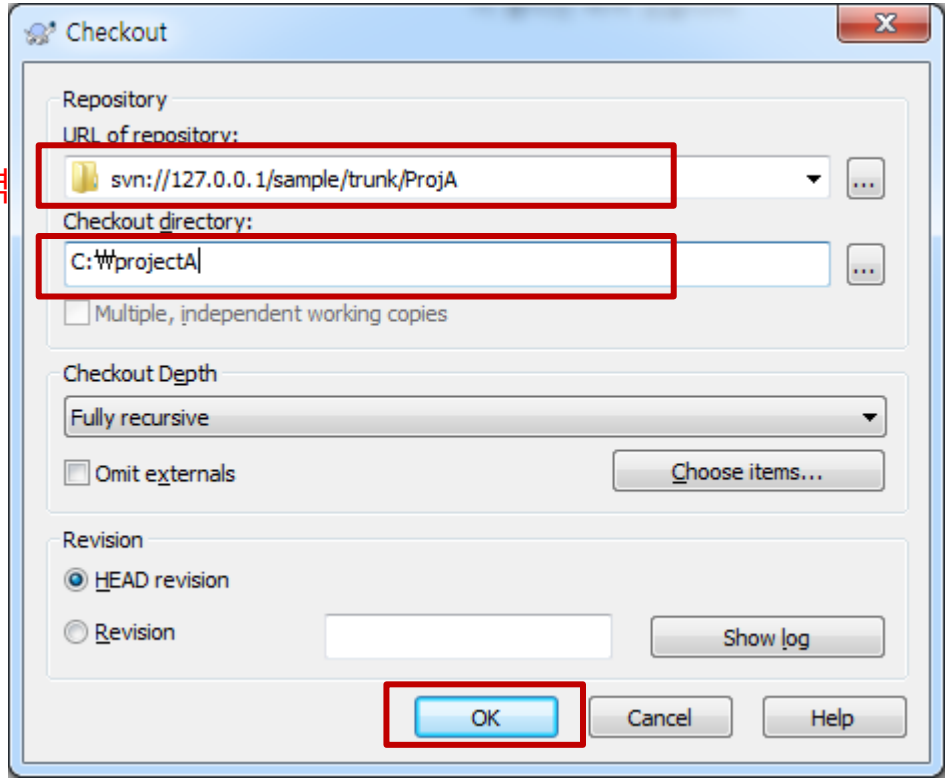
4. 활용 예제

4.2 Project 생성 (7/8) – 기존소스 Checkout

- Checkout 창에서 Repository URL을 입력한다.
- Checkout 대상 폴더를 확인한 후 OK 를 클릭한다.

기존소스를 등록한 Repository URL 입력

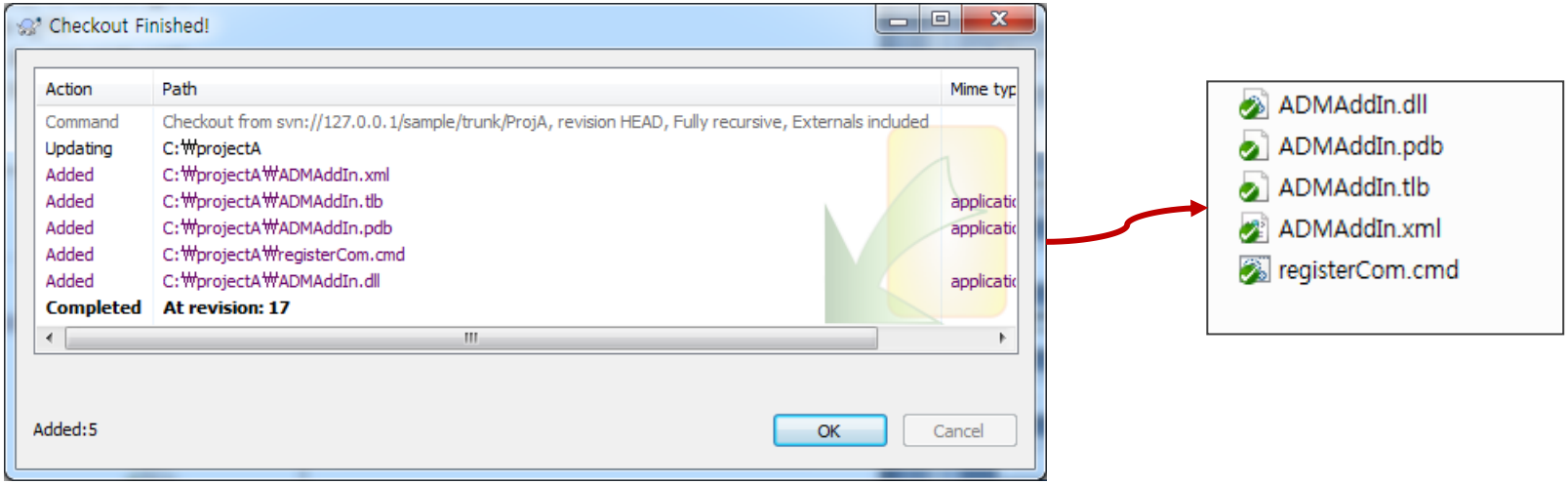
Checkout 대상 폴더 확인



4. 활용 예제

4.2 Project 생성 (8/8) – 기존소스 Checkout

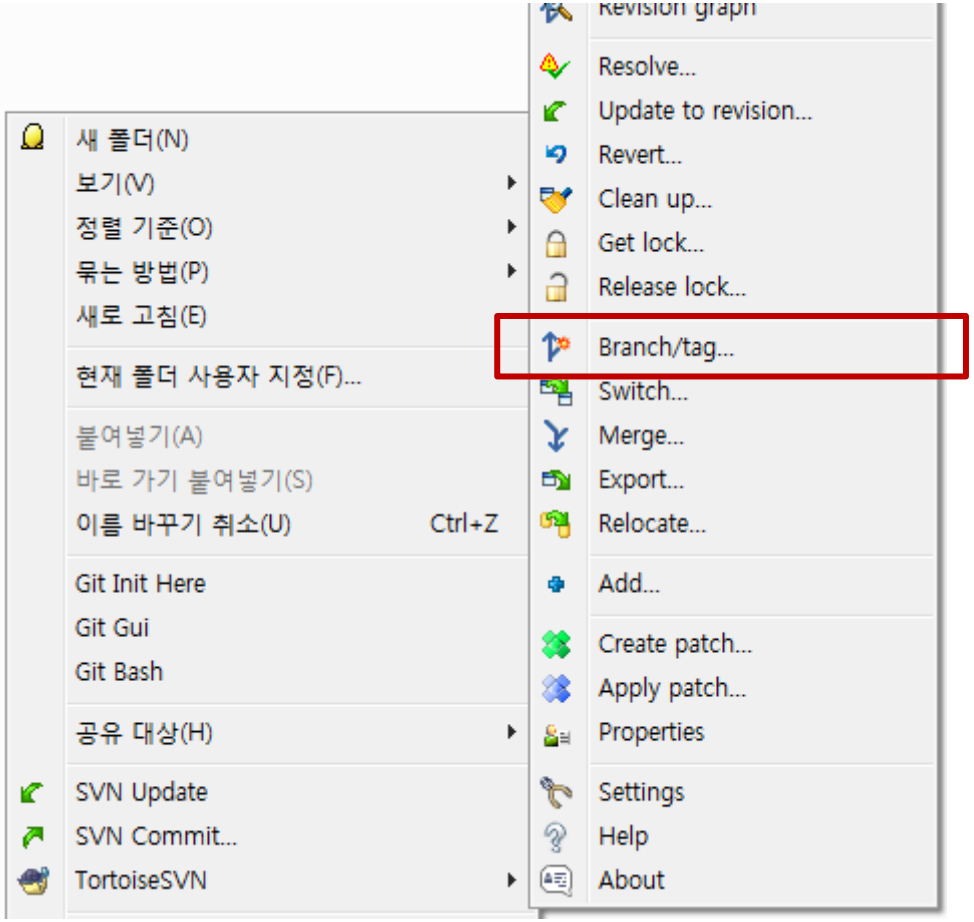
- Checkout 결과창을 확인하고 OK 를 클릭한다.
- 탐색기에 소스가 생성된 것을 확인한다.



4. 활용 예제

4.3 Branch 생성 (1/6) – 신규코드 Branch 생성

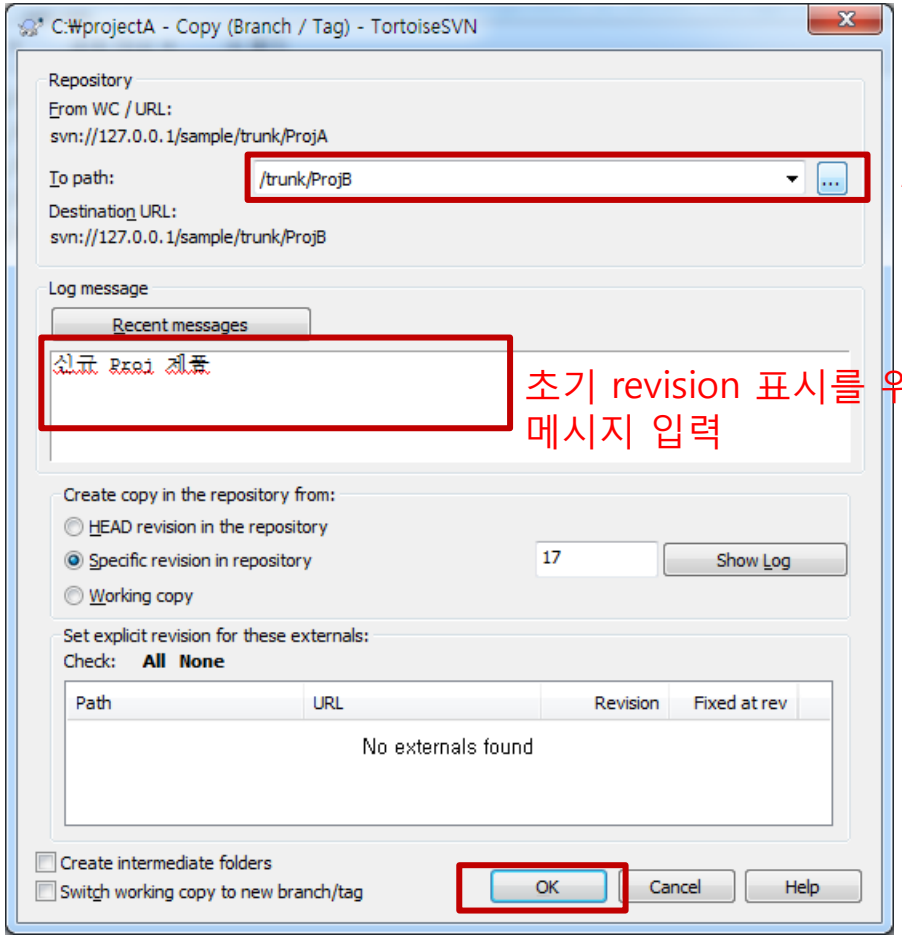
- 신규소스를 SVN 서버에 등록하기 위해 Create Branch 를 수행한다.
- 마우스 오른쪽 버튼을 클릭한다.
- TortoiseSVN -> Branch/tag... 를 클릭한다.



4. 활용 예제

4.3 Branch 생성 (2/6) – 신규코드 Branch 생성

- Branch를 생성할 Repository URL 을 입력한다.
- Log message 를 입력하고 OK 를 클릭한다.



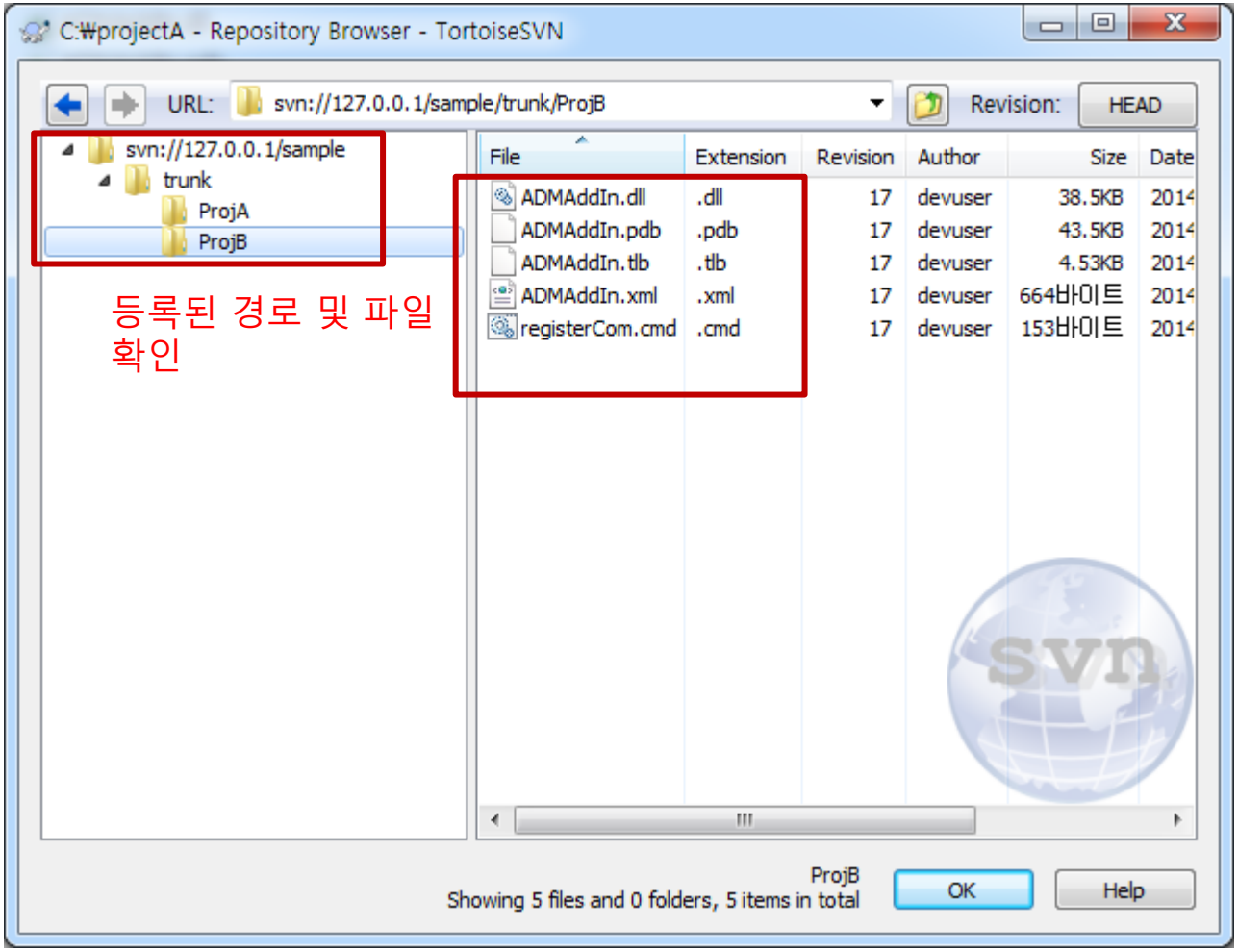
SVN 서버 URL에 /trunk/ProjB(예시)를 추가하여 Repository URL을 입력

초기 revision 표시를 위한 메시지 입력

4. 활용 예제

4.3 Branch 생성 (3/6) – 신규코드 Branch 생성

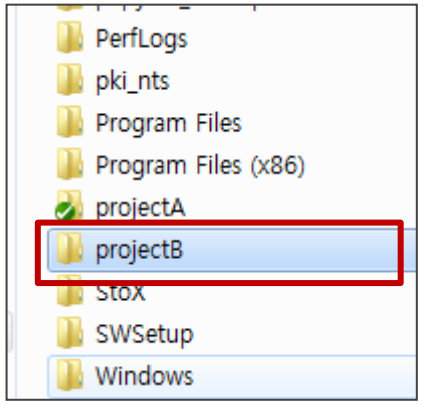
- Repository Browser를 통해 해당 Repository 경로 및 파일이 등록된 것을 확인한다.

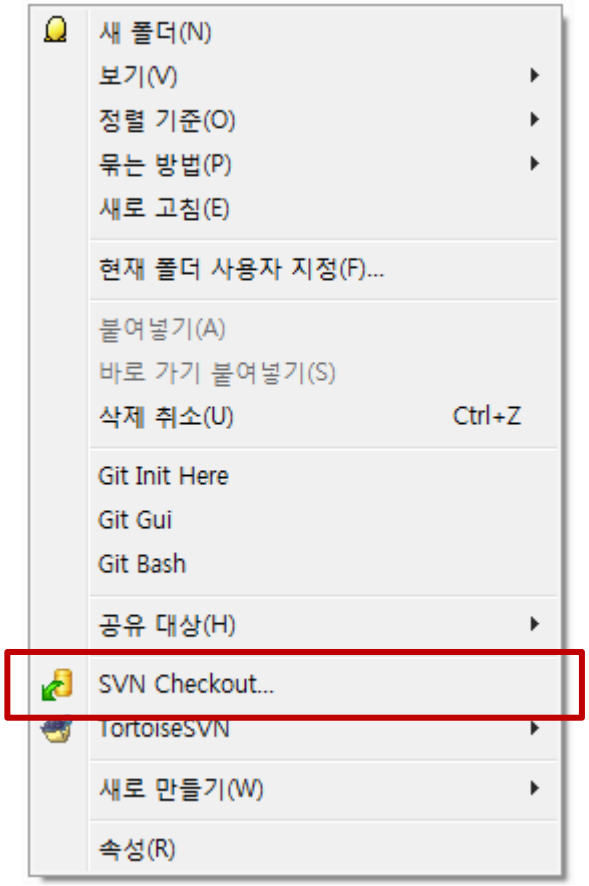


4. 활용 예제

4.3 Branch 생성 (4/6) – 신규소스 Checkout

- 신규소스를 형상관리하기 위해 checkout을 수행한다.
- 신규소스를 관리할 폴더를 생성한 후 만들어진 폴더로 이동한다.
- 마우스 오른쪽 버튼을 클릭하여 SVN Checkout을 클릭한다.





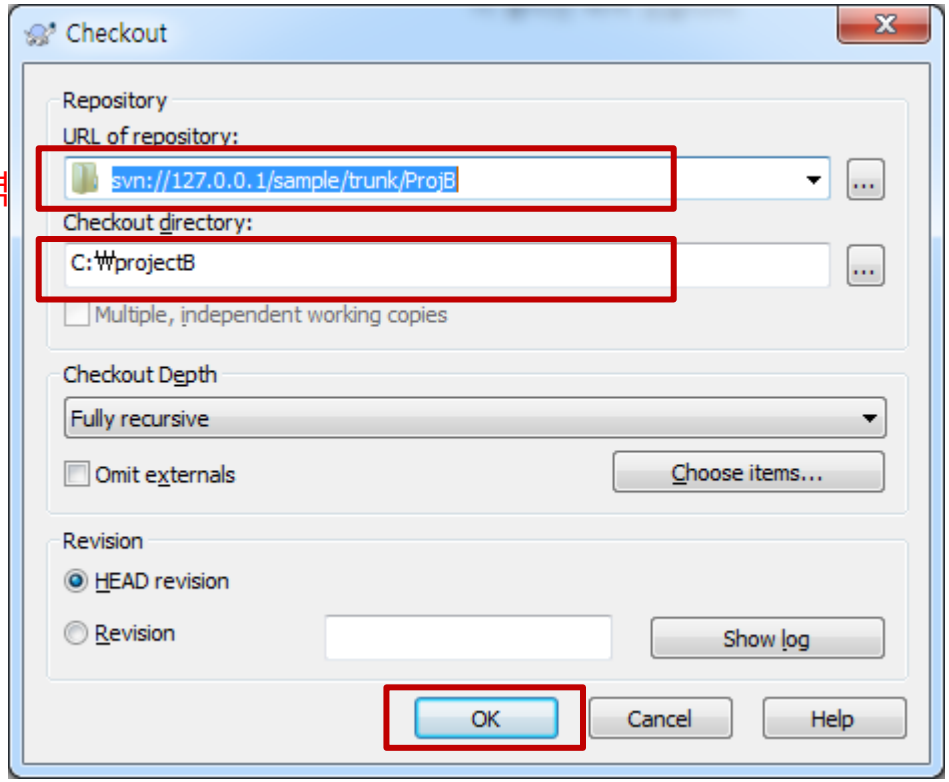
4. 활용 예제

4.3 Branch 생성 (5/6) – 신규소스 Checkout

- Checkout 창에서 Repository URL을 입력한다.
- Checkout 대상 폴더를 확인한 후 OK 를 클릭한다.

기존소스를 등록한 Repository URL 입력

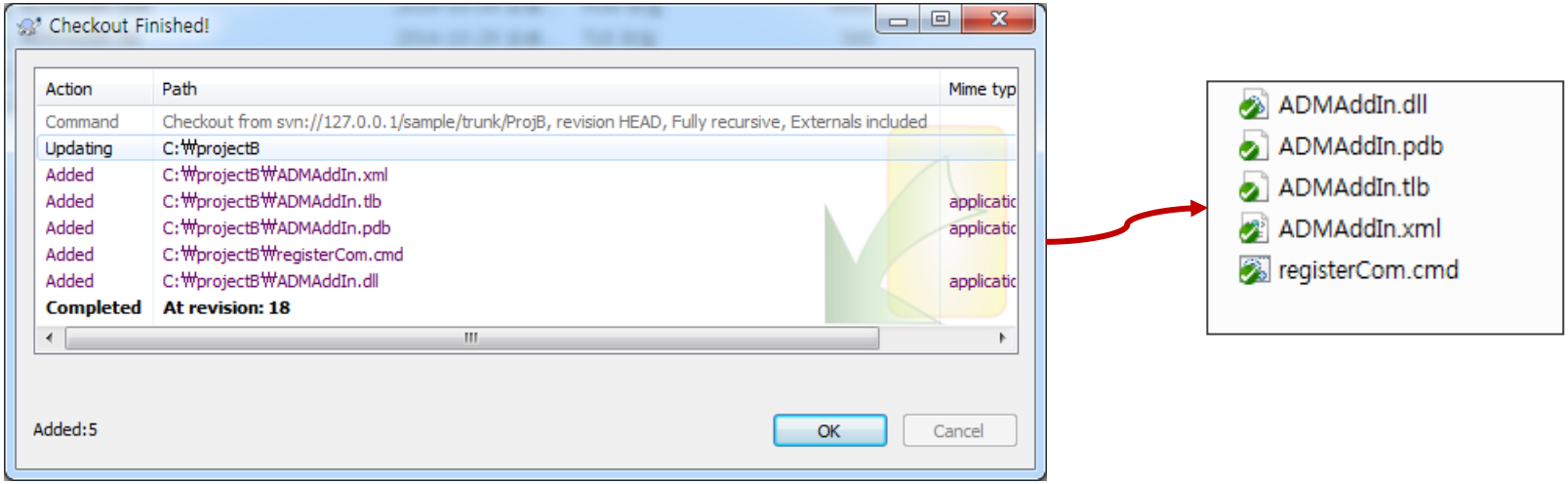
Checkout 대상 폴더 확인



4. 활용 예제

4.3 Branch 생성 (6/6) – 신규소스 Checkout

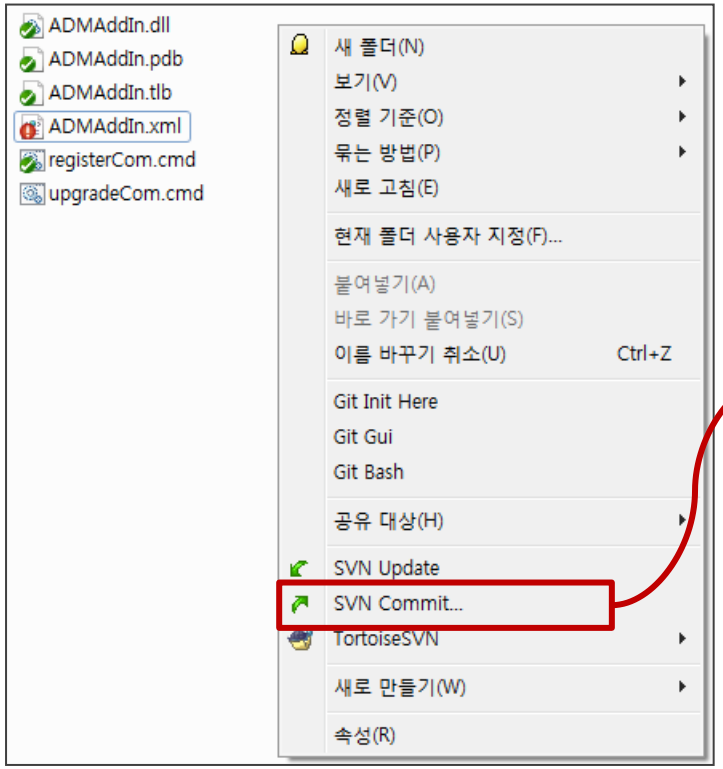
- Checkout 결과창을 확인하고 OK 를 클릭한다.
- 탐색기에 소스가 생성된 것을 확인한다.

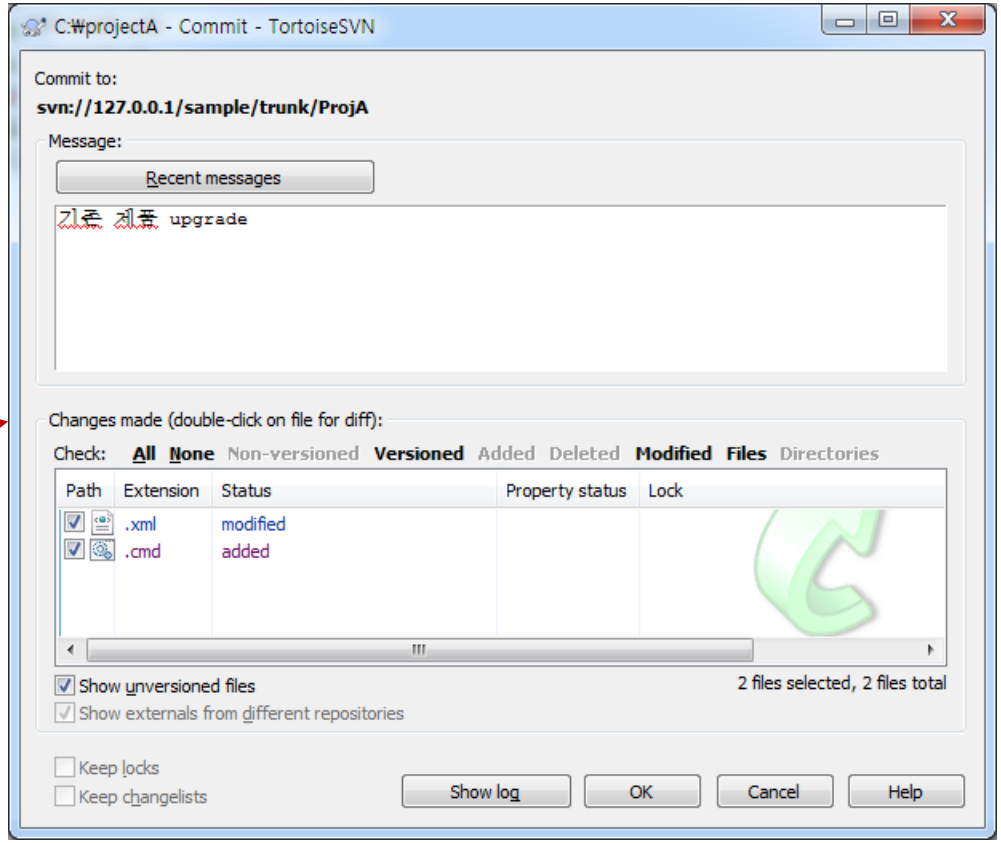


4. 활용 예제

4.4 Branch별 형상관리 (1/4) - 기존소스

- 기존소스를 관리하는 폴더(ProjectA)에서 소스를 수정 후 Commit을 수행한다.

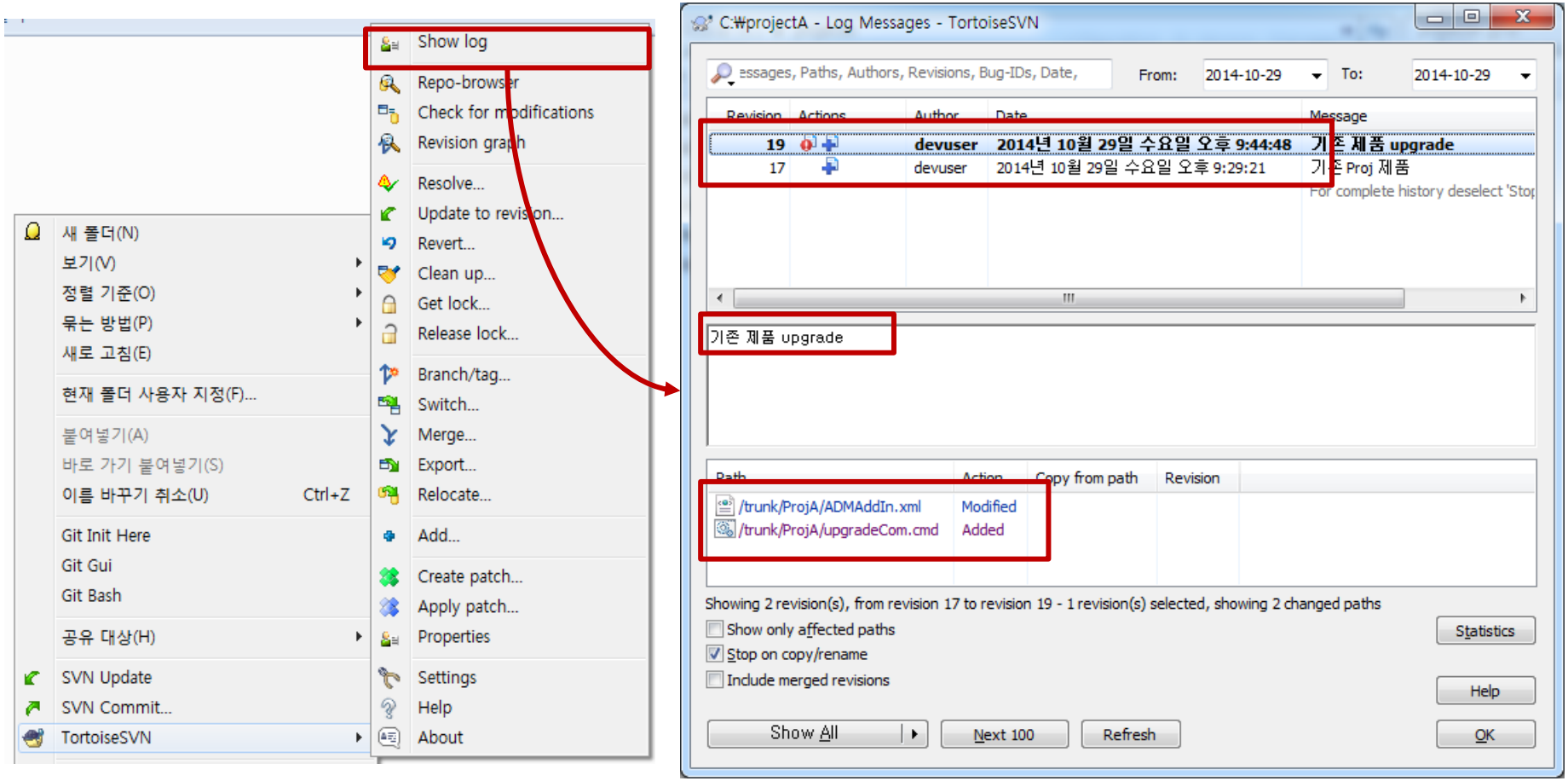




4. 활용 예제

4.4 Branch별 형상관리 (2/4) - 기존소스

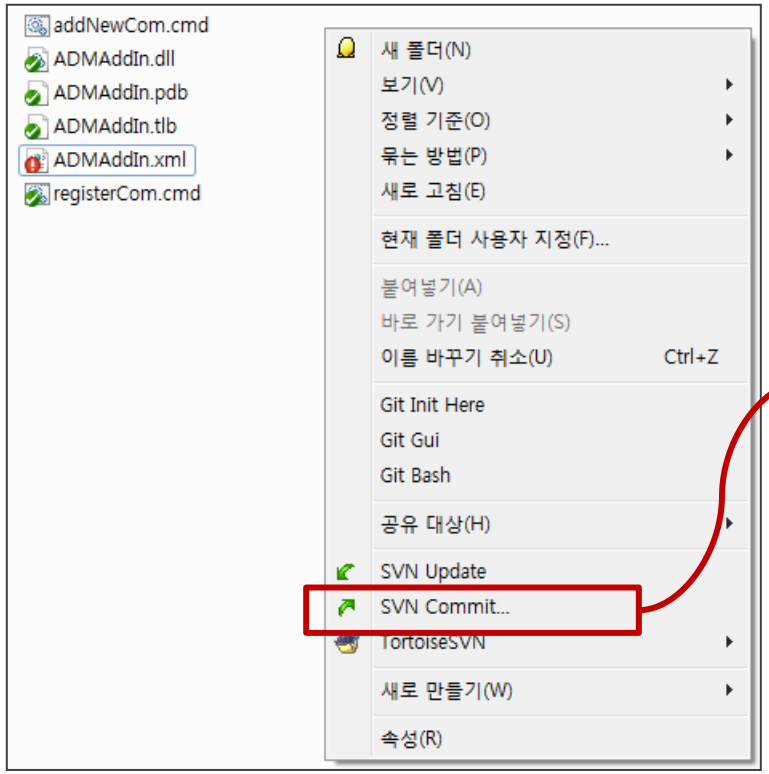
- Show log 를 실행하여 변경 내역, Revision별 Action, Message 내용을 확인하면 해당 Branch 내에서 독립적인 형상관리가 수행되고 있다.

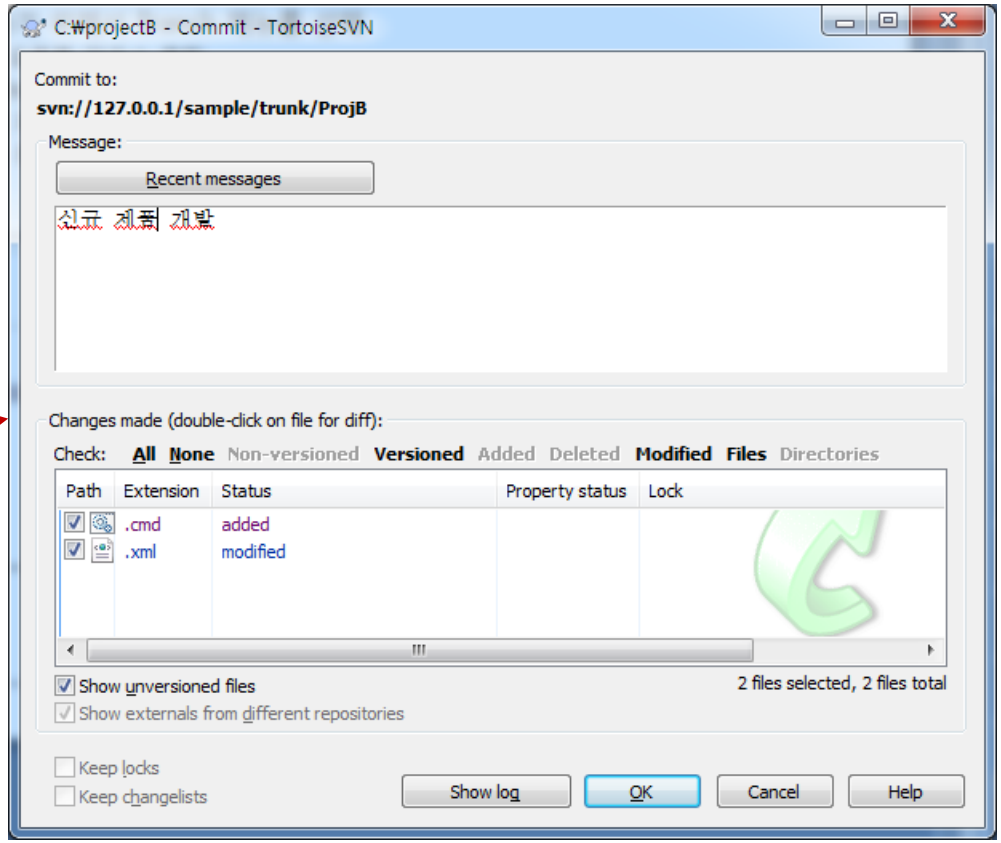


4. 활용 예제

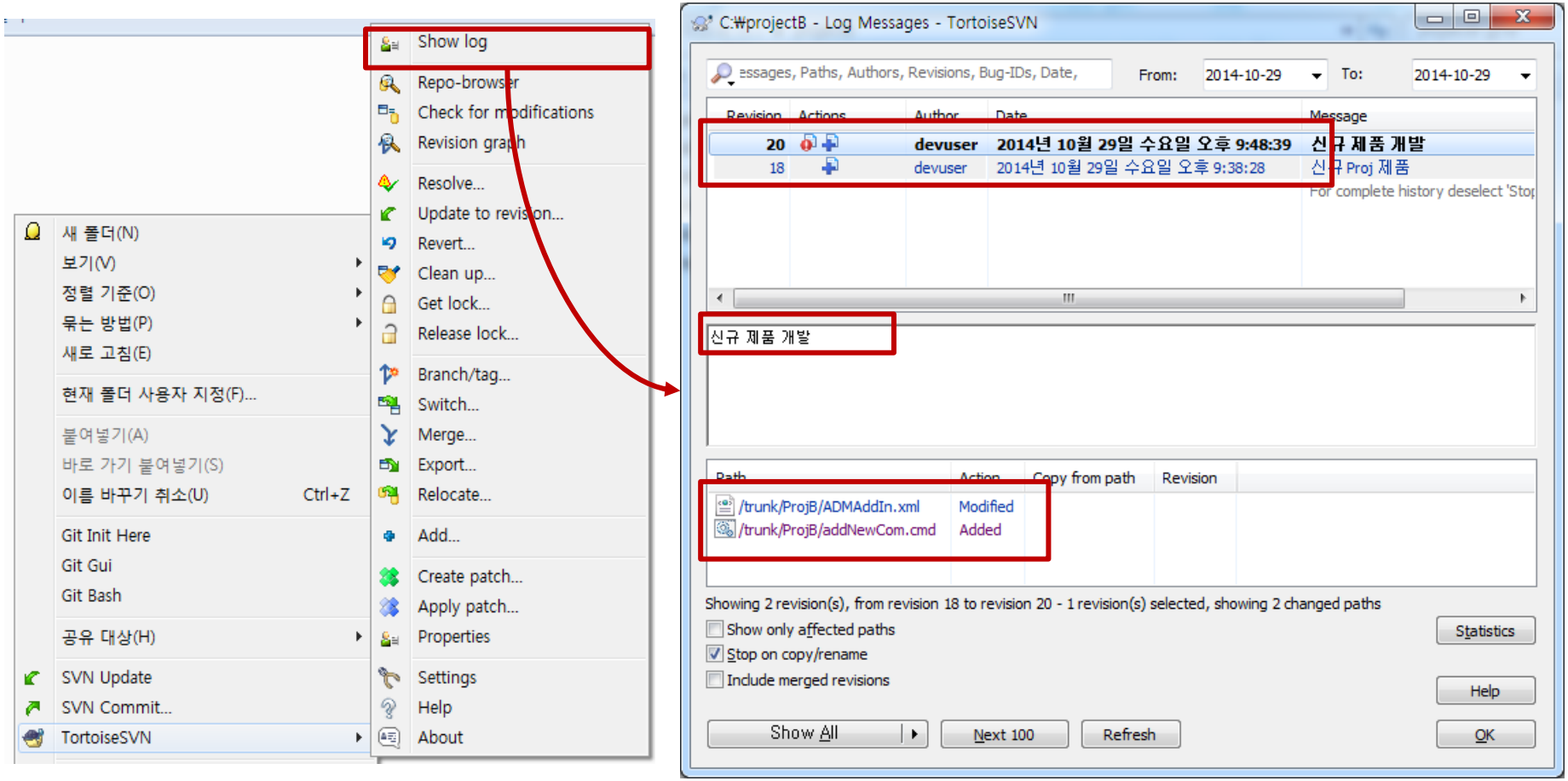
4.4 Branch별 형상관리 (3/4) - 신규소스

- 신규소스를 관리하는 폴더(ProjectB)에서 소스를 수정 후 Commit 을 수행한다.





4. 활용 예제

4.4 Branch별 형상관리 (4/4) - 신규소스

- Show log 를 실행하여 변경 내역, Revision별 Action, Message 내용을 확인하면 해당 Branch 내에서 독립적인 형상관리가 수행되고 있다.