○「New MQL4（Build 600 以降）；基礎（その５）OnChartEvent()［２／２］」2014.10.02
（　EventChartCutom()と組み合わせて使う　）

・アメンボです、

本稿は「OnChartEvent()」の内の、チョット特殊な**カスタム・イベント**の使い方です。
前回述べたように、イベントは大きく下記の「２種類」に分類されており、

① MQL5（システム）備え付けのイベント
② ユーザーが任意に設定するカスタム・イベント

本稿は「②」の検証結果の解説です。
調査当初、とにかく資料が少ない（あっても、何のこっちゃ？）と言う状況でした、
しかしながら、判ってしまえば「なんてことはなく」、コロンブスの卵、と言うところです。

・さてと、

カスタム・イベントで、NewMQL4 で新規に使えるようになった主だった
「イベント・ハンドリング関数」の解説は終わってしまいます。(小物は残りますが)

・ところで、

既にご存知のように、New_MT4 は次から次へと目まぐるしくバージョンアップしており、
この記事を書いてる最中に（英語版）最新バージョンは「Build711」になったとの、
ニュースを読みました。(どこまで行くんでしょう？)

・どうせなら、

Old_MQL4(MT4)の機能・関数は極力そののままで、且つ MQL5(MT5)の機能・関数も全て使える、
のが、個人的にはうれしいのですが！（でも、入門書が無いのが悩ましい！）

＜本稿で使用した MQL4 コード＞

※使用コード添付；「new_mql4_2014_10_01.zip」；New MQL4 ChartEvent() [2/2]」(ZIP 形式圧縮)
※本稿は「MT4；version 4.00 Build670」「MetaEditor；version 5.00 Buid966」にて確認済み。

目次：

1．イベント「ハンドリング関数とトリガ」一覧（MQL5 との比較）

※「On****()」形式のハンドリング関数のこと

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

		機能ｻﾎﾟｰﾄ		New MQL4 のｻﾎﾟｰﾄ範囲			確認
				EA	Indicator	Script	
ﾊﾝﾄﾞﾘﾝｸﾞ関数	ｲﾍﾞﾝﾄ・ﾄﾘｶﾞ と ﾓｰﾄﾞ別	MQL5	New MQL4	関数 使用	ｲﾝﾃﾞｨｹｰﾀ 表示	ｽｸﾘﾌﾟﾄ 実行	
OnStart()	—	○	○	○	—	○	
OnInit()	開始	○	○	○	○	—	
OnDeinit()	終了	○	○	○	○	—	
OnTick()	ﾃｨｯｸ	○	○	○	—	—	
	ﾏﾙﾁｶﾚﾝｼｰ・ﾓｰﾄﾞ	○	?	?	—	—	
OnTimer()	ﾀｲﾏｰ	○	○	○	○	—	**済**
OnTrade()	order・deal・position	○	—	—	—	—	
OnTradeTransaction()		○	—	—	—	—	
OnTester()	ｽﾄﾗﾃｼﾞｰ・ﾃｽﾀｰ	○	○	○	—	—	**済**
OnBookEvent()	板(DOM)情報	○	—	—	—	—	
OnChartEvent()	10 種類	○	○	○	○	—	**済**
	ｶｽﾀﾑ・ｲﾍﾞﾝﾄ	**○**	**○**	**○**	**○**	—	**本稿**
OnCalculate()	ｲﾝﾃﾞｨｹｰﾀ表示計算	○	○	—	○	—	**済**
	簡略ﾀｲﾌﾟ	○	—	—	?	—	

2．Chart Event の種類（OnChartEvent()が呼出されるイベント）

・OnChartEvent()を呼び出すイベントの内、システム備え付けのものは「10 種類」あります。

＜イベント；ID とパラメータ＞

	ｲﾍﾞﾝﾄ（割込）発生	ID	概要	返し値		
				lparam	dparama	sparam
1	キーが押された	CHARTEVENT_KEYDOWN	どのｷｰが押されたか	ｷｰ・ｺｰﾄﾞ	—	—
2	マウスが動いた	CHARTEVENT_MOUSE_MOVE	ﾏｳｽの動きをﾌｫﾛｰ	X 座標	Y 座標	ﾋﾞｯﾄ・ﾏｽｸ値 ﾎﾞﾀﾝ検出用
3	ｸﾞﾗﾌｨｶﾙ・ｵﾌﾞｼﾞｪｸﾄの作成	CHARTEVENT_OBJECT_CREATE	—	—	—	作成された ｵﾌﾞｼﾞｪｸﾄ名
4	ｸﾞﾗﾌｨｶﾙ・ｵﾌﾞｼﾞｪｸﾄの変更	CHARTEVENT_OBJECT_CHANGE	—	—	—	変更された ｵﾌﾞｼﾞｪｸﾄ名
5	ｸﾞﾗﾌｨｶﾙ・ｵﾌﾞｼﾞｪｸﾄの削除	CHARTEVENT_OBJECT_DELETE	—	—	—	削除された ｵﾌﾞｼﾞｪｸﾄ名
6	ﾁｬｰﾄ上で ﾏｳｽがｸﾘｯｸされた	CHARTEVENT_CLICK	ｸﾘｯｸした座標検出	X 座標	Y 座標	—
7	ｸﾞﾗﾌｨｶﾙ・ｵﾌﾞｼﾞｪｸﾄ上でﾏｳｽ・ｸﾘｯｸされた	CHARTEVENT_OBJECT_CLICK	ｵﾌﾞｼﾞｪｸﾄがある ﾁｬｰﾄ上の座標検出	X 座標	Y 座標	ｸﾘｯｸされた ｵﾌﾞｼﾞｪｸﾄ名
8	ｸﾞﾗﾌｨｶﾙ・ｵﾌﾞｼﾞｪｸﾄがﾏｳｽでﾄﾞﾗｯｸﾞされた	CHARTEVENT_OBJECT_DRAG	—	—	—	ﾄﾞﾗｯｸﾞされた ｵﾌﾞｼﾞｪｸﾄ名
9	ｵﾌﾞｼﾞｪｸﾄのﾗﾍﾞﾙが編集された	CHARTEVENT_OBJECT_ENDEDIT	—	—	—	ﾗﾍﾞﾙ編集済み ｵﾌﾞｼﾞｪｸﾄ名
10	チャート変更	CHARTEVENT_CHART_CHANGE	表示ﾁｬｰﾄの変更	—	—	—
11	ﾕｰｻﾞｰ定義ｲﾍﾞﾝﾄが発生した	CHARTEVENT_CUSOM+N （カスタム・イベント）	EventChartCustom()が実行された時に起動する	※1	※1	※1

※1；EventChartCustom() によって設定した値「long, double, string」が 1 個づつ渡される。

N=0 の場合は「ID= CHARTEVENT_CUSOM」、N=LastNo の場合は「ID= CHARTEVENT_CUSOM_LAST」

３．関数書式と引数

（１）OnChartEvent()；　再確認

```
void OnChartEvent (
                const int id,           //イベント ID（識別子）
                const long& lparam,     //イベント・パラメータ（long タイプ）
                const double& dparama,  //イベント・パラメータ（double タイプ）
                const string& sparam    //イベント・パラメータ（string タイプ）
                  )
```

※「id」により、どの様なイベントが発生したかを判別することが可能であり、また
「パラメータ；lparam、dparam、sparam」により更に詳細な情報を得ることが出来る。
・・例えば、「id」によりマウスがチャート上で「クリック」されたことを判別し、
「パラメータ」により、クリックされたチャート上の「位置」を知る事が出来る。

（２）EventChartCustom()　関数

```
bool  EventChartCustom(
   long     chart_id,           // イベントを受取る側のチャート ID を指定する
   ushort   custom_event_id,    // ユーザー設定カスタム・イベントの ID（識別子）
   long     lparam,             // イベント・パラメータ（long タイプ）
   double   dparam,             // イベント・パラメータ（double タイプ）
   string   sparam              // イベント・パラメータ（string タイプ）
   );
```

パラメータ

パラメータ		指　定　内　容
chart_id	[in]	カスタム・イベントの受取側チャート ID を指定する； ・「0」とすると現在のチャート、つまり EventChartCustom を含む mql4 コードが実行されているチャートが送信先（受信側）になります。 ・通常は ChartFirst（）や ChartNext（）等で取得した、別チャートの チャート ID（多分、ハンドル値）を指定することで、カスタム・イベントを 別のチャートに送信し、そのチャートに設定されたＥＡやインディケータ 上に「OnChartEvent」を発生させることが可能となります。
custom_event_id	[in]	ユーザー設定カスタム・イベントの ID で、追加で設定したときに、 特に指定しなければ自動的に「CHARTEVENT_CUSTOM」は「プラス１」される。　この ID は「CHARTEVENT_CUSTOM」から「CHARTEVENT_CUSTOM_LAST」までの「65536」通りが可能。
lparam	[in]	受信先の OnChartEvent()の「const long& lparam」に渡す、 イベント・パラメータ（long タイプ）を、ここで設定する。
dparam	[in]	受信先の OnChartEvent()の「const double& dparama」に渡す、 イベント・パラメータ（double タイプ）を、ここで設定する。
sparam	[in]	受信先の OnChartEvent()の「const string& sparam」に渡す、 イベント・パラメータ（string タイプ）を、ここで設定する。 もし、string が「63」キャラクター以上の場合は切り詰められる。

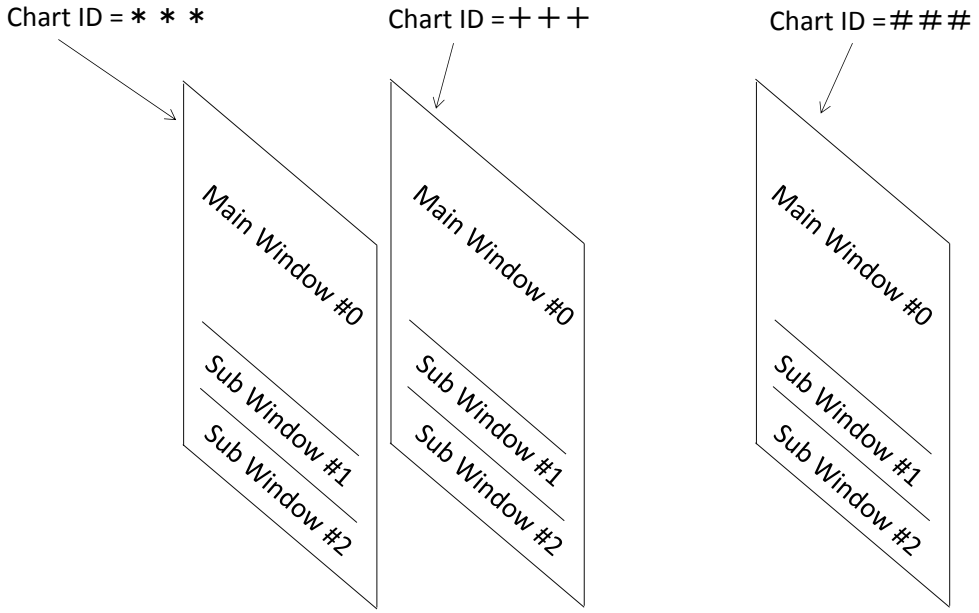
返し値；　成功すると「true」を、失敗すると「false」を返す。
エラーコードは「GetLastError()」で入手する。
ノート；　OnChartEvent()コードが使えるのは「Expert Advisor かインディケータ」。

４．EventChartCustom()の使い方

※カスタム・イベントは、解析に非常に手間取りました、
英語版 MQL5 にもロクな資料（と言うか、判りやすい資料）が、見当たらないのです。
ただ、判ってしまえば「実に簡単な構造（仕掛け）」です。（コロンブスの卵、ですね）
以下、なるべく判りやすく！？解説したつもりです。

（１）予備知識・・「chart ID」について

NewMT4 では、MT5 と類似の「Chart ID（チャート ID)」と言う概念があります。
＜概念図＞



・NewMT4 では、チャートを同時に「１００枚」まで開くことができます。
このチャートを NewMQL4 では「Chart ID」（多分ハンドルです）によって管理・識別します。
・「Chart ID」の値（long 型）は、開かれた時期によって「一定の規則」に従って決定されますが、筆者にはその規則が良く判りません。
ただし、一番最初に開かれたチャートが一番若い番号に、最後に開かれたチャートが一番大きな番号になります。（チャートと閉じてしまうと、その順番はリセットされるようです）
・「Chart ID」と「メイン・ウインドウ、サブ・ウインドウ」の概念は異なります。

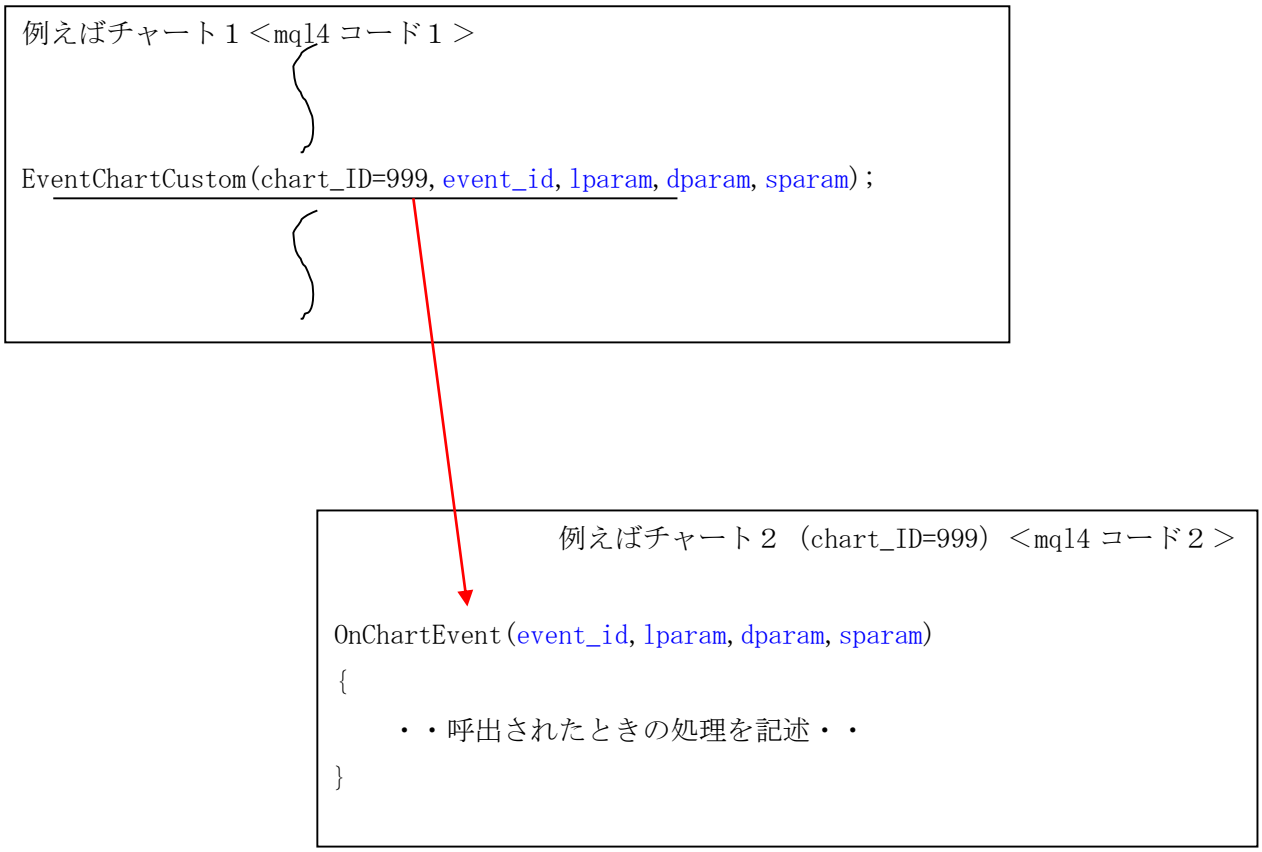
※本稿で使用した「Chart」関係の関数（NewMQL4）を下記に記載しておきます。

型・書式	機能（内容）
long ChartID()	**現チャートのID（多分、ハンドル値だと思う）を返す**
long ChartFirst()	最初に開いたチャートのＩＤを返す
long ChartNext(long chart_id // Chart ID);	・「long chart_id 」で指定したチャートの 次に開かれたチャートのＩＤを返す
string ChartSymbol(long chart_id // Chart ID);	**・「long chart_id」で指定したチャートの Symbol（為替ペア名）を返す ・「chart_id=0」とすると、コードを実行した現チャートの Symbol（為替ペア名）を返してくる。**

（２）基本動作の概念図

・「EventChartCustom(chart_ID,,,,)」が実行されると、このイベントがトリガとなり
「chart_ID」で指定するチャート上のＥＡかインディケータの「OnChartEvent()」が実行されます。
（ゆえに、チャート・イベントとして分類している様です）

＜イメージ図＞



①EventChartCustom()の行が実行されると、これをトリガとして、
②chart_ID で指定されたチャート上の OnChartEvent()コードが実行されます。

・「コード１」と「コード２」は同じチャート上でも、
あるいは、別々のチャート上に設定されていても構いません。
・「青書」のパラメータ値がそのまま引き渡されます。
・「chart_ID」で引渡し先のチャートを指定するのがポイントです。
※発信側と受信側の組合せ；

	発信側 EventChartCustom()	受信側 OnChartEvent()	備考
1	インディケータ	ＥＡ	動作確認済み
2	インディケータ	インディケータ	動作確認済み
3	ＥＡ	インディケータ	未確認
4	ＥＡ	ＥＡ	未確認

「３、４」は、諸兄にて確認してみてください。

（注）バックテストで動作するのかは、「１～４」全てで未確認です！

５．実例；「他チャート（為替ペア）」の最新「Open 値」を監視する

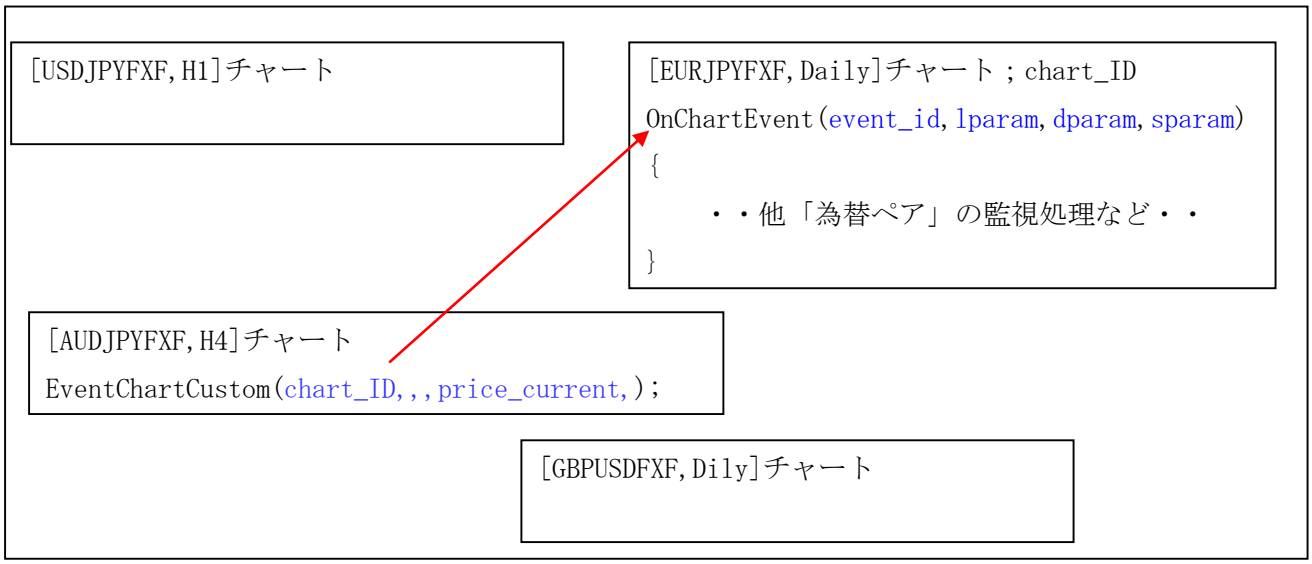
※他チャートの「Open 値」などは、「iOpen()」を使えば判るので、ちっとも面白くないのですが、
一例として採用しました。（実を言うと、面白い例を思いつかなかっただけ、なのですが）

（１）本稿での実現仕様

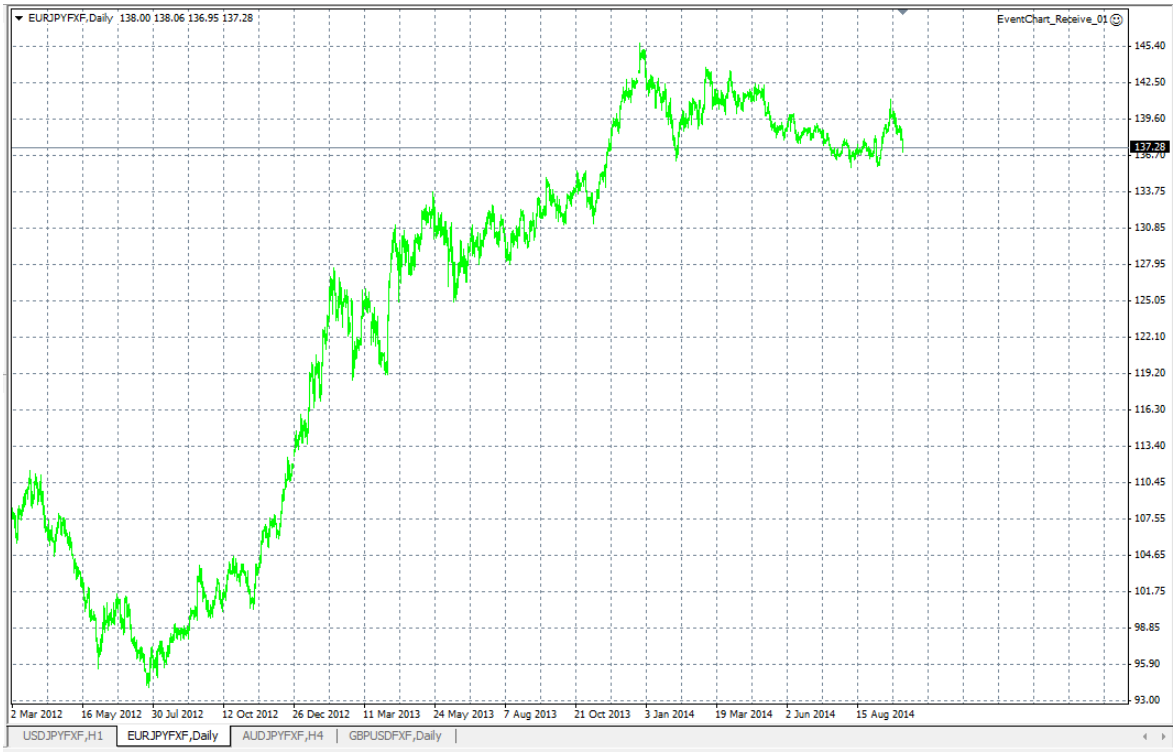
①「送信側」；　[AUDJPYFXF,H4] チャート上に、「EventChartCustom()を含むコード」を記載し、
chart_ID を「EURJPYFXF,Daily」に設定する。

②「受信側」；　[EURJPYFXF,Daily] チャート上に、「OnChartEvent()を含むコード」を設定、
if(event_id>=CHARTEVENT_CUSTOM)　でカスタム・イベントを検出する。

＜イメージ図（例）＞



・チャートは「４つ」開いています。
開いた順番は［USDJPYFXF］→［EURJPYFXF］→［AUDJPYFXF］→［GBPUSDFXF］



（2）送信側コード（インディケータ）

※本稿では［AUDJPYFXF,H4］チャート上に設定します（本当はどこでも良い）
　また動作確認用に、余分な「Print()と PlaySound()」を入れてあります。

```
//+------------------------------------------------------------------+
//|                                          EventChart_Send_02.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
//
input ushort   custom_event_id=0; // event id
input string target="EURJPYFXF";
//
int End;
//string target="USDJPYFXF";
//string target="EURJPYFXF";
//
long chart_id_array[50];
string chart_pare_name[50];
//
//+------------------------------------------------------------------+
//| Custom indicator initialization function                         |
//+------------------------------------------------------------------+
int OnInit()
  {
   //--- indicator buffers mapping

   //---
   return(INIT_SUCCEEDED);
  }
//+------------------------------------------------------------------+
//| 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[])
  {
   //  本来はここにインディケータ本体を書く
   //（本稿ではカスタム・イベントの例のみ記載）
   //----------------------------------------
   //  実施手順；
   //  １．全チャートの ChartID を取得する
   //  ２．ChartID と ChartSymbol の対応をつける
   //  ３．特定の ChartSymbol に EventChartCustom を送信する
   //----------------------------------------
   //
```

```
   //  ＜１．全チャートの ChartID を取得する＞
   //  ＜２．ChartID と ChartSymbol の対応をつける＞
   //
   chart_id_array[0]=ChartFirst();
   chart_pare_name[0]=ChartSymbol(chart_id_array[0]);
   //
   Print(" [",chart_pare_name[0],"] の ChartID は「",chart_id_array[0],"」です");
   //
   //
   for(int i=1;i<10;i++)   {
      //
      chart_id_array[i]=ChartNext(chart_id_array[i-1]);
      //
      if(chart_id_array[i]==-1)
      {
         End=i;
         break;
       }
      //
      chart_pare_name[i]=ChartSymbol(chart_id_array[i]);
      //
      Print(" [",chart_pare_name[i],"] の ChartID は「",chart_id_array[i],"」です");
      //
    }
   //
   // ＜３．特定の ChartSymbol に EventChartCustom を送信する＞
   //  データの準備
   //  あまり面白くないけど、「Open 値」を送ってみる
   double price_current=Open[0];
   //
   for(int j=0;j<End;j++)
   {
      if(chart_pare_name[j]==target)
      {
         Print(chart_pare_name[j],"を見つけた！");
         //

EventChartCustom(chart_id_array[j],(ushort)(custom_event_id+1),(long)_Period,price_current,
_Symbol+"から発信：");
         //
         Print(_Symbol,"：カスタム・イベント発信しました");
         //
         PlaySound("tick");
         //
         break;
      }
         Print(_Symbol,"：まだ、チャートが見つからない");
   }
   //
   //--- return value of prev_calculated for next call
   return(rates_total);
  }
//+------------------------------------------------------------------+
```

（3）受信側コード（ＥＡ）

※本稿では [EURJPYFXF,Daily] チャート上に設定します。

送信側で「string target="EURJPYFXF";」を書換えれば、自由に設定できる。

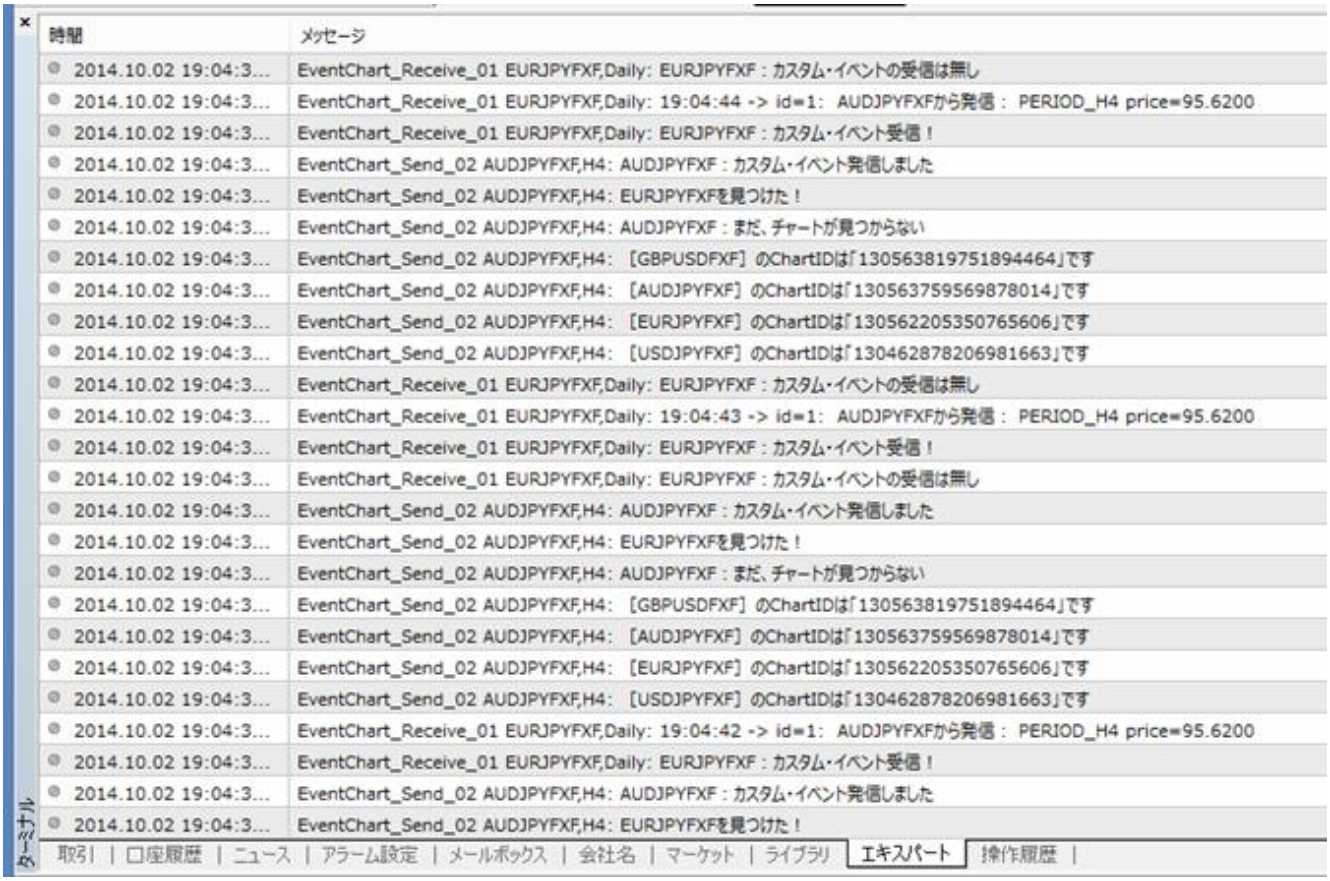
また動作確認用に、余分な「Print()と PlaySound()」を入れてあります。

```
//+------------------------------------------------------------------+
//|                                        EventChart_Receive_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 initialization function                                   |
//+------------------------------------------------------------------+
int OnInit()
  {
   //---

   //---
   return(INIT_SUCCEEDED);
  }
//+------------------------------------------------------------------+
//| Expert deinitialization function                                 |
//+------------------------------------------------------------------+
void OnDeinit(const int reason)
  {
   //---

  }
//+------------------------------------------------------------------+
//| Expert tick function                                             |
//+------------------------------------------------------------------+
void OnTick()
  {
   //---
   // ここには通常動かすＥＡの本体を書く
  }
//+------------------------------------------------------------------+
//| ChartEvent function                                              |
//+------------------------------------------------------------------+
void OnChartEvent(const int id,
                  const long &lparam,
                  const double &dparam,
                  const string &sparam)
  {
   //---
   if(id>=CHARTEVENT_CUSTOM)
     {
      //
         Print(_Symbol,"：カスタム・イベント受信！");
         //
      Print(TimeToString(TimeCurrent(),TIME_SECONDS)," -> id=",
            id-CHARTEVENT_CUSTOM,":  ",sparam," ",
            EnumToString((ENUM_TIMEFRAMES)lparam)," price=",DoubleToString(dparam,4));
            /*
```

```
            ここに「他の為替ペア」チャートからの「データ（例えば価格）」を
            参照・利用する本体コードを記述する。
          */
          Sleep(1000);
          PlaySound("alert2");
          //
      }
    //
    Print(_Symbol,"：カスタム・イベントの受信は無し");
    //
  }
//+------------------------------------------------------------------+
```

（4）実行結果

※とりあえず、どのように動作しているかの確認用です。



時間	メッセージ
2014.10.02 19:04:3...	EventChart_Receive_01 EURJPYFXF,Daily: EURJPYFXF：カスタム・イベントの受信は無し
2014.10.02 19:04:3...	EventChart_Receive_01 EURJPYFXF,Daily: 19:04:44 -> id=1: AUDJPYFXFから発信： PERIOD_H4 price=95.6200
2014.10.02 19:04:3...	EventChart_Receive_01 EURJPYFXF,Daily: EURJPYFXF：カスタム・イベント受信！
2014.10.02 19:04:3...	EventChart_Send_02 AUDJPYFXF,H4: AUDJPYFXF：カスタム・イベント発信しました
2014.10.02 19:04:3...	EventChart_Send_02 AUDJPYFXF,H4: EURJPYFXFを見つけた！
2014.10.02 19:04:3...	EventChart_Send_02 AUDJPYFXF,H4: AUDJPYFXF：まだ、チャートが見つからない
2014.10.02 19:04:3...	EventChart_Send_02 AUDJPYFXF,H4: [GBPUSDFXF] のChartIDは「130563819751894464」です
2014.10.02 19:04:3...	EventChart_Send_02 AUDJPYFXF,H4: [AUDJPYFXF] のChartIDは「130563759569878014」です
2014.10.02 19:04:3...	EventChart_Send_02 AUDJPYFXF,H4: [EURJPYFXF] のChartIDは「130562205350765606」です
2014.10.02 19:04:3...	EventChart_Send_02 AUDJPYFXF,H4: [USDJPYFXF] のChartIDは「130462878206981663」です
2014.10.02 19:04:3...	EventChart_Receive_01 EURJPYFXF,Daily: EURJPYFXF：カスタム・イベントの受信は無し
2014.10.02 19:04:3...	EventChart_Receive_01 EURJPYFXF,Daily: 19:04:43 -> id=1: AUDJPYFXFから発信： PERIOD_H4 price=95.6200
2014.10.02 19:04:3...	EventChart_Receive_01 EURJPYFXF,Daily: EURJPYFXF：カスタム・イベント受信！
2014.10.02 19:04:3...	EventChart_Receive_01 EURJPYFXF,Daily: EURJPYFXF：カスタム・イベントの受信は無し
2014.10.02 19:04:3...	EventChart_Send_02 AUDJPYFXF,H4: AUDJPYFXF：カスタム・イベント発信しました
2014.10.02 19:04:3...	EventChart_Send_02 AUDJPYFXF,H4: EURJPYFXFを見つけた！
2014.10.02 19:04:3...	EventChart_Send_02 AUDJPYFXF,H4: AUDJPYFXF：まだ、チャートが見つからない
2014.10.02 19:04:3...	EventChart_Send_02 AUDJPYFXF,H4: [GBPUSDFXF] のChartIDは「130563819751894464」です
2014.10.02 19:04:3...	EventChart_Send_02 AUDJPYFXF,H4: [AUDJPYFXF] のChartIDは「130563759569878014」です
2014.10.02 19:04:3...	EventChart_Send_02 AUDJPYFXF,H4: [EURJPYFXF] のChartIDは「130562205350765606」です
2014.10.02 19:04:3...	EventChart_Send_02 AUDJPYFXF,H4: [USDJPYFXF] のChartIDは「130462878206981663」です
2014.10.02 19:04:3...	EventChart_Receive_01 EURJPYFXF,Daily: 19:04:42 -> id=1: AUDJPYFXFから発信： PERIOD_H4 price=95.6200
2014.10.02 19:04:3...	EventChart_Receive_01 EURJPYFXF,Daily: EURJPYFXF：カスタム・イベント受信！
2014.10.02 19:04:3...	EventChart_Send_02 AUDJPYFXF,H4: AUDJPYFXF：カスタム・イベント発信しました
2014.10.02 19:04:3...	EventChart_Send_02 AUDJPYFXF,H4: EURJPYFXFを見つけた！

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 |

※多少、PlaySound()音がうるさいことはご容赦！

◎諸兄にて、何か面白い使い方を考えて、発表してください！

以　上