How to use

# Include the source in you project:

1. Copy Paste the ‘src’ folder to your project folder.
2. Include “src/main” directory in your *additional include directory* settings.

# For Thread Application:

This component does not depend on any nonstandard library, but if you use this component in a thread system (As up to now C++ does not have any standard Thread library) you have to define two Macros: “\_CLThreadLock()” to lock and “\_CLThreadUnLock()” to unlock the logger for a unit log request.   
To define those macros go to: src/main/com/afrigis/util/thread/ThreadLock.h file.

Already there are some commented out codes for posix thread and boost thread. If you any of these two (POSIX or Boost) thread system then you can just uncomment appropriate piece of code block and you are good to go. (I assume you already added necessary header, lib and dll files in your project for any of those thread system)  
  
And if you are using this component on a nonThread system then you can just forget about it.

# Start Using:

Include com/afrigis/util/logsystem/ChronicleLogger.h in your main source file and you are good to go to start logging your application by using one of the following macros:

\_CLogFatal("Fatal Error Cause"); //Use this log for a situation when the application must be terminated.

\_CLogCritical("This is a critical error"); //For a very harmful error but the system can still continue working

\_CLogWarning ("Warning for the user");

\_CLogInfo ("This is an information massage");

\_CLogDebug ("This is a debug message");

# Change OutputDevice and Output format of log:

With the default setting the all log message will be in Plain text format and log messages will be printed on the console. In order to change this, lets say you want to send the log message to a file in HTML formate (Html formatter is given as one of the default format implementation). You have to do the following:

com::afrigis::util::logsystem::ChronicleLogger::getLogger().setDevice(new com::afrigis::util::logsystem::DiskFile(“MYLogFile.html”)); //change the device to DiskFile

com::afrigis::util::logsystem::ChronicleLogger::getLogger ().setLogFormatter (new com::afrigis::util::logsystem::Html()); //Now Change the format to HTML

Or you can use the following macro to do the same thing:   
  
\_CLDiskFileHtml(“MYLogFile.html”);

There is also for PlainText in disk file macro:

\_CLDiskFilePlain(“MYLogFile.txt”);

Remember in order to use the Formatter and Device correctly you should 1st change to Device then the formatter.

Change Log level:

There are 6 log levels in the logger.

Level 0: No Log level, that is nothing will be logged  
Level 1: Only Fatal error messages will be loged  
Level 2: Critical Error and Fatal error  
Level 3: This is up to Warning Level.  
Level 4: User information level.  
Level 5: This is Debug level, Everything will be logged

By default this logger is set to log level 5. In order to change it, (may be just before release you may want to change the logger level to Level 3 or 4) write this line of code:

logSystemName::ChronicleLogger::getLogger ().setLevel (3);

Or

\_CLsetLevel(3);

This is it for normal use of this component.

Please go throught the class diagram and doxigen documentation for further upgradation of this component.

Thanks  
-Sowrov