
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 UsrcLog 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 UsrcLog 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
