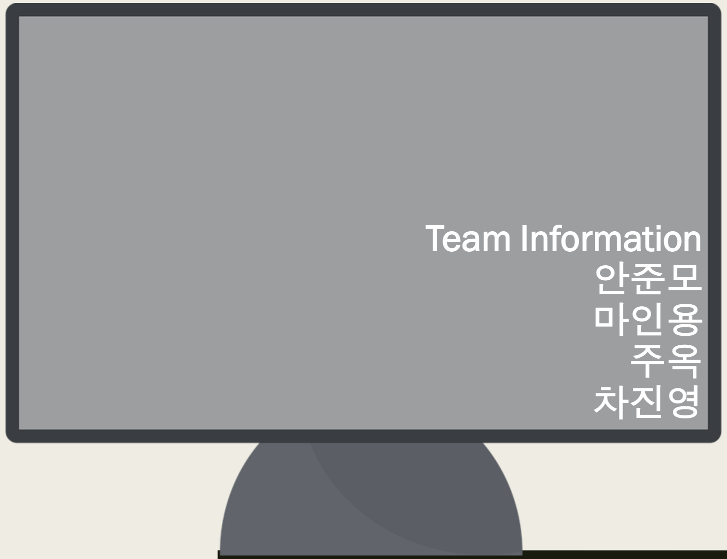
UNIT TEST PLAN FOR

POS

TEAM 4



Team Information
안준모
마인용
주옥
차진영

Contents

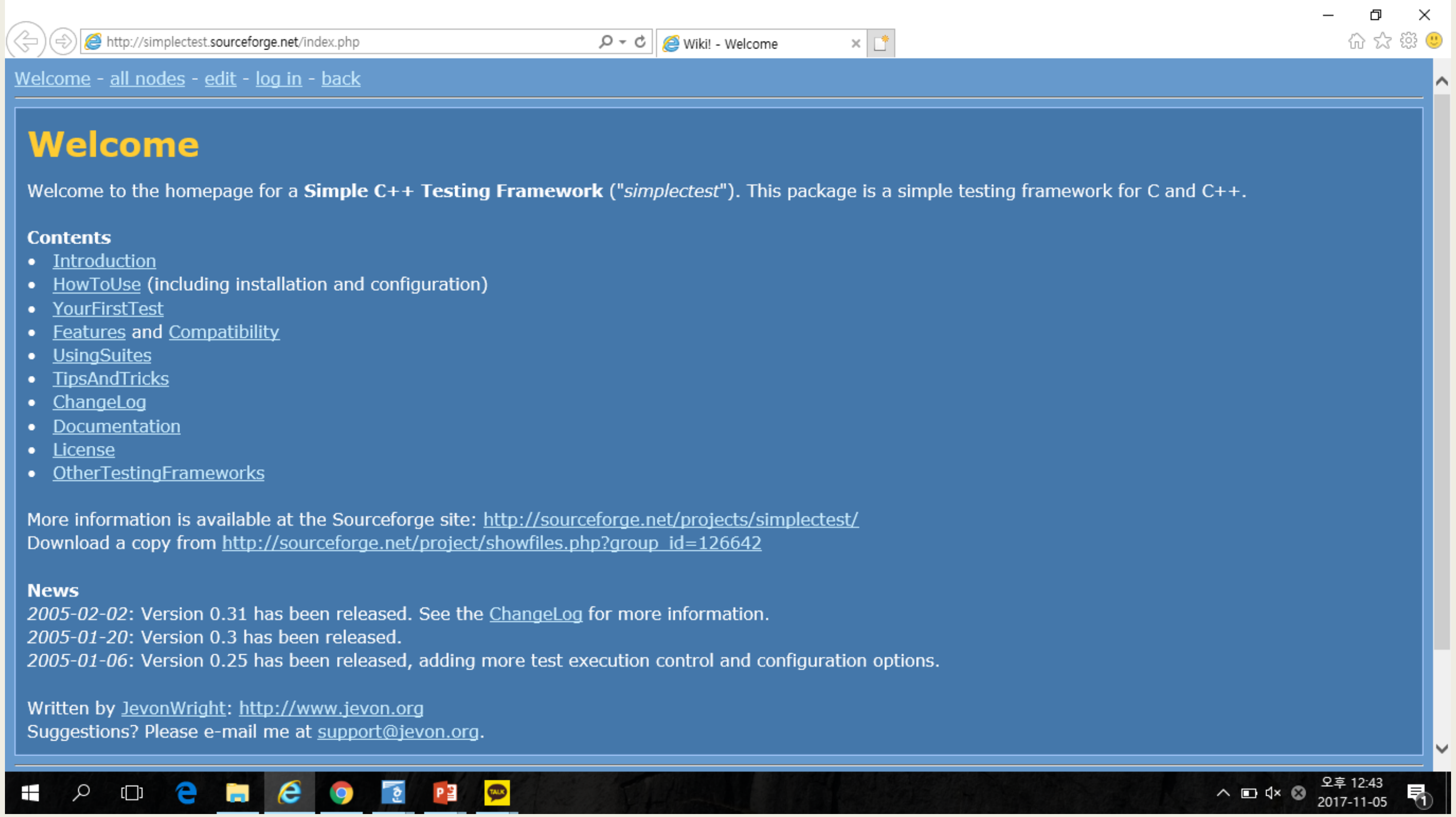
- Enviorment
- Testing Unit
- Point of Sale
- Featured to be Tested
- Test Case Specification

Enviorment

- Gcc Compile
- Cygwin
- Simple ctest

Testing Unit

(Simple Ctest)



Welcome - all nodes - edit - log in - back

Welcome

Welcome to the homepage for a **Simple C++ Testing Framework** ("*simplectest*"). This package is a simple testing framework for C and C++.

Contents

- Introduction
- HowToUse (including installation and configuration)
- YourFirstTest
- Features and Compatibility
- UsingSuites
- TipsAndTricks
- ChangeLog
- Documentation
- License
- OtherTestingFrameworks

More information is available at the Sourceforge site: http://sourceforge.net/projects/simplectest/
Download a copy from http://sourceforge.net/project/showfiles.php?group_id=126642

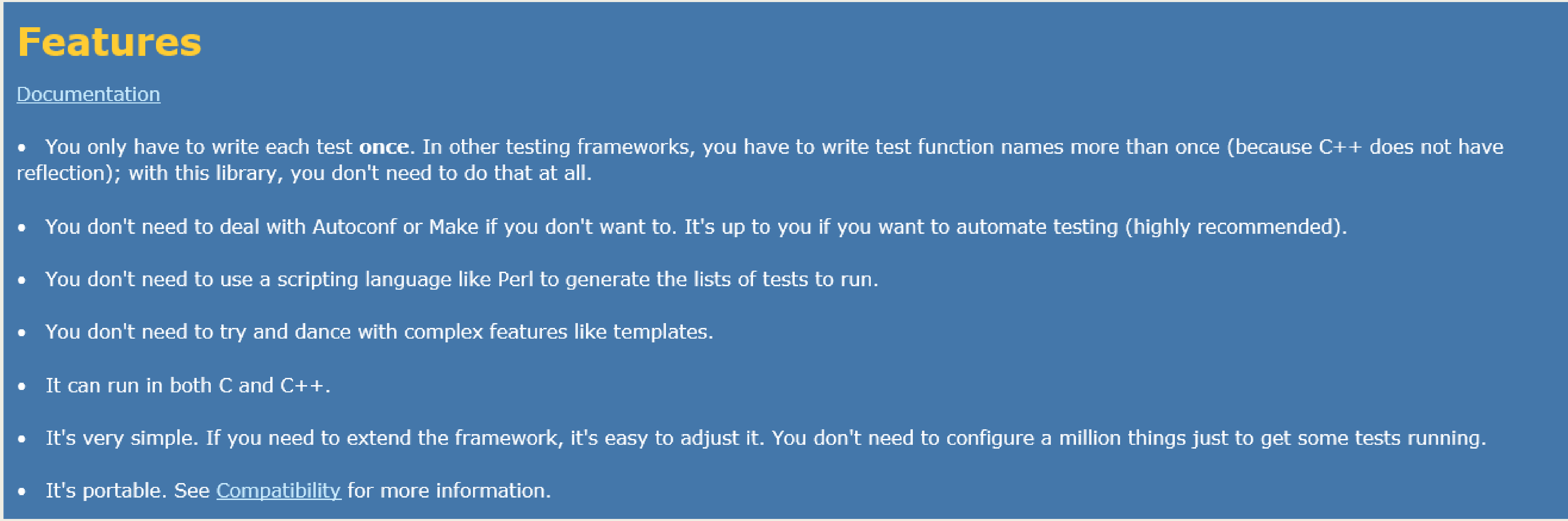
News
2005-02-02: Version 0.31 has been released. See the ChangeLog for more information.
2005-01-20: Version 0.3 has been released.
2005-01-06: Version 0.25 has been released, adding more test execution control and configuration options.

Written by JevonWright: http://www.jevon.org
Suggestions? Please e-mail me at support@jevon.org.

https://sourceforge.net/projects/simplectest/files/

Testing Unit

(Simple Ctest-Features)

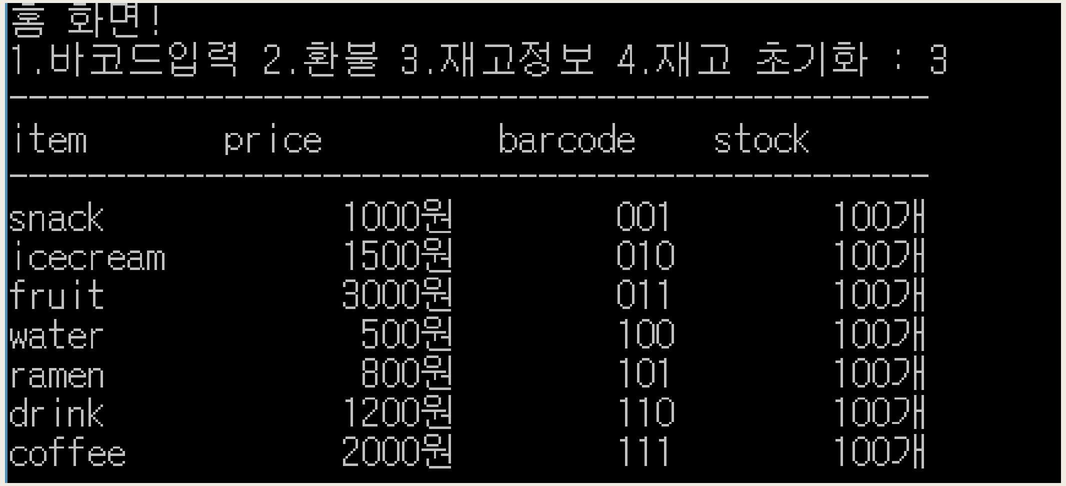


Features

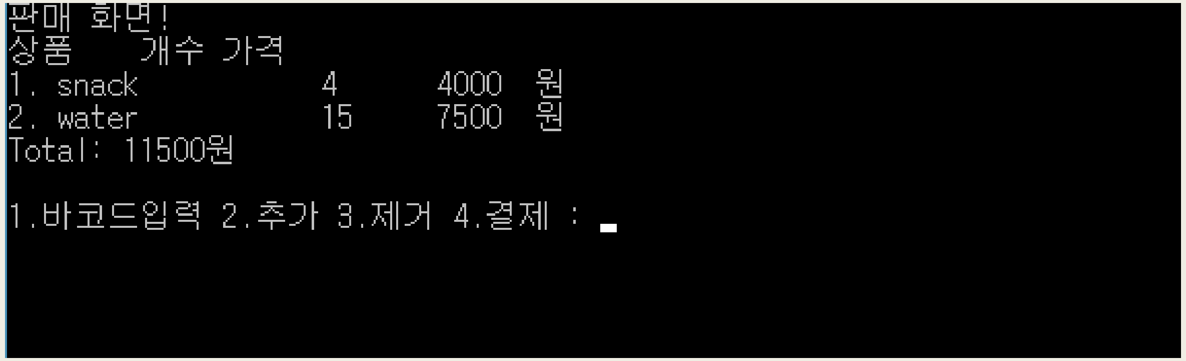
Documentation

- You only have to write each test **once**. In other testing frameworks, you have to write test function names more than once (because C++ does not have reflection); with this library, you don't need to do that at all.
- You don't need to deal with Autoconf or Make if you don't want to. It's up to you if you want to automate testing (highly recommended).
- You don't need to use a scripting language like Perl to generate the lists of tests to run.
- You don't need to try and dance with complex features like templates.
- It can run in both C and C++.
- It's very simple. If you need to extend the framework, it's easy to adjust it. You don't need to configure a million things just to get some tests running.
- It's portable. See Compatibility for more information.

Point of Sale (POS)



```
홈 화면!
1.바코드입력 2.환불 3.재고정보 4.재고 초기화 : 3
----------------------------------------------
item         price            barcode     stock
----------------------------------------------
snack                 1000원      001         100개
icecream              1500원      010         100개
fruit                 3000원      011         100개
water                  500원      100         100개
ramen                  800원      101         100개
drink                 1200원      110         100개
coffee                2000원      111         100개
```



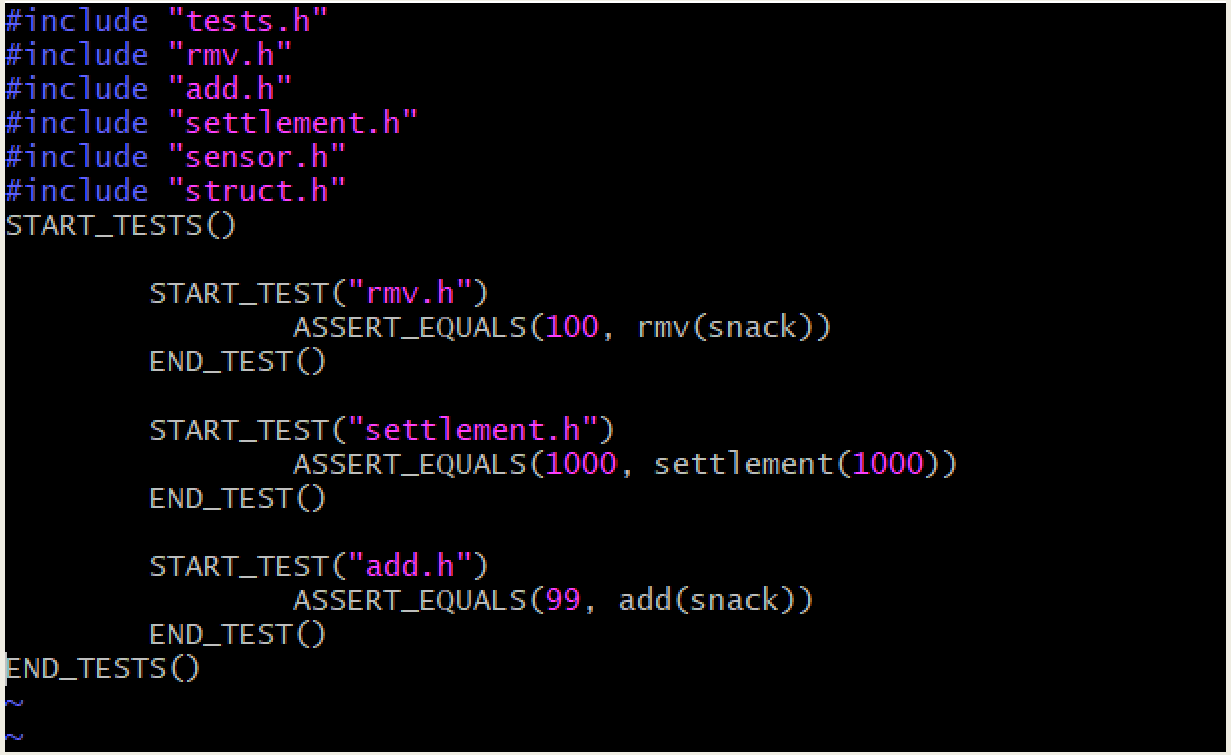
```
판매 화면!
상품    개수 가격
1. snack             4      4000  원
2. water             15     7500  원
Total: 11500원

1.바코드입력 2.추가 3.제거 4.결제 :
```

Featured to be tested

Name	Description
showStock(Item*)	재고확인요청이 들어왔을 때 재고 수량을 보여준다.
barcodeSence(int)	바코드 센서가 스캔했을 때 올바른 물품을 인식하는지 확인한다.
add(Item*)	add버튼을 눌렀을 때 판매수량이 증가하는지 확인한다.
remove(Item*)	remove버튼을 눌렀을 때 판매수량이 감소하는지 확인한다.
initialize(Item*)	재고수량 초기화가 정상적으로 되어지는지 확인한다.
showList(list*)	판매화면으로 정상정으로 전환되는지 확인한다.

Simple Ctest

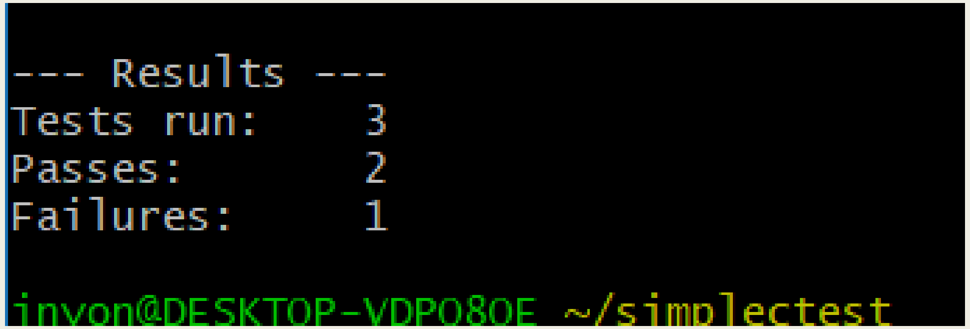


```
#include "tests.h"
#include "rmv.h"
#include "add.h"
#include "settlement.h"
#include "sensor.h"
#include "struct.h"
START_TESTS()

        START_TEST("rmv.h")
                ASSERT_EQUALS(100, rmv(snack))
        END_TEST()

        START_TEST("settlement.h")
                ASSERT_EQUALS(1000, settlement(1000))
        END_TEST()

        START_TEST("add.h")
                ASSERT_EQUALS(99, add(snack))
        END_TEST()
END_TESTS()
~
~
```



```
--- Results ---
Tests run:     3
Passes:        2
Failures:      1

invon@DESKTOP-VDPO8OE ~/simplectest
```

Test Case Specification

ID	Name	Input specification / Feature	Output specification
POS.UTC.101	showStock(Item*)	showStock(Item*) /재고확인요청	(showStock(pItem))==100
POS.UTC.102	barcodeSence(int)	char* barcodeSense(int); /바코드에 따른 물품인식	barcodeSense(001)==“snack”
POS.UTC.103	add(Item*)	int add(item*) /add버튼에 따른 변화	add(“snack”) == 1
POS.UTC.104	remove(Item*)	int remove(item*) /remove버튼에 따른 변화	remove(“water”) == 0
POS.UTC.105	initialize(Item*)	int initialize(item*) /재고수량 초기화	initailize(pItem) == 100
POS.UTC.106	showList(list*)	void showList(list*) /판매화면	showList(pList) == void