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.