

CTIP for C & Unit Test

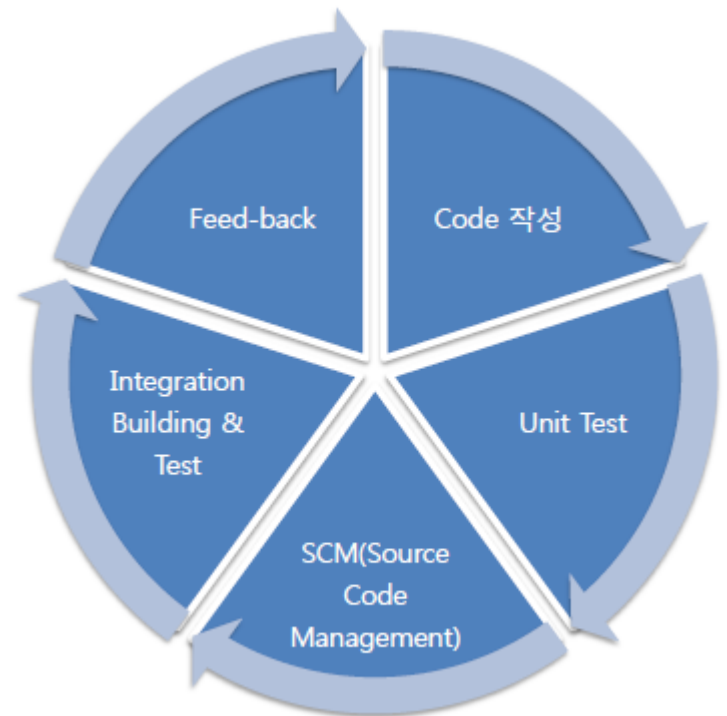
200511305 김성규
200511306 김성훈
200614164 김효석
200611124 유성배
200518036 곡진화

Establish CTIP Enviroment

CTIP

❖ CTIP

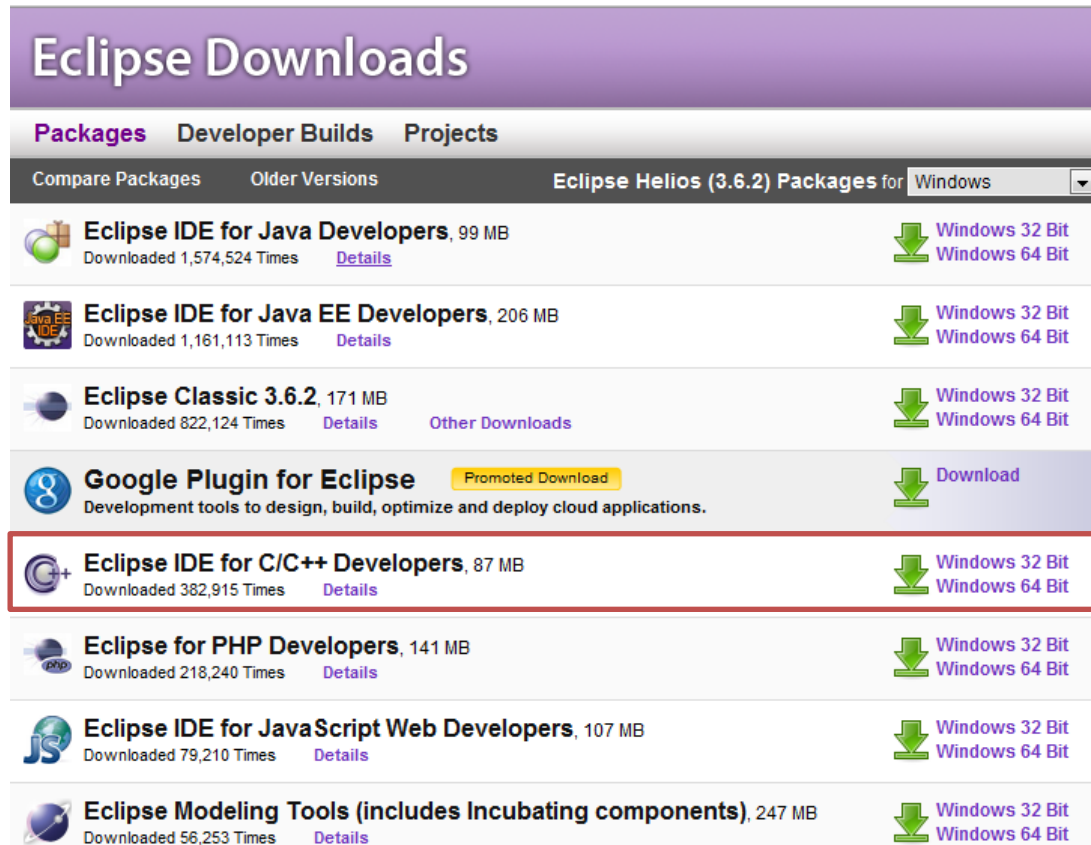
- Continuous Test & Integration Platform
- Code Editor
 - Eclipse CDT
- Unit Test
 - CUnit
- Source Code Management
 - Subclipse
- Integration Building
 - make command & makefile
- CI Tool
 - Hudson



Code Editor

❖ Eclipse IDE for C/C++ Developers

















■ <http://www.eclipse.org/downloads/>



Eclipse Downloads

Packages Developer Builds Projects

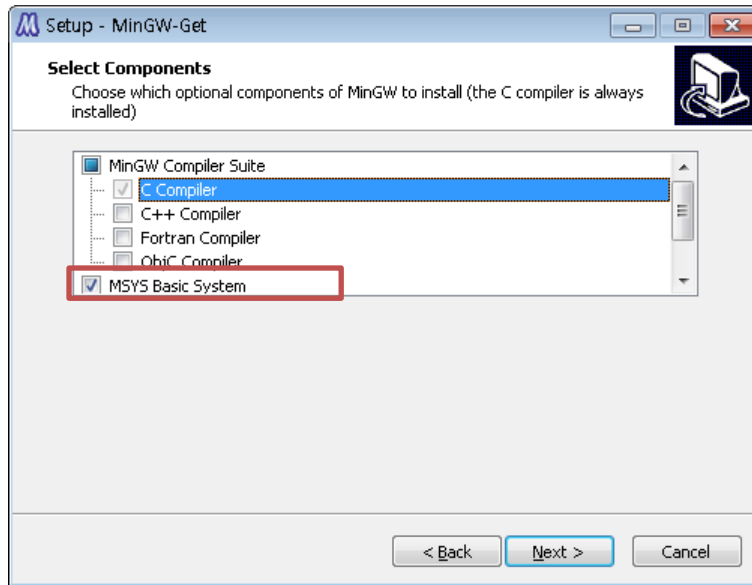
Compare Packages Older Versions **Eclipse Helios (3.6.2) Packages for Windows**

 Eclipse IDE for Java Developers , 99 MB Downloaded 1,574,524 Times Details	 Windows 32 Bit Windows 64 Bit
 Eclipse IDE for Java EE Developers , 206 MB Downloaded 1,161,113 Times Details	 Windows 32 Bit Windows 64 Bit
 Eclipse Classic 3.6.2 , 171 MB Downloaded 822,124 Times Details Other Downloads	 Windows 32 Bit Windows 64 Bit
 Google Plugin for Eclipse Promoted Download Development tools to design, build, optimize and deploy cloud applications.	 Download
 Eclipse IDE for C/C++ Developers , 87 MB Downloaded 382,915 Times Details	 Windows 32 Bit Windows 64 Bit
 Eclipse for PHP Developers , 141 MB Downloaded 218,240 Times Details	 Windows 32 Bit Windows 64 Bit
 Eclipse IDE for JavaScript Web Developers , 107 MB Downloaded 79,210 Times Details	 Windows 32 Bit Windows 64 Bit
 Eclipse Modeling Tools (includes Incubating components) , 247 MB Downloaded 56,253 Times Details	 Windows 32 Bit Windows 64 Bit

But Eclipse not support Compiler!!

Code Editor

- ❖ C Compiler for Windows
 - Mingw(Minimalist GNU for Windows)
 - <http://sourceforge.net/projects/mingw/>

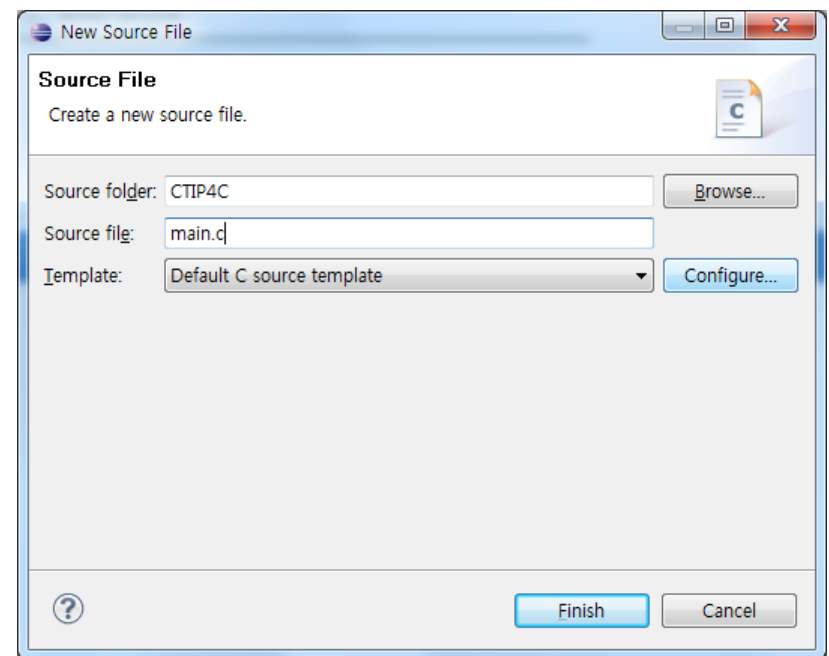
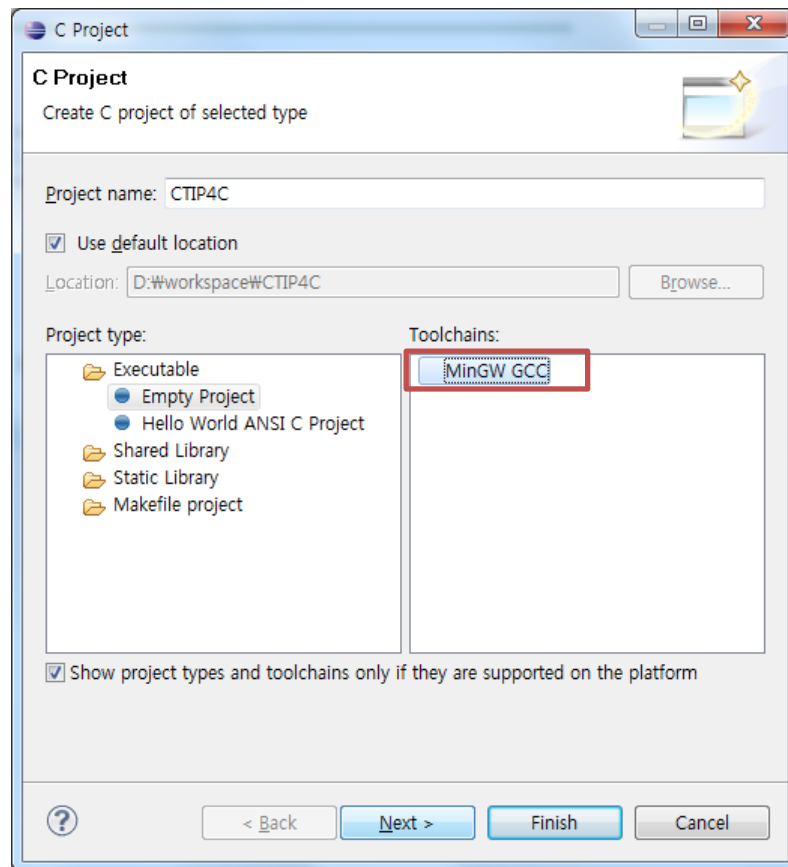


- MSYS(Minimal System)
- Provide shell environment
 - Install for installing CUnit

- Adding Mingw's bin folder to Path environment variables
 - C:\MinGW\bin
 - C:\MinGW\msys\1.0\bin

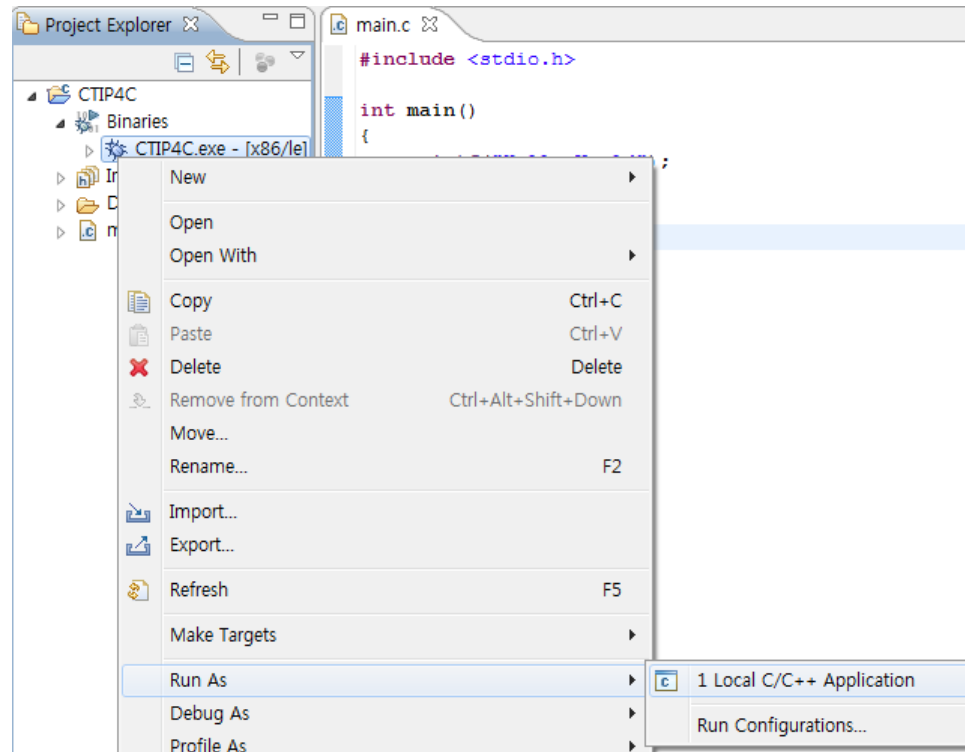
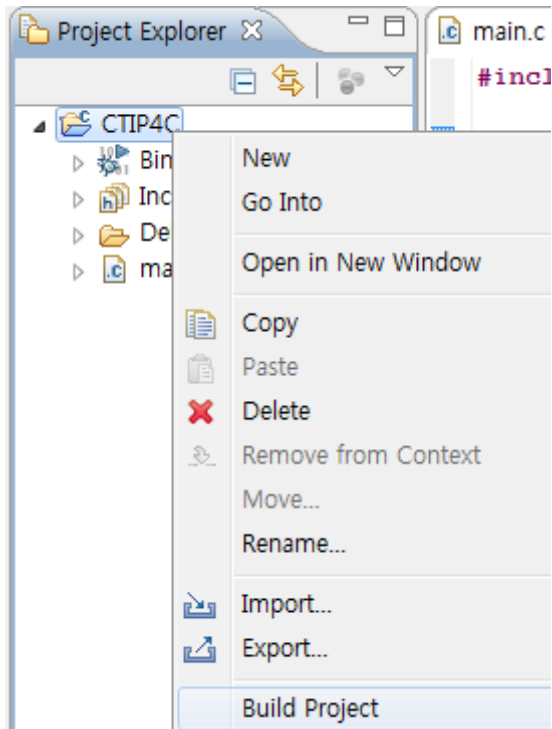
Code Editor

- ❖ Create Project, file
 - File>New>C Project, Source File



Code Editor

❖ Build / Run



Unit Test

❖ CUnit

- Unit Testing Framework for C
- Platform dependent (Curses on Unix)
- <http://cunit.sourceforge.net/>

CUnit

A Unit Testing Framework for C

Overview
[Documentation](#)
[Screenshots](#)
[Contacts](#)
[Example Code](#)

Download Link
[Project Home](#)

SOURCEFORGE.NET®

W3C HTML 4.01 ✓

CUnit is a lightweight system for writing, administering, and running unit tests in C. It provides C programmers a basic testing functionality with a flexible variety of user interfaces.

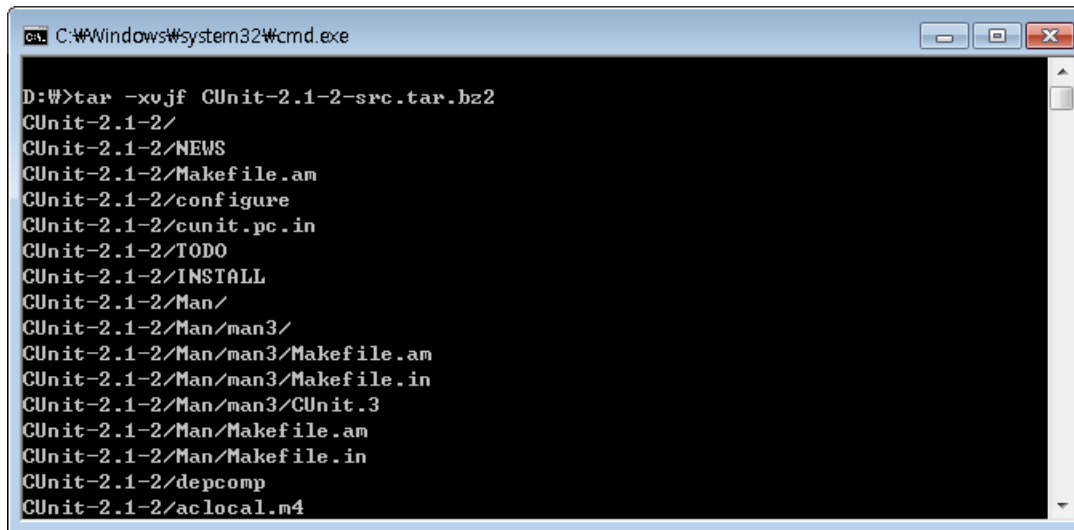
CUnit is built as a static library which is linked with the user's testing code. It uses a simple framework for building test structures, and provides a rich set of assertions for testing common data types. In addition, several different interfaces are provided for running tests and reporting results. These interfaces currently include:

Automated	Output to xml file	Non-interactive
Basic	Flexible programming interface	Non-interactive
Console	Console interface (ansi C)	Interactive
Curses	Graphical interface (Unix)	Interactive

Unit Test

❖ Install CUnit(1/3)

- Download "CUnit-2.1-2-src.tar.bz2"
 - <http://sourceforge.net/projects/cunit/>
- Decompression File
 - `tar -xvjf CUnit-2.1-2-src.tar.bz2`



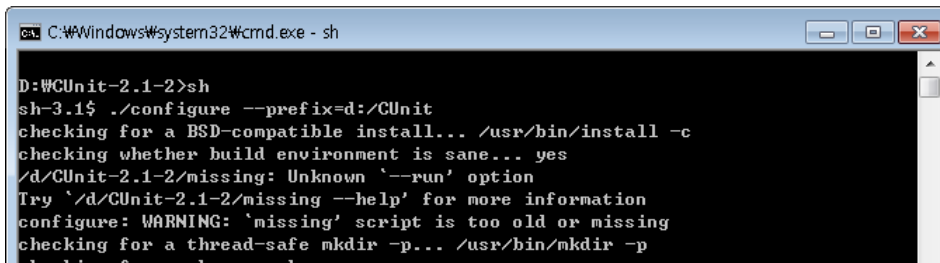
```
C:\Windows\system32\cmd.exe

D:\W>tar -xvjf CUnit-2.1-2-src.tar.bz2
CUnit-2.1-2/
CUnit-2.1-2/NEWS
CUnit-2.1-2/Makefile.am
CUnit-2.1-2/configure
CUnit-2.1-2/cunit.pc.in
CUnit-2.1-2/TODO
CUnit-2.1-2/INSTALL
CUnit-2.1-2/Man/
CUnit-2.1-2/Man/man3/
CUnit-2.1-2/Man/man3/Makefile.am
CUnit-2.1-2/Man/man3/Makefile.in
CUnit-2.1-2/Man/man3/CUnit.3
CUnit-2.1-2/Man/Makefile.am
CUnit-2.1-2/Man/Makefile.in
CUnit-2.1-2/depcomp
CUnit-2.1-2/aclocal.m4
```

Unit Test

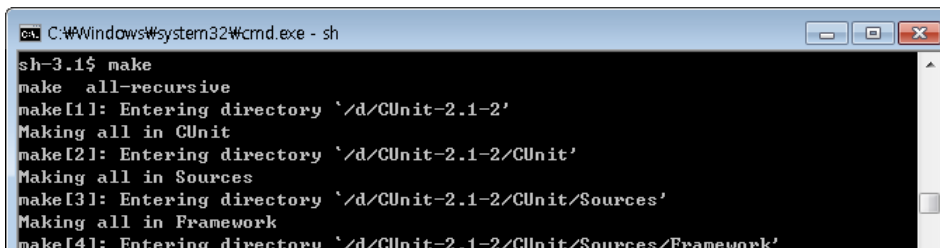
❖ Install CUnit(2/3)

- shell mode
 - sh (MSYS command)
- set Install configure
 - ./configure --prefix=<Install path>



```
C:\Windows\system32\cmd.exe - sh
D:\CUnit-2.1-2>sh
sh-3.1$ ./configure --prefix=d:/CUnit
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
/d/CUnit-2.1-2/missing: Unknown '--run' option
Try '/d/CUnit-2.1-2/missing --help' for more information
configure: WARNING: 'missing' script is too old or missing
checking for a thread-safe mkdir -p... /usr/bin/mkdir -p
checking for awk...
```

- Generate Install makefile by configure
 - make



```
C:\Windows\system32\cmd.exe - sh
sh-3.1$ make
make all-recursive
make[1]: Entering directory '/d/CUnit-2.1-2'
Making all in CUnit
make[2]: Entering directory '/d/CUnit-2.1-2/CUnit'
Making all in Sources
make[3]: Entering directory '/d/CUnit-2.1-2/CUnit/Sources'
Making all in Framework
make[4]: Entering directory '/d/CUnit-2.1-2/CUnit/Sources/Framework'
```

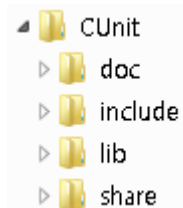
Unit Test

❖ Install CUnit(3/3)

■ Install CUnit

- make install

```
ca: C:\Windows\system32\cmd.exe - sh
sh-3.1$ make install
Making install in CUnit
make[1]: Entering directory `/d/CUnit-2.1-2/CUnit'
Making install in Sources
make[2]: Entering directory `/d/CUnit-2.1-2/CUnit/Sources'
Making install in Framework
make[3]: Entering directory `/d/CUnit-2.1-2/CUnit/Sources/Framework'
make[4]: Entering directory `/d/CUnit-2.1-2/CUnit/Sources/Framework'
make[4]: Nothing to be done for `install-exec-am'.
make[4]: Nothing to be done for `install-data-am'.
make[4]: Leaving directory `/d/CUnit-2.1-2/CUnit/Sources/Framework'
make[3]: Leaving directory `/d/CUnit-2.1-2/CUnit/Sources/Framework'
Making install in Automated
make[3]: Entering directory `/d/CUnit-2.1-2/CUnit/Sources/Automated'
make[4]: Entering directory `/d/CUnit-2.1-2/CUnit/Sources/Automated'
make[4]: Nothing to be done for `install-exec-am'.
make[4]: Nothing to be done for `install-data-am'.
make[4]: Leaving directory `/d/CUnit-2.1-2/CUnit/Sources/Automated'
```



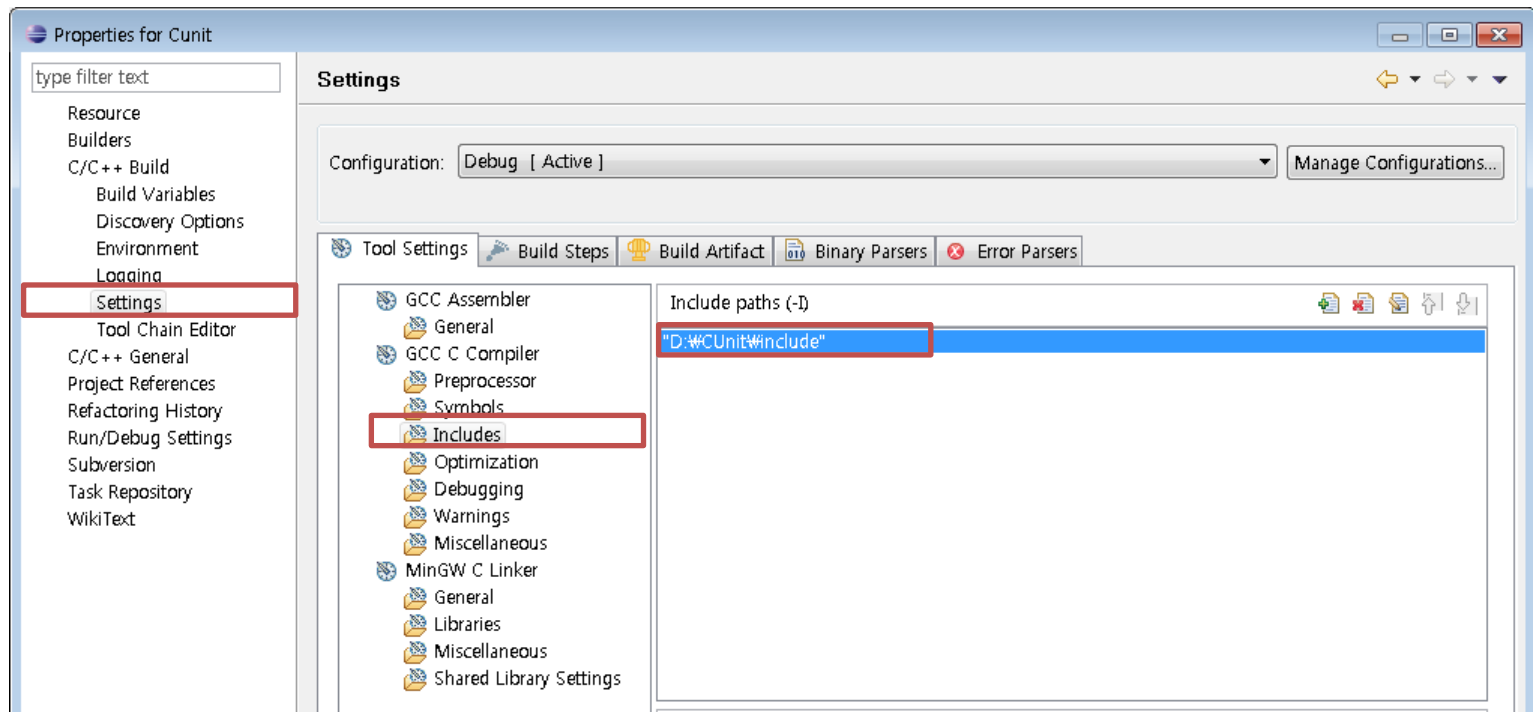
- doc : CUnit document(Html)
- include : Header files
- lib : Libraries
- share : XML DTD, XSL, man files

Unit Test

❖ Setting for Eclipse

■ Setting Compiler's Include path

- Project > Properties > C/C++ Build > Settings
- GCC C Compiler > Includes
- add CUnit's include folder

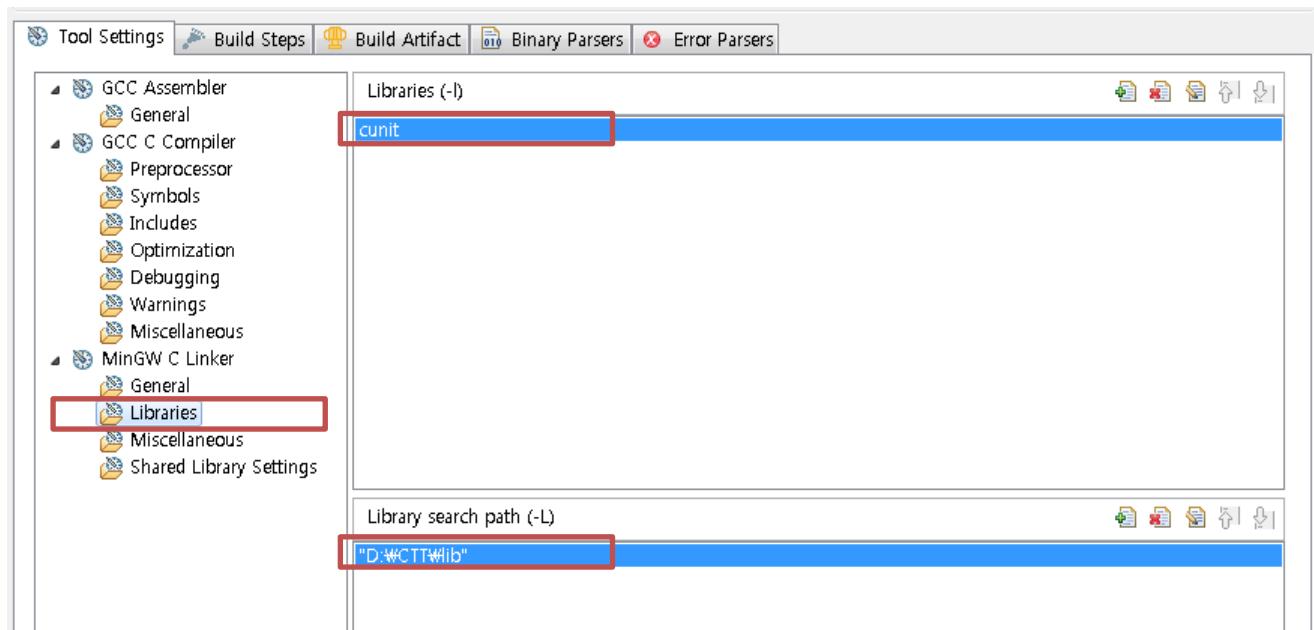


Unit Test

❖ Setting for Eclipse

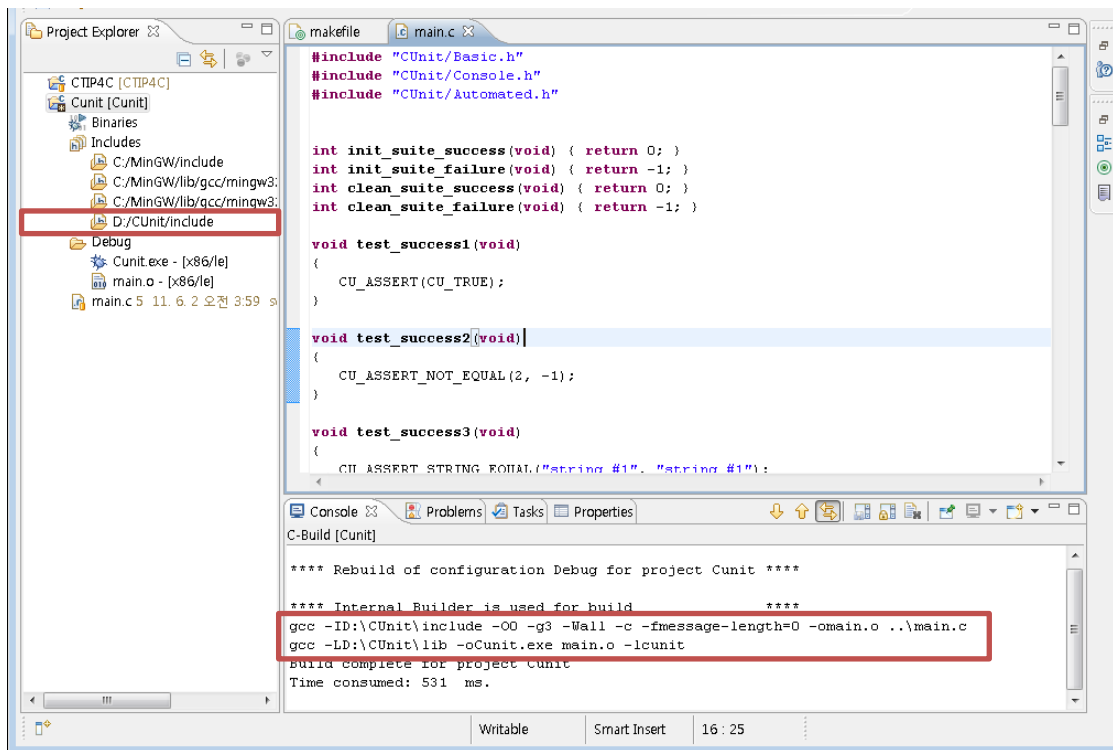
■ Setting Linker's Libraries

- Project > Properties > C/C++ Build > Settings
- MinGW C Linker > Libraries
- add Libraries "cunit"
- add CUnit's lib folder



Unit Test

- ❖ Test CUnit(1/3)
 - Using CUnit's Example Code
 - Build



Unit Test

❖ Test CUnit(2/3)

■ Run Executable file

- Using Basic.h

```
CUnit - A unit testing framework for C - Version 2.1-2
http://cunit.sourceforge.net/

Suite: Suite_success
  Test: successful_test_1 ...passed
  Test: successful_test_2 ...passed
  Test: successful_test_3 ...passed
WARNING - Suite initialization failed for 'Suite_init_failure'.
Suite: Suite_clean_failure
  Test: successful_test_4 ...passed
  Test: failed_test_2 ...FAILED
    1. ..\main.c:38 - CU_ASSERT_EQUAL(2,3)
  Test: successful_test_1 ...passed
WARNING - Suite cleanup failed for 'Suite_clean_failure'.
Suite: Suite_mixed
  Test: successful_test_2 ...passed
  Test: failed_test_4 ...FAILED]
    1. ..\main.c:48 - CU_ASSERT_STRING_EQUAL("string #1","string #2")
  Test: failed_test_2 ...FAILED
    1. ..\main.c:38 - CU_ASSERT_EQUAL(2,3)
  Test: successful_test_4 ...passed

Run Summary:
  Type   Total   Ran Passed Failed Inactive
  suites    4     3   n/a     2     0
  tests   13    10     7     3     0
  asserts  10    10     7     3     n/a

Elapsed time = 0.001 seconds

1. CUnit System:0 - Suite Initialization failed - Suite Skipped
2. ..\main.c:38 - CU_ASSERT_EQUAL(2,3)
3. CUnit System:0 - Suite cleanup failed.
4. ..\main.c:48 - CU_ASSERT_STRING_EQUAL("string #1","string #2")
5. ..\main.c:38 - CU_ASSERT_EQUAL(2,3)
```

- Using Console.h

```
CUnit - A Unit testing framework for C - Version 2.1-2
http://cunit.sourceforge.net/

***** CUNIT CONSOLE - MAIN MENU *****
(R)un (S)elect (L)ist (A)ctivate (F)ailures (O)ptions (H)elp (Q)uit
Enter command:

***** CUNIT CONSOLE - MAIN MENU *****
(R)un (S)elect (L)ist (A)ctivate (F)ailures (O)ptions (H)elp (Q)uit
Enter command: F

----- Test Run Failures -----
src_file:line#: (suite:test) : failure_condition

1. CUnit System:0 : (Suite_init_failure : ) : Suite Initialization failed -
2. ..\main.c:38 : (Suite_clean_failure : failed_test_2) : CU_ASSERT_EQUAL(2
3. CUnit System:0 : (Suite_clean_failure : ) : Suite cleanup failed.
4. ..\main.c:48 : (Suite_mixed : failed_test_4) : CU_ASSERT_STRING_EQUAL("s
5. ..\main.c:38 : (Suite_mixed : failed_test_2) : CU_ASSERT_EQUAL(2,3)

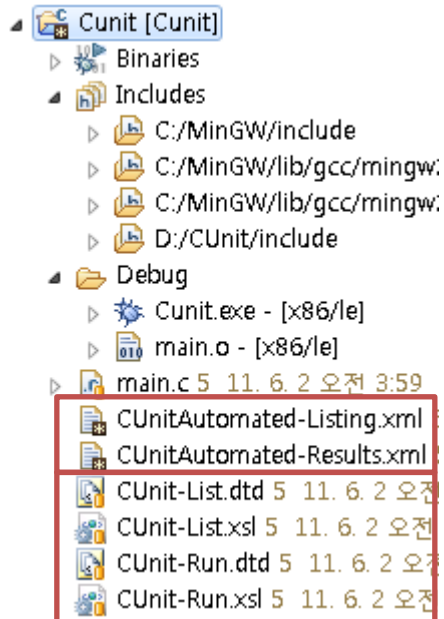
-----
Total Number of Failures : 5
```

Unit Test

❖ Test CUnit(3/3)

■ Run Executable file

- Using Automated.h



- Executable file generate XML file

- Use CUnit/share's DTD, XSL files

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" href="CUnit-Run.xsl" ?>
<!DOCTYPE CUNIT_TEST_RUN_REPORT SYSTEM "CUnit-Run.dtd">
<CUNIT_TEST_RUN_REPORT>
  <CUNIT_HEADER/>
  <CUNIT_RESULT_LISTING>
    <CUNIT_RUN_SUITE>
      <CUNIT_RUN_SUITE_SUCCESS>
        <SUITE_NAME> Suite_success </SUITE_NAME>
        <CUNIT_RUN_TEST_RECORD>
          <CUNIT_RUN_TEST_SUCCESS>
            <TEST_NAME> successful_test_1 </TEST_NAME>
          </CUNIT_RUN_TEST_SUCCESS>
        </CUNIT_RUN_TEST_RECORD>
        <CUNIT_RUN_TEST_RECORD>
          <CUNIT_RUN_TEST_SUCCESS>
            <TEST_NAME> successful_test_2 </TEST_NAME>
          </CUNIT_RUN_TEST_SUCCESS>
        </CUNIT_RUN_TEST_RECORD>
        <CUNIT_RUN_TEST_RECORD>
          <CUNIT_RUN_TEST_SUCCESS>
            <TEST_NAME> successful_test_3 </TEST_NAME>
          </CUNIT_RUN_TEST_SUCCESS>
        </CUNIT_RUN_TEST_RECORD>
      </CUNIT_RUN_SUITE_SUCCESS>
    </CUNIT_RUN_SUITE>
  </CUNIT_RESULT_LISTING>
</CUNIT_TEST_RUN_REPORT>
```

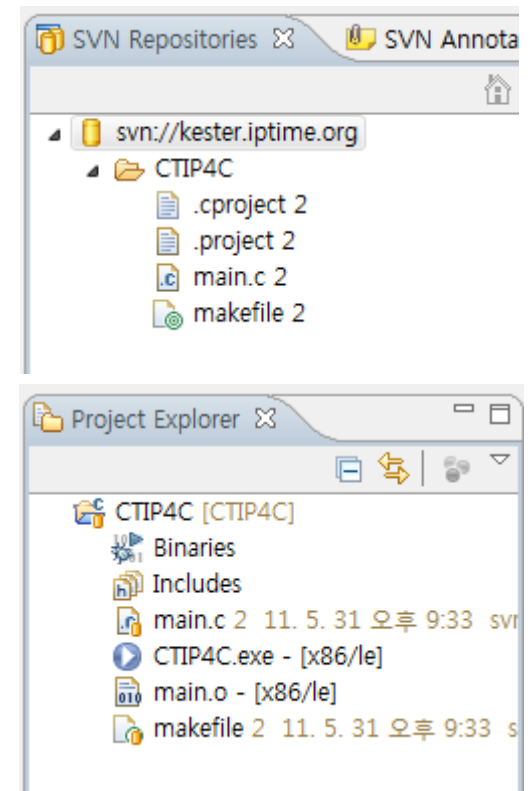
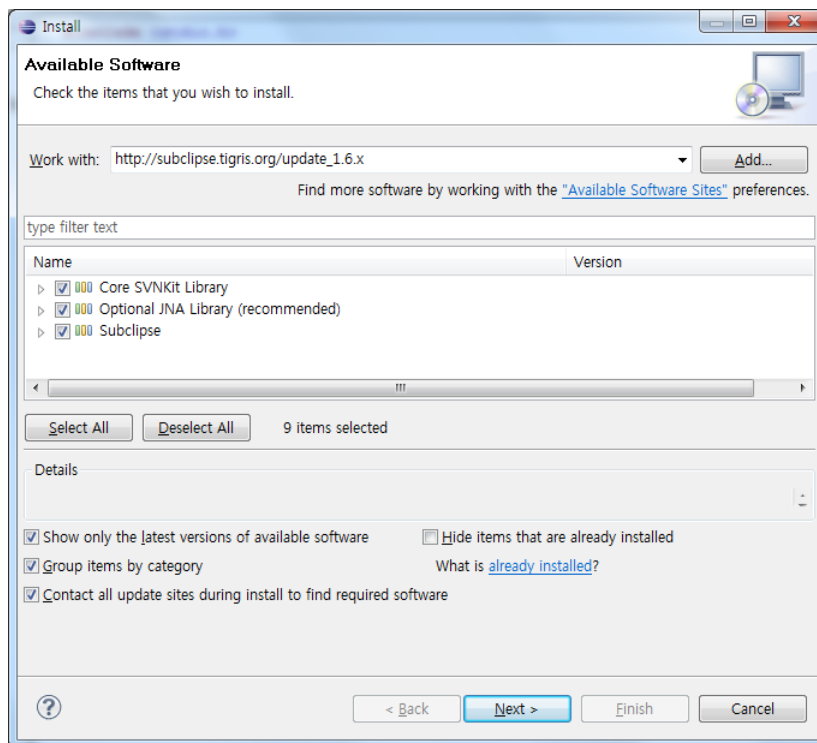
Automated Test Run Results		
Running Suite Suite_success		
Running test successful_test_1 ...	Passed	
Running test successful_test_2 ...	Passed	
Running test successful_test_3 ...	Passed	
Running Suite Suite_init_failure ...		
	Suite Initialization	Failed
Running Suite Suite_clean_failure		
Running test successful_test_4 ...	Passed	
Running test failed_test_2 ...	Failed	
File Name .\main.c	Line Number 38	
Condition CU_ASSERT_EQUAL(2,3)		
Running test successful_test_1 ...	Passed	
Running Suite Suite_clean_failure ...		
	Suite Cleanup	Failed

Listing of Registered Suites & Tests				
	Initialize Function?	Cleanup Function?	Test Count	Active?
Suite Suite_success	Yes	Yes	3	Yes
Test successful_test_1				Yes
Test successful_test_2				Yes
Test successful_test_3				Yes
Suite Suite_init_failure	Yes	No	3	Yes
Test successful_test_1				Yes
Test successful_test_2				Yes
Test successful_test_3				Yes
Suite Suite_clean_failure	No	Yes	3	Yes
Test successful_test_4				Yes
Test failed_test_2				Yes
Test successful test 1				Yes

SCM

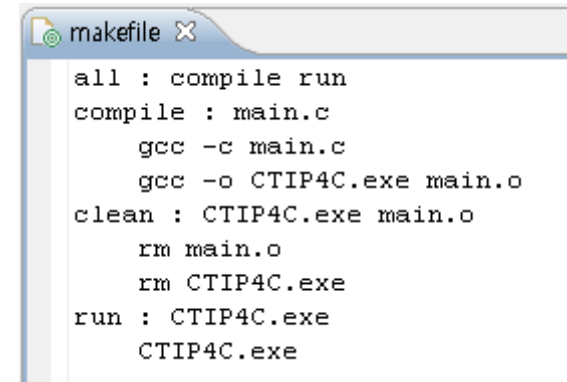
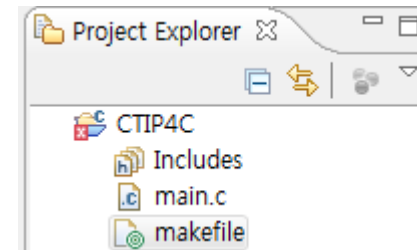
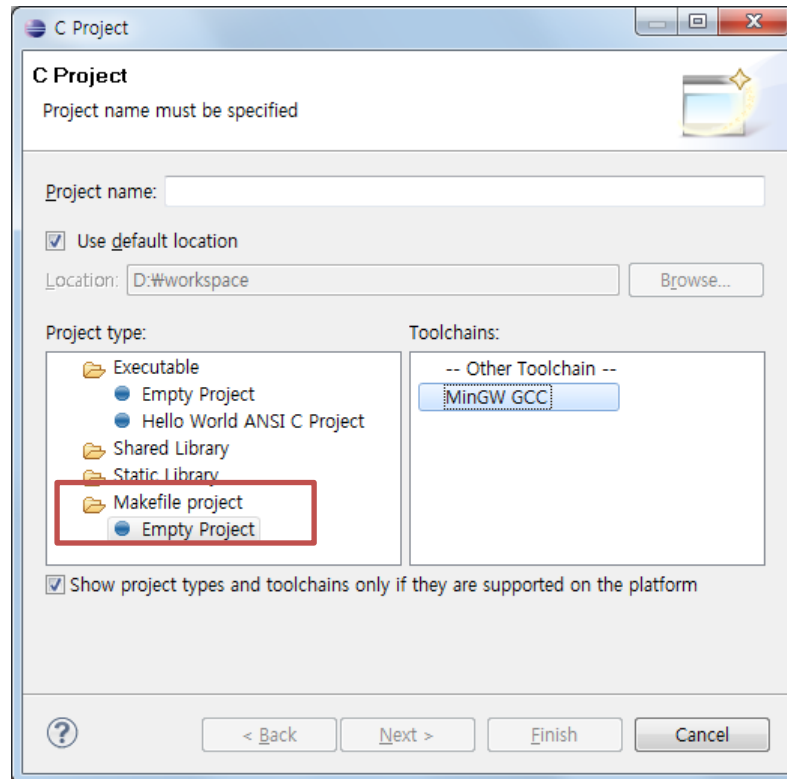
❖ Source Code Management

- Subversion - Subclipse
- Help>Install New Software
- http://subclipse.tigris.org/update_1.6.x



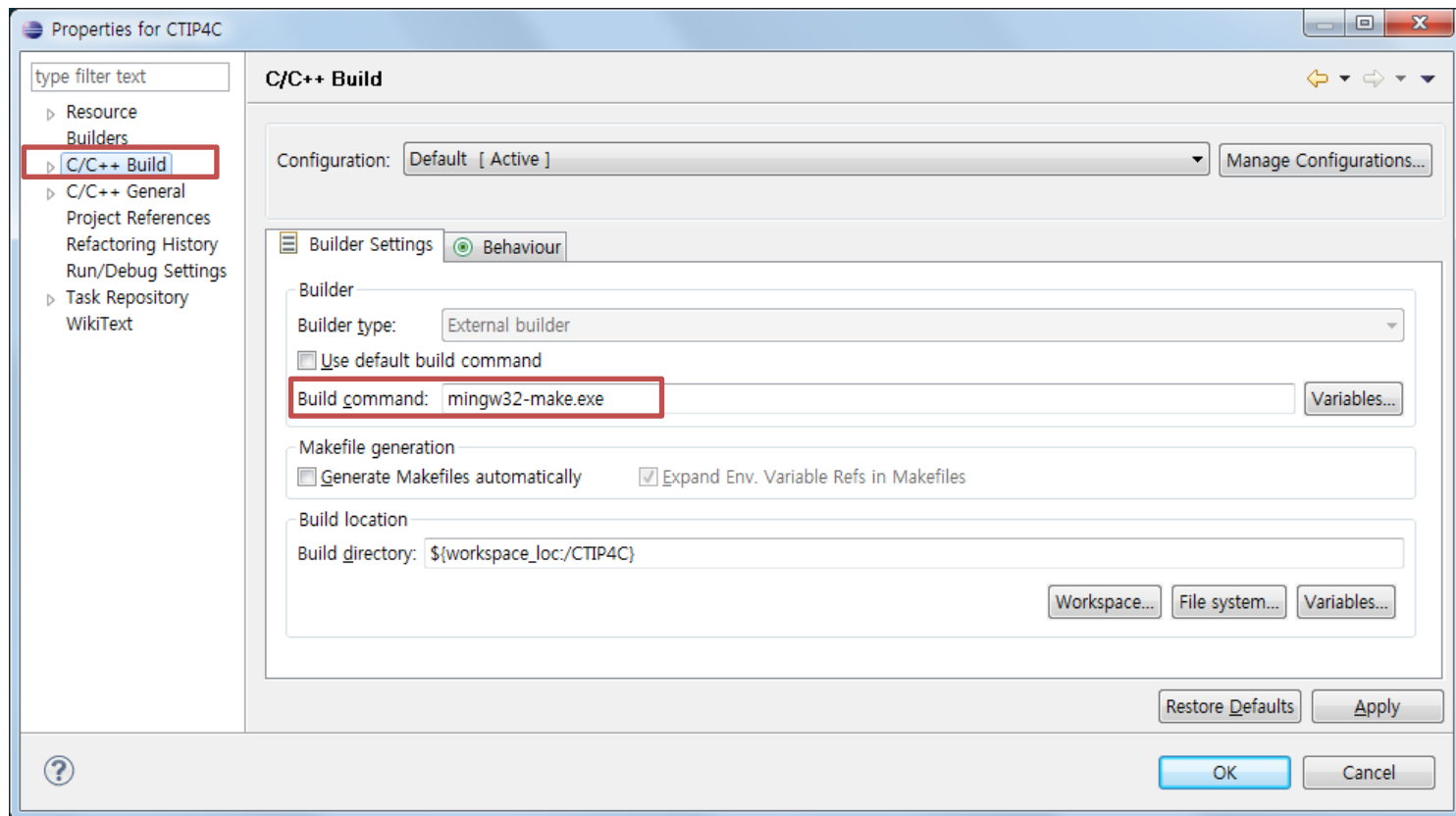
Integration Building

- ❖ Use makefile for Integration building
 - similar to build script
 - Create Makefile Project, makefile
 - New>C Project, File



Integration Building

- ❖ Use mingw32-make.exe
 - Project > Properties > C/C++ Build > Builder Settings
 - Set Build command : mingw32-make.exe



Integration Building

❖ Edit makefile

- Build Default : Label "all"

The image shows a code editor window titled 'makefile' containing the following content:

```
all : compile run
compile : main.c
    gcc -c main.c
    gcc -o CTIP4C.exe main.o
clean : CTIP4C.exe main.o
    rm main.o
    rm CTIP4C.exe
run : CTIP4C.exe
    CTIP4C.exe
```

Annotations in the image:

- A red box highlights the 'compile' label and its dependency 'main.c'. An arrow points from the text 'dependency' to 'main.c'.
- A blue box highlights the two gcc commands. An arrow points from the text 'execute statement' to the first command.
- An arrow points from the text 'Label' to the 'all' label.
- An arrow points from the text 'must use tab' to the indentation of the gcc commands.

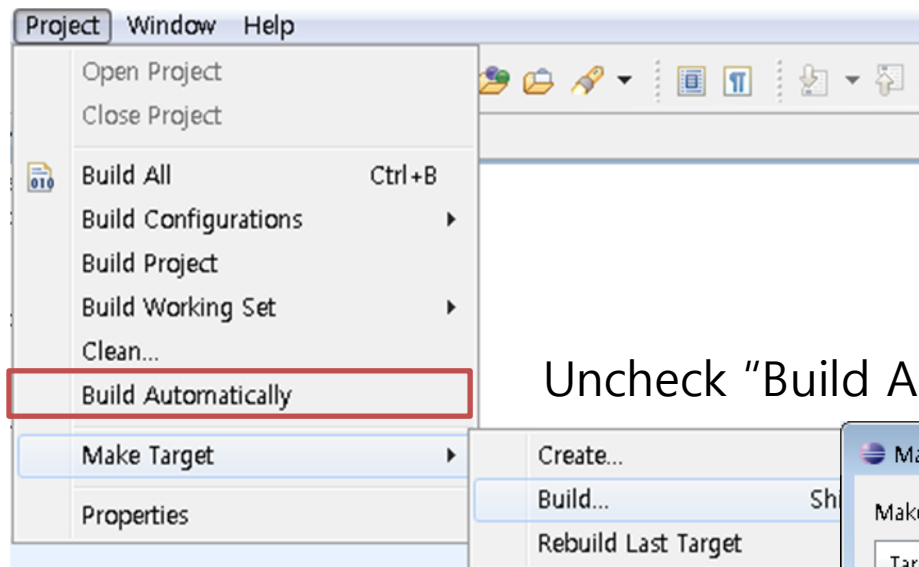
A green arrow labeled 'Build' points from the makefile editor to a console window titled 'C-Build [CTIP4C]'. The console output is:

```
**** Build of configuration Default
mingw32-make.exe all
gcc -c main.c
gcc -o CTIP4C.exe main.o
CTIP4C.exe
Hello World
```

Integration Building

❖ Selective Build

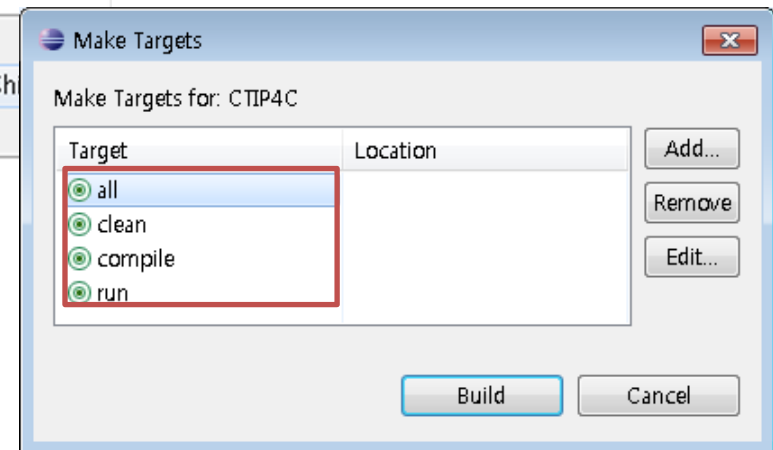
- Project > Make Target > Build



Uncheck "Build Automatically"

Target is makefile's label

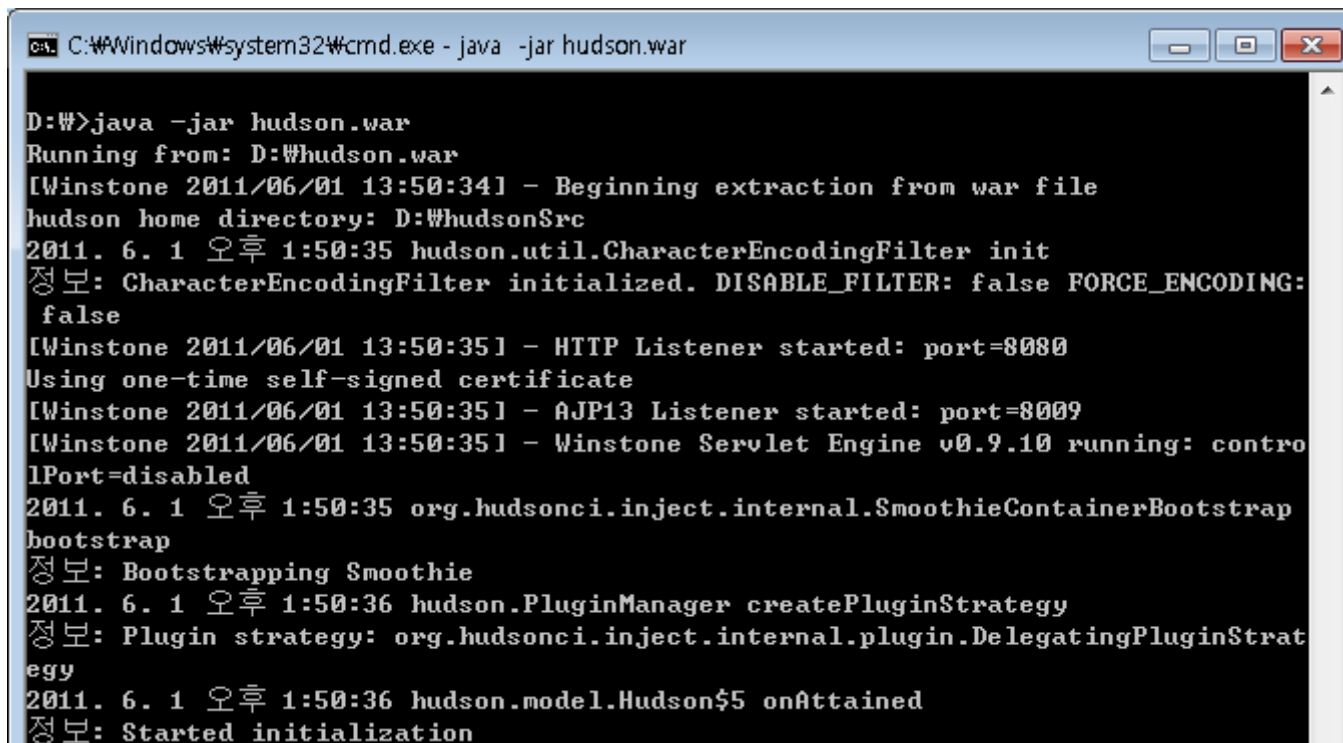
Selective Build in Command-line
: mingw32-make <label>



CI Server

❖ Hudson

- <http://hudson-ci.org/>
- Download "hudson.war"
- `java -jar hudson.war`



```
C:\Windows\system32\cmd.exe - java -jar hudson.war

D:\>java -jar hudson.war
Running from: D:\hudson.war
[Winstone 2011/06/01 13:50:34] - Beginning extraction from war file
hudson home directory: D:\hudsonSrc
2011. 6. 1 오후 1:50:35 hudson.util.CharacterEncodingFilter init
정보: CharacterEncodingFilter initialized. DISABLE_FILTER: false FORCE_ENCODING:
false
[Winstone 2011/06/01 13:50:35] - HTTP Listener started: port=8080
Using one-time self-signed certificate
[Winstone 2011/06/01 13:50:35] - AJP13 Listener started: port=8009
[Winstone 2011/06/01 13:50:35] - Winstone Servlet Engine v0.9.10 running: contro
lPort=disabled
2011. 6. 1 오후 1:50:35 org.hudsonci.inject.internal.SmoothieContainerBootstrap
bootstrap
정보: Bootstrapping Smoothie
2011. 6. 1 오후 1:50:36 hudson.PluginManager createPluginStrategy
정보: Plugin strategy: org.hudsonci.inject.internal.plugin.DelegatingPluginStrat
egy
2011. 6. 1 오후 1:50:36 hudson.model.Hudson$5 onAttained
정보: Started initialization
```

CI Server

❖ Hudson

- http://<ip address>:8080
- Create New Job
- Select "Build a free-style software project"

The screenshot shows the Hudson web interface. At the top, there's a navigation bar with the 'Hudson' logo, a search box, and the user 'hmanager' with a '로그아웃' (Logout) link. Below the navigation bar, there's a breadcrumb 'Hudson > All' and a sidebar with icons for '새 작업' (New Job), 'Hudson 관리' (Hudson Management), '개발자' (Developer), '빌드 기록' (Build History), and 'My Views'. The main content area is titled '작업명' (Job Name) and contains a text input field with 'CTIP4C'. Below this, there are several radio button options for job types: 'Build a free-style software project' (selected), 'Build a maven2/3 project', 'Monitor an external job', 'Build multi-configuration project', and '기존 작업 복사' (Copy Existing Job). Each option has a brief description. At the bottom, there's an 'OK' button and a 'Copy from' field with a text input box.

Hudson > All

새 작업
Hudson 관리
개발자
빌드 기록
My Views

빌드 대기 목록
빌드 대기 항목이 없습니다.

빌드 실행 상태

#	상태
1	대기 중
2	대기 중

작업명

- Build a free-style software project**
이것은 Hudson의 주요 기능입니다. Hudson은 어느 빌드 시스템과 어떤 SCM(형상관리)으로 묶인 당신의 프로젝트를 빌드할 것이고, 소프트웨어 빌드보다 다른 어떤 것에 자주 사용될 수 있습니다.
- Build a maven2/3 project**
Maven2/3 프로젝트를 빌드합니다. Hudson은 POM 파일의 이점을 가지고 있고 급격히 설정을 줄입니다.
- Monitor an external job**
이 유형의 작업은 원격 장비처럼 Hudson 외부에서 동작하는 프로세스의 실행을 기록하는 것을 허용합니다. 그렇게 설계되어서, 기존의 자동 시스템의 대시보드로서 Hudson을 사용할 수 있습니다. 자세한 설명은 [여기\(영문\)](#)을 보세요.
- Build multi-configuration project**
다양한 환경에서의 테스트, 플랫폼 특성 빌드, 기타 등을 처럼 다수의 서로다른 환경설정이 필요한 프로젝트에 적합함.
- 기존 작업 복사**
Copy from

OK

CI Server

❖ Job Configure

Project name

Description

Discard Old Builds

This build is parameterized

Disable Build (No new builds will be executed until the project is re-enabled.)

Execute concurrent builds if necessary (beta)

Advanced Project Options

Source Code Management

None

CVS

Subversion

Modules

Local module directory (optional)

Enroll Repository

Ignore externals option

Check-out Strategy

Use 'svn update' whenever possible, making the build faster. But this causes the artifacts from the previous build to remain when a new build starts.

Repository browser

Build Triggers

Build after other projects are built

Trigger builds remotely (e.g., from scripts)

Poll SCM

Build periodically

Build

Execute Windows batch command

Command

See [the list of available environment variables](#)

Execute Windows batch command

Invoke Ant

Execute shell

Invoke top-level Maven targets

Perform Subversion tagging on successful build

Use Excute Windows batch command : "mingw32-make.exe" build project using makefile

CI Server

❖ Build

The screenshot displays the Hudson web interface for the project 'CTIPForC'. On the left sidebar, the 'Build Now' button is highlighted with a red box. Below it, the 'Build History' table shows a list of builds, with build #9 (dated 2011. 6. 1 오후 1:58:12) selected and highlighted with a red box. A green arrow points from this build entry to the main view of build #9. In the main view, the 'Console 출력' (Console Output) link is also highlighted with a red box, and a green arrow points down to the console output area. The console output shows the build process starting with 'Started by user huser', updating SVN, and successfully compiling and running the program 'CTIP4C.exe', which prints 'Hello World'.

Hudson
Hudson > CTIPForC

Back to Dashboard
Status
Changes
Workspace
Build Now

Project CTIPForC

Hudson > CTIPForC > #9

프로젝트로 돌아가기
상태
변경사항
콘솔 출력
Configure
Take this build
Build again

빌드 #9 (2011. 6. 1 오후 1:58:12)

Revision: 3
No changes.
사용자 k에 의해 시작됨

콘솔 출력

```
Started by user huser
Updating svn://kester.iptime.org/CTIP4C revision: 2011. 6. 1 오후 1:58:12 depth:infinity ignoreExternals: false
At revision 3
no change for svn://kester.iptime.org/CTIP4C since the previous build
[workspace] $ cmd /c call C:\Users\kester\n\AppData\Local\Temp\hudson1622363393178990809.bat

D:\hudsonSrc\jobs\CTIPForC\workspace>mingw32-make
gcc -c main.c
gcc -o CTIP4C.exe main.o
CTIP4C.exe
Hello World
D:\hudsonSrc\jobs\CTIPForC\workspace>exit 0
Finished: SUCCESS
```

Unit Test - CUnit

CUnit

❖ Test Process

1. Write functions for testcases
 - suite init/cleanup if necessary
2. Initialize the test registry
 - CU_initialize_registry()
3. Add suites to the test registry
 - CU_add_suite()
4. Add tests to the suites
 - CU_add_test()
5. Run tests using an appropriate interface
 - CU_basic_run_tests()
 - CU_automated_run_tests()
 - CU_console_run_tests()
6. Cleanup the test registry
 - CU_cleanup_registry

CUnit

❖ Write functions for Testcases

■ Using CUnit Assertions

- Similar with JUnit
- Defined by CUnit.h

❖ Assertion Functions (1/2)

CU_ASSERT (int expression) CU_TEST (int expression)	Assert that <i>expression</i> is TRUE (non-zero)
CU_ASSERT_TRUE (value) CU_ASSERT_FALSE (value)	Assert that <i>value</i> is TRUE (non-zero) Assert that <i>value</i> is FALSE (zero)
CU_ASSERT_EQUAL (actual, expected) CU_ASSERT_NOT_EQUAL (actual, expected))	Assert that <i>actual</i> == <i>expected</i> Assert that <i>actual</i> != <i>expected</i>
CU_PASS (message)	Register a passing assertion with the specified message. No logical test is performed.
CU_FAIL (message)	Register a failed assertion with the specified message. No logical test is performed.

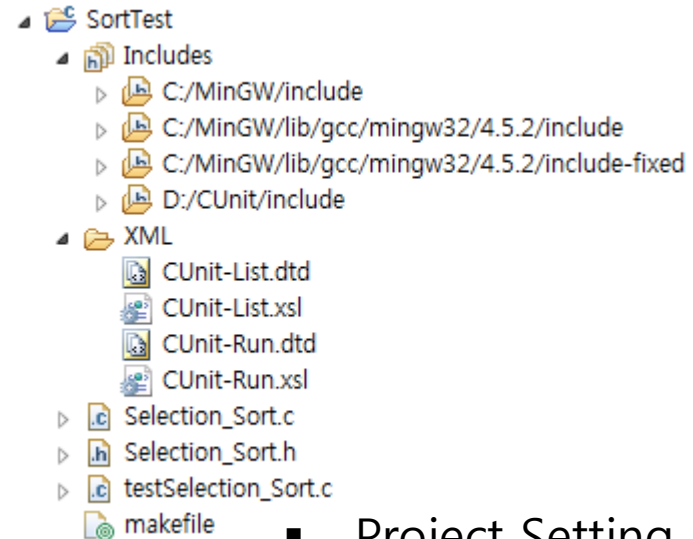
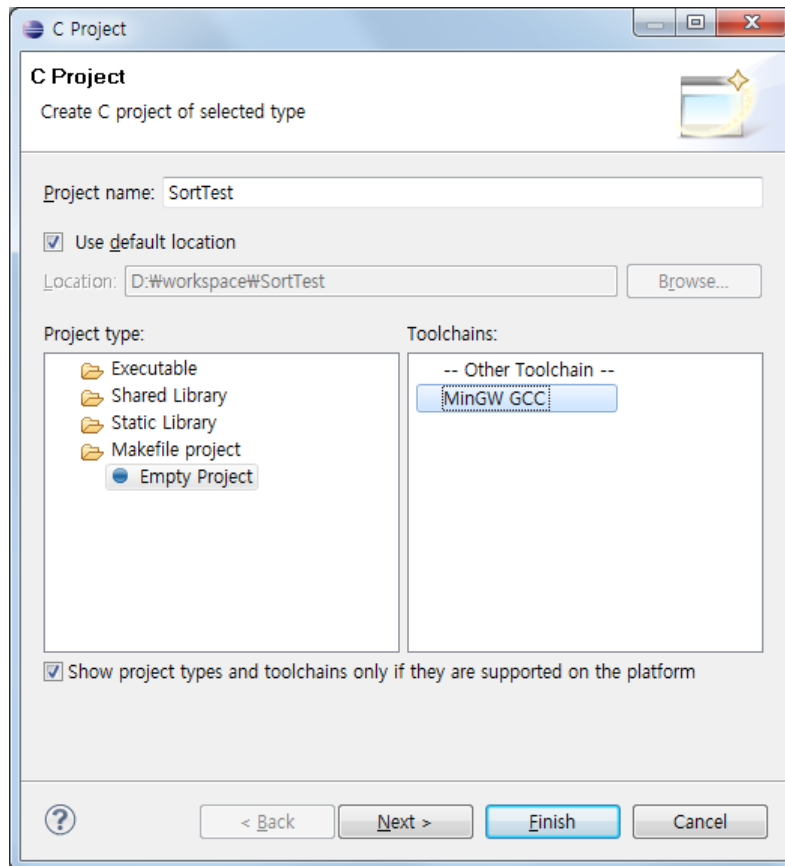
CUnit

❖ Assertion Functions (2/2)

CU_ASSERT_PTR_EQUAL (<i>actual</i> , <i>expected</i>) CU_ASSERT_PTR_NOT_EQUAL (<i>actual</i> , <i>expected</i>)	Assert that pointers <i>actual</i> == <i>expected</i> Assert that pointers <i>actual</i> != <i>expected</i>
CU_ASSERT_PTR_NULL (<i>value</i>) CU_ASSERT_PTR_NOT_NULL (<i>value</i>)	Assert that pointer <i>value</i> == NULL Assert that pointer <i>value</i> != NULL
CU_ASSERT_STRING_EQUAL (<i>actual</i> , <i>expected</i>) CU_ASSERT_STRING_NOT_EQUAL (<i>actual</i> , <i>expected</i>)	Assert that strings <i>actual</i> and <i>expected</i> are equivalent Assert that strings <i>actual</i> and <i>expected</i> differ
CU_ASSERT_NSTRING_EQUAL (<i>actual</i> , <i>expected</i> , <i>count</i>) CU_ASSERT_NSTRING_NOT_EQUAL (<i>actual</i> , <i>expected</i> , <i>count</i>)	Assert that 1st count chars of <i>actual</i> and <i>expected</i> are the same Assert that 1st count chars of <i>actual</i> and <i>expected</i> differ
CU_ASSERT_DOUBLE_EQUAL (<i>actual</i> , <i>expected</i> , <i>granularity</i>) CU_ASSERT_DOUBLE_NOT_EQUAL (<i>actual</i> , <i>expected</i> , <i>granularity</i>)	Assert that $ actual - expected \leq granularity $ Assert that $ actual - expected > granularity $ <i>Math library must be linked in for this assertion.</i>

Sort Test

❖ Create Project



- Project Setting
 - include directory
 - library path
- copy from CUnit/share
- add Files to Test
- add Testing file
- add makefile for Build

Sort Test

❖ Write Testcase

■ Testsuite1

```
void test_exp(void) {
    char array[MAX_SIZE]="abedea";
    selection_sort(array);
    CU_ASSERT(array[0] == 'a');
    CU_ASSERT(array[1] == 'a');
    CU_ASSERT(array[2] == 'b');
    CU_ASSERT(array[3] == 'd');
    CU_ASSERT(array[4] == 'e');
    CU_ASSERT(array[5] == 'e');
}

void test_true(void) {
    int isEqual;
    char array[MAX_SIZE]="adebcf";
    selection_sort(array);
    if(strcmp(array,"abcdef")==0)
        isEqual = TRUE;
    else
        isEqual = FALSE;
    CU_ASSERT_TRUE(isEqual);
}

void test_equal(void) {
    char array[MAX_SIZE]="abedea";
    selection_sort(array);
    CU_ASSERT_EQUAL(array[0], 'a');
    CU_ASSERT_EQUAL(array[1], 'a');
    CU_ASSERT_EQUAL(array[2], 'b');
    CU_ASSERT_EQUAL(array[3], 'd');
    CU_ASSERT_EQUAL(array[4], 'e');
    CU_ASSERT_EQUAL(array[5], 'e');
}

void test_string(void) {
    char array[MAX_SIZE]="abedea";
    selection_sort(array);
    CU_ASSERT_STRING_EQUAL(array, "abcdef");
}

void test_fail(void) {
    char array[MAX_SIZE]="abedea";
    selection_sort(array);
    CU_FAIL("fail test_fail");
}
```

■ Testsuite2

```
void test_int(void) {
    char array[MAX_SIZE]="582547";
    selection_sort(array);
    CU_ASSERT_STRING_EQUAL(array, "245578");
}

void test_capital(void) {
    char array[MAX_SIZE]="EMCDOH";
    selection_sort(array);
    CU_ASSERT_STRING_EQUAL(array, "CDEHMO");
}

void test_special(void) {
    char array[MAX_SIZE]="(^@=#>";
    selection_sort(array);
    CU_ASSERT_STRING_EQUAL(array, "#(*=>@^");
}

void test_mix(void) {
    char array[MAX_SIZE]="1a6V#e";
    selection_sort(array);
    CU_ASSERT_STRING_EQUAL(array, "#16Vae");
}
```

Sort Test

❖ Write main Function for Testing

```
int main()
{
    CU_pSuite testSuite=NULL;
    /* initialize the CUnit test registry */
    CU_initialize_registry();
    /* add a suite to the registry */
    testSuite = CU_add_suite("ASSERT Test", NULL, NULL);
    /* add the tests to the suite */
    CU_add_test(testSuite, "ASSERT TEST", test_exp);
    CU_add_test(testSuite, "ASSERT TRUE TEST", test_true);
    CU_add_test(testSuite, "ASSERT EQUAL TEST", test_equal);
    CU_add_test(testSuite, "ASSERT STRING TEST", test_string);
    CU_add_test(testSuite, "ASSERT Fail Test", test_fail);

    testSuite = CU_add_suite("Type Test", NULL, NULL);
    CU_add_test(testSuite, "TYPE INT TEST", test_true);
    CU_add_test(testSuite, "TYPE CAPITAL TEST", test_equal);
    CU_add_test(testSuite, "TYPE SPECIAL TEST", test_string);
    CU_add_test(testSuite, "TYPE MIX TESTt", test_fail);

    /* Run all tests using the automated interface */
    CU_automated_run_tests();
    CU_list_tests_to_file();

    /* Clean up registry and return */
    CU_cleanup_registry();
    return CU_get_error();
}
```

Declaration Suite Variable

Initialize Registry

add Suite to Registry

add Testcases to Suite

Run Test - XML Output

Clean up Registry

Sort Test

❖ Write makefile

```
all : compile link run move
compile : Selection_Sort.c testSelection_Sort.c
    gcc -ID:\CUnit\include -o Selection_Sort.o -c Selection_Sort.c
    gcc -ID:\CUnit\include -o testSelection_Sort.o -c testSelection_Sort.c
link : testSelection_Sort.o Selection_Sort.o
    gcc -LD:\CUnit\lib -o SortTest.exe testSelection_Sort.o Selection_Sort.o -lcunit
run : SortTest.exe
    SortTest.exe
move : CUnitAutomated-Listing.xml CUnitAutomated-Results.xml
    mv CUnitAutomated-Listing.xml ./XML/Listing.xml
    mv CUnitAutomated-Results.xml ./XML/Results.xml
clean : Selection_Sort.o testSelection_Sort.o SortTest.exe
    rm Selection_Sort.o
    rm testSelection_Sort.o
    rm SortTest.exe
```

- compile : gcc with -I option
- link : gcc with -L option
- run : generate Test Result(XML)
- move : moving XML files
- clean : delete object, executable files

Sort Test

❖ Build makefile

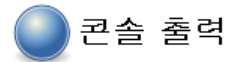
In Eclipse

```
**** Build of configuration Default for project SortTest

mingw32-make all
gcc -ID:\CUnit\include -o Selection_Sort.o -c Selection_Sort.c
gcc -ID:\CUnit\include -o testSelection_Sort.o -c testSelection_Sort.c
gcc -LD:\CUnit\lib -o SortTest.exe testSelection_Sort.o Selection_Sort.o
mv CUnitAutomated-Listing.xml ../XML/Listing.xml
mv CUnitAutomated-Results.xml ../XML/Results.xml
```

- SortTest
 - Binaries
 - Includes
 - XML
 - CUnit-List.dtd
 - CUnit-List.xml
 - CUnit-Run.dtd
 - CUnit-Run.xml
 - Listing.xml
 - Results.xml
 - Selection_Sort.c
 - Selection_Sort.h
 - testSelection_Sort.c
 - Selection_Sort.o - [x86/le]
 - SortTest.exe - [x86/le]
 - testSelection_Sort.o - [x86/le]
 - makefile

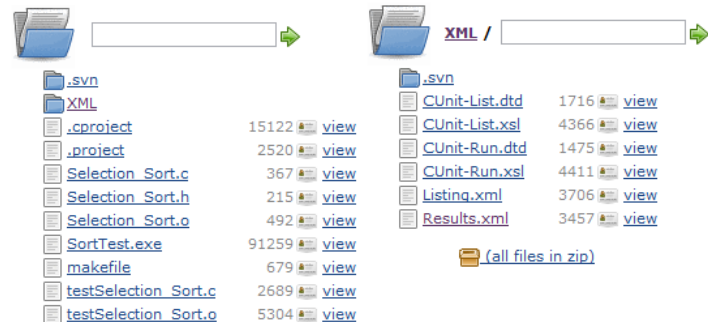
In Hudson



```
Started by user hmanager
Updating svn://kester.iptime.org/SortTest revision: 2011. 6. 3 오전 4:43:54 dep
At revision 9
no change for svn://kester.iptime.org/SortTest since the previous build
[workspace] $ cmd /c call C:\Users\kester\AppData\Local\Temp\Hudson17335361446:

D:\hudsonSrc\jobs\SortTest\workspace>mingw32-make,exe
gcc -ID:\CUnit\include -o Selection_Sort.o -c Selection_Sort.c
gcc -ID:\CUnit\include -o testSelection_Sort.o -c testSelection_Sort.c
gcc -LD:\CUnit\lib -o SortTest.exe testSelection_Sort.o Selection_Sort.o -lcunit
SortTest.exe
mv CUnitAutomated-Listing.xml ../XML/Listing.xml
mv CUnitAutomated-Results.xml ../XML/Results.xml

D:\hudsonSrc\jobs\SortTest\workspace>exit 0
Finished: SUCCESS
```



Sort Test

❖ Test Run Result

CUnit - A Unit testing framework for C.
<http://cunit.sourceforge.net/>

Automated Test Run Results

Running Suite ASSERT_Test

Running test ASSERT TEST ...	Passed
Running test ASSERT TRUE TEST ...	Passed
Running test ASSERT EQUAL TEST ...	Passed
Running test ASSERT STRING TEST ...	Passed
Running test ASSERT Fail Test ...	Failed

File Name	testSelection_Sort.c	Line Number	46
Condition	CU_FAIL("fail test_fail")		

Running Suite Type_Test

Running test TYPE INT TEST ...	Passed
Running test TYPE CAPITAL TEST ...	Passed
Running test TYPE SPECIAL TEST ...	Passed
Running test TYPE MIX TEST ...	Passed

Cumulative Summary for Run					
Type	Total	Run	Succeeded	Failed	Inactive
Suites	2	2	- NA -	0	0
Test Cases	9	9	8	1	0
Assertions	24	24	23	1	n/a

File Generated By CUnit v2.1-2 - Fri Jun 03 04:43:55 2011

Sort Test

❖ List of Test

CUnit - A Unit testing framework for C
<http://cunit.sourceforge.net/>

Total Number of Suites 2
Total Number of Test Cases 9

Listing of Registered Suites & Tests

	Initialize Function?	Cleanup Function?	Test Count	Active?
Suite ASSERT_Test	No	No	5	Yes
Test ASSERT TEST				Yes
Test ASSERT TRUE TEST				Yes
Test ASSERT EQUAL TEST				Yes
Test ASSERT STRING TEST				Yes
Test ASSERT Fail Test				Yes
Suite Type_Test	No	No	4	Yes
Test TYPE INT TEST				Yes
Test TYPE CAPITAL TEST				Yes
Test TYPE SPECIAL TEST				Yes
Test TYPE MIX TEST				Yes

File Generated By CUnit v2.1-2 - Fri Jun 03 04:43:55 2011



END