
Subject: U++ logging refactored
Posted by [mirek](#) on Mon, 02 Jul 2012 15:34:24 GMT
[View Forum Message](#) <> [Reply to Message](#)

While finishing Skylark, I have found certain features of U++ logging inadequate.

First of all, output from multiple threads depended on locks in LOGing macros, which was fine for macros, but e.g. Sql is not using these macros (and thus MT logging was basically broken in Sql). Fixed by complete refactoring of LogStream, which is now handling locking internally (serializes issuing completed lines to the output).

The I have removed secondary logs UsrLog and BugLog and integrated them to the main log output.

I have added new standard log options:

LOG_SYS - in Posix, log is outputted to syslog too.

LOG_ROTATE(n) - when .log file is about to be replaced, up to 'n' older logs are preserved (using file extensions '.1', '.2', ... '.n').

LOG_ROTATE_GZIP - preserved logs '.2' and higher are gzipped.

I have also introduced a new concept, 'modular' logging, which usually should be bound to program config (e.g. .ini file) and immediately used it to replace UsrLog concept (two lines of code better than 300 words :):

```
#define LOG_(flag, x)    do { if(flag) RLOG(x); } while(false)
```

```
namespace Ini {  
extern Bool user_log;  
};
```

```
#define USRLOG(x)        LOG_(Ini::user_log, x)
```

Last but not least, logging is now documented

UPDATE: Config->Ini

Subject: Re: U++ logging refactored
Posted by [Zbych](#) on Tue, 03 Jul 2012 06:38:39 GMT
[View Forum Message](#) <> [Reply to Message](#)

If it is not a problem, can you add UTC timestamp (LOG_TIMESTAMP_UTC)?

Subject: Re: U++ logging refactored

Posted by [mirek](#) on Tue, 03 Jul 2012 07:47:49 GMT

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

Done.

Subject: Re: U++ logging refactored

Posted by [Zbych](#) on Tue, 03 Jul 2012 08:24:48 GMT

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

Thanks.

Subject: Re: U++ logging refactored

Posted by [dolik.rce](#) on Fri, 06 Jul 2012 17:19:58 GMT

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

Hi Mirek,

I think there is a bug in the refactored logging - or maybe it is a feature The logging is now not

"atomic", e.g.:bool fn(){

LOG("fn called");

return true;

}

GUI_APP_MAIN{

DUMP(fn());

};

producesfn() = fn called

true

That is IMHO wrong behavior.

Best regards,

Honza

Subject: Re: U++ logging refactored

Posted by [mirek](#) on Sat, 07 Jul 2012 09:54:31 GMT

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

This is nothing new, it was always working this way.

It is possible to fix it, but with somewhat lowered performance (because we would need to create String first, then output to stream, now everything goes directly to stream).

Not sure it is worth it...

(Note that MT behaviour is fine, it is "non-atomic" only in single thread).

Subject: Re: U++ logging refactored

Posted by [dolik.rce](#) on Sat, 07 Jul 2012 10:22:22 GMT

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

mirek wrote on Sat, 07 July 2012 11:54 This is nothing new, it was always working this way.

It is possible to fix it, but with somewhat lowered performance (because we would need to create String first, then output to stream, now everything goes directly to stream).

Not sure it is worth it...

(Note that MT behaviour is fine, it is "non-atomic" only in single thread).

Hm, interesting, I never noticed before... I think that when using logging for debug purposes, the performance is not important. Would it be possible to add some mechanism that would allow to differentiate between debug logging and other cases?

The simple way that comes to my mind is to make it work fast in release mode and correct in debug, but that might lead to confusion in the non-debug usage. What about a flag/macro, that would let me say "now I want to use the 'atomic' logging"? It could even work on per file basis...

Honza

Subject: Re: U++ logging refactored

Posted by [mirek](#) on Sat, 07 Jul 2012 11:01:19 GMT

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

dolik.rce wrote on Sat, 07 July 2012 06:22 mirek wrote on Sat, 07 July 2012 11:54 This is nothing new, it was always working this way.

It is possible to fix it, but with somewhat lowered performance (because we would need to create String first, then output to stream, now everything goes directly to stream).

Not sure it is worth it...

(Note that MT behaviour is fine, it is "non-atomic" only in single thread).

Hm, interesting, I never noticed before... I think that when using logging for debug purposes

But the changes in logging are intended for logging even for production purposes...

Mirek

Subject: Re: U++ logging refactored
Posted by [dolik.rce](#) on Sat, 07 Jul 2012 11:54:17 GMT
[View Forum Message](#) <> [Reply to Message](#)

mirek wrote on Sat, 07 July 2012 13:01 But the changes in logging are intended for logging even for production purposes...

Mirek
I'm aware of that, that is why I was thinking about doing it settable - to allow the current, high performance version where needed.

Honza

Subject: Re: U++ logging refactored
Posted by [NilaT](#) on Thu, 06 Dec 2012 11:41:46 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

and sorry for digging out this old thread again but I have a question to the new logging feature.

A few days ago I went from 2010 source to 2012 and saw, that UsrLog and BugLog etc. isn't available anymore.
I searched and luckily found this thread.

In my program I added "StdLogSetup(LOG_FILE | LOG_TIMESTAMP, sFile);" as suggested in here:
[http://www.ultimatepp.org/src\\$Core\\$Logging\\$en-us.html](http://www.ultimatepp.org/src$Core$Logging$en-us.html)

BUT, how do I create two different logfiles?
I want files with the .log extension and another with .buglog or something.
In 2010 source I used a USRLOG or BUGLOG Macro which get me UsrLog() in Debug.cpp

But now it's a bit more complicated with struct IniBool and stuff... (don't know exactly what the user_log flag is for, but as it seems LOG_ calls RLOG calls UPP::VppLog calls UppLog calls StdLog calls StdLogStream)

Seems like I have to add a new Stream Method oder something?
Any help appreciated.

Thanks.

Subject: Re: U++ logging refactored
Posted by [mirek](#) on Thu, 06 Dec 2012 11:45:03 GMT
[View Forum Message](#) <> [Reply to Message](#)

I am sorry, but UsrLog and BugLog were removed.

I think you can easily bring them back for your code even with USRLOG and BUGLOG macros.
Even copying from 2010 sources might work.

Maybe we should this BW support to bazaar?

Mirek

Subject: Re: U++ logging refactored
Posted by [NilaT](#) on Thu, 13 Dec 2012 12:37:46 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

well I tried it, but I gave up.
Now I'm doing the following:

Mutex LogFileMutex;

```
void WriteLogfile(String text) {  
    LogFileMutex.Enter();  
    Time zeit = GetSysTime();  
    String zt = Format("[%i/%i/%i/ %i:%i:%i]", zeit.day, zeit.month, zeit.year, zeit.hour, zeit.minute,  
zeit.second);  
    FileAppend file(m_LogFilename);  
    file.PutLine(Format("%s %s", zt, text));  
    file.Close();  
    LogFileMutex.Leave();  
}
```
