

○ 「New MQL4 (Build 600 以降) ; 基礎 (その 2) OnTimer() を使う」 2014. 08. 07

・ アメンボです、

New_MQL4 の特徴とはなんでしょう？

極論を言うと、MQL5 から受け継いだ「OOP」、つまり「On***()」関数群と「標準クラス」ライブラリではないかと、小生は勝手に考えています。

・ 「On***()」関数群は、関連資料が比較的読みやすい！

と、言うわけで簡単なものからコツコツと解析・実証結果を報告します。

(一方、標準クラス・ライブラリは解析が難題です。機能は結構凄いのですが！)

<本稿で使用した MQL4 コード>

※単純なコードですが添付しました。「new_mql4_08_01.zip ; New MQL4 基礎 (その 2)」
(ZIP 形式で圧縮)

※本稿は「MT4 ; version 4.00 Build670」「MetaEditor ; version 5.00 Build966」にて確認済み。

目次：

1. 「On****()」ハンドリング関数 (MQL5 との比較) ・ ・ P 1
2. 「OnTimer()」の使い方 ・ ・ P 2 ~ P 5
 - (1) 基本事項
 - (2) EA での使用例
 - (3) インディケータでの使用例

1. 「On****()」ハンドリング関数 (MQL5 との比較)

New_MQL4 で使用可能な「ハンドリング関数」を MQL5 の場合と比較しながら、使用方法を解説します

		機能サポート		New MQL4 のサポート範囲			確認
		MQL5	New MQL4	EA 関数 使用	Indicator インディケータ 表示	Script スクリプト 実行	
ハンドリング関数	イベント・トリガと モード別						
OnStart()	—	○	○	○	—	○	
OnInit()	開始	○	○	○	○	—	
OnDeinit()	終了	○	○	○	○	—	
OnTick()	ティック	○	○	○	—	—	
	マルチレンジャー・モード	○	?	?	—	—	
OnTimer()	タイマー	○	○	○	○	—	本稿
OnTrade()	order・deal・position	○	—	—	—	—	
OnTradeTransaction()		○	—	—	—	—	
OnTester()	ストラテジー・テスター	○	○	○	—	—	
OnBookEvent()	板(DOM)情報	○	—	—	—	—	
OnChartEvent()	未確認	○	?	?	?	—	
	カスタム・イベント	○	○	○	○	—	
OnCalculate()	インディケータ表示計算	○	○	—	○	—	済
	簡略タイプ	○	?	—	?	—	

※OnTimer() はインディケータでも使用可能です。

2. OnTimer() の使い方

本稿では、「OnTimer()」を理解することを中心に解説しましたので、EAとインディケータの記述は形式的なものです。ただ、どちらの場合でも「OnTimer()」が使えることを示しました。

(1) 基本事項

- 1) 動作タイミング； タイマーにセットした時間が経過した時
- 2) 引数； なし
- 3) 戻り値型； void ・ ・ 要は戻り値は「なし」
- 4) 関連関数一覧；

関数	引数	戻り値	機能	備考
void OnTimer();	なし	なし	タイマーにセットされた時間に達したら実行される	OnInit()内でEventSetTimer()を呼出した場合は、
bool EventSetTimer(int seconds);	int seconds 秒数	論理 ・成功→true ・失敗→false	タイマーをセットする	OnDeinit()でEventKillTimer()を呼び出しておくことを推奨
void EventKillTimer();	なし	なし	タイマーを止める	

(2) EAでの使用例

- 1) コード；

```

//+-----+
//|                                     OnTimer_01.mq4 |
//|                                     amenbo |
//|                                     泉の森の弁財天池
//+-----+
#property copyright "amenbo"
#property link      "泉の森の弁財天池"
#property version   "1.00"
#property strict

// Expert Advisor
bool start = true;
int count;

//
int OnInit()
{
    //
    count=0;
    //--- create timer//普通はここで設定することが多い
    // EventSetTimer(60);
    return(INIT_SUCCEEDED);
}

//
void OnDeinit(const int reason)
{
    //--- destroy timer
    EventKillTimer();
}

// Expert tick function

```

```

void OnTick()
{
    if(start){
        bool setTimer = EventSetTimer(1);//OnTimer()を「1秒」ごとにセット
        if(setTimer){
            printf("OnTick()が来て、タイマー設定をした");
            start = false;
        }else{
            printf("タイマー設定に失敗");
        }
    }else{
        printf("OnTick()が来たけど、タイマー設定しない");
    }
}

// Timer function
void OnTimer()
{
    count=count+1;

    printf("OnTimer()が呼ばれた: %d 回目",count);
    PlaySound("alert2");

    if(count==2)
    {
        //--- destroy timer
        EventKillTimer();
        count=0;
        start = true;
    }
}
//+-----+

```

2) 結果 (例) ;

時間	メッセージ
2014.08.07 00:40:5...	OnTimer_01 USDJPYFXF,Daily: OnTimer()が呼ばれた : 2 回目
2014.08.07 00:40:5...	OnTimer_01 USDJPYFXF,Daily: OnTimer()が呼ばれた : 1 回目
2014.08.07 00:40:4...	OnTimer_01 USDJPYFXF,Daily: OnTick()が来て、タイマー設定をした
2014.08.07 00:40:2...	OnTimer_01 USDJPYFXF,Daily: OnTimer()が呼ばれた : 2 回目
2014.08.07 00:40:2...	OnTimer_01 USDJPYFXF,Daily: OnTimer()が呼ばれた : 1 回目
2014.08.07 00:40:1...	OnTimer_01 USDJPYFXF,Daily: OnTick()が来て、タイマー設定をした
2014.08.07 00:40:1...	OnTimer_01 USDJPYFXF,Daily: initialized
2014.08.07 00:40:1...	Expert OnTimer_01 USDJPYFXF,Daily: loaded successfully

ターミナル

取引 | 口座履歴 | ニュース | アラーム設定 | メールボックス | 会社名 | マーケット | ライブラリ | エキスパート | 操作履歴 |

(3) インディケータでの使用例

1) コード ;

```

//+-----+
//|                                     OnTimer_ind_01.mq4 |
//|                                     amenbo |
//|                                     泉の森の弁財天池
//+-----+
#property copyright "amenbo"
#property link      "泉の森の弁財天池"
#property version   "1.00"
#property strict
#property indicator_chart_window

// Custom indicator
bool start = true;
int count;

// initialization function
int OnInit()
{
    //--- indicator buffers mapping
    count=0;
    //
    return(INIT_SUCCEEDED);
}

//
void OnDeinit(const int reason)
{
    //--- destroy timer
    EventKillTimer();
}

// Custom indicator iteration function
int OnCalculate(const int rates_total,
                const int prev_calculated,
                const datetime &time[],
                const double &open[],
                const double &high[],
                const double &low[],
                const double &close[],
                const long &tick_volume[],
                const long &volume[],
                const int &spread[])
{
    if(start){
        bool setTimer = EventSetTimer(1); // OnTimer()を「1秒」ごとにセット
        if(setTimer){
            printf("OnCalculate()が来て、タイマー設定をした");
            start = false;
        }else{
            printf("タイマー設定に失敗");
        }
    }else{
        printf("OnCalculate()が来たけど、タイマー設定しない");
    }
}

```

