

○ 「 Win32API ; 任意フォルダ内のファイル・アクセス 」

・アメンボです、

実は、ある大問題解決のために「任意フォルダ」中のファイルをアクセス・解析するための DLL を時間をかけて何とか完成したのですが、その作成中に調べた資料で「Win32API (kernel32)」を使えば MQL4 からでも「絶対パス」が使えることに気が付いてチョット試したところ簡単に出来てしまい、気が抜けてしまいました。

( せっかく苦勞して作った DLL が〜・・・ )

・大問題解決のためには、更に「大きすぎる壁」が立ちはだかっているのが判ったので、この解決は後回しとして、本稿では「Win32API (kernel32)」を使った「任意フォルダ中」のファイル・アクセス方法を解説することにしました。

(アメンボは、今回初めて「Win32API」に触った超初心者ですので、内容不備はご容赦)

※なお、本コードの動作は「Windows XP」と「MT4」の組合せで試したのもです。

(恥ずかしながら、アメンボのメインPCは未だにXPなのです、もっともMT4との相性は最適)

※本稿「MQL4 コード」はダウンロード用に同時掲載してあります。

目次：

1. MQL4 のファイル・アクセス可能範囲	・・・	1 頁
2. Win32API によるファイル・アクセス		
(1) 疑問	・・・	2 頁
(2) 準備	・・・	3 頁
(3) 実施例	・・・	4 頁
3. MQL4 コード一覧	・・・	5 頁

1. MQL4 のファイル・アクセス可能範囲

・投稿済みの「MQL4 ; 基礎の確認 (その3)」でも解説したように、アクセス可能なフォルダは限定されています。

(使いやすい、と言え言えなくも無いですが)

資料を読んでいたら、MT 4 システム保護?のためとか書いてありましたが、ホントかどうかは不明です。

「ファイル・オープン関数」と「アクセス可能フォルダ」の関係 (再確認) ;

ファイル・オープン関数	アクセス可能な 「¥MetaTrader 4 at ブローカー名」 以下のフォルダ・パス名
FileOpen()	¥experts¥files
	¥tester¥files
FileOpenHistory()	¥history¥ブローカー server_name

## 2. Win32API によるファイル・アクセス

※本稿では、読み書きするデータは「テキスト（文字列）」との前提で解説します。

（絶対パス指定による任意の場所のファイル・アクセスを目指します）

### (1) 疑問

？「**user 32.dll**」はどこにある？と思った方、アメンボ以外にいませんか？

そう、MT4 に初めから付属している「**Period\_Converter.mq4**」のコードに出てくる DLL です。

・アメンボは散々、探してしまいましたが、とうとう判りませんでした。

でも、「**Period\_Converter.mq4**」をコンパイルしてみると、何の問題も無くコンパイル出来てしまうのです。

「**Period\_Converter.mq4**」の中身、抜粋；

```
#property copyright "Copyright © 2007, MetaQuotes Software Corp."
#property link      "http://www.metaquotes.net"
#property show_inputs
#include <WinUser32.mqh>
. . . . .
```

「**WinUser32.mqh**」の中身、抜粋；

```
#define  copyright "Copyright © 2004, MetaQuotes Software Corp."
#define  link      "http://www.metaquotes.net/"
#import "user32.dll"
//--- messages
int      SendMessageA(int hWnd,int Msg,int wParam,int lParam);
int      SendNotifyMessageA(int hWnd,int Msg,int wParam,int lParam);
int      PostMessageA(int hWnd,int Msg,int wParam,int lParam);
void     keybd_event(int bVk,int bScan,int dwFlags,int dwExtraInfo);
void     mouse_event(int dwFlags,int dx,int dy,int dwData,int dwExtraInfo);
. . . . .
```

・こんな話をするのは、Win32API の各種 DLL の内の、「**kernel32.dll**」が

ファイル・アクセス機能を提供することを知ったためです。

アメンボは当所、この「**kernel32.dll**」を散々探したのに見つからず、

どうしようか迷っていたところ、「**kernel32.dll**」や「**user 32.dll**」は個別の DLL として存在するわけではない！との記事を読みました。

・そこで、「**Period\_Converter.mq4**」の「**user 32.dll**」のように、

「**kernel32.dll**」もただ宣言するだけで、コンパイルしてみたところ、なんと OK でした。

（Win32API 超初心者のアメンボは遠周りが多い！！）

・「**kernel32.dll**」等は、PC が 16 ビットだった時代の古い遺産？？らしいのですが

使えるものは、ありがたく使わせて頂くこととしました。

## (2) 準備

- まず、MQL4 コード先頭の部分で、最低限使用する「関数類」をインポート宣言します。

```
#import "kernel32.dll"
int _lopen (string path, int of);
int _lcreat (string path, int attrib);
int _llseek (int handle, int offset, int origin);
int _lread (int handle, string buffer, int bytes);
int _lwrite (int handle, string buffer, int bytes);
int _lclose (int handle);
#import
```

- 引数と戻り値を解説； (必要最小限の記述のみ；ここには不記載のモード有り)

戻り値 (全て int)	関数	引数
失敗；マックス値<0	_lopen (string path, int of) ・ファイル・オープン	第1；ファイルへの絶対パス
成功；ファイルハンドル≥0		第2；オープン・モード 0-read only 1-write only 2-read/write
失敗；マックス値<0	_lcreat (string path, int attrib) ・ファイル新規作成	第1；ファイルへの絶対パス
成功；ファイルハンドル≥0		第2；作成モード 0-read/write 1-read only
失敗；マックス値<0	_llseek (int handle, int offset, int origin) ・カーソル基点 (初期) と位置 (シフト数) 指定	第1；ファイル・ハンドル
成功；カーソルシフト数≥0		第2；バイト単位で設定、 カーソル基点からのシフト数
		第3；カーソル基点 0-ファイルの先頭 1-現在位置 2-ファイルの最後尾
失敗；マックス値<0	_lread (int handle, string buffer, int bytes) ・ファイル内容を読み取る	第1；ファイル・ハンドル
成功；読取りバイト数 ファイル最後尾なら「0」		第2；読取り内容の格納先
		第3；読取るバイト数
失敗；マックス値<0	_lwrite (int handle, string buffer, int bytes) ・ファイルへ書き込む	第1；ファイル・ハンドル
成功；書込みバイト数		第2；書込む内容の格納先
		第3；書込むバイト数
成功；「0」	_lclose (int handle) ・ファイルを閉じる	ファイル・ハンドル
失敗；マックス値<0		

※「絶対パス」指定時の注意；

- ・「C:\myLog\API.txt」の様に記載します。・・フォントを「Century」等にした時の見え方  
(「C:¥¥myLog¥¥API.txt」と見える場合もあり・・フォントを「MS 明朝」等にしたとき)
- ・「\」(「¥」)を一つ余計に書くのは、初めの「\」(「¥」)はエスケープ・シーケンスで、  
2番目の「\」(「¥」)が渡される文字となるためです。

※フォルダの作成については、本稿では記載しません。

## (3) 実施例

※MQL4 コード詳細は「3. MQL4 コード」を参照ください

※実際に試してみる場合は「C:\myLog」フォルダを事前に作っておいてください。

－ 1. ファイルへの書き込み； 「Win32API\_write.mq4」

仕様： string path 先にファイルの有無を確認し、  
 ①無かった場合は新規に「ファイル」を作成して、  
 ②存在する場合は、その「ファイル」内に、  
 string buffer の文字列データを書き込む。

－ 2. ファイル内容の読み込み（その1）； 「Win32API\_read.mq4」

仕様： string path 先のファイルから文字列データを読み取り、その内容を  
 string buffer（文字列バッファ；固定長）に書き込む。

－ 3. ファイル内容の読み込み（その2）； 「Win32API\_read\_02.mq4」

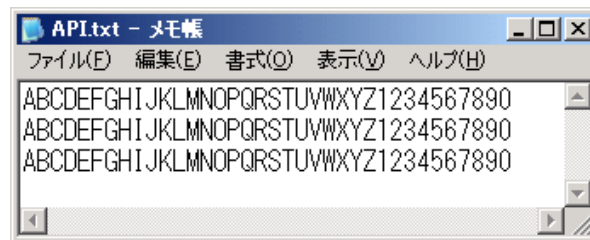
仕様： string path 先のファイルから文字列データを読み取り、その内容を  
 string buffer+add（文字列バッファ；可変長）に書き込む。

実施：①3回「Win32API\_write」スクリプトを実行した後、

②1回「Win32API\_read」を実施、更に、

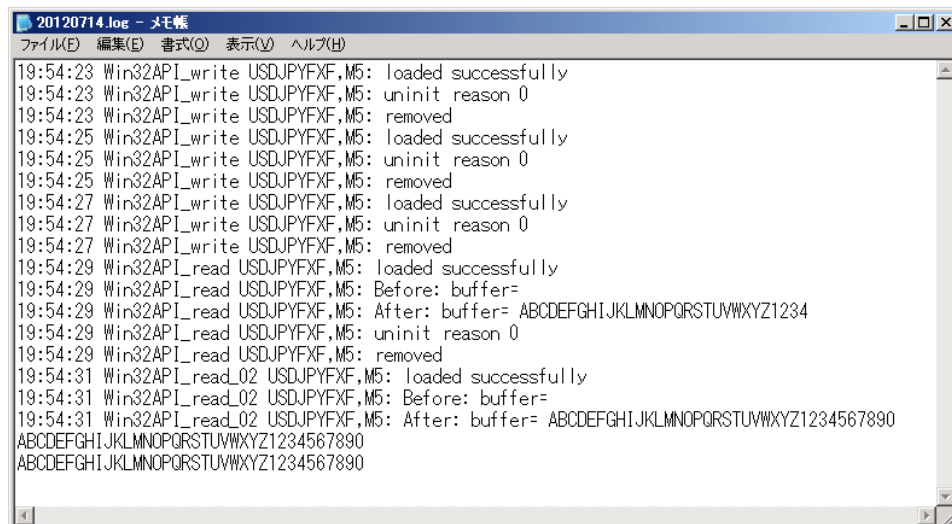
③1回「Win32API\_read\_02.」を実施して、終了する。

結果（その1）： 「C:\myLog\API.txt」の内容



結果（その2）：

C:\Program Files\MetaTrader 4 at ブローカ名.com\experts\logs\ 20120714.log の内容



### 3. MQL4 コード一覧

- 以下のファイル・アクセスは、テキスト・データを対象にしています。
- 以下のコードは、全て「スクリプト」です。

#### (1) ファイル書き込み例

- ファイルが存在しない場合は指定フォルダー中に「ファイル」を新規作成し、データを書き込みます。
- ファイルが存在する場合は、その中にデータを追記します。

(注) ; 新規にフォルダーを作る機能はありません

```
//+-----+
//|                               Win32API_write.mq4 |
//|                               amenbo |
//|                               |
//+-----+
#property copyright "amenbo"
//
#import "kernel32.dll"
    int _lopen (string path, int of);
    int _lcreat (string path, int attrib);
    int _llseek (int handle, int offset, int origin);
    int _lread (int handle, string buffer, int bytes);
    int _lwrite (int handle, string buffer, int bytes);
    int _lclose (int handle);
#import
//
//+-----+
//| script program start function |
//+-----+
int start()
{
//----
string path="C:\\myLog\\API.txt";
string buffer="ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890";
string cr="\r\n";//改行&リターン
    int count=StringLen (buffer);
    int result_o;
    int result_o2;
    int result_o3;
    int result_o4;

    int handle_o=_lopen (path,1);//write_mode
    //
    if(handle_o<0)
    {
        int handle_c=_lcreat (path,0);//create file(mode:reading or writing)
        int result_c=_lclose (handle_c);//close file
        handle_o= -1;
    }
    //
    if (handle_o<0)
```

```

{
    int handle_o2=_lopen (path,1);//try open_file again(write_mode)
    if(handle_o2<0)
    {
        Print("Error opening file ",path);
        return(0);
    }
}
//
if(handle_o>0)
{
    //
    result_o2=_llseek (handle_o,0,2);
    result_o3=_lwrite (handle_o,cr,2);// 2 バイト書き込み
    result_o3=_lwrite (handle_o,buffer,count);
    result_o4=_lclose (handle_o);

    //
}
else{
    //
    result_o2=_llseek (handle_o2,0,0);
    result_o3=_lwrite (handle_o2,buffer,count);
    result_o4=_lclose (handle_o2);
    //
}
//-----
//
if(result_o2<0)
{
    Print("Error placing cursor");
    return(0);
}
//
if(result_o3<0)
{
    Print("Error writing to file ",path," ",count," bytes");
    result_o4=_lclose (handle_o2);
}
//
if(result_o4<0)
{
    Print("Error closing file ",path);
    return(0);
}
//----
    PlaySound("alert2.wav");
//
//-----
return(0);
}

```

(2) 指定ファイル内から、string buffer (固定長) 内にデータを読み込みます

```
//+-----+
//|                               Win32API_read.mq4 |
//|                               amenbo |
//|                               type1 |
//+-----+
#property copyright "amenbo"
//
#import "kernel32.dll"
    int _lopen (string path, int of);
    int _lcreat (string path, int attrib);
    int _llseek (int handle, int offset, int origin);
    int _lread (int handle, string buffer, int bytes);
    int _lwrite (int handle, string buffer, int bytes);
    int _lclose (int handle);
#import
//+-----+
//| script program start function |
//+-----+
int start()
{
//---
    string path="C:\\myLog\\API.txt";
    string buffer="";

    Print("Before: buffer= ",buffer);
    int count=30;
    int result_o;
    int handle_o=_lopen (path,0);//read_mode
    //
    //
    if(handle_o<0)
    {
        Print("Error openig file",path);
        return(0);
    }
    //
    result_o=_llseek (handle_o,0,0);//set cursor top of file
    //
    if (result_o<0)
    {
        Print("Error placing cursor",path);
        return(0);
    }
    //
    //-----
    int result_o2=_llseek(handle_o,0,0);
    //-----
    //
    if(result_o2<0)
    {
        Print("Error placing cursor");
        return(0);
    }
    //
    //-----

```

```
    int result_o3=_lread(handle_o,buffer,count);
//-----
//
if(result_o3<0)
{
    Print("Error reading from file ",path," ",count," bytes");
    return(0);
}
//
    int result_o4=_lclose(handle_o);
//
if(result_o4<0)
{
    Print("Error closing file ",path);
    return(0);
}
//----
    Print("After: buffer= ",buffer);
//
    PlaySound("alert2.wav");
//
//-----
return(0);
}
```



(3) 指定ファイル内から、string buffer (可変長) 内にデータを読み込みます

```
//+-----+
//|          Win32API_read_02.mq4 |
//|          amenbo |
//|          type2 |
//+-----+
#property copyright "amenbo"
//
#import "kernel32.dll"
    int _lopen (string path, int of);
    int _lcreat (string path, int attrib);
    int _llseek (int handle, int offset, int origin);
    int _lread (int handle, string buffer, int bytes);
    int _lwrite (int handle, string buffer, int bytes);
    int _lclose (int handle);
#import
//+-----+
//| script program start function |
//+-----+
int start()
{
//---
    string path="C:\\myLog\\API.txt";
    string buffer="";//type2
    string add="K";//type2

    Print("Before: buffer= ",buffer);
    //int count=30;
    int count=0;//type2
    int result_o;
    int handle_o=_lopen (path,0);//read_mode
    //
    //
    if(handle_o<0)
    {
        Print("Error openig file",path);
        return(0);
    }
    //
    result_o=_llseek (handle_o,0,0);//set cursor top of file
    //
    if (result_o<0)
    {
        Print("Error placing cursor",path);
        return(0);
    }
    //
    //-----
    int result_o2=_llseek(handle_o,0,0);
    //-----
    //
    if(result_o2<0)
    {
        Print("Error placing cursor");
        return(0);
    }
}
```

```
    }
    //
    //-----
    //int result_o3=_lread(handle_o,buffer,count);
    int result_o3=_lread(handle_o,add,1);
    while(result_o3>0)
    {
        buffer=buffer+add;
        add="A";
        result_o3=_lread(handle_o,add,1);
    }
    //-----
    //
    /*
if(result_o3<0)
{
    Print("Error reading from file ",path," ",count," bytes");
    return(0);
}
//
*/
    int result_o4=_lclose(handle_o);
    //
if(result_o4<0)
{
    Print("Error closing file ",path);
    return(0);
}
//----
    Print("After: buffer= ",buffer);
    //
    PlaySound("alert2.wav");
    //
    //-----
    return(0);
}
```

以 上