

회사에서 많이 사용하는 소스코드 관리툴들



하이테크 과정에 대한 당부 말씀

■ 여러분은 이제 엔지니어입니다.

- 엔지니어 : 우리말로 ()입니다.
- <https://namu.wiki/w/%EA%B8%B0%EC%88%A0%EC%9E%90>
- 기술인의 분류 : 작업자, 기능인/숙련공, 기사, 기술인, 공학자 → 어디에 속하고 싶으세요

■ 엔지니어의 자세

- 책만 보면서 공부하는 습관을 탈피하자.
 - 대학때까지의 습관으로 책을 정독하면서 지식을 얻고자 하는 경우가 있음.
 - 모든 지식은 실습을 통해 체득하면서 얻도록 하자.
- 항상 궁금해 하자.
 - 의문이 없으면 고민이 없고 고민이 없으면 발전이 없다.
- 모르는 것을 물어보는 습관, 아는 것을 가르쳐주는 습관을 가지자.
- 시간 투자와 반복을 통해 기술을 體化 하자.
- 문서작업을 습관화하자.
 - 조각지식들을 모아 체계적인 파일 형태로 가지고 있자.
- 발표를 잘하는 엔지니어가 인정 받는다.
- 버전관리툴, 소스 코드 분석툴과 같은 보조툴을 잘 사용하자.
 - 기억력의 한계 극복과 협업을 위해서는 버전 관리툴 사용이 필수
- 인터넷에서 자료를 잘 찾는 엔지니어가 인정 받는다.(know where)

체화³ (體化) [체화]  

[명사]

1. 물체로 변화함. 또는 물체로 변화하게 함.
2. 생각, 사상, 이론 따위가 몸에 배어서 자기 것이 됨.

소스 코드 관리툴들

■ 소스 코드란?

- 개발자가 직접 편집 수정하는 텍스트 형식의 파일들
- 예) Healthcare 프로젝트
 - C파일과 H파일들로 이루어짐

이름	수정한 날짜
Dc_Motor.c	2015-01-23 오후.
Dc_Motor.h	2015-01-23 오전.
Health_Care.apr	2015-01-24 오후.
health_care.aws	2018-07-16 오후.
Key.c	2015-01-23 오후.
Key.h	2015-01-23 오후.
Lcd.c	2016-05-18 오후.
Lcd.h	2016-05-24 오후.
Led.c	2015-01-23 오전.
Led.h	2015-01-23 오전.
Main.c	2015-12-15 오전.
Var.c	2015-01-23 오전.
Var.h	2015-01-22 오후.

■ 소스 코드 관리 툴의 필요성

- 소스코드의 양이 많음
- 소스의 양이 적더라도 지속적인 수정으로 기억의 한계에 부딪힘
 - 일주일 작업했는데 문제가 다른 데서 발견되어 일주일 전 상태로 돌리고 싶다.
- 여러 개발자가 동시에 같은 소스나 프로젝트를 사용
 - Main.c 파일을 개발자 30명이 수정하려고 한다.

소스 코드 관리툴들

■ 관리툴의 종류

- 소스 코드 버전관리 툴
- 소스 코드 분석 툴
- 소스 코드 비교 툴

■ 회사에서 많이 사용하는 관리툴

- 소스 코드 버전 관리 툴
 - Subversion, 최근에는 git을 많이 사용하는 추세
- 소스 코드 분석 툴
 - 윈도우 환경에서는 주로 Source Insight를 많이 사용
- 소스 코드 비교 툴
 - Beyond compare, winmerge, araxis등

버전 관리 프로그램

■ 대표적인 버전관리 프로그램

■ CVS

- Concurrent Versions System, GNU에서 개발, 배포, 현재는 Subversion으로 대체됨

■ Subversion

- SVN이라고 줄여서 부르기도 함.
- 서버-클라이언트 모델
- http, https, svn, svn+ssh의 규약으로 통신
- 대표적인 서버 프로그램: visual SVN (<https://www.visualsvn.com/>)
- 대표적인 클라이언트 프로그램 : tortoise SVN (<https://tortoisesvn.net/>)

■ GIT

- "기트" 또는 "깃"이라고 읽음
- <https://git-scm.com/>
- 리눅스의 창시자 리누스 토발즈가 개발함
- 서버-클라이언트 모델을 사용하지 않고 분산관리 시스템개념을 도입
- 안드로이드 소스코드등을 git로 관리 배포함 (<http://source.android.com/source/index.html>)

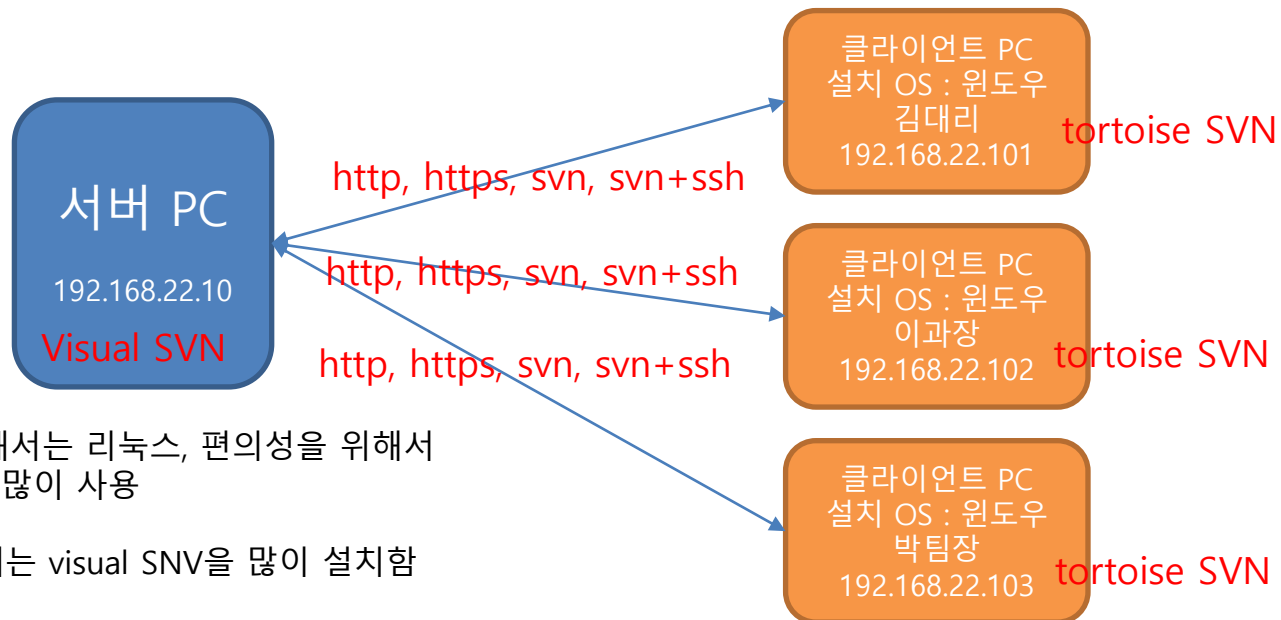
■ Github

- Git 을 사용하는 프로젝트를 지원하는 웹호스팅 서비스
- <https://ko.wikipedia.org/wiki/%EA%B9%83%ED%97%88%EB%B8%8C>
- 최근에 많이 사용되며 수많은 공개 무료 소스를 얻을 수 있는 창구

버전 관리 프로그램

■ Subversion

- 최근 기업에서 git을 많이 사용하는 추세이나 상대적으로 사용이 너무 어려워 처음 배울때는 subversion이 유리함.
- github는 직관적으로 소스를 다운로드 받을 수 있어 굳이 사용법을 배울 필요 없음.
- Subversion은 서버, 클라이언트 모델
 - 실습에서 서버는 visual SVN을 클라이언트로는 tortoise SVN을 사용할 예정
- 실제 구현의 예



안정성을 위해서는 리눅스, 편의성을 위해서는 윈도우를 많이 사용

윈도우PC일때는 visual SNV을 많이 설치함

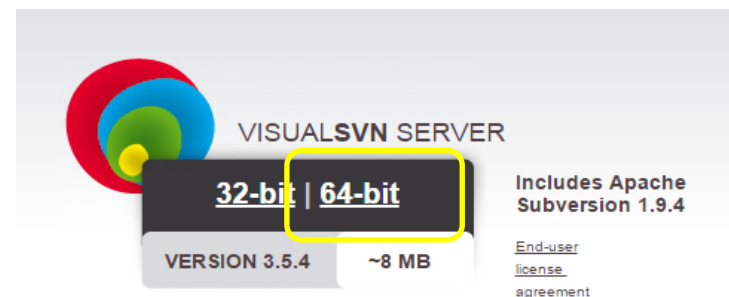
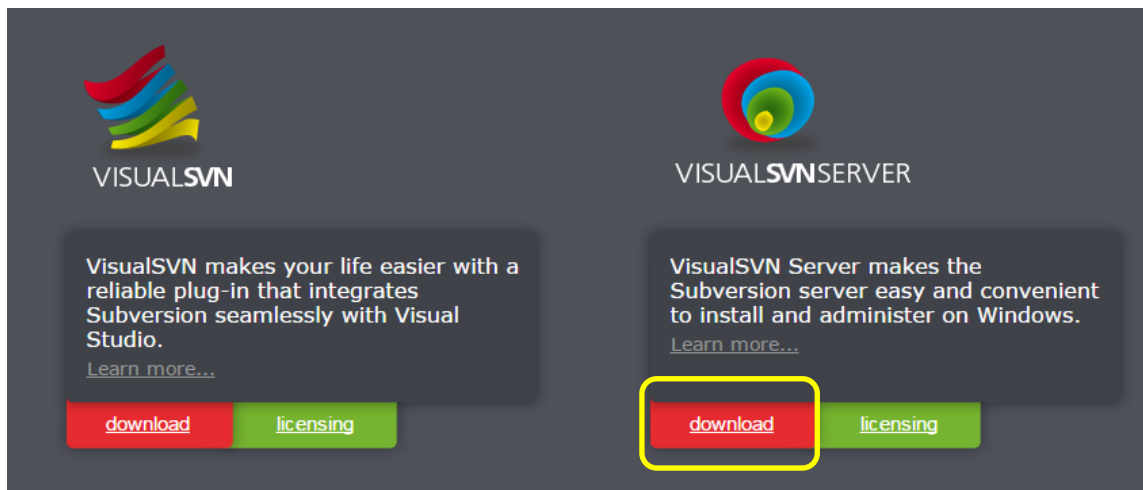
Visual Subversion

■ Visual SVN

■ 서버 역할

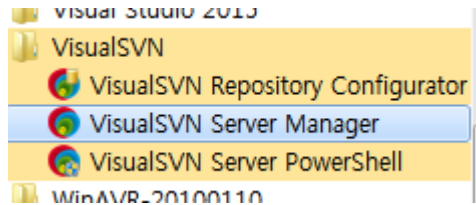
- 회사에서는 팀당 하나만 설치하고 모든 팀 구성원이 그 서버에 접속해서 사용하지만 실습을 위해 각자의 PC에 하나씩 설치

- <https://www.visualsvn.com/>

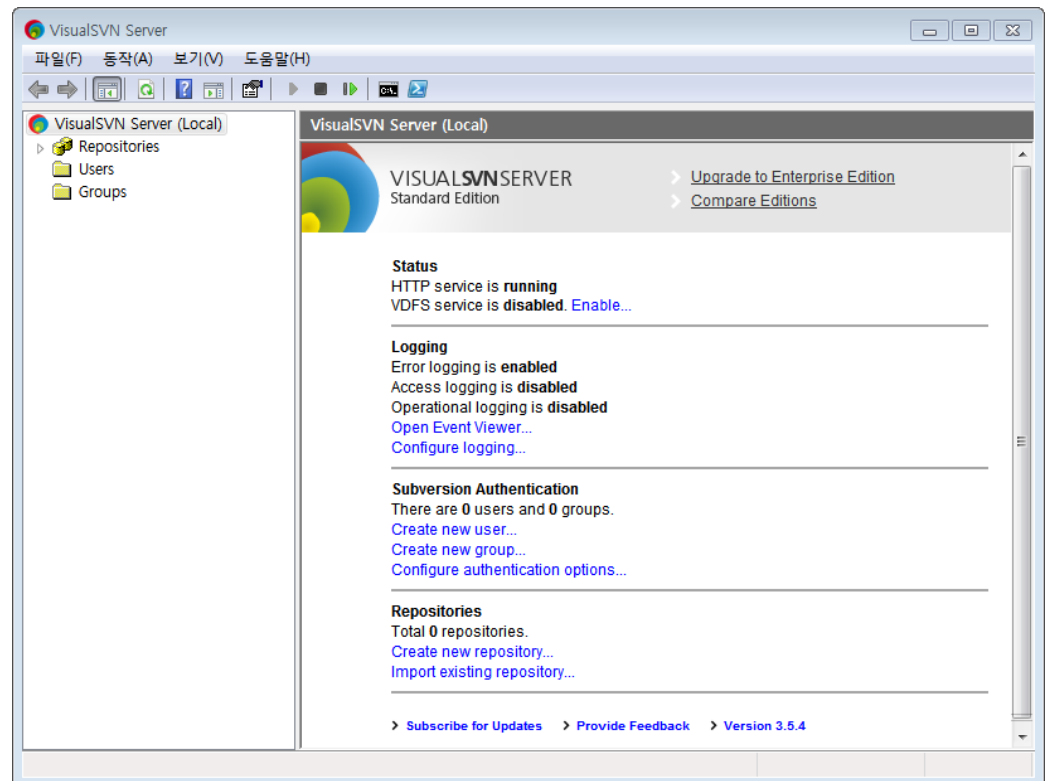


Visual Subversion

- VisualSVN-Server-x.x.x-x64.msi 설치
- 모두 "Next"로 설치



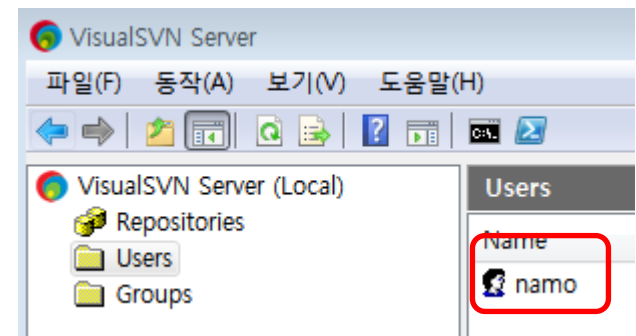
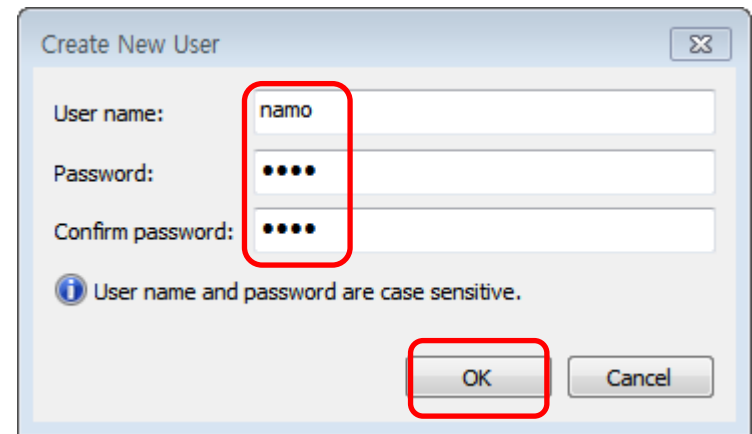
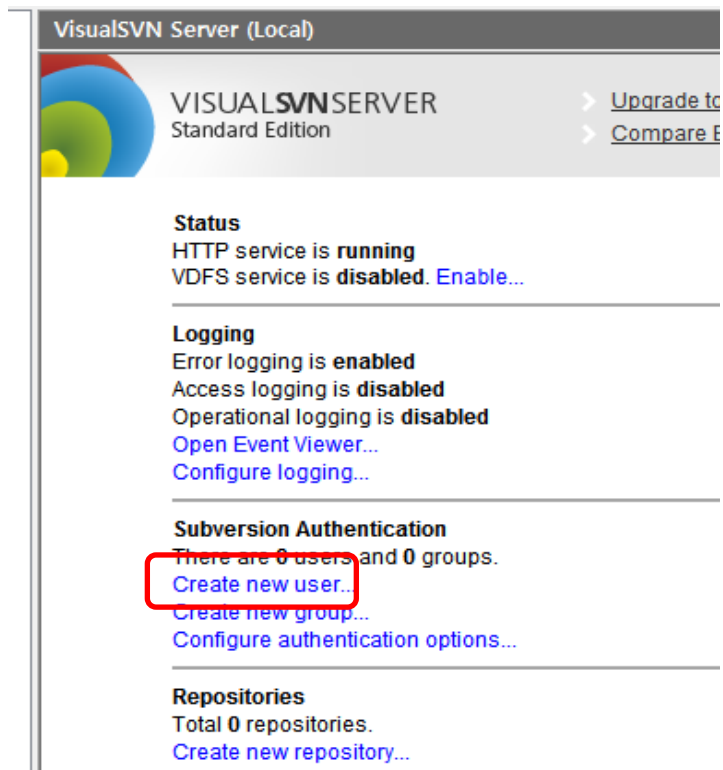
- Visual Server Manager실행



Visual Subversion

■ 사용자 추가

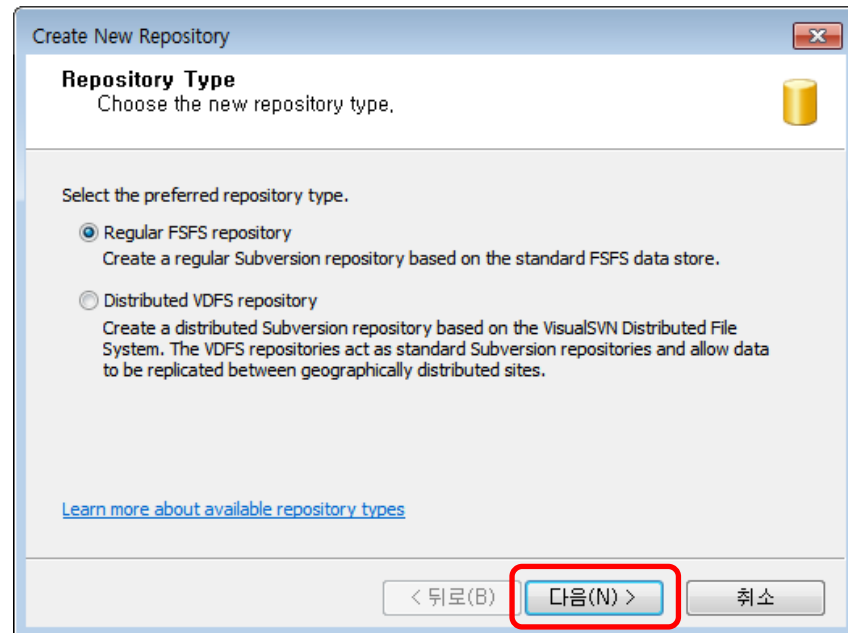
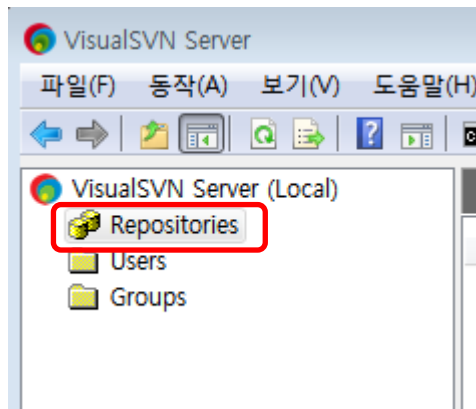
- 서버에 접속해 코드를 사용할 사용자들 추가
- Visual Server Manager의 Create new user...실행
- 일단 자신의 NAS ID를 추가해보자



Visual Subversion

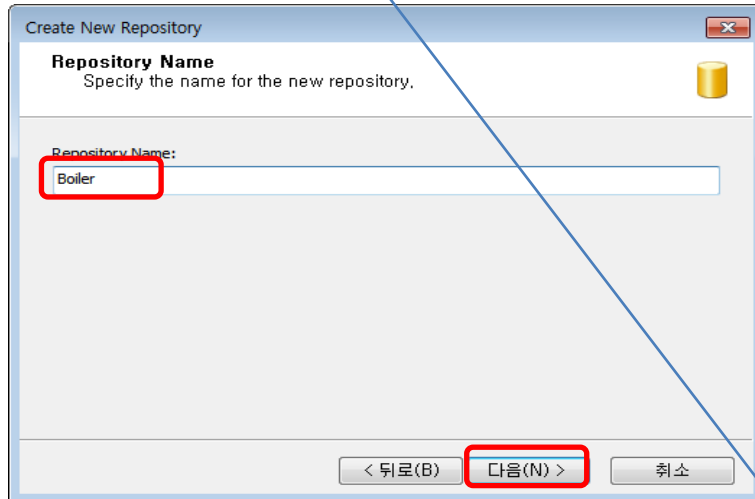
■ Repositories추가

- Repositories = 저장소
- 소스코드를 서버에 저장하는데 이 저장소를 Repository라고 함
- Boiler 소스(AVR 소스코드)를 저장소에 추가해 보자.
 - 차후에 Boiler.zip 에 있는 소스 코드들을 이 저장소에 추가할 예정
 - Repositories에 마우스 오른쪽 버튼 클릭하여 "Create New Repositories"선택



Visual Subversion

- 저장소 이름 추가
- Repository URL에 주목하자

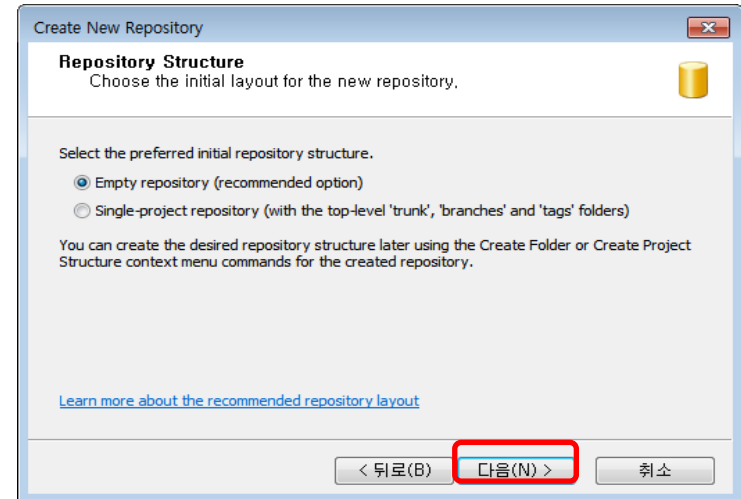


Create New Repository

Repository Name
Specify the name for the new repository.

Repository Name:
Boiler

< 뒤로(B) 다음(N) > 취소



Create New Repository

Repository Structure
Choose the initial layout for the new repository.

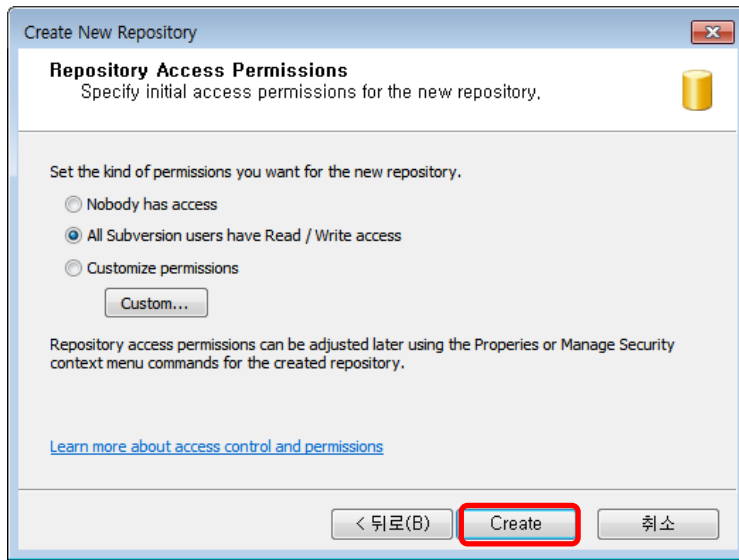
Select the preferred initial repository structure.

☒ Empty repository (recommended option)
☐ Single-project repository (with the top-level 'trunk', 'branches' and 'tags' folders)

You can create the desired repository structure later using the Create Folder or Create Project Structure context menu commands for the created repository.

[Learn more about the recommended repository layout](#)

< 뒤로(B) 다음(N) > 취소



Create New Repository

Repository Access Permissions
Specify initial access permissions for the new repository.

Set the kind of permissions you want for the new repository.

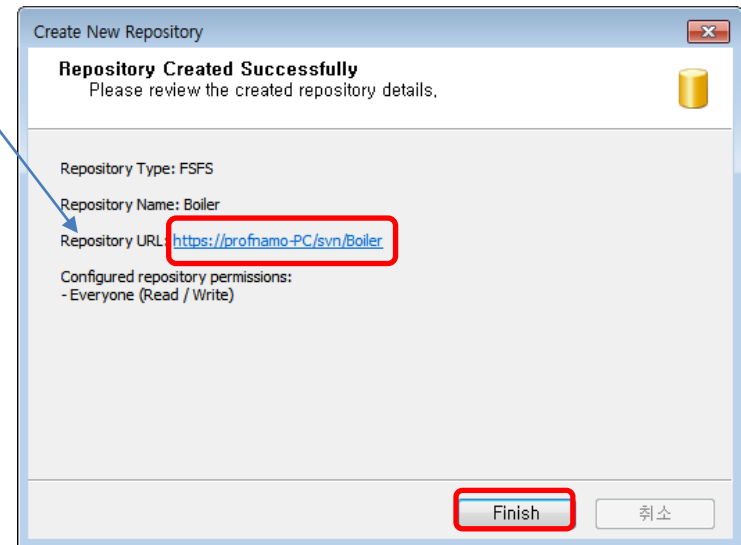
☐ Nobody has access
☒ All Subversion users have Read / Write access
☐ Customize permissions

Custom...

Repository access permissions can be adjusted later using the Properties or Manage Security context menu commands for the created repository.

[Learn more about access control and permissions](#)

< 뒤로(B) Create 취소



Create New Repository

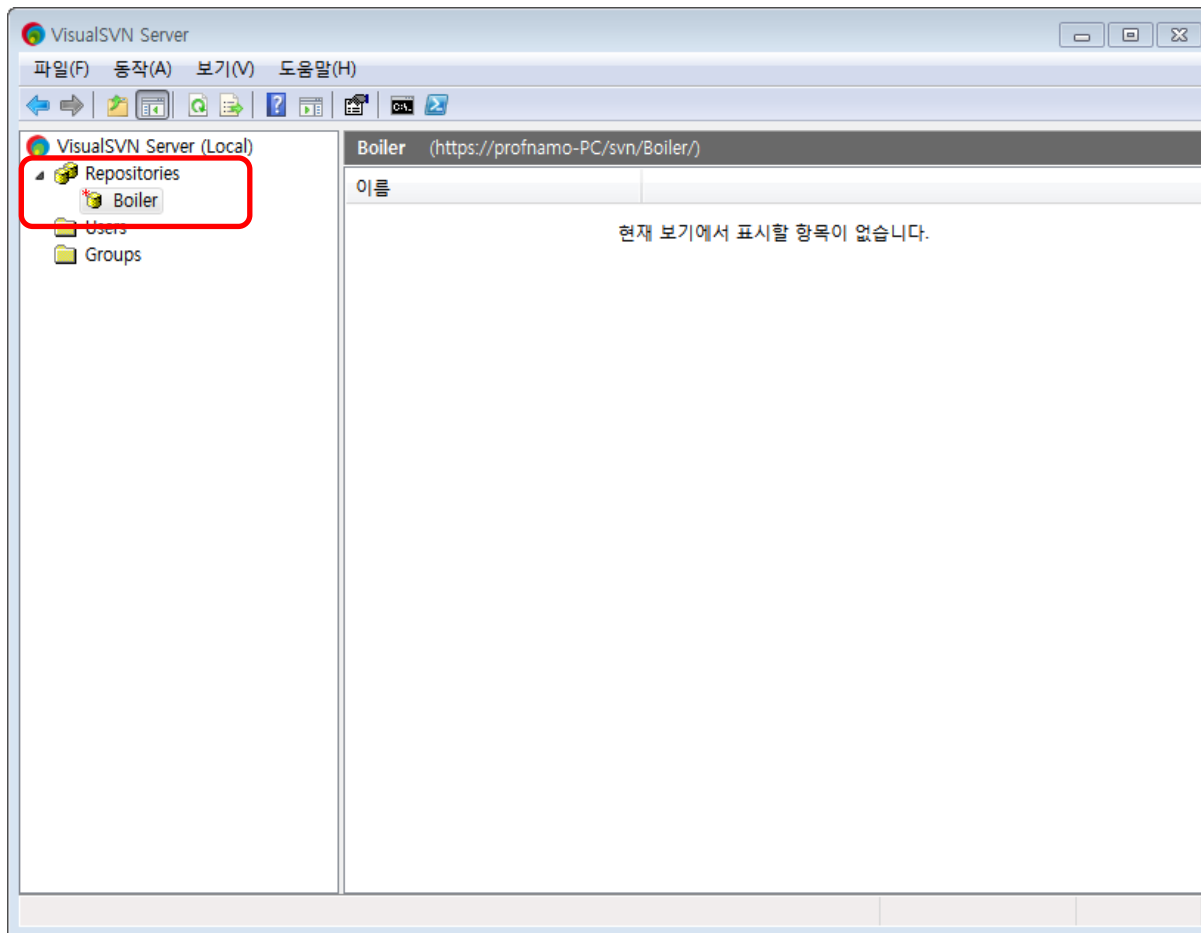
Repository Created Successfully
Please review the created repository details.

Repository Type: FSFS
Repository Name: Boiler
Repository URL: <https://profnano-PC/svn/Boiler>
Configured repository permissions:
- Everyone (Read / Write)

Finish 취소

Visual Subversion

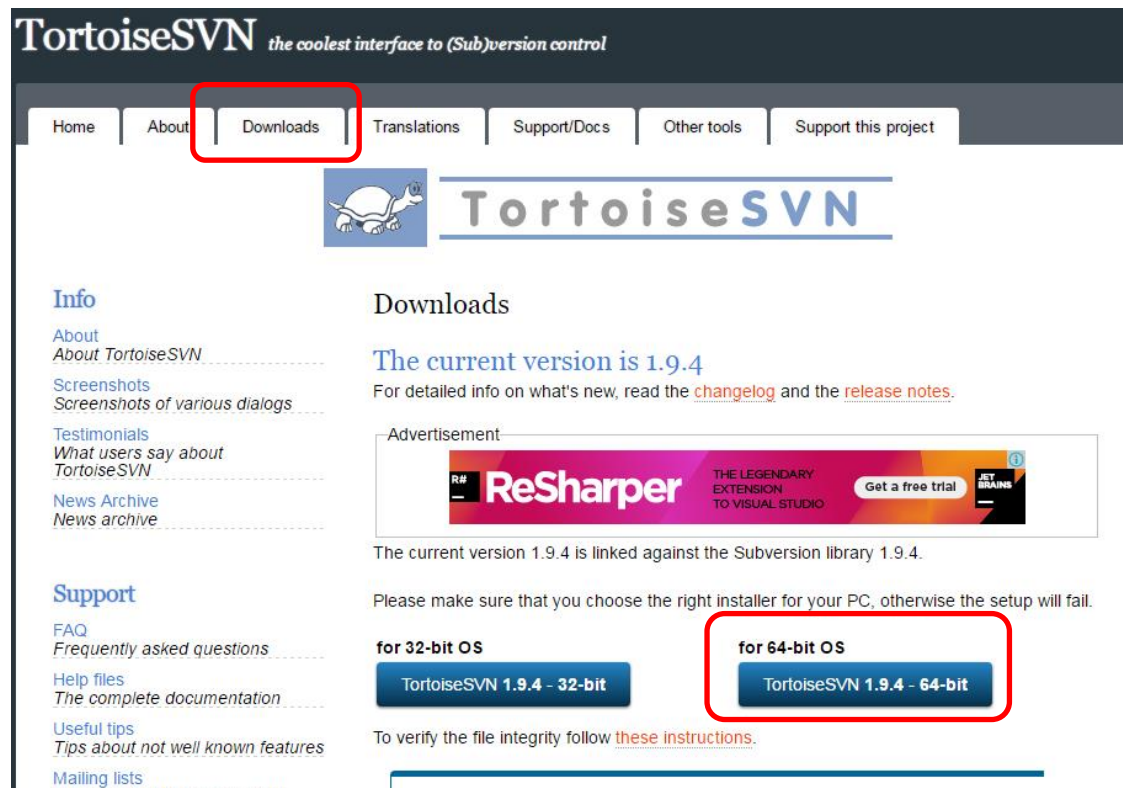
- Boiler 저장소 생성됨
- 서버 설치 완료
- 이제 서버의 저장소는 마련됨, 다음으로 클라이언트의 소스 코드를 이 저장소에 저장할 예정



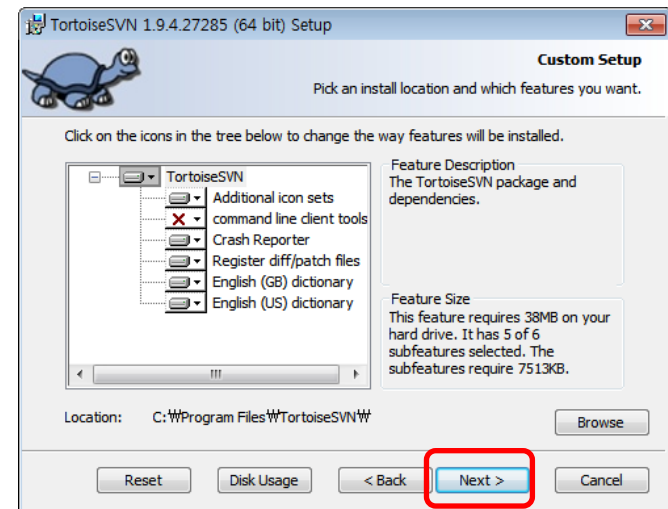
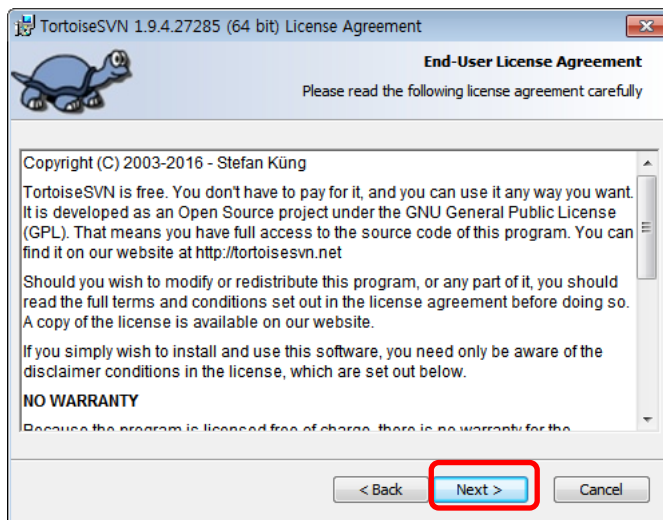
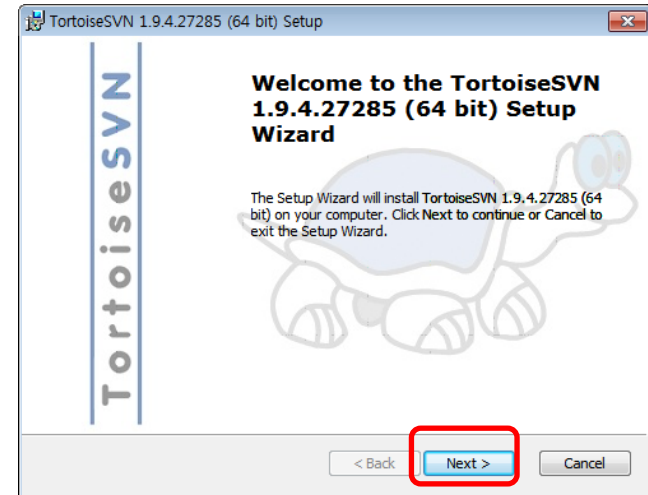
Tortoise SVN

■ <https://tortoisesvn.net/>

- Subversion 클라이언트로 가장 많이 사용됨, 이 프로그램외에도 수많은 SVN 클라이언트 프로그램이 존재함.
- 윈도우 탐색기와 연동하기 때문에 많이 사용됨
- Downloads선택 → TortoiseSVN-x.x.x.x-x64-svn-1.9.4.msi

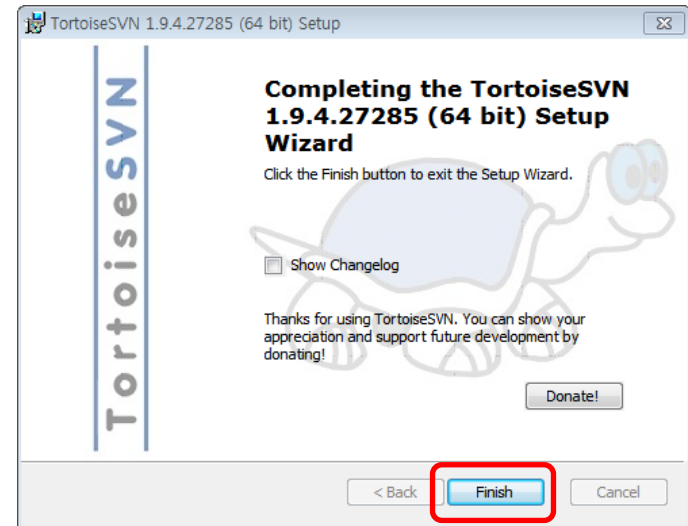
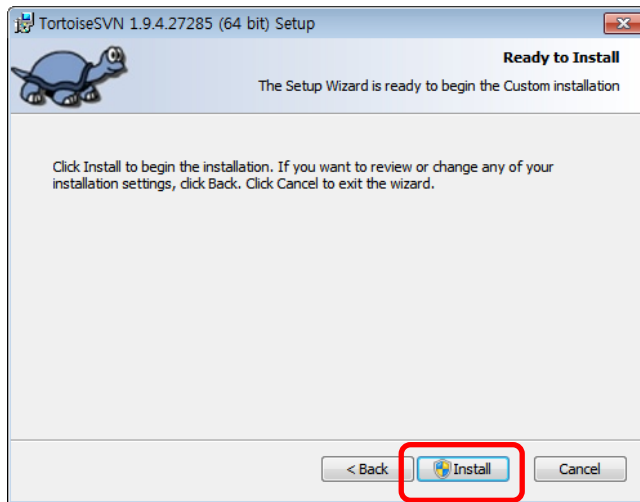


Tortoise SVN



Tortoise SVN

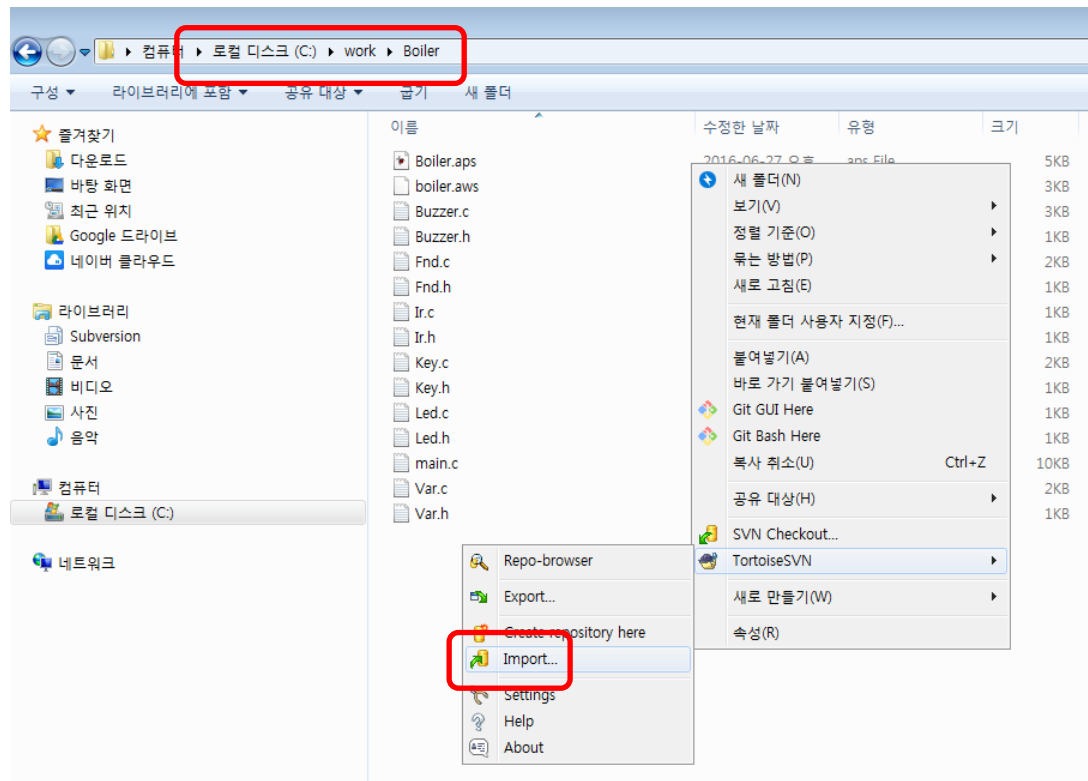
- 설치 완료



Tortoise SVN

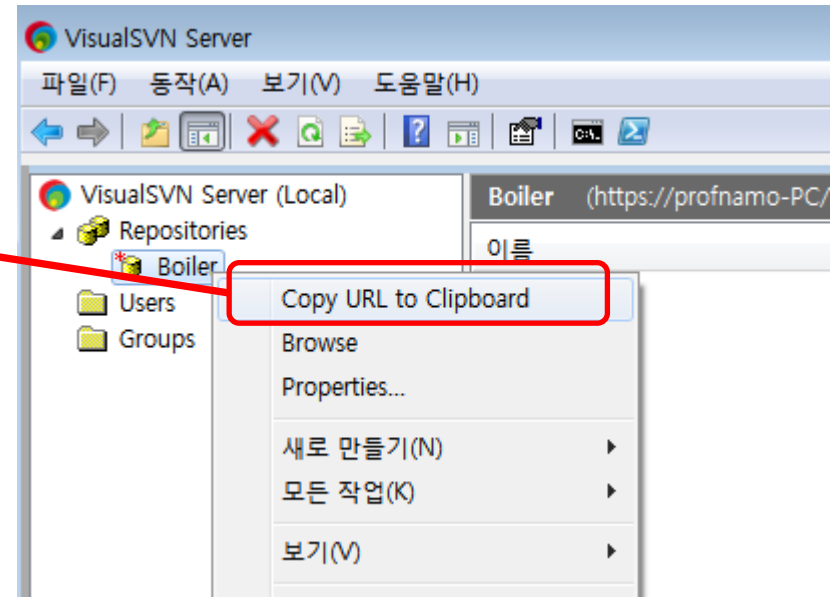
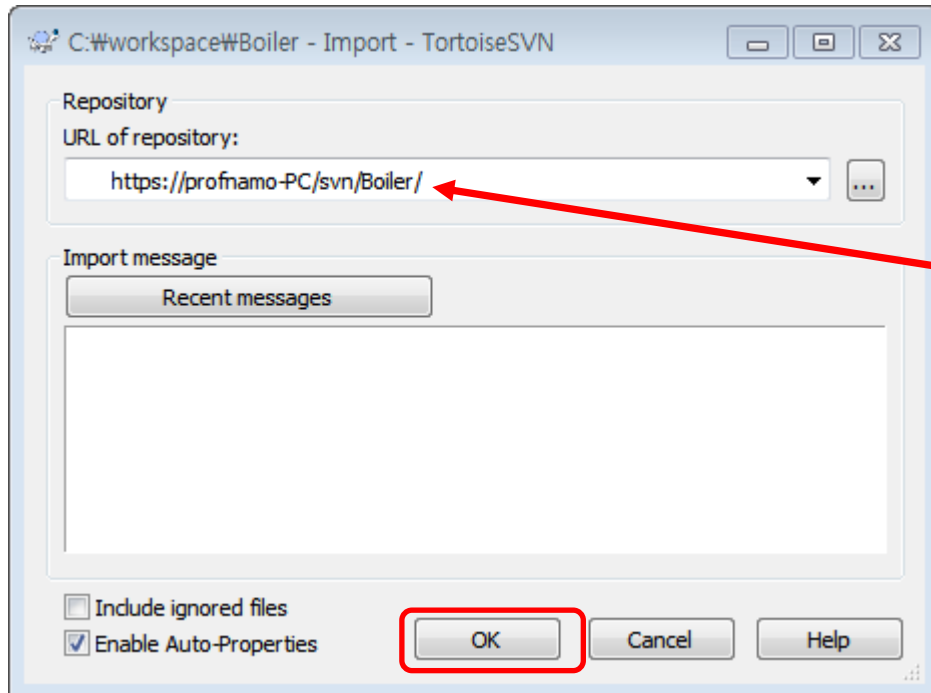
■ Boiler 프로젝트 소스 코드들을 서버 저장소에 저장

- 이와 같은 작업을 import 한다고 함
- Boiler 소스 import
 - Boiler.zip 파일을 C:\work\Boiler라는 폴더에 푼다.
 - 버전관리할 소스를 visual SVN 서버에 등록, 최초 한번만 하면 됨.
 - Boiler 폴더에서 마우스 오른쪽 버튼 선택하여 import선택



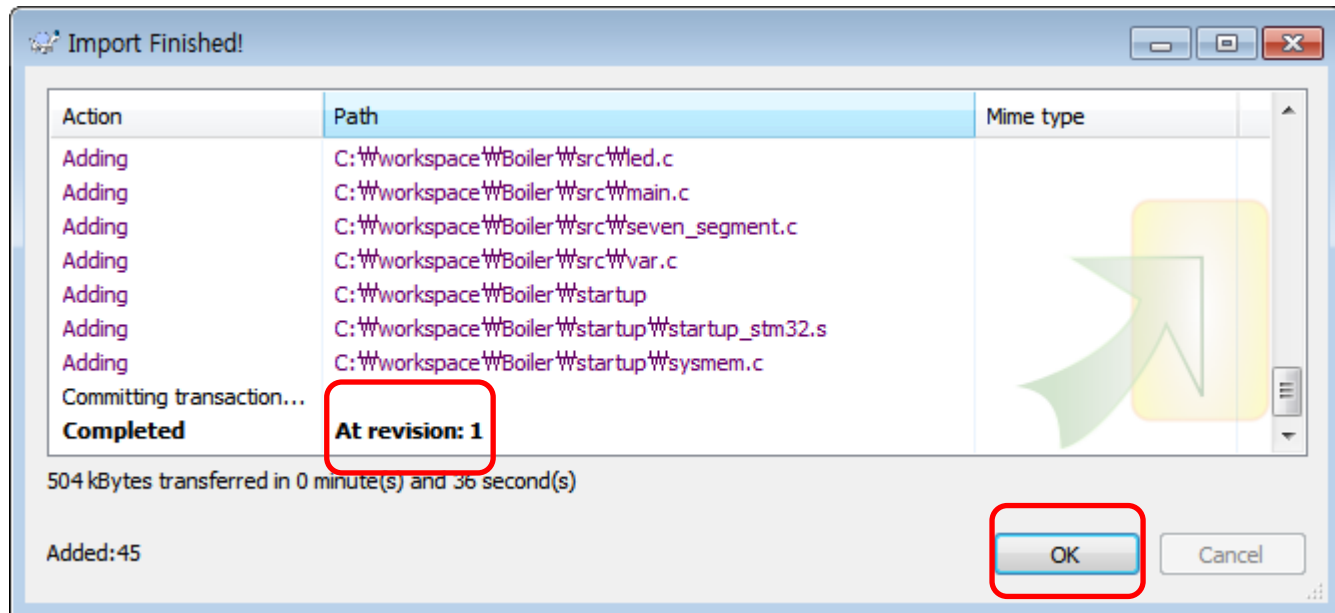
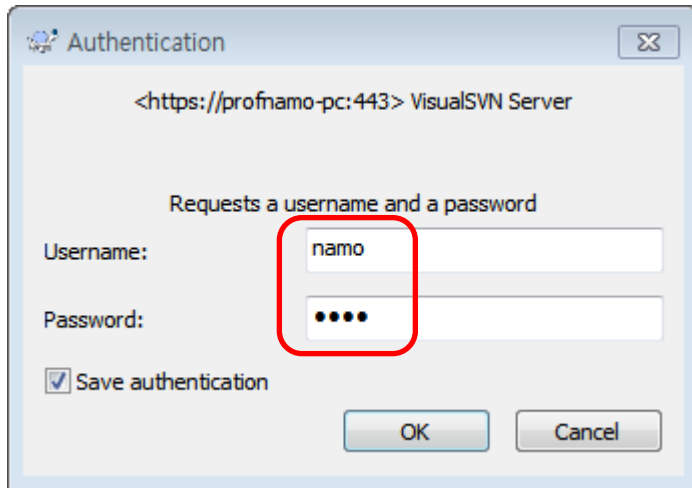
Tortoise SVN

- Visual SVN의 URL을 복사하여 Tortoise SVN에 복사



Tortoise SVN

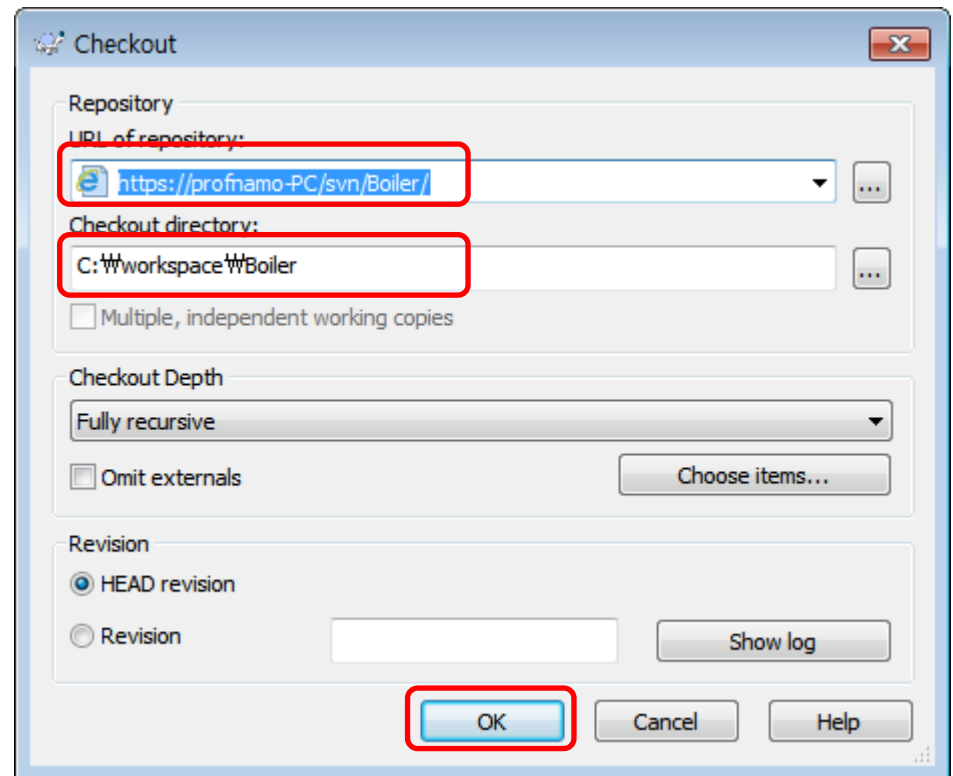
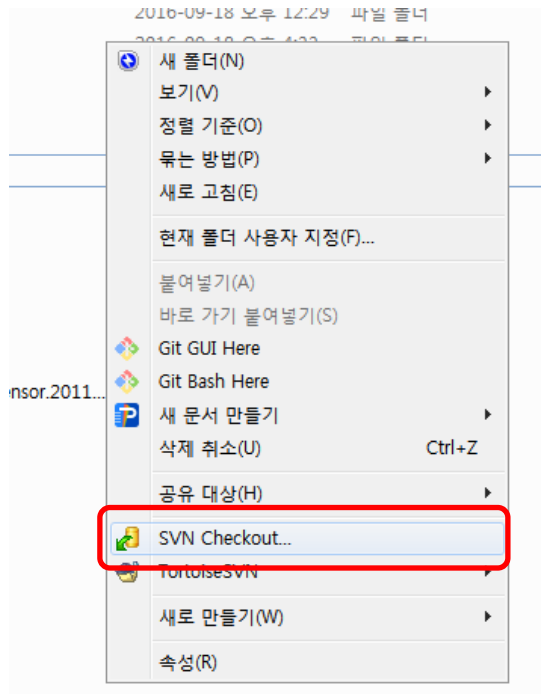
- Revision number 1로 Boiler 프로젝트가 서버에 등록된 화면



Tortoise SVN

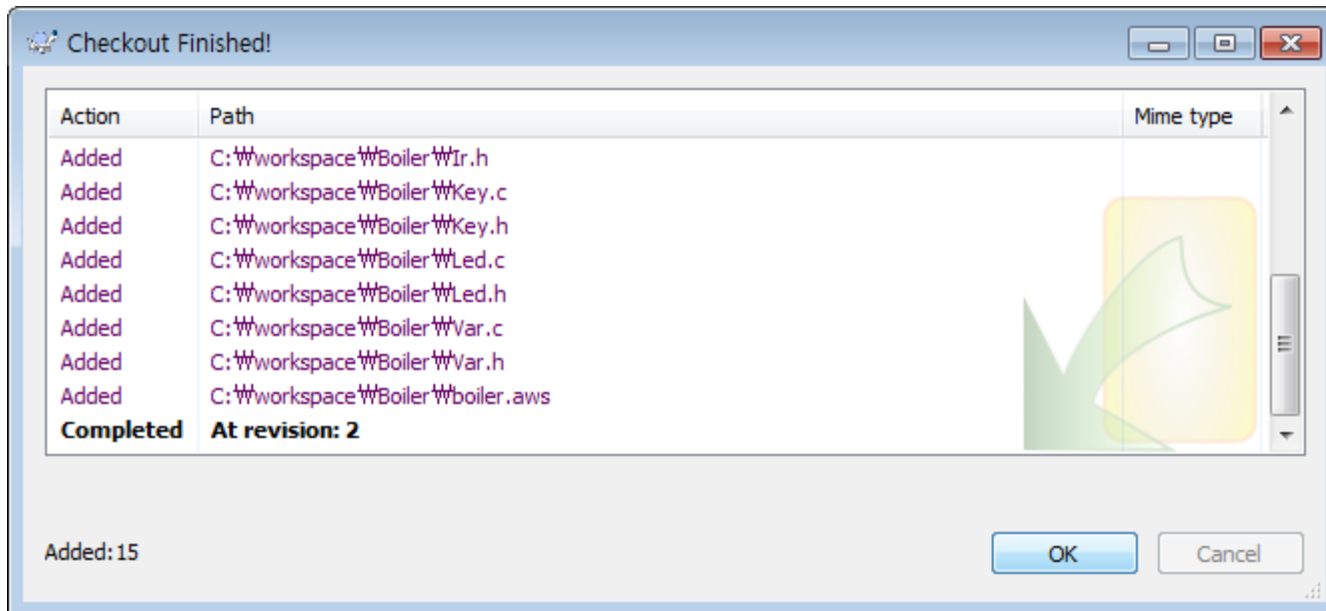
■ SVN 서버에 저장되어 있는 소스 코드들 가져오기

- 전문 용어로 check out한다고 함
- C:\workspace에 Boiler소스코드를 가져온다고 하자
 - 윈도우 탐색기로 C:\workspace 폴더에 가서 마우스 오른쪽 버튼 클릭
 - SVN Checkout... 선택



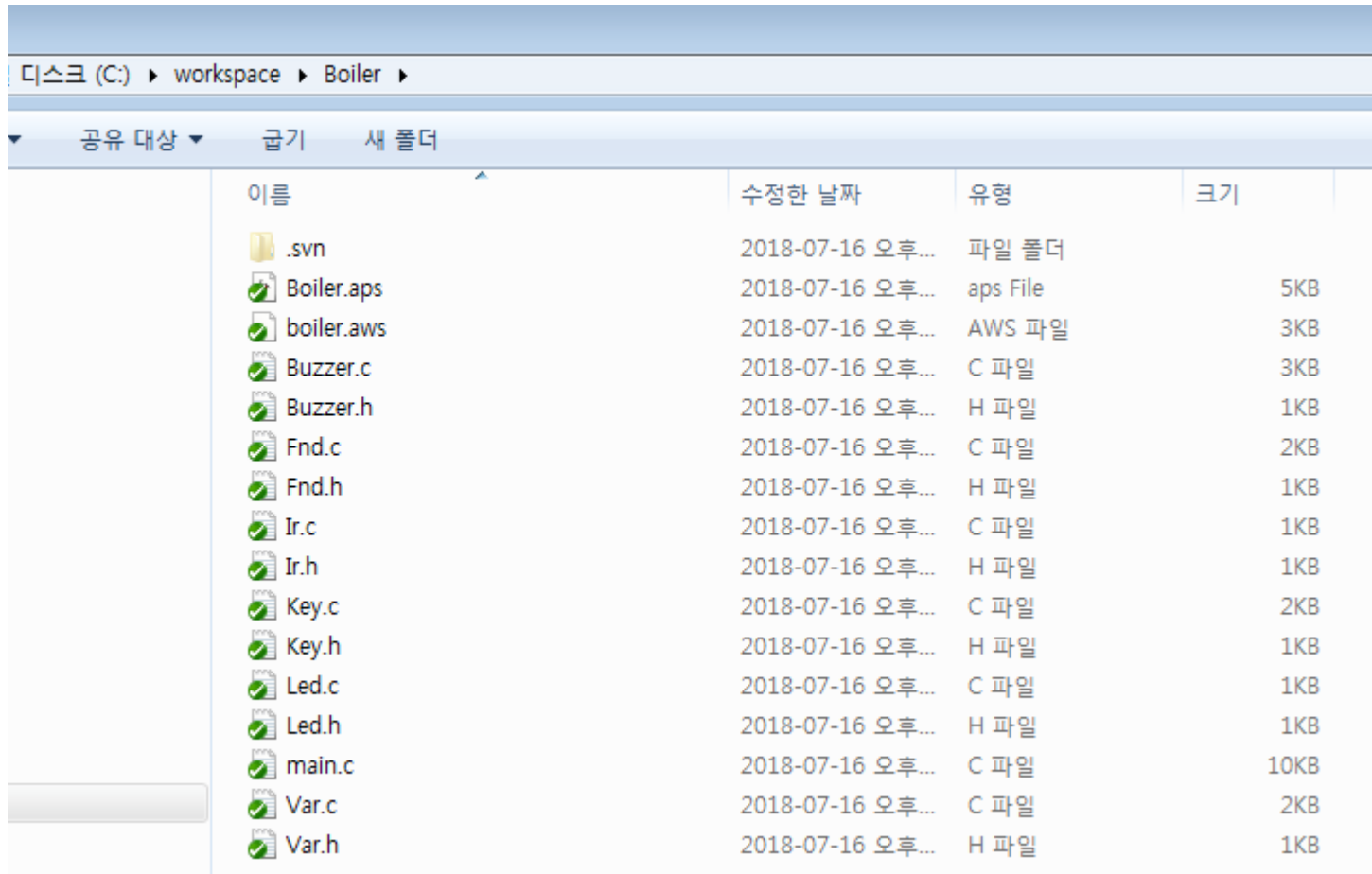
Tortoise SVN

- Boiler라는 폴더가 생기면서 소스가 복사됨



Tortoise SVN

- 아래와 같이 파일 아이콘에 녹색 마크가 생겨야 함.
- 마크가 없다면 컴퓨터를 재부팅



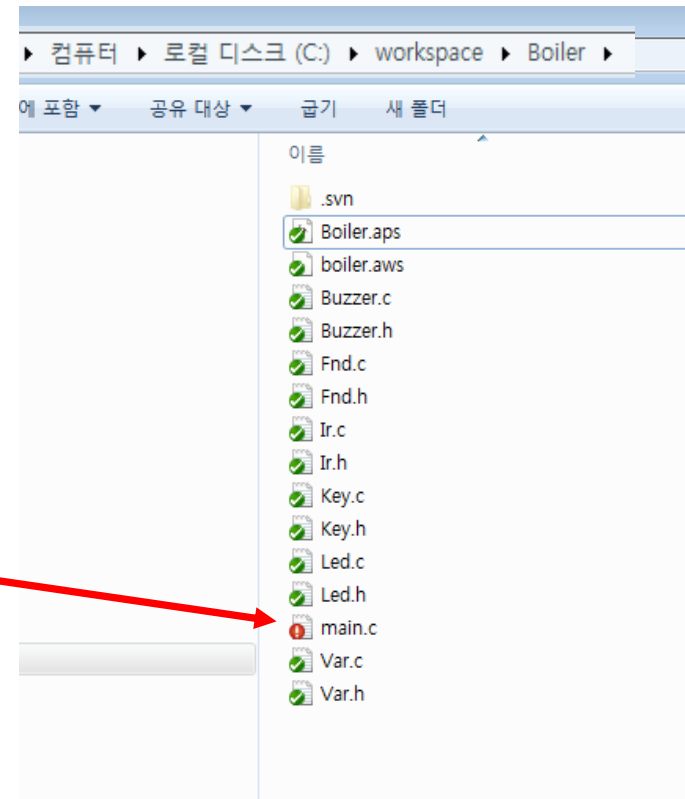
이름	수정한 날짜	유형	크기
.svn	2018-07-16 오후...	파일 폴더	
Boiler.aps	2018-07-16 오후...	aps File	5KB
boiler.aws	2018-07-16 오후...	AWS 파일	3KB
Buzzer.c	2018-07-16 오후...	C 파일	3KB
Buzzer.h	2018-07-16 오후...	H 파일	1KB
Fnd.c	2018-07-16 오후...	C 파일	2KB
Fnd.h	2018-07-16 오후...	H 파일	1KB
Ir.c	2018-07-16 오후...	C 파일	1KB
Ir.h	2018-07-16 오후...	H 파일	1KB
Key.c	2018-07-16 오후...	C 파일	2KB
Key.h	2018-07-16 오후...	H 파일	1KB
Led.c	2018-07-16 오후...	C 파일	1KB
Led.h	2018-07-16 오후...	H 파일	1KB
main.c	2018-07-16 오후...	C 파일	10KB
Var.c	2018-07-16 오후...	C 파일	2KB
Var.h	2018-07-16 오후...	H 파일	1KB

Tortoise SVN

- 이제 소스 코드를 수정한 후 서버에 저장해 보자.
- C:\workspace\Boiler\Boiler.aps를 실행하여 프로젝트를 연다.
- Main.c에서 다음과 같이 main함수에 변수 하나를 추가하고 저장한다.

```
}  
int main(void){  
    unsigned char key = 0;  
    unsigned char test_var = 1;  
  
    mcu_init(); //함수 실행  
    start_boiler(); //함수 실행  
  
    while(1){ //무한 루프  
        key = getkey(key); //읽어온 스위치 값  
        if(key_flag){ //key_flag 값이 1일때
```

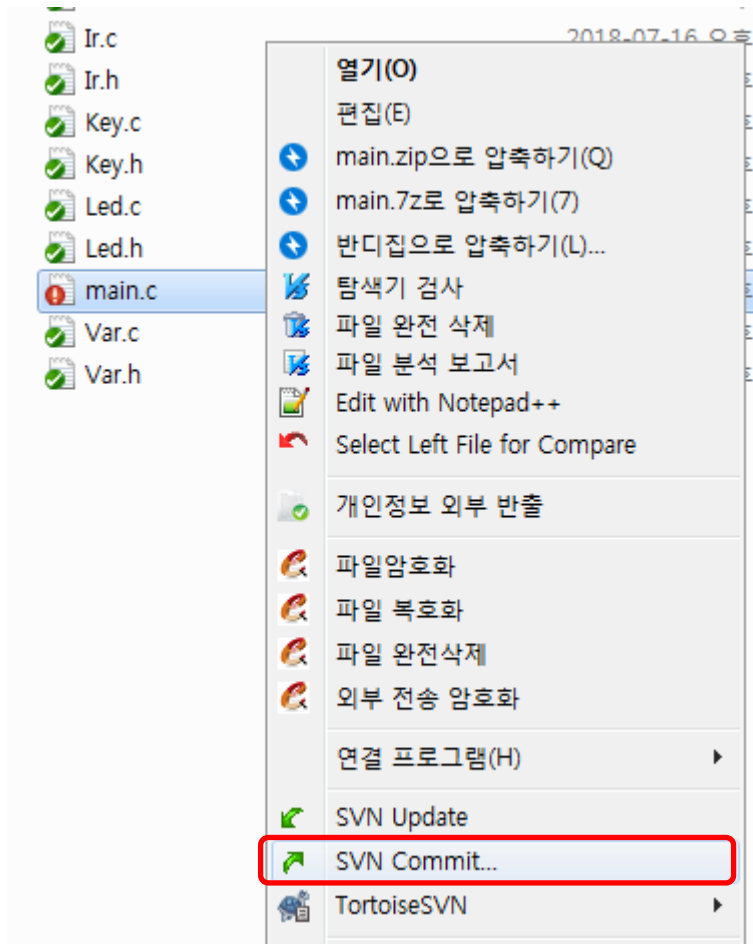
그러면 다음과 같이 윈도우 탐색기에서
main.c 파일 아이콘에 빨간색 마크가 생김
즉, 빨간색 마크는 서버에 저장되어 있는 소스 코드
와 버전이 달라졌다는 표시



Tortoise SVN

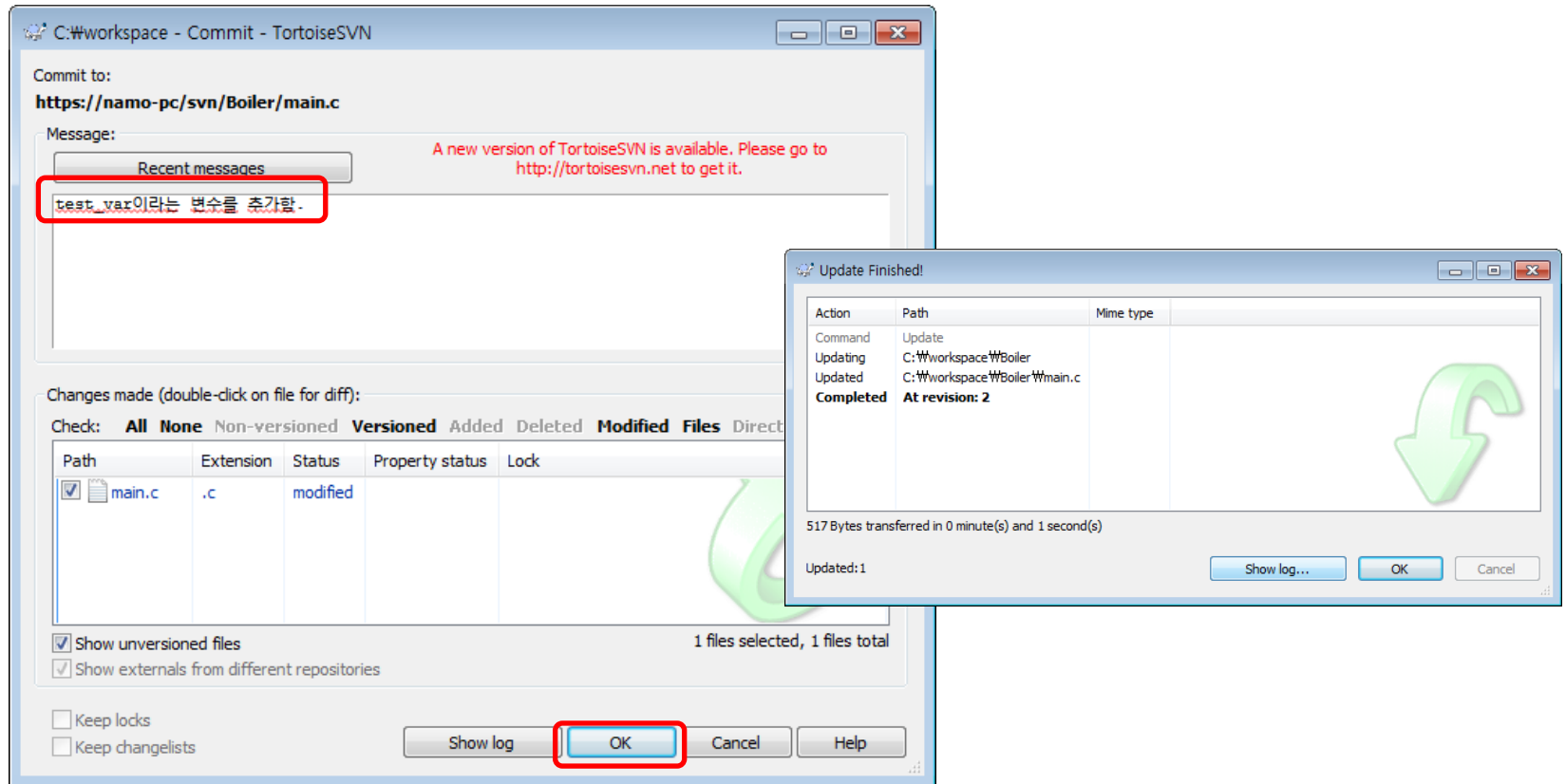
- 이 수정된 내용을 서버에 반영

- 이와 같은 동작을 check in 또는 commit 한다고 함
- Commit할 소스인 main.c 아이콘에 오른쪽 마우스 버튼을 클릭한 SVN Commit... 선택



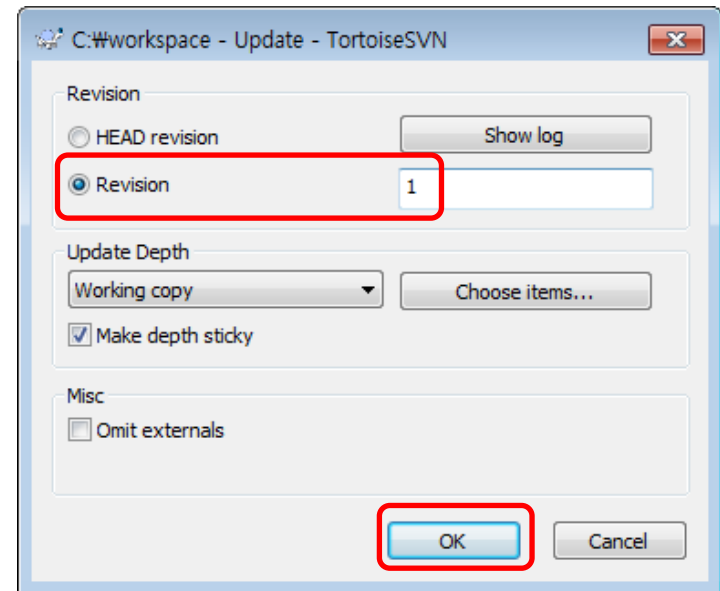
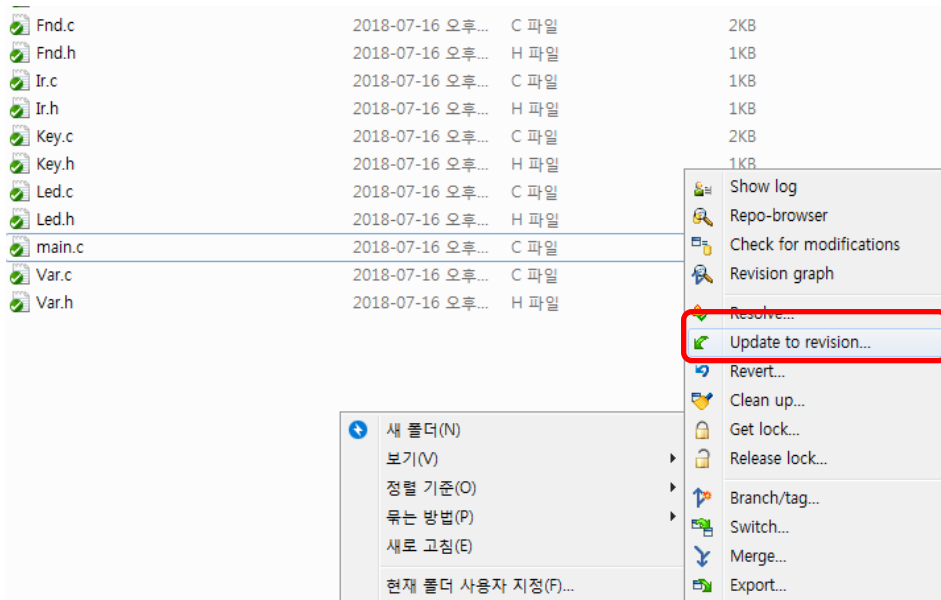
Tortoise SVN

- 이와 같은 message화면이 나오면 수정한 설명문을 추가하면 됨
- OK를 선택하여 최종 commit
- Revision number가 1에서 2를 증가함.



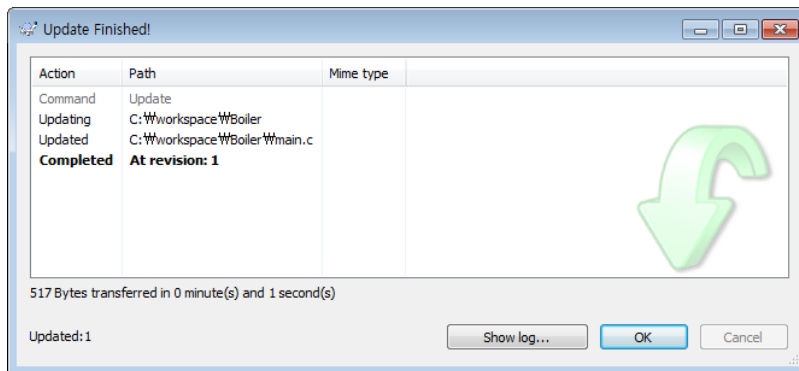
Tortoise SVN

- 이 revision 2는 수정한 main.c뿐만 아니라 프로젝트 전체에 있는 소스코드들의 수정을 저장하고 있음.
- 이제 수정하기 전 상태인 revision 1으로 프로젝트를 돌려보자.
 - 마우스 오른쪽 버튼을 클릭하여 update to revision...을 선택
 - HEAD revision은 최신 버전을 의미함. Revision을 선택하여 1을 선택하고 OK를 선택



Tortoise SVN

- Revision 1으로 수정되었다는 화면을 확인
- Main.c파일의 main함수에서 좀 전에 추가했던 코드가 없어지고 원 상태로 돌아갔음을 확인



```
}  
int main(void){  
    unsigned char key = 0;  
    mcd_init(); //함수 실행  
    start_boiler(); //함수 실행  
    while(1){ //무한 루프, ... }  
}
```

Activity 1.

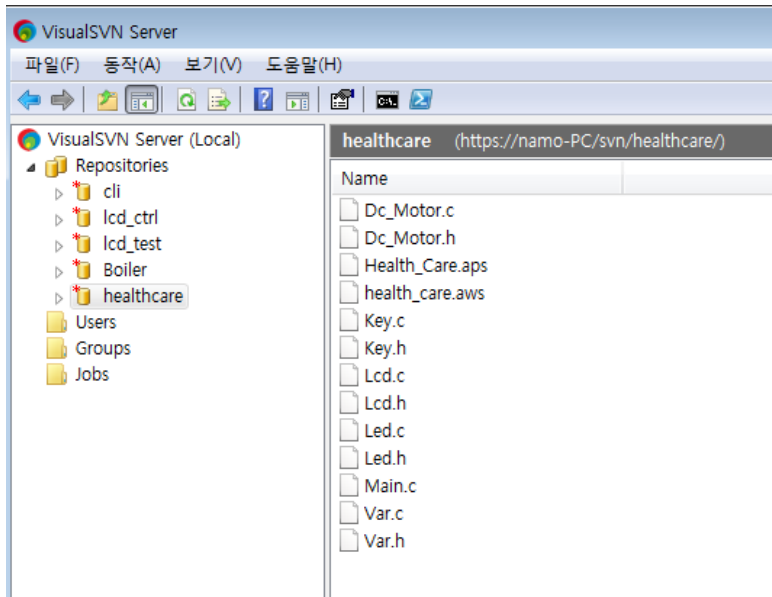
① Healthcare 소스로 자신의 subversion 서버에 아래 작업을 실시하고 각 단계를 사진 캡처 하시오.

- 1) Healthcare.zip파일을 C:\work\healthcare 폴더에 푼다.
- 2) Visual SVN에 healthcare 라는 저장소를 생성한다.
- 3) C:\work\healthcare 폴더를 healthcare 저장소에 import 한다.
- 4) C:\workspace\healthcare 폴더에 checkout한다.
- 5) Main함수에 int test_var; 이라는 변수를 추가한다.
- 6) Main.c 파일을 check in(=commit)한다.
- 7) Main.c 파일을 revision 1으로 수정한다.

Activity 1.

■ 예제 사진 파일

■ Healthcare 저장소 파일



■ Healthcare 폴더 파일

C:\workspace\healthcare				
이름	수정한 날짜	유형	크기	
.svn	2018-07-16 오후...	파일 폴더		
Dc_Motor.c	2018-07-16 오후...	C 파일	1KB	
Dc_Motor.h	2018-07-16 오후...	H 파일	1KB	
Health_Care.aps	2018-07-16 오후...	aps File	5KB	
health_care.aws	2018-07-16 오후...	AWS 파일	3KB	
Key.c	2018-07-16 오후...	C 파일	1KB	
Key.h	2018-07-16 오후...	H 파일	1KB	
Lcd.c	2018-07-16 오후...	C 파일	2KB	
Lcd.h	2018-07-16 오후...	H 파일	1KB	
Led.c	2018-07-16 오후...	C 파일	1KB	
Led.h	2018-07-16 오후...	H 파일	1KB	
Main.c	2018-07-16 오후...	C 파일	6KB	
Var.c	2018-07-16 오후...	C 파일	1KB	
Var.h	2018-07-16 오후...	H 파일	1KB	

Activity 1.

② 이제 회사처럼 팀 플레이로 버전관리툴을 사용해보자.

- 1) 3인 1조로 팀을 구성한다. 3번 학생(팀장)은 1번,2번 학생의 ID를 사용자로 추가한다.
- 2) 1번, 2번 학생은 3번 학생의 서버에 접속하여 healthcare 소스를 가져온다.
- 3) 1번 학생은 main함수에 `int first_st`라는 변수를 추가하여 commit 한다.
- 4) 2번 학생은 main함수에 `int second_st` 라는 변수를 추가하여 commit 한다.
- 5) 작업 과정의 캡쳐파일들을 각자 저장하여 제출한다.

Source Insight

- 대표적인 소스 코드 분석 툴
 - 회사에서 많이들 사용하는 에디터
- C 코드 분석에 아주 뛰어난 툴
- Source Insight
 - 다운로드 : <http://www.sourceinsight.com/down35.html>



Source Insight

■ 소스 코드 분석이란?

- 간단한 소스 코드는 분석없이 직관적으로 파악이 가능하다.
- 하지만 소스 코드가 길고 복잡해지고 파일의 숫자가 늘어나면 인간의 능력으로는 분석이 불가능하게 된다.
- 예를 들어 AVR 코드에서 DDRA와 같은 매크로는 어떻게 정의되어 있을까?
- C코드는 함수가 sub 함수를 호출하고 sub함수를 호출하는 형태로 복잡하게 되어 있는 경우가 많음. 따라가다가 결국 포기하게 됨.

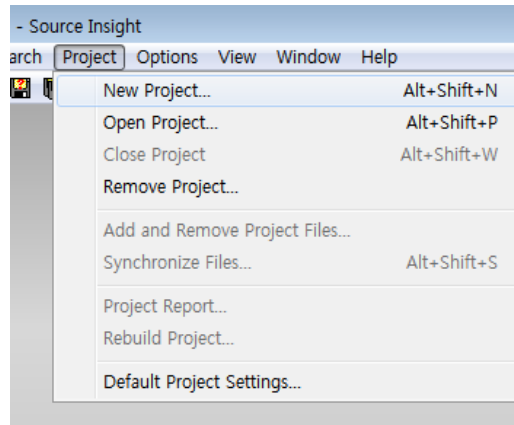
■ 경진대회 소스인 healthcare 프로젝트를 예를 들어 분석해보자.

- Lcd.c의 LCD_OUT 은 PORTC를 매크로로 정의한 것이다.
- PORTC는 어디에 정의되어 있을까?
 - AVR Studio 소스 코드인 C:\WinAVR-20100110의 어딘가에 정의되어 있다.
 - 이를 찾아내기 위해 source insight 프로젝트에 C:\WinAVR-20100110와 healthcare 소스코드를 추가하여 분석

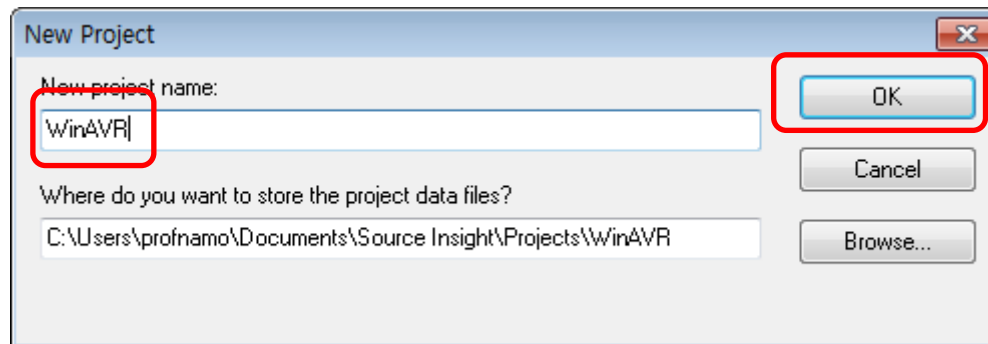
Source Insight

■ 프로젝트 만들기

- [Project] → [New Project...]

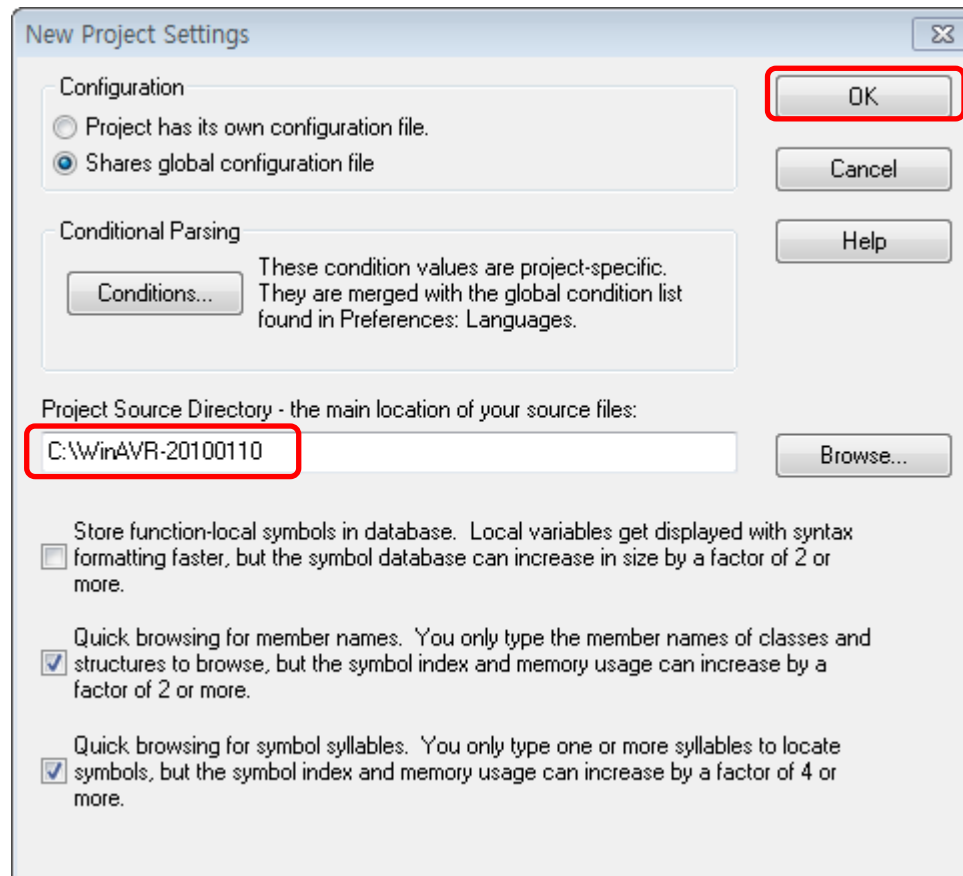


- New project name: 에 WinAVR 입력하고 [OK]선택



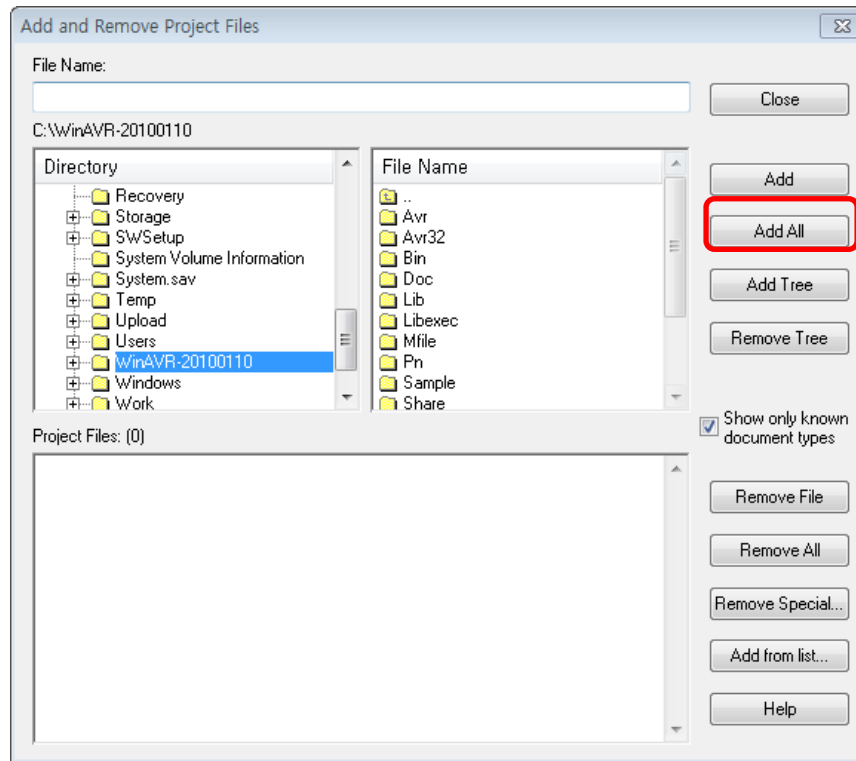
Source Insight

- Project Source Directory에 C:\WinAVR-20100110 입력하고 [OK] 선택

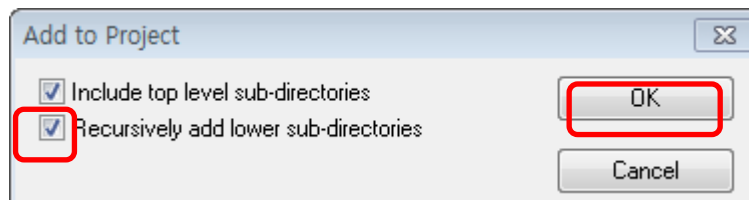


Source Insight

- [Add All] 선택

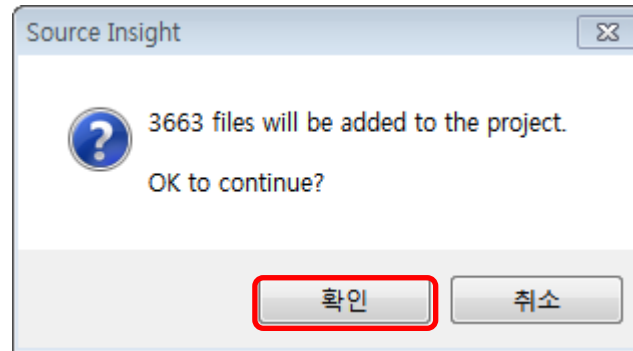


- Recursively add lower sub-directories를 클릭하고 OK선택

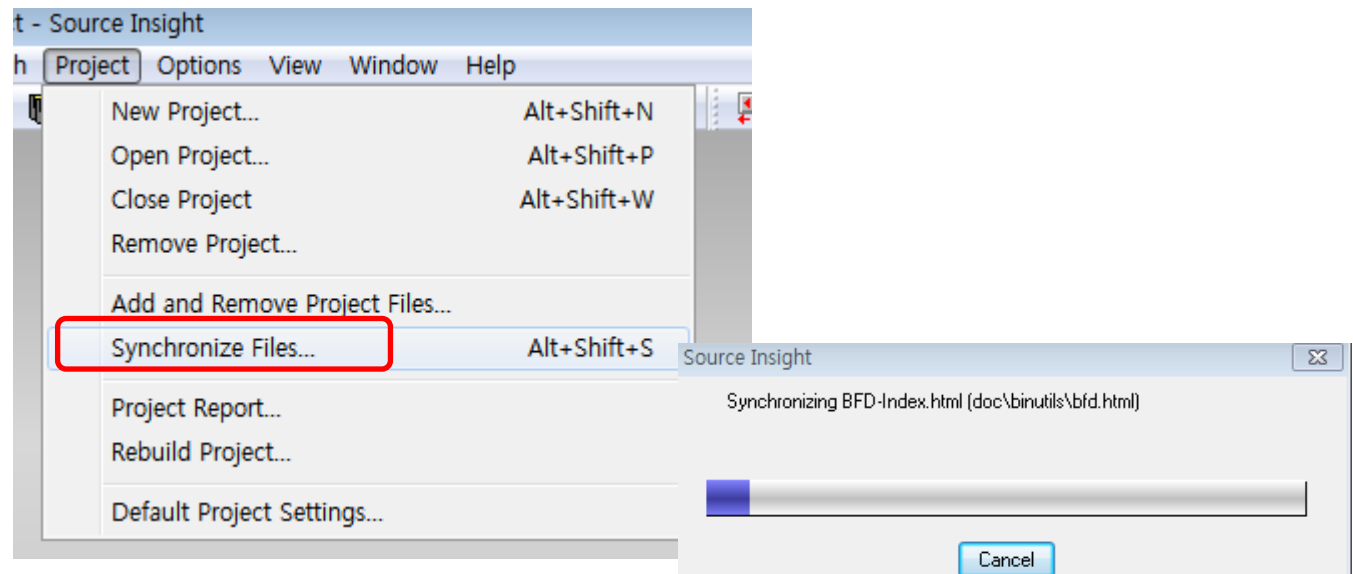


Source Insight

- 확인 선택

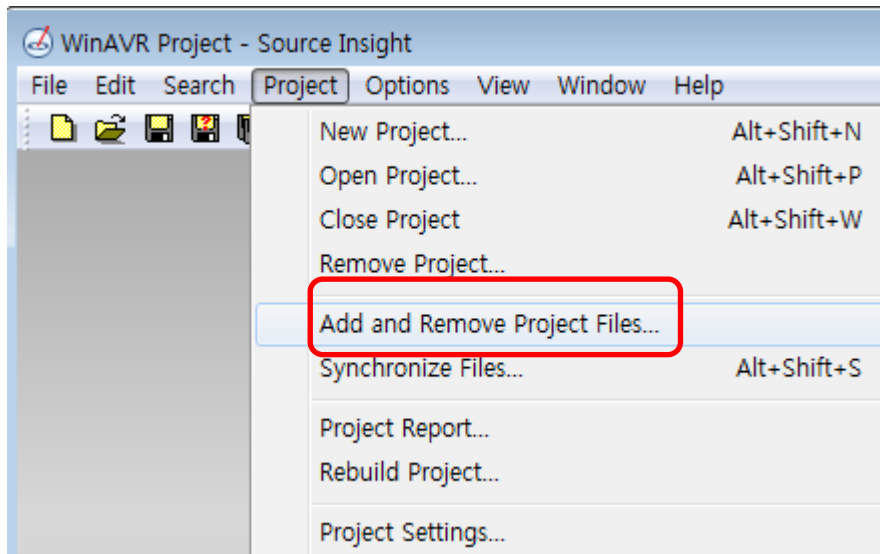


- [Project] → [Synchronize Files...] 선택
- 파일 개수에 따라 시간이 좀 걸림.



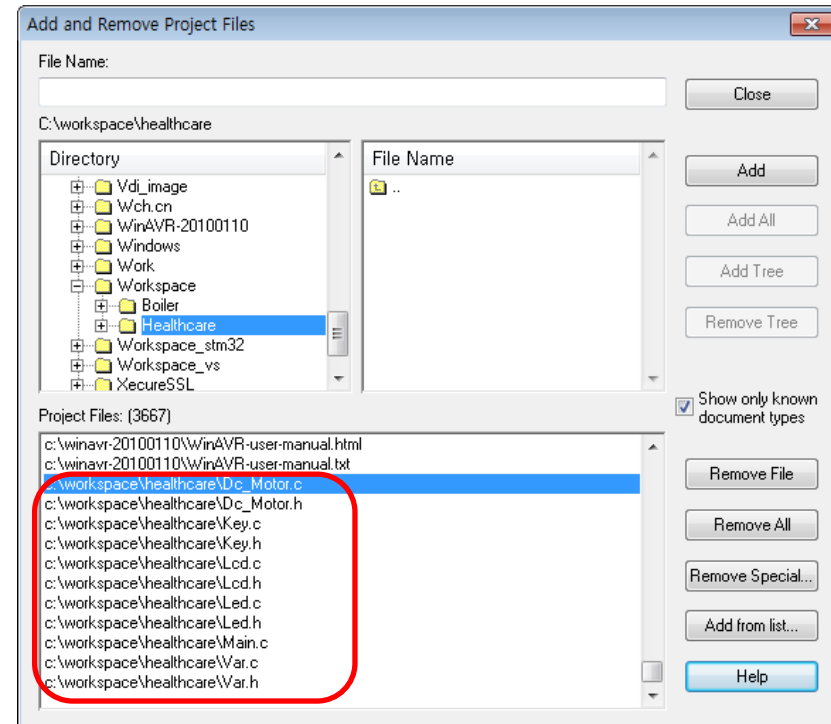
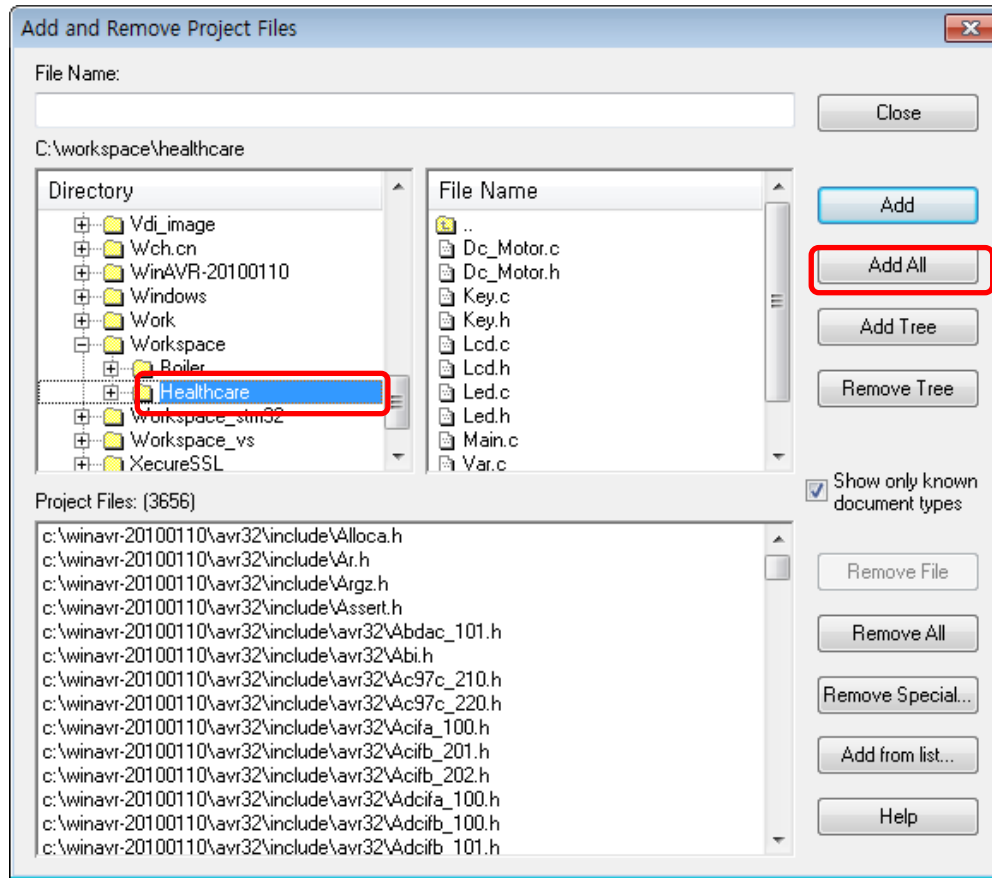
Source Insight

- C:\WinAVR-20100110 밑에 있는 파일들은 모두 프로젝트에 추가된 것임.
- 여기에 C:\workspace\healthcare 소스코드들을 추가해보자.
 - Project 메뉴에서 Add and Remove...를 선택



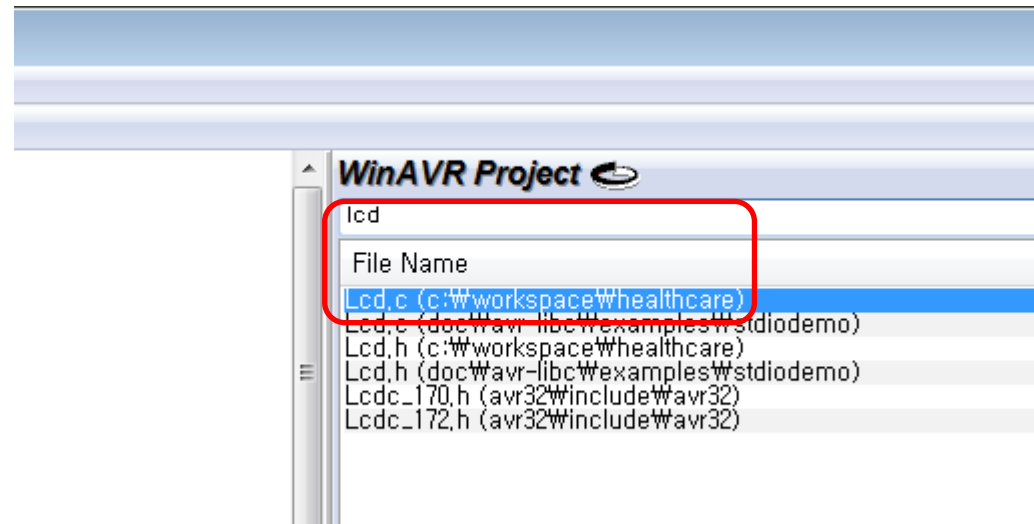
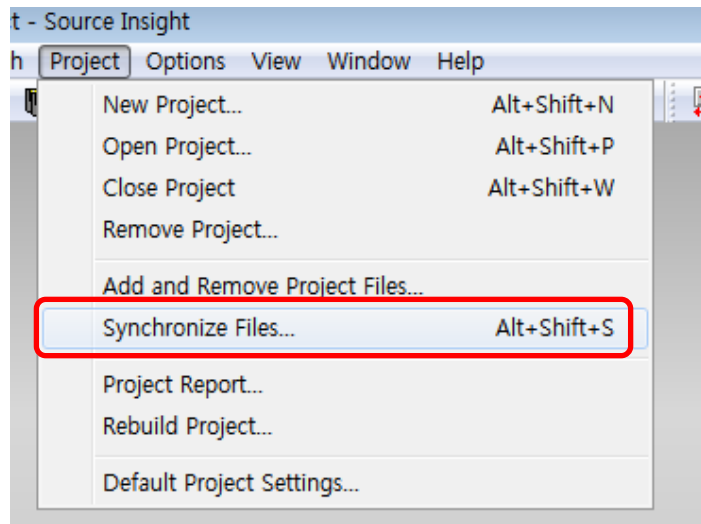
Source Insight

- Directory창에서 C:\workspace\healthcare를 선택한 후 Add All 을 선택



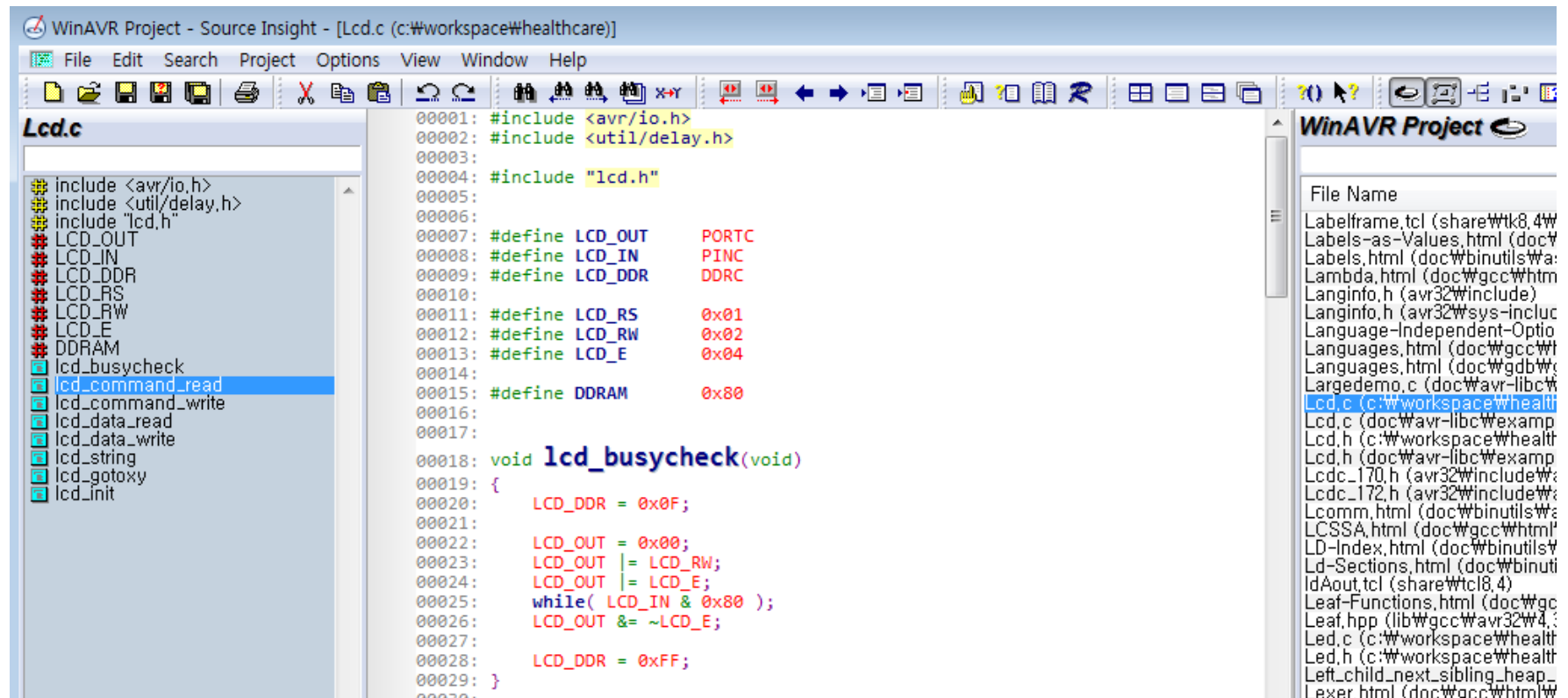
Source Insight

- 새로 추가된 소스의 동기를 맞추기 위해 다시 Synchronize Files...를 선택
- lcd.c파일을 열기 위해 오른쪽 상단 창에 lcd를 입력
- 더블 클릭하여 lcd.c를 연다.



Source Insight

- 그림과 같은 화면이 뜨면 모두 준비됨
- 정의된 함수나 매크로를 찾아가려면 키보드 ctrl를 선택한 상태에서 코드를 마우스로 선택하면 됨



Source Insight

- 예를 들어 lcd_init()함수에 있는 LCD_DDR나 lcd_command_write()함수의 정의된 값을 알아보자
- 키보드 ctrl를 선택한 상태에서 코드를 마우스로 선택

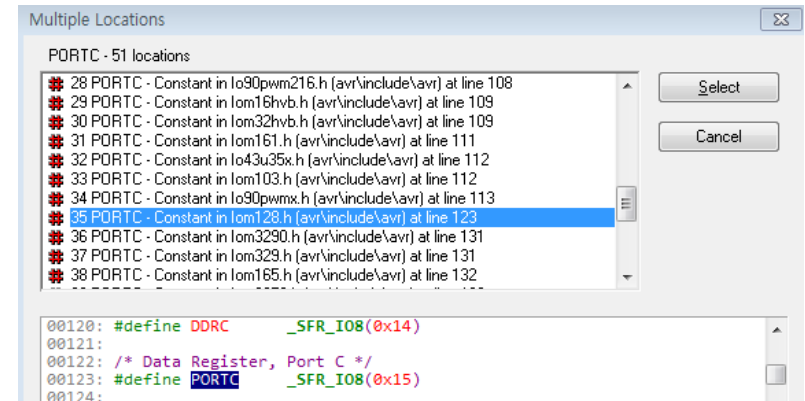
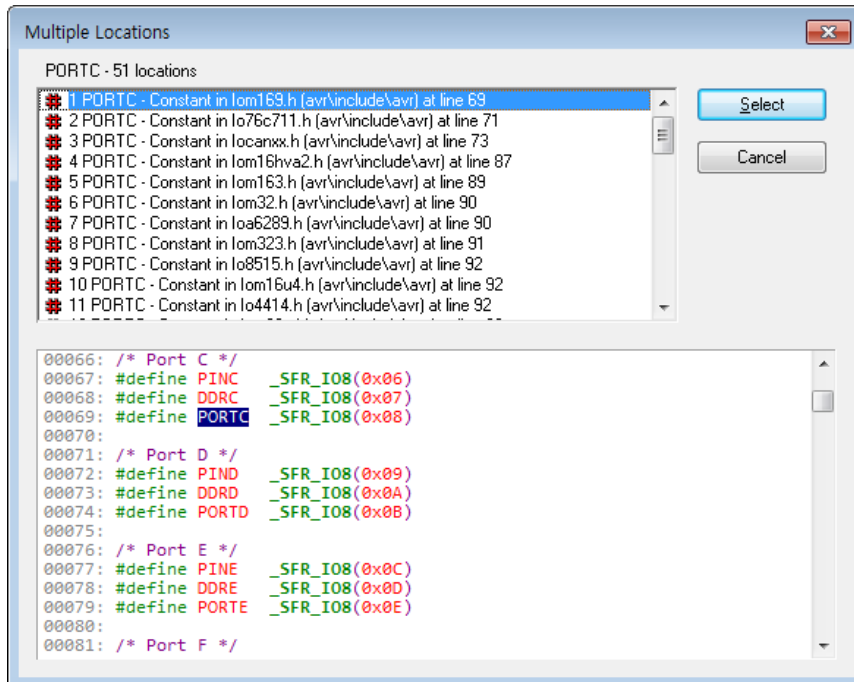
```
00122: void lcd_init(void)
00123: {
00124:     LCD_DDR = 0xFF;
00125:
00126:     lcd_command_write(0x20);
00127:     _delay_ms(10);
00128:     lcd_command_write(0x20);
00129:     _delay_ms(10);
00130:     lcd_command_write(0x20);
00131:     _delay_ms(10);
00132:     lcd_command_write(0x28);
00133:     lcd_command_write(0x08);
00134:     lcd_command_write(0x01);
00135:     lcd_command_write(0x06);
00136:     _delay_ms(10);
00137:     lcd_command_write(0x0C);
00138: }
```

```
00006:
00007: #define LCD_OUT    PORTC
00008: #define LCD_IN     PINC
00009: #define LCD_DDR    DDRC
00010:
00011: #define LCD_RS     0x01
00012: #define LCD_RW     0x02
00013: #define LCD_E      0x04
00014:
00015: #define DDRAM      0x80
00016:
```

```
00053: void lcd_command_write(unsigned char cmd)
00054: {
00055:     lcd_bussycheck();
00056:
00057:     LCD_OUT = 0x00;
00058:
00059:     LCD_OUT |= cmd & 0xF0;
00060:     LCD_OUT |= LCD_E;
00061:     LCD_OUT &= ~LCD_E;
00062:
00063:     LCD_OUT &= 0x0F;
00064:     LCD_OUT |= cmd << 4;
00065:     LCD_OUT |= LCD_E;
00066:     LCD_OUT &= ~LCD_E;
00067:
00068:     _delay_ms(2);
00069: }
00070:
```

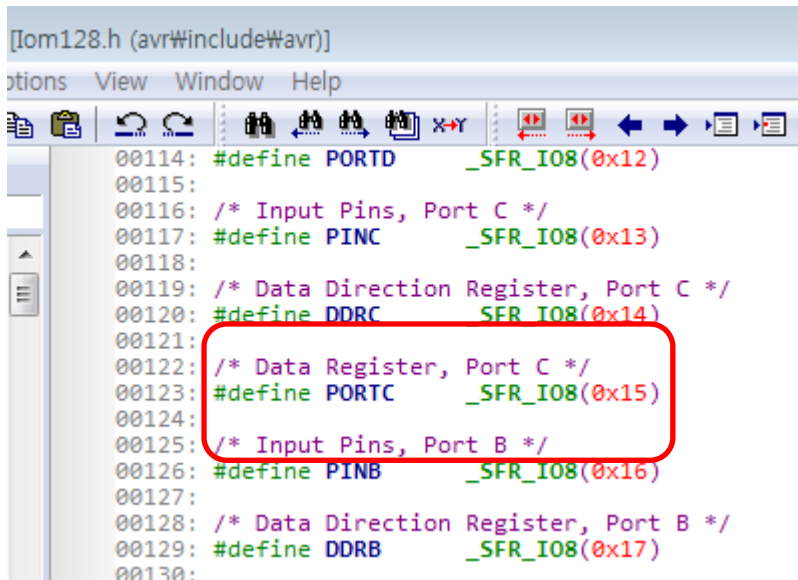

Source Insight

- LCD_DDR나 lcd_command_write()함수는 개발자가 정의한 것이나 PORTC와 같은 매크로는 개발환경에서 정의한 값이다.
- 이제 PORTC와 같은 매크로가 정의된 값을 찾아보자.
- 키보드 ctrl을 선택하고 마우스로 PORTC를 클릭해보자
 - 그림과 같이 정의된 위치가 무려 51개나 있다. 이중에 우리가 찾고자 하는 값은?



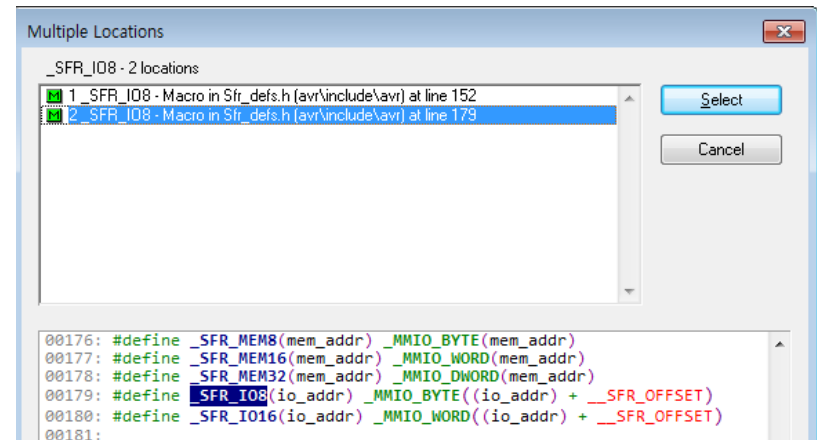
Source Insight

- 51개나 있는 이유는 MCU의 종류가 그 만큼 있는 것이고 우리는 ATmega128을 쓴다
- 분석이 필요하나 일단 눈치확상 iom128.h를 선택
- PORTC가 _SFR_IO8()이라는 매크로에 0x15값을 입력한 값이라는 것을 알 수 있다.
- 0x15는 무엇일까?
- _SFR_IO8() 매크로는 어떤 정의일까?
- _SFR_IO8() 매크로를 찾았더니 2가지 정의가 존재하는데 일단 아래 쪽을 선택해보자.
- _MMIO_BYTE()에 io_addr 을 입력하고 __SFR_OFFSET을 더한 값이다.



```
[Iom128.h (avr\include\avr)]
Options View Window Help

00114: #define PORTD      _SFR_IO8(0x12)
00115:
00116: /* Input Pins, Port C */
00117: #define PINC       _SFR_IO8(0x13)
00118:
00119: /* Data Direction Register, Port C */
00120: #define DDRC       _SFR_IO8(0x14)
00121:
00122: /* Data Register, Port C */
00123: #define PORTC      _SFR_IO8(0x15)
00124:
00125: /* Input Pins, Port B */
00126: #define PINB       _SFR_IO8(0x16)
00127:
00128: /* Data Direction Register, Port B */
00129: #define DDRB       _SFR_IO8(0x17)
00130:
```



Multiple Locations

_SFR_IO8 - 2 locations

- 1 _SFR_IO8 - Macro in Sfr_defs.h (avr\include\avr) at line 152
- 2 _SFR_IO8 - Macro in Sfr_defs.h (avr\include\avr) at line 179

Select

Cancel

```
00176: #define _SFR_MEM8(mem_addr) _MMIO_BYTE(mem_addr)
00177: #define _SFR_MEM16(mem_addr) _MMIO_WORD(mem_addr)
00178: #define _SFR_MEM32(mem_addr) _MMIO_DWORD(mem_addr)
00179: #define _SFR_IO8(io_addr) _MMIO_BYTE((io_addr) + __SFR_OFFSET)
00180: #define _SFR_IO16(io_addr) _MMIO_WORD((io_addr) + __SFR_OFFSET)
00181:
```

Source Insight

- `_MMIO_BYTE`는 `mem_addr`이라는 값에 `uint8_t` 포인터형의 내용을 의미한다.

```
00127: #define _MMIO_BYTE(mem_addr) (*(volatile uint8_t*)(mem_addr))  
00128: #define _MMIO_WORD(mem_addr) (*(volatile uint16_t*)(mem_addr))  
00129: #define _MMIO_DWORD(mem_addr) (*(volatile uint32_t*)(mem_addr))  
00130: #endif  
00131: ----  
00132: ----
```

- `#define LCD_OUT PORTC`
 - `#define PORTC _SFR_IO8(0x15)`
 - `_SFR_IO8(io_addr) _MMIO_BYTE((io_addr) + __SFR_OFFSET)`
 - `#define _MMIO_BYTE(mem_addr) (*(volatile uint8_t*)(mem_addr))`
-
- 한마디로 `__SFR_OFFSET`이 0이라면 `PORTC`는
 - `*(volatile uint8_t*)(0x15)`
 - 이를 설명하면 0x15번지를 가지는 메모리의 내용을 의미함.

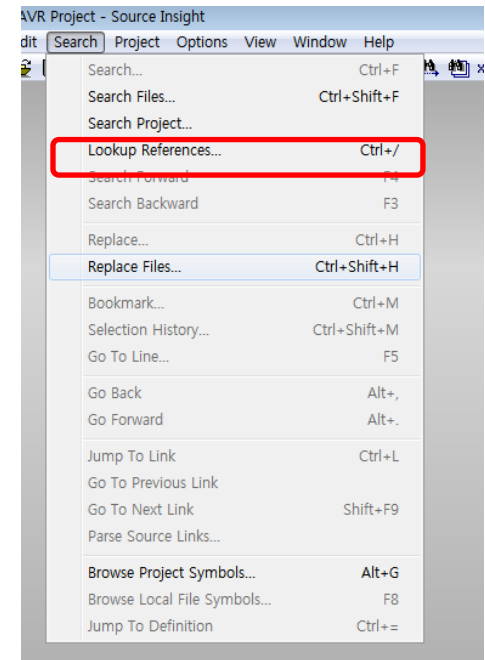
Source Insight

- 이제 다른 방법으로 macro값을 찾아보자

```
#include <avr/io.h>

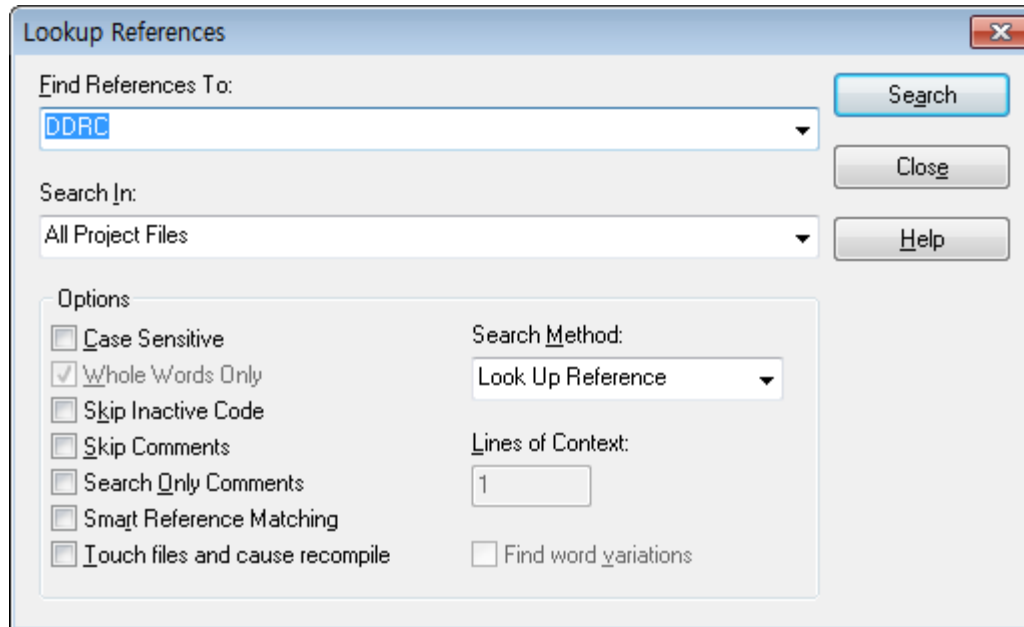
int main(void)
{
    DDRC = 0xFF;
    while(1){
        PORTC = PINB;
    }
    return 0;
}
```

- DDRC, PORTC, PINB가 어떤 macro인지?
- 찾고자하는 심볼을 [Search] → [Lookup Reference...]



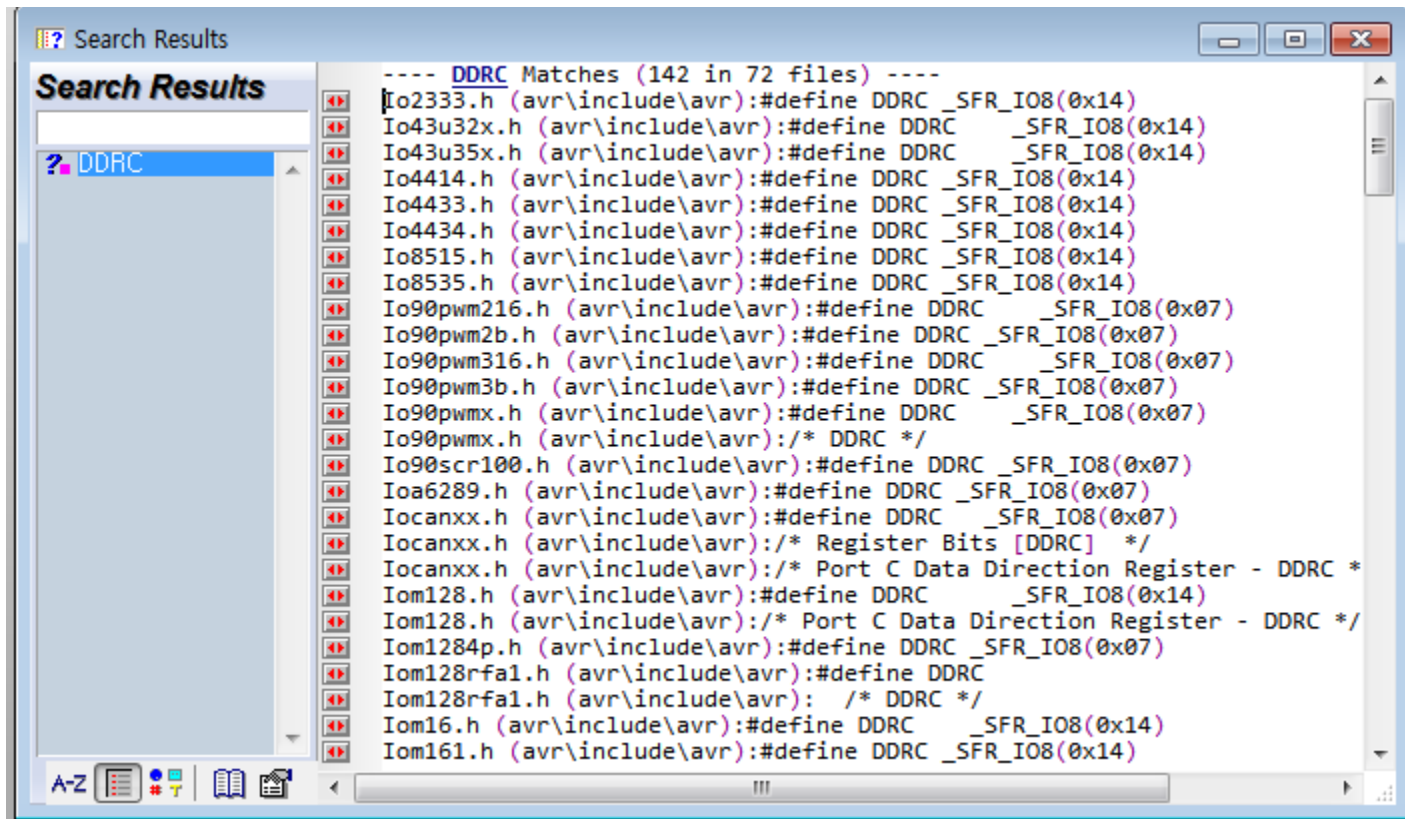
Source Insight

■ DDRC입력 후 [Search]선택



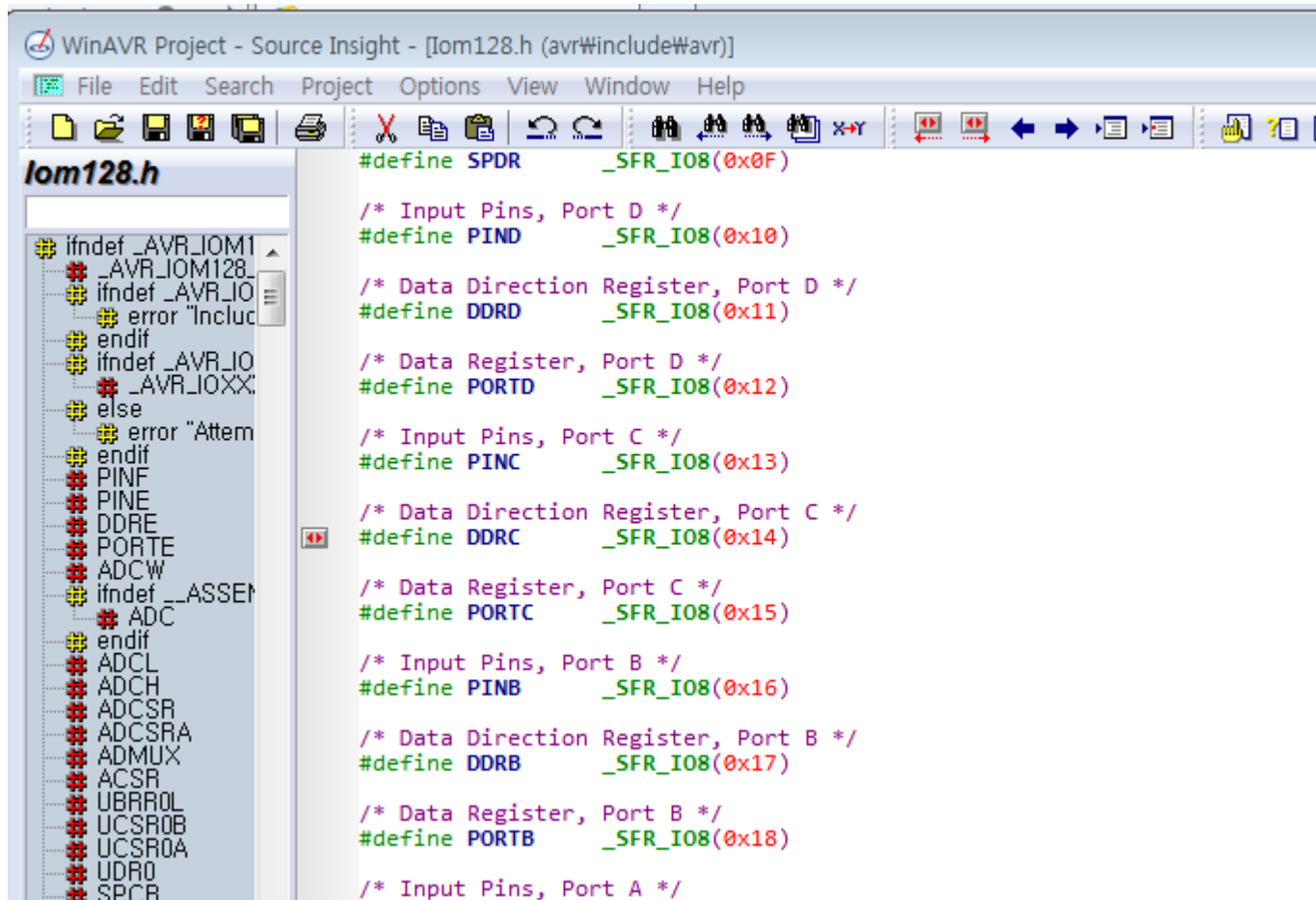
Source Insight

- 전체 프로젝트 (C:\WinAVR-20100110)에서 DDRC라고 쓰여진 모든 문장을 보여줌



Source Insight

- 먼저 io.h을 찾고 iom128.h에서 DDRC 를 분석해보시오.



```
WinAVR Project - Source Insight - [Iom128.h (avr#include#avr)]
File Edit Search Project Options View Window Help

Iom128.h
#ifndef _AVR_IOM128_
#define _AVR_IOM128_
/* Input Pins, Port D */
#define PIND _SFR_I08(0x10)
/* Data Direction Register, Port D */
#define DDRD _SFR_I08(0x11)
/* Data Register, Port D */
#define PORTD _SFR_I08(0x12)
/* Input Pins, Port C */
#define PINC _SFR_I08(0x13)
/* Data Direction Register, Port C */
#define DDRC _SFR_I08(0x14)
/* Data Register, Port C */
#define PORTC _SFR_I08(0x15)
/* Input Pins, Port B */
#define PINB _SFR_I08(0x16)
/* Data Direction Register, Port B */
#define DDRB _SFR_I08(0x17)
/* Data Register, Port B */
#define PORTB _SFR_I08(0x18)
/* Input Pins, Port A */
```

Activity 2.

① Health프로젝트를 분석해보자.

- 1) KEY_START 값은 얼마인가?
- 2) NO_MODE 값은 얼마인가?
- 3) timer0_init()함수의 TCCR0의 매크로를 정리하시오.
- 4) Main함수에서 mcu_init()함수가 호출되는 함수들의 이름만 sub 함수가 없을 때 까지 찾아 source_code_analysis.xlsx에 정리해보자.