

○MQL 5 ; 翻訳まとめ「OnBookEvent()の使い方、他(その1)」 翻訳のみ実施 2012. 02. 12

本稿の翻訳対象は「OnInit()、OnDeinit()、OnTimer()、OnBookEvent()」4関数です。

注意 ; ・本資料は、まだMT 5での動作・検証を行っていません、

近々の検証用資料として、英文資料を意識しながら纏めたもの(メモ)です。
訳した資料がある程度たまったところで、MT 5をダウンロードして確認して
いくつもりです。・・・すいません、まだMT 5は使ったことが無いのです！

(実機で未検証の内容ですので、誤訳があるかもしれません)

・以上の状況を理解されたうえで、本稿内容を参照ください。

○本稿を「(その1)」としたのは、基本内容のみを記述したので、
別の機会に応用や実施例等を報告しようと考えているからです。

目次 :

1. イベント「ハンドリング関数とトリガ」	・・・ P 1
2. OnInit()とOnDeinit()のチョットした知識	・・・ P 2
3. OnTimer()の使い方	・・・ P 4
4. OnBookEvent()の使い方	・・・ P 5

1. イベント「ハンドリング関数とトリガ」

※MQL4の3個「init()、deinit()、start()」に比較して、MQL5ではイベント・ハンドリング関数がやたら増えています。MQL4に慣れた身には混乱の種ですが、一つ一つ解析・解説していくことにしました。まず、全体と一応解説済みのものを整理してみます。

ハンドリング関数	イベント・トリガ	EA 実行	インディケータ表示	スクリプト実行	解説
OnStart	—	—	—	○	? 改めて説解説必要
OnInit	開始	○	○	—	本稿
OnDeinit	終了	○	○	—	本稿
OnTick	ティック	○	—	—	? 改めて解説必要
OnTimer	タイマー	○	—	—	本稿
OnTrade	注文・ポジション	○	—	—	未<別途>
OnTester	ストラテジー・テスター	○	—	—? 調査中	済; その1
OnBookEvent	板(DOM)情報	○	—	○? 調査中	本稿
OnChartEvent	未確認	○			未<別途>
OnCalculate	インディケータ表示計算		○	—	済; その1

※DOM : Depth of Market ・ ・ 要するに「板情報」のこと

2. OnInit() と OnDeinit() のちょっとした知識

(1) OnInit()

1) 動作タイミング ;

- ① E A やインディケータがチャートにダウンロードされたときに、一番最初に実行される。(初期化)
- ② 設定チャートの Symbol (為替ペア)、Period (周期) が変更されたとき
- ③ インプット・パラメータ (extern 宣言されたパラメータ) が変更されたとき

2) 引数 ; なし

3) 戻り値型 ; 「void と int」の2種類ある

	戻り値タイプ	戻り値	意味	void Deinit() への影響 (イベント発生)
OnInit	void OnInit()	なし	常に正常初期化	なし
	int OnInit()	0	正常初期化	なし
		0 以外	異常あり	REASON_INITFAILED を発生

4) 使用例 「int OnInit()」

```
double   MA[];           // array for the indicator iMA
int      MA_handle;     // handle of the indicator iMA
//-----
int OnInit()
{
    MA_handle=iMA(NULL,0,21,0,MODE_EMA,PRICE_CLOSE); //creation of the indicator iMA
    //--- report if there was an error in object creation
    if(MA_handle<0)
    {
        Print("The creation of iMA has failed: MA_handle=",INVALID_HANDLE);
        Print("Runtime error = ",GetLastError());
        return(-1);
    }
    return(0);
}
```

(2) OnDeInit()

1) 動作タイミング ;

- ① E A やインディケータがチャートから削除されたとき
- ② 設定チャートの Symbol (為替ペア)、Period (周期) が変更されたとき
- ③ インプット・パラメータ (extern 宣言されたパラメータ) が変更されたとき

2) 引数 ; `const int reason`

`reason` (=OnDeinit() イベントが発生した理由) が渡される。

3) 戻り値型 ; `void`

4) 使用例

```
void OnDeinit(const int reason)
{
//--- Get the uninitialization reason code
Print(_FUNCTION_, "_Uninitialization reason code = ", reason);
}
```

(補足) `_FUNCTION_` ; コードが書かれている「関数名」

5) OnDeinit() イベントが発生した理由 (reason) のコード一覧

識別定数名	値 reason	意味
REASON_PROGRAM	0	プログラム中の ExpertRemove() 関数によって EAが止まり、チャートからアンロードされた
REASON_REMOVE	1	プログラムがチャートから削除された
REASON_RECOMPILE	2	プログラムが再度コンパイルされた
REASON_CHARTCHANGE	3	為替ペアかチャート周期が変更された
REASON_CHARTCLOSE	4	チャートが閉じられた
REASON_PARAMETERS	5	インプット・パラメータが変更された
REASON_ACCOUNT	6	新たなアカウントに権限が移った
REASON_TEMPLATE	7	新たなテンプレートが適用された
REASON_INITFAILED	8	OnInit() ハンドラーが「0」以外の値を返した
REASON_CLOSE	9	ターミナルが閉じられた

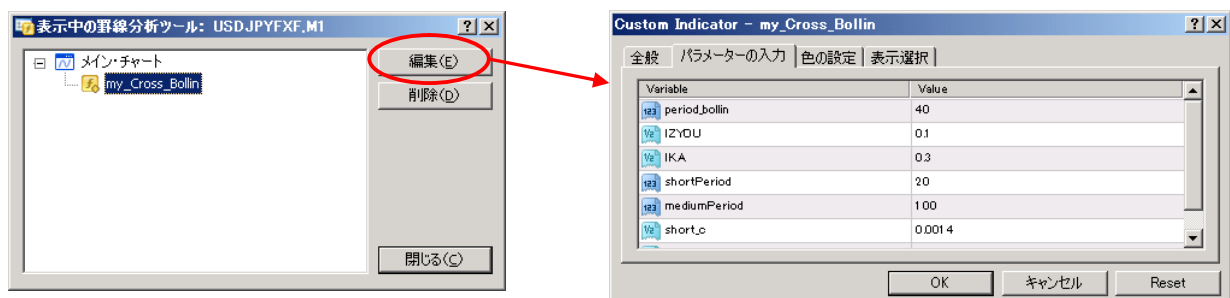
※動作タイミングについての補足；

- ・設定チャートの Symbol (為替ペア)、Period (周期) が変更されたとき
- ・インプット・パラメータ (extern 宣言されたパラメータ) が変更されたとき

上記の場合には、「OnInit() から始まり、OnDeinit() で終わる」1 サイクルが繰り返されることに注意。

MT 4 の場合で言えば、

下記の手順等で「インプット・パラメータ」が変更された場合に相当する、と思う。



3. OnTimer() の使い方

- 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、 OnDeinit()で EventKillTimer()を 呼び出しておくことを推奨
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	
void EventKillTimer();	なし	なし	タイマーを止める	

5) 使用例

```

bool start = true;
//
int OnInit()
{
    return(0);
}
//
void OnDeinit(const int reason)
{
}
// Expert tick function 要はEA本体
void OnTick()
{
//---
    if(start){
        bool setTimer = EventSetTimer(2);
        if(setTimer){
            printf("success set timer");
            start = false;
        }else{
            printf("failed set timer");
        }
    }else{
        printf("no need to set timer");
    }
}
//-----
void OnTimer(){
    printf("on timer");
}

```

※一定の時間が経過したら、「SendMail()」するなど色々な応用例が可能。

4. OnBookEvent()の使い方

<機能はいったい何?>

※アメンボが関数名を観たときの第一印象は「??」でした。(アメンボだけか?)

(通常は「関数名」からは、漠然ではあっても使用目的がイメージできるのですが)
イメージを掴むのに適した言葉としては「ブッキング」が良い、と、後で思いついた。

※「OnBookEvent()」は、「DOM(Depth of Market) (またまた訳の判らない用語!の出現) に、
何らかの変化(イベント)があったときに呼び出される、とありました。
(ここまで調べても、まだ具体的なイメージは全く湧きませんでした!)

<ようやく判った!(つもり?かな)>

●要は、**板情報**(下記)に何らかの変化(イベント)が起こったときに、呼び出されます。

(MT5では板情報が使える?、FX会社に依存するのでは?、など疑問が多々残りますが)

●Depth of Market (DOM; 要は板情報)の表示例;

メニュー;「Market Watch」-「Depth of Market」で表示される。

Buy	Bid	Price	Ask	Sell
		183700	1	
		183695	3	
		183690	1	
		183655	12	
		183650	2	
	1	183610		
	20	183600		
	6	183595		
1	1	183590		
	1	183580		

上側のボタン

ボタン名	実行内容(pending オーダーキャンセル)
Cancel Buy	・Buy Limit、Buy Stop オーダーをキャンセル
Cancel All	・全ての pending オーダーをキャンセル
Cancel Sell	・Sell Limit、Sell Stop オーダーをキャンセル

中段の表示

表示	内容
Buy	トレード者自信の買いオーダー・ロット数を表示 他者からは見えない
Bid	各 Bid(買値)のロット数
Price	売買価格
Ask	各 Ask(売値)のロット数
Sell	トレード者自信の売りオーダー・ロット数を表示 他者からは見えない

下側のボタン

ボタン名・ロット指定 BOX	実行内容
Sell ボタン	買いオーダーの発行(最も有利な価格でのオーダーを出す)
Volume 設定 BOX	売買ロット数の設定
Buy ボタン	売りオーダーの発行(最も有利な価格でのオーダーを出す)
Close ボタン	ポジション手仕舞い(「買い」または「売り」のオープン・ポジションがあるときのみ表示される)

- 1) 動作タイミング ; 板情報 (Depth of Market ; DOM) に何らかの変化が起きたとき
- 2) 引数 ; なし
- 3) 戻り値型 ; void ・ ・ 要は戻り値は「なし」
- 4) 関連関数

関数	引数	戻り値	機能
void OnBookEvent(const string& symbol);	・symbol 利用するDOM情報の 通貨ペア	なし	DOMに変化が起きたとき 実行される
bool MarketBookAdd(string symbol);	・symbol 利用する 通貨ペア	論理 ・成功→true ・失敗→false	利用したいDOMを EA やスクリプト中で オープンする(設定する)
bool MarketBookGet(string symbol, MqlBookInfo& book[]);	・symbol 利用する通貨ペア ・book[] 利用する構造体配列	論理 ・成功→true ・失敗→false	指定する通貨ペアのDOM 情報を構造体配列に 取り込む (※1)
bool MarketBookRelease(string symbol);	・symbol 利用していた 通貨ペア	論理 ・成功→true ・失敗→false	オープン(設定)した DOMを閉じる

5) 関連構造体・識別定数名

①構造体 ;

構造体	メンバー(DOMの内容に対応)
struct MqlBookInfo{ ENUM_BOOK_TYPE type; double price; long volume; }	・type 買い・売りのタイプ ・price 板上に表示された価格 ・volume 板上に表示されたボリューム

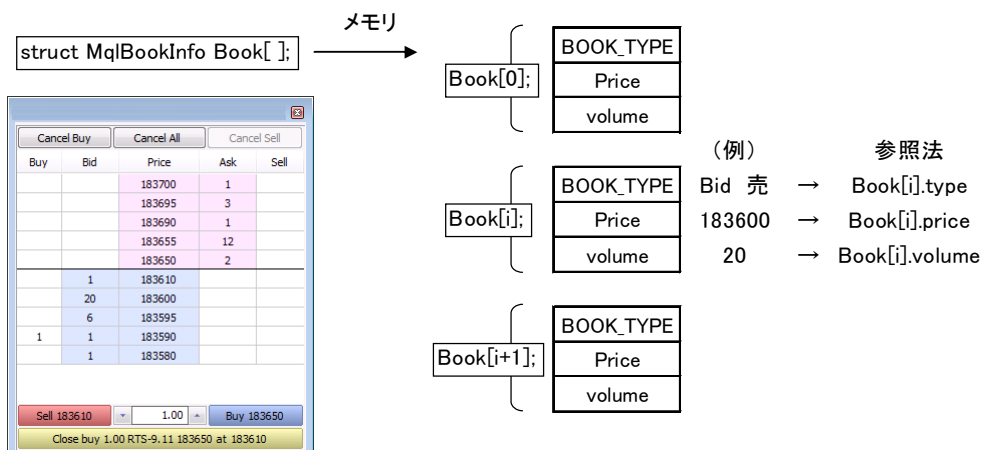
②ENUM_BOOK_TYPE ;

識別定数名	内容
BOOK_TYPE_BUY	買いオーダー (Bid:ディーラ側)
BOOK_TYPE_SELL	売りオーダー (Offer:ディーラ側)

※いつもトレード側とディーラ側からの表現で混乱するのはアメンボだけ？

③構造体配列 ;

MqlBookInfo Book[] : MqlBookInfo という構造を持つ Book[] という「配列」を宣言
(※1) MarketBookGet(NULL,Book) によって DOM 情報を Book 配列にコピーする



6) 使用例 ;

①単純な機能確認用コード (翻訳)

```

int OnInit()
{
    MarketBookAdd(USDJPYFXF);
    ....
    return(0);
}
void Deinit()
{
    MarketBookRelease(USDJPYFXF);
    ....
}
void OnTick()
}
    ....
    Tick 値が更新されるのに伴う処理を記述 (EA の本体記述)
{
void OnBookEvent(const string& symbol)
{
    MqlBookInfo myBook[];
    bool getBook=MarketBookGet(NULL,myBook);
    if(getBook)
    {
        int size=ArraySize(myBook);
        Print("MarketBookInfo for ",Symbol());
        for(int i=0;i<size;i++)
        {
            Print(i+":",myBook[i].price
                +" Volume = "+ myBook[i].volume, " type = ", myBook[i].type);
        }
    }
    else
    {
        Print("Could not get contents of the symbol DOM ",Symbol());
    }
    ....
}

```

②実際に使用する場合の骨格は、こんな感じかな・・・??

```

MqlBookInfo myBook[], prevBook[];//2つの構造体配列を準備する
.....
int OnInit()
{
    MarketBookAdd(USDJPYFXF);
    MarketBookGet(NULL,prevBook);//初期値としての DOM 情報を prevBook[]に取得
    .....
    return(0);
}
void Deinit()
{
    MarketBookRelease(USDJPYFXF);
    .....
}
void OnTick()
}
.....
Tick 値が更新されるのに伴う処理を記述 (EA の本体記述)
{
void OnBookEvent(const string& symbol)
{
    bool getBook=MarketBookGet(NULL,myBook);//変化した DOM 情報を myBook[]に取得
    if(getBook)
    {
        ・prevBook[]と myBook[]を比較して、変化した内容を把握する
        ・意図する処理を実行・・
        ・myBook[]の内容を prevBook[]にコピーする
    }
    .....
}
}

```

7) 特記事項 ; ブロードキャスト

- ・ BookEvent は他のイベントとは異なり、ブロードキャストです。
(翻訳してるだけで、アメンボは未確認ですけど)

どういうことかと言うと、

- ① 1つのEAが「MarketBookAdd」により BookEvent を呼び出すと、
- ②他のEAでも「OnBookEvent()」によって、イベントの受け取りが可能となる。

従って、各EAは「const string& symbol」パラメータによって、自身の対象とするイベントか否かを判断する必要があります。

以上