
Subject: ide: Compare with log and new 'unit' testing method

Posted by [mirek](#) on Wed, 19 Aug 2020 08:59:59 GMT

[View Forum Message](#) <> [Reply to Message](#)

So I have got an idea how to simplify unit testing, or at least what I consider unittesting in U++ context...

Traditionally, checks in unit testing are something similar to a tedious list of ASSERTs. So far, in many tests, this is exactly what I was doing in autotest too.

Recently I have introduced a new approach: Instead of individual ASSERTs, test normally LOGs values out, creating standard U++ log. Then, at the end of test, this log is compared with "Etanol.log" in the test package.

When creating the test, you simply output all values of interest, like

```
#include <Core/Core.h>

using namespace Upp;

CONSOLE_APP_MAIN
{
    StdLogSetup(LOG_COUT|LOG_FILE);

    { Vector<int> x(20); DUMP(x.GetCount()); }
    { Vector<int> x(20, 123); DUMP(x); }
    { Vector<String> x(20, "123"); DUMP(x); }

    { Array<int> x(20); DUMP(x.GetCount()); }
    { Array<int> x(20, 123); DUMP(x); }
    { Array<String> x(20, "123"); DUMP(x); }
}
```

You check that the output is correct, then create the file "Etalon.log", copy current log into it and add CheckLogEtalon(); call at the end of MAIN.

```
CONSOLE_APP_MAIN
{
    StdLogSetup(LOG_COUT|LOG_FILE);

    { Vector<int> x(20); DUMP(x.GetCount()); }
    { Vector<int> x(20, 123); DUMP(x); }
    { Vector<String> x(20, "123"); DUMP(x); }

    { Array<int> x(20); DUMP(x.GetCount()); }
```

```
{ Array<int> x(20, 123); DUMP(x); }
{ Array<String> x(20, "123"); DUMP(x); }

CheckLogEtalon();
}
```

CheckLogEtalon loads current log, etalon log, fixes them a bit (removes first line which is in principle different, removes '\r' to avoid problems) and compares. If they are equal, test passes. (CheckLogEtalon is now in Core/Diag.h).

To support this, I have also added "Compare with log" function to the ide - if the test fails, you can use that to quickly find the culprit.

Subject: Re: ide: Compare with log and new 'unit' testing method
Posted by [Didier](#) on Thu, 20 Aug 2020 08:06:31 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Mirek,

This is a simple and yet efficient approach: make a reference result and compare to reference. This should be very easy to implement and use for unit testing (only if there is no multithread and time doesn't matter)
