```
//+------------------------------------------------------------------+
//|                            EA_RecordSpreadStopAndFreezeLevel.mq4 |
//|                                         Copyright 2017, M Wilson |
//|                                      https://www.algotrader.blog |
//+------------------------------------------------------------------+
#include <C_LOG.mqh>

#property copyright "Copyright 2017, M Wilson"
#property link      "https://www.algotrader.blog"
#property version   "1.00"
#property strict
//+------------------------------------------------------------------+
//| Inputs                                                           |
//+------------------------------------------------------------------+
input int I_RecordPeriodInMinutes=1;
//How often the EA records the spread etc.
input string I_BeginningOfLogFile="MktData";
//Beginning of Log File Name to record data.
//+------------------------------------------------------------------+
//| Global Variables                                                 |
//+------------------------------------------------------------------+
string g_strSymbols[];
C_LOG *g_objLog;
//+------------------------------------------------------------------+
//| Expert initialization function                                   |
//+------------------------------------------------------------------+
int OnInit()
  {
//--- create timer
   EventSetTimer(I_RecordPeriodInMinutes*60);

//--- Load Global Variable Array with the Symbols that we are interested in.
   PopulateSymbolArray();

//--- Initiate the Log File
   string strFileName=I_BeginningOfLogFile+"_Acc_"+IntegerToString(AccountNumber())+
".txt";
   g_objLog=new C_LOG(strFileName);

//--- Archive any previous Log Files
   g_objLog.ArchiveAndRemoveLogFile();

//--- Add titles to the log file
   addTitlesToLogFile();

//---
   return(INIT_SUCCEEDED);
  }
void PopulateSymbolArray()
{
   //Clear the array.
   if(ArraySize(g_strSymbols)>0) ArrayFree(g_strSymbols);

   //Add the symbols that we are interested in into the global array.
   addSymbol("EURGBP");
   addSymbol("UK100");

}
void addSymbol(const string strSymbol="")
{
   //Define variable for
   int intSize;

   //Add the symbol
   intSize=ArraySize(g_strSymbols);
   ArrayResize(g_strSymbols,intSize+1);
   g_strSymbols[intSize]=strSymbol;
}
void addTitlesToLogFile()
{
   string strWrite="TimeLocal;TimeBroker;IsConnected;";

   int intCount=ArraySize(g_strSymbols);
   for(int i=0;i<intCount;i++)
   {
      string strSymb=g_strSymbols[i];

      strWrite+=(strSymb+"_Spread;");
      strWrite+=(strSymb+"_StopLevel;");
      strWrite+=(strSymb+"_FreezeLevel;");
   }

   g_objLog.AppendStringToLog(strWrite);
}
//+------------------------------------------------------------------+
//| Expert deinitialization function                                 |
//+------------------------------------------------------------------+
void OnDeinit(const int reason)
  {
//--- destroy timer
   EventKillTimer();

//--- clear the global variable array.
   if(ArraySize(g_strSymbols)>0) ArrayFree(g_strSymbols);

//--- clear the Log File.
   if(g_objLog!=NULL)   delete g_objLog;

  }
//+------------------------------------------------------------------+
//| Timer function                                                   |
//+------------------------------------------------------------------+
void OnTimer()
  {
//---

   string strWrite=TimeToString(TimeLocal())+";"+TimeToString(TimeCurrent())+";";

   string strAppend="True;";
   if(!IsConnected())   strAppend="False;";
   strWrite+=strAppend;

   int intCount=ArraySize(g_strSymbols);
   for(int i=0;i<intCount;i++)
   {
      string strSymb=g_strSymbols[i];

      double dblBid=MarketInfo(strSymb,MODE_BID);
      double dblAsk=MarketInfo(strSymb,MODE_ASK);
      double dblStop=MarketInfo(strSymb,MODE_STOPLEVEL);
      double dblFreeze=MarketInfo(strSymb,MODE_FREEZELEVEL);

      if(dblBid>0 && dblAsk>0)
      {
         strWrite+=DoubleToString(dblAsk-dblBid)+";";
         strWrite+=DoubleToString(dblStop)+";";
         strWrite+=DoubleToString(dblFreeze)+";";
      }
      else
      {
         strWrite+=("N/A;N/A;N/A;");
      }
   }

   g_objLog.AppendStringToLog(strWrite);

  }
//+------------------------------------------------------------------+
```