Subject: Extended Logging Package
Posted by captainc on Tue, 05 Aug 2008 12:31:33 GMT
View Forum Message <> Reply to Message

I created a package to extend the logging capabilities of Upp. I encountered a situation where I wanted to use custom log files for release programs to log Successful and Failed transfers. It is built as a singleton with lazy creation. I built it with an archiving capability so it uses Upp bz2 package to archive logs older than a user-defined time frame. There are special cases for debug and release logs so that the archiving and dating capabilities can also apply to them. Finally, there is some Topic++ documentation for it, but the in-code documentation in the header file is best. Let me know what you think.

Simple Example:void QuickExample(){
 // Logger singleton instance created automatically and lazily on first use.
 UppLog::LOGGER->BeginLogs();
 UppLog::LOGGER->EnableCout(UppLog::RELEASE);
 UppLog::UPPLOG(UppLog::BOTH, "This message is sent to both log files.");
 UppLog::UPPLOG(UppLog::DEBUG, "This message is sent to debug log.");
 UppLog::UPPLOG(UppLog::RELEASE, "This message is sent to release log (and Cout()).");

 UppLog::LOGGER->CreateLog("Successes");
 UppLog::LOGGER->CreateLog("Failures", "C:\\failures");
 UppLog::UPPLOG("Successes", "My successful transaction message.");
 UppLog::UPPLOG("Failures", "My failed transaction message.");
};

Edit: Attachment deleted. Use the newer version below.

Subject: Re: Extended Logging Package
Posted by cbpporter on Tue, 05 Aug 2008 12:50:23 GMT
View Forum Message <> Reply to Message

Really nice functionality!

But I would make it a a normal class, and if I need singleton design pattern I would just use the Single template provided by U++. And macros? There has got to be another way to do the logging without those terrible macros . What if I have a function somewhere named UPPLOG or LOGGER?

But the rest is nice and I'll give it a try (after I remove the macros ) since I use DUMP a lot, especially under Linux, and I've been wishing for something a little more powerful.

Subject: Re: Extended Logging Package
Posted by captainc on Tue, 05 Aug 2008 13:32:42 GMT
View Forum Message <> Reply to Message

Quote: if I need singleton design pattern I would just use the Single template provided by U++
Yeah, I had never used the singleton template from U++, I forgot it was even there.

Quote:And macros? There has got to be another way to do the logging without those terrible macros
There is, but the naming makes it really long. The macros are only a shortcut for it.
You can use something like this instead:

```
using namespace UppLog;
Logger::Instance()->Log("mylog", "my log message");
```

I'll rethink the naming conventions...

---

Subject: Re: Extended Logging Package
Posted by captainc on Tue, 05 Aug 2008 16:42:33 GMT
View Forum Message <> Reply to Message

Changed macros to inline void functions.
Fixed bug in calculating getting seconds from time interval.
Changed massage arguments to const string references.
Sample using namespace:

```
using namespace UppLog;
Logger::Instance()->BeginLogs();
UPPLOG(BOTH, "log message");
```

## File Attachments

1) UppLogging.zip, downloaded 393 times

---