

```

//+-----+
//|                                     C_LOG.mqh |
//|                                     Copyright 2017, M Wilson |
//|                                     https://www.algotrader.blog |
//+-----+
#property copyright "Copyright 2017, M Wilson"
#property link      "https://www.algotrader.blog"
#property version   "1.00"
#property strict

//+-----+
//| C_LOG Class |
//+-----+
class C_LOG
{
private:
    //Private Variables
    string m_strFileName;
public:
    //Public Variables
    string m_strWhereIsTheLog;
    string m_strWhereIsTheStrategyTesterLog;
    //Constructor and Destructor
    C_LOG();
    C_LOG(const string strFileName="LogFile.txt");
    ~C_LOG();
    //Public Functions
    void AppendStringToLog(const string strInput);
    void PrintLocationOfLogFiles();
    void RemoveLogFile();
    void ArchiveAndRemoveLogFile();
    bool FileExists();
};

//+-----+
//| Constructor |
//+-----+
C_LOG::C_LOG()
{
    this.m_strFileName="";
    this.m_strWhereIsTheLog="";
    this.m_strWhereIsTheStrategyTesterLog="";
}
void C_LOG::C_LOG(const string strFileName="LogFile.txt")
{
    this.m_strFileName=strFileName;

//Update WhereIsTheLog so that the developer can find out where the actual log file should be
    string strTerminalPath = TerminalInfoString(TERMINAL_DATA_PATH);
    this.m_strWhereIsTheLog=strTerminalPath+"\\MQL4\\Files\\"+this.m_strFileName;
    this.m_strWhereIsTheStrategyTesterLog=strTerminalPath+"\\tester\\files\\"+this
.m_strFileName;
}

//+-----+
//| Destructor |
//+-----+
C_LOG::~C_LOG()
{
}

//+-----+
//| Public Functions |
//+-----+

void C_LOG::AppendStringToLog(const string strInput)
{

//This function opens the Log File, moves to the end of the file and then appends the input
//string to the file. Returns (ie \r\n) are added by this routine onto the end of the stri

```

```

//Reset the last error
ResetLastError();

//Open the file - must be read and write for appending to files.
int intFileHandle=FileOpen(this.m_strFileName,FILE_READ|FILE_WRITE|FILE_TXT);

//If the file has been opened successfully, write to it
if(intFileHandle!=INVALID_HANDLE)
{
    //Find the End of the file
    if(!FileSeek(intFileHandle,0,SEEK_END))    Print(__FUNCTION__,"File Seek Error ",
GetLastError());

    //Write the String
    if(FileWriteString(intFileHandle,strInput+"\r\n")<=0) Print(__FUNCTION__,
"File Write Error ",GetLastError());

    //Close the file
    FileClose(intFileHandle);
}
else
{
    Print(__FUNCTION__,"Failed to open file ",this.m_strFileName," ",GetLastError());
}

return;
}
void C_LOG::PrintLocationOfLogFiles()
{
//Call this function at the end of an EA to let the developer/user know where the log file is
Print(this.m_strWhereIsTheStrategyTesterLog);
Print("Location of Strategy Tester Log File:");
Print(this.m_strWhereIsTheLog);
Print("Location of Standard Log File:");

return;
}
void C_LOG::RemoveLogFile()
{
//This just deletes the log file
FileDelete(this.m_strFileName);
return;
}
void C_LOG::ArchiveAndRemoveLogFile()
{
//If there is not a BUP directory it creates it.  It then deletes any BUP versions of the file
//and copies the file over it.  A new log file is then created.

//It is best if this is called during the Init stage of an EA so that we have prepared a new
//directory

if(!FileIsExist("BUP"))
{
    FolderCreate("BUP");
}

FileDelete("Bup\\"+this.m_strFileName);
FileCopy(this.m_strFileName,0,"BUP\\"+this.m_strFileName,FILE_REWRITE);
FileDelete(this.m_strFileName);
return;
}
bool C_LOG::FileExists()
{
string strFile=this.m_strFileName;
bool boolRet=FileIsExist(strFile);
return boolRet;
}

```