Subject: Added Google Test
Posted by Klugier on Sun, 24 Jan 2016 13:15:04 GMT
View Forum Message <> Reply to Message

Hello,

Google Test is a unit test framework for C++. More information about library you can find on official GitHub project site: https://github.com/google/googletest.

This is native upp plugin, so it doesn't require any additional dependency. In addition it offers fast creating of test package via TheIDE template mechanism. Currently it's supporting MS Windows and GNU/Linux.

Current u++ port require dependency with Core package. This is because Google Test in one place has got memory leak and I manually turn it off via Upp::MemoryIgnoreLeaksBegin() and Upp::MemoryIgnoreLeaksEnd().

Below is the simply example that shows how to test Upp Core package (bazzar/GoogleTestExample):
StringTest.h:

```
#ifndef _GoogleTestExample_VectorTest_h_
#define _GoogleTestExample_VectorTest_h_

#include <Core/Core.h>
#include <plugin/gtest/gtest.h>

NAMESPACE_UPP

class StringTest : public testing::Test {
protected:
 virtual void SetUp();

protected:
 String sEmpty;
 String sCat;
 String sDog;
};

END_UPP_NAMESPACE

#endif
```

StringTest.h:

```
#include "StringTest.h"
```

```
NAMESPACE_UPP

void StringTest::SetUp()
{
 sCat = "Cat";
 sDog = "Dog";
}

TEST_F(StringTest, TestDefaultConstructor)
{
 String a;

 ASSERT_EQ(a, "");
}

TEST_F(StringTest, TestConstructor)
{
 String a("Test");
 ASSERT_EQ(a, "Test");
}

TEST_F(StringTest, TestGetCount)
{
 ASSERT_EQ(sEmpty.GetCount(), 0);
 ASSERT_EQ(sCat.GetCount(), 3);
 ASSERT_EQ(sDog.GetCount(), 3);
}

TEST_F(StringTest, TestClear)
{
 sCat.Clear();

 ASSERT_EQ(sCat, "");
 ASSERT_EQ(sCat.GetCount(), 0);
}

TEST_F(StringTest, TestCompare)
{
 ASSERT_EQ(sCat.Compare(sCat), 0);
 ASSERT_EQ(sCat.Compare(sDog), -1);
}

TEST_F(StringTest, TestEqual)
{
 ASSERT_TRUE(sCat.IsEqual(sCat));
 ASSERT_FALSE(sCat.IsEqual(sDog));
}
```

```
END_UPP_NAMESPACE
```

main.cpp

```
#include <plugin/gtest/gtest.h>

int main(int argc, char *argv[])
{
 testing::InitGoogleTest(&argc, argv);

 return RUN_ALL_TESTS();
}
```

Sincerely,
Klugier

Subject: Re: Added Google Test
Posted by dolik.rce on Sun, 24 Jan 2016 19:20:31 GMT
View Forum Message <> Reply to Message

Hi Klugier,

This is good news :) I have used google test in past for some of my projects and I was always satisfied with how easy it was to write the tests with it.

One great thing to add to this package would be a macro for the main() function. It is almost always the same, so having TEST_APP_MAIN would remove the need to remember the two lines that always need to be there.

Thanks for the good work!
Honza

Subject: Re: Added Google Test
Posted by Klugier on Sun, 24 Jan 2016 20:01:20 GMT
View Forum Message <> Reply to Message

Hello Honza,

Well I will think about your proposition, but please notice that you don't need to remember that, because you can generate package with above main via TheIDE template in "New package" creator. I added bazaar/upt/GoogleTest.upt file. In the future I plan to remove int main and replace with CONSOLE_APP_MAIN for more upp style. In addition CONSOLE_APP_MAIN allows checking memory leaks.

Sincerely,
Klugier

Subject: Re: Added Google Test
Posted by dolik.rce on Sun, 24 Jan 2016 20:36:41 GMT
View Forum Message <> Reply to Message

Klugier wrote on Sun, 24 January 2016 21:01Well I will think about your proposition, but please notice that you don't need to remember that, because you can generate package with above main via TheIDE template in "New package" creator. I added bazaar/upt/GoogleTest.upt file. In the future I plan to remove int main and replace with CONSOLE_APP_MAIN for more upp style. In addition CONSOLE_APP_MAIN allows checking memory leaks.
You're right, using the template makes sense. Unless someone wants to keep tests in same package as the code (e.g. using build flag). One reason for this might be to avoid switching between the app and tests all the time.

Thinking of it now, another nice feature would be integrating the tests into theide using macro, so the can be easily launched immediately after changing the application code. What do you think?

Honza

Subject: Re: Added Google Test
Posted by mr_ped on Sun, 24 Jan 2016 20:57:45 GMT
View Forum Message <> Reply to Message

Did you try the UnitTest++ from Bazaar too? So can you compare them?

I'm using UnitTest++ for all my C++ projects, and so far I find it best between different languages, but I never tried other C++ libraries.

Ped

Subject: Re: Added Google Test
Posted by dolik.rce on Sun, 24 Jan 2016 21:10:29 GMT
View Forum Message <> Reply to Message

mr_ped wrote on Sun, 24 January 2016 21:57Did you try the UnitTest++ from Bazaar too? So can you compare them?
No, the first C++ test framework I met was the one from google and I never needed anything it couldn't do :) I can only compare with nosetest and pytest I used for python and I'd said that all three of them provide about the same level of comfort when writing tests. Which is definitely a

good thing to say for gtest, since the other two have advantage of being written in dynamic language :)

Honza

Subject: Re: Added Google Test
Posted by Klugier on Thu, 09 Mar 2017 22:55:03 GMT
View Forum Message <> Reply to Message

Hello,

I made progress in Google Test areas. Now the mocking framework is added. Of course not without problem. We need to live with global constructors leak problem on Windows platform - so memory debugging is temporary disable (I hope this can be fixed in the future - but not soon). On POSIX memory debugging works great.

Moreover, I made progress in ide integration and now you can execute all tests or one specific one. I will work on group support, too. But, it would probably require changes in Esc ide integration.

The polished version can be used in version: #10926.

Sincerely,
Klugier