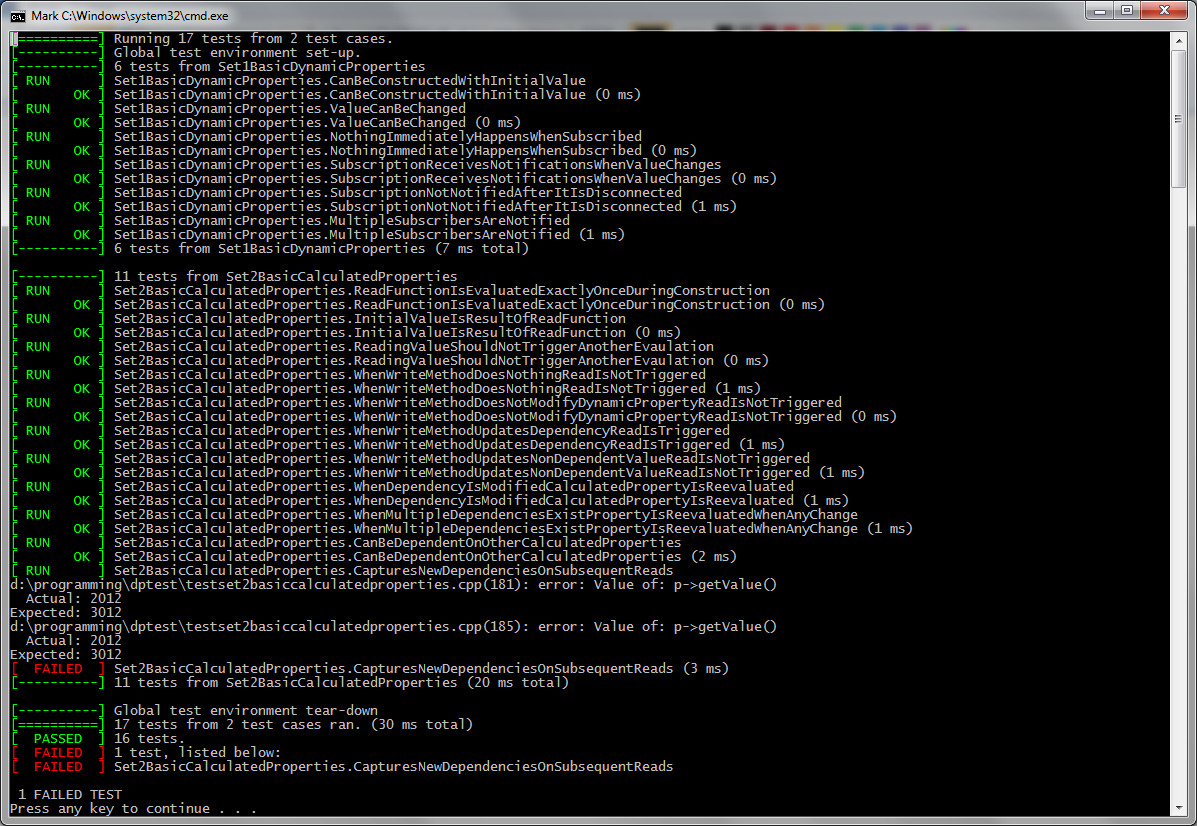
Dynamic Property Test Report

# **Tester – Zengjia Yan, 05/11/2014**

# **Test Result**

Basic tests: 16/17 passed (Intermediate and Advanced tests are not done)

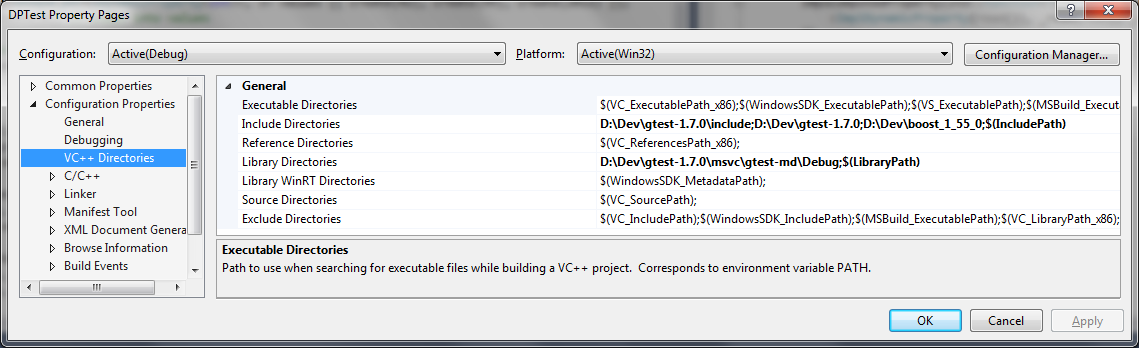


# **Coding Environment**

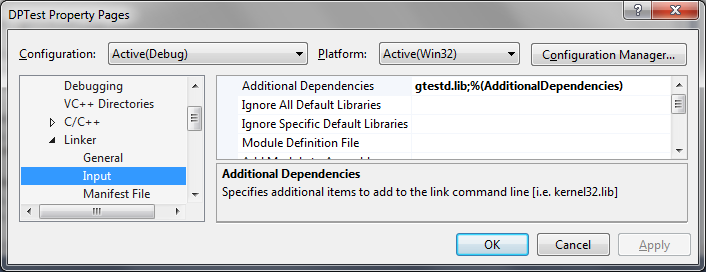
Window OS 7, Visual Studio 2013, Google Test 1.70, Boost 1.55

# **Compiling Prerequisite**

1. Install google test and boost in Windows
2. Open solution file “DPTest.sln” in Visual Studio 2013
3. Set VC++ Directories to include g-test and boost in the “Include Directories” and “Library Directories”



1. Add gtest.lib into Linker->Input->Additional Dependencies



1. Build and run, the result of basic test will be printed.

# **Change to Original Source Code**

1. DynamicProperty.h - Register the object to BindingController before use in both create().
2. IDynamicProperty.h - Intact.
3. Main.cpp – Intact.
4. TestSet1BasicCalculatedProperties.cpp – Intact.
5. TestSet2BasicCalculatedProperties.cpp

I know I’m not supposed to change it at all, but this line below behaves different with different compilers: I had to add a cast to make it compile. The rest part remains untouched.

EXPECT\_EQ(std::list<int>({100 + 99 + static\_cast<int>(strlen("forty-two")), 100 + 99}), notifications); // notification was sent out

# **Class Design**

1. Communication is maintained using modified Observer pattern

**Observer**

**Observer**

**Observer**

**Observer**

**Observable**

Message:

1. GetValue
2. SetValue
3. Dispose

Message:

Dependent observable changed

**BindingController (also an observer)**

**BindingMap**

**(Observable X Observers)**

1. ImplDynamicProperty is observable to broadcast status update.

**IDynamicProperty**

**IObservable**

**ImplDynamicProperty**

1. ImplComputedProperty inherits from ImplDynamicProperty. So, it’s both observable and observer, and both DynamicProperty and ComputedProperty, to allow ComputedProperty to depend on other ComputedProperty.

**ImplDynamicProperty**

**IObserver**

**ImplDynamicProperty**

1. Find dependencies (observables) of a ComputedProperty (observer).

BindingController turns on its “Radar” to find which observables sent GetValue messages during the construction of ComputedProperty when the read function was called.

One observable calls GetValue() and notify BindingController

End of construction

/

Turn off “Radar”

Start of construction

/

Turn on “Radar”