

○ 「 ExpertAdvisorWizard ; 自分専用テンプレートを作るには 」

・アメンボです、

アメンボのみならず、諸兄も「MQL4 コード」を作成する際に、「ExpertAdvisorWizard」を使う事が多いと推測しますが、生成されたコード中に余計な「記述」が沢山出てきて、「ウザッタイ」と思ったことはありませんか？

アメンボは生成されたコードから、いらぬものを一々削除しながら使っていました。漠然と、何とかならないものかと感じていたのですが、たまたま英語版サイトで、ヒントを見つけたので少し調べることにしました。

・本稿では、「ExpertAdvisorWizard」のテンプレートについて、確認できた内容を報告いたします。ただ、うまく動作しない内容などアメンボの調査範囲では「未解決問題」が多々有りますので、興味のある諸兄は是非とも取り組んでみてはいかがでしょうか。

・メタトレーダー&エディターは「version 4.00 BUILD 451」での確認内容です。

<同時掲載資料>・・・ダウンロード用

・アメンボが動作確認に使用したテンプレートの一つ； 「 Indicator_tag1_amenbo.mqt 」
「experts¥templates」フォルダー中にコピーして使います

目次： 1. 再確認

(1) 本稿で言うウィザードとは	・・・	2 頁
(2) ウィザードのテンプレート・ファイルを確認する	・・・	2 頁
(3) テンプレート・ファイル (MQT) と生成ファイル (MQ4) の対比		2 頁
2. 調査1 ; チェック用テンプレートを試してみる		
(1) チェック用テンプレート「Indicator_tag1_amenbo.mqt」の内容		3 頁
(2) ウィザードからテンプレートを開いていく	・・・	4 頁
(3) 生成された「MQ4」ファイルを確認します	・・・	7 頁
3. 調査2 ; ウィザードに表示される「ファイル名」とタグ記述		
(1) 下記のMQT ファイルを作成した・・・スクリプトとEAを意識		9 頁
(2) ウィザードから見えるファイル名を確認	・・・	9 頁
4. 判明事項 (「MQT」と「MQ4」ファイルの対比より)		
(1) 基本	・・・	9 頁
(2) 各種パラメータ・タグと生成コード (及びウィザード項目)		10 頁
(3) 展開記号と内容	・・・	11 頁
(4) ウィザードとファイルの関係	・・・	13 頁
(5) 「MACD.mqt」内容を参考に記載しておきます	・・・	14 頁

1. 再確認

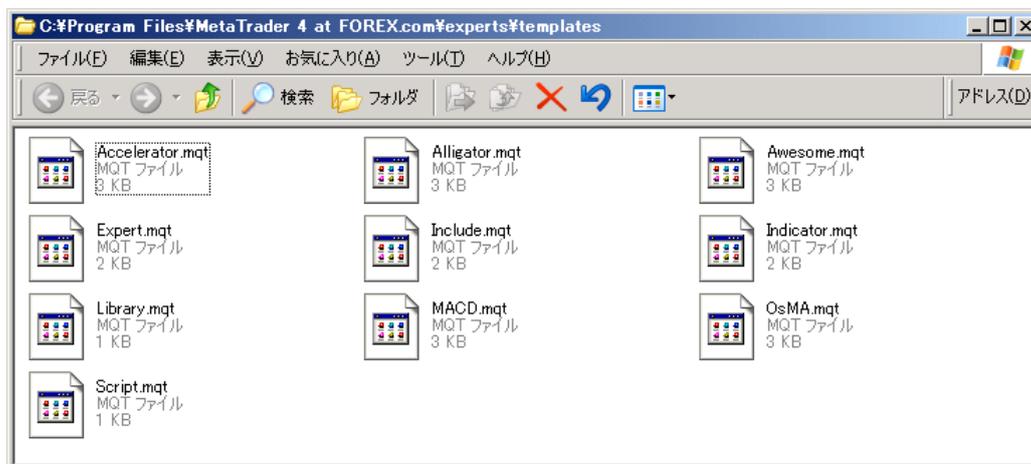
(1) 本稿で言うウィザードとは

- ・「ウィザード」と言うと、一瞬「なんだったっけ」と思うかもしれませんが、要はこれ！です。



(2) ウィザード (ExpertAdvisorWizard) のテンプレート・ファイル (MQT ファイル) を確認する

- ・ウィザードで、「Expert Advisor」などと指定 (チェックを入れる) と、対応するテンプレート・ファイルが選択されて「MQL4 コードの雛形」が作成されるのは、了解済みとします。
- ・先ず、テンプレート・ファイルの在り場所の「experts\templates」フォルダ内を観てみます。
<FOREX 使用の場合>・・・10 ファイルありました (ブローカーにより異なるかも！)



(3) 次に「テンプレート・ファイル (MQT)」とウィザードで生成される「MQL4 ファイル (MQ4)」を対比させてみます

- － 1. 「Script.mqt」を「テキスト・エディタ」で開くと、

```
<expert>
type=SCRIPT_ADVISOR
</expert>
#header#
#property copyright "#copyright#"
#property link      "#link#"

#extern_variables#
//+-----+
//| script program start function |
```

```
//+-----+
int start()
{
//----

//----
    return(0);
}
//+-----+
```

※見慣れた記述と、見慣れない「<expert>・・・</expert>」「#・・・#」と言う表現があります
(他のテンプレート・ファイルも確認すると色々判ってきます)

- ー 2. ウィザード (ExpertAdvisorWizard) で「Script」を選択すると、おなじみの以下のコードが生成されます。

```
//+-----+
//|                                     Script_Tag_amenbo.mq4 |
//|                                     amenbo |
//|                                     泉の森の弃財天池 |
//+-----+
#property copyright "amenbo"
#property link      "泉の森の弃財天池"

//+-----+
//| script program start function |
//+-----+
int start()
{
//----

//----
    return(0);
}
//+-----+
```

※一定の文法に従って「テンプレート」⇒「MQL4 コード」へと変換されているように見えますが！

2. 調査1 ; チェック用テンプレートを試してみる

- (1) チェック用テンプレート「Indicator_tag1_amenbo.mqt」の内容 (「MQT」ファイル)

```
//-----
<expert>

type=INDICATOR_ADVISOR
description=Check Tag
separate_window=1
used_buffers=2
ind_minimum=200
ind_maximum=5000

<param>
type=int
name=FastEMA
```

```

value=12
</param>

<param>
type=int
name=SlowEMA
value=26
</param>

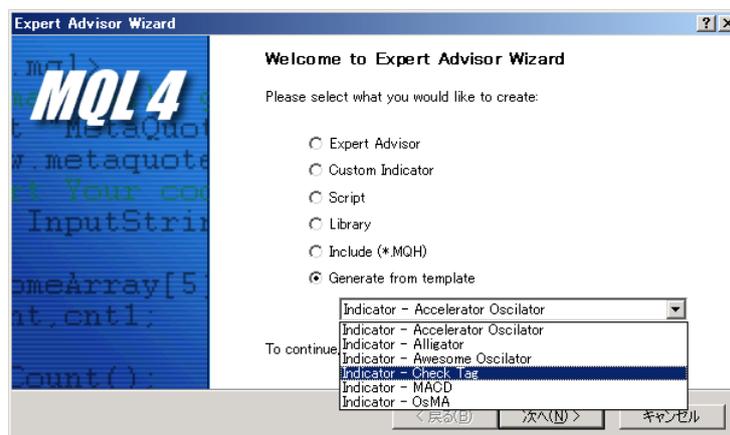
<param>
type=int
name=SignalSMA
value=9
</param>

</expert>
////-----
//// header
#header#
//// copyright
#copyright#
//// link
#link#
//// indicator_properties
#indicator_properties#
//// extern_variables
#extern_variables#
//// mapping_buffers
#mapping_buffers#
//// buffers_used
#buffers_used#
//// indicators_init
#indicators_init#
////-----

```

(2) ウィザードからテンプレートを開いていく

- － 1. 「Generate from template」 にチェックを入れてテンプレートを指定します

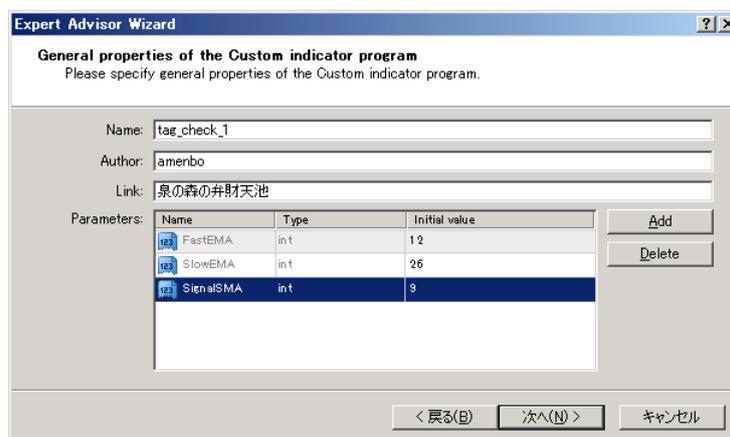


※先に手の内を示すと、実はテンプレート中に

- ・「type=INDICATOR_ADVISOR」と、
- ・「description=Check Tag」の記述があるため、

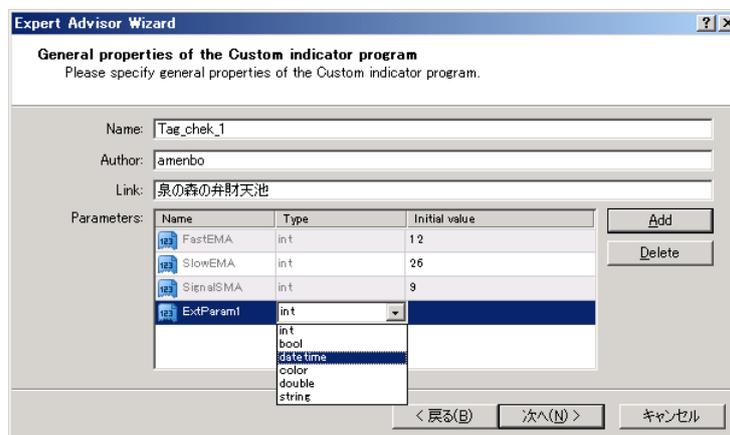
「Indicator-Check Tag」と表示されます。

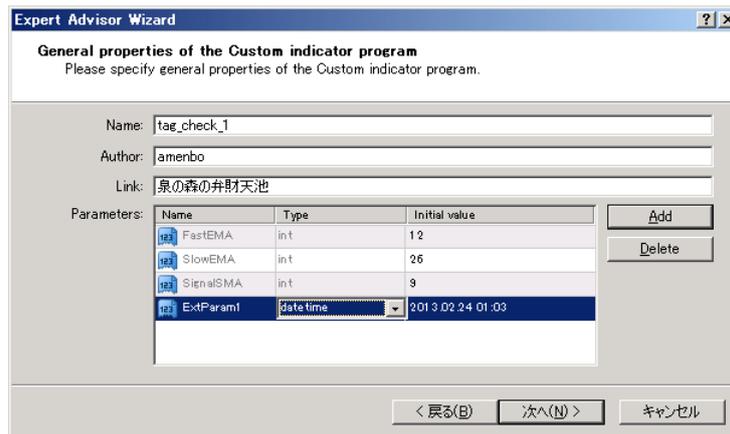
- ・もし、「//description=Check Tag」とコメントアウトすると、「Indicator-Indicator_tag1_amenbo」と、ファイル名が表示されます。(試してみてください)
- － 2. 「Indicator-Check Tag」を選択して操作(いつも通り)を継続する



※ 3つの「<param> </param>」タグで挟んだ「extern」パラメータに対応した変数が、3つ作成されてウィザードの「Parameters」に表示されます。

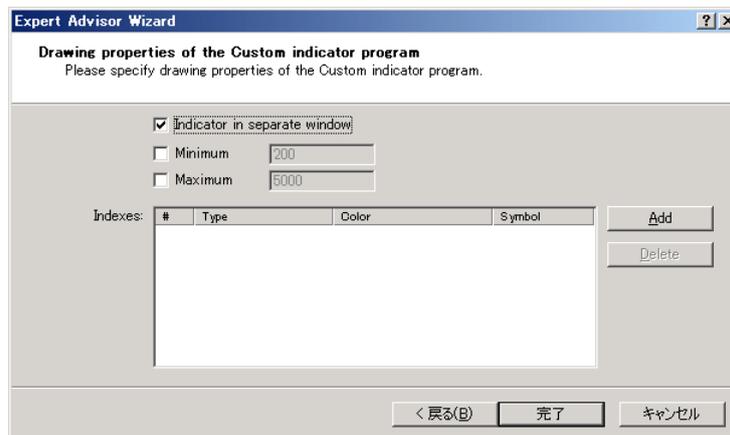
- － 3. [Add] を選択して、変数を1つ増やしてみます
- ・ [Type]をダブルクリックし、「datetime」を選択してみます



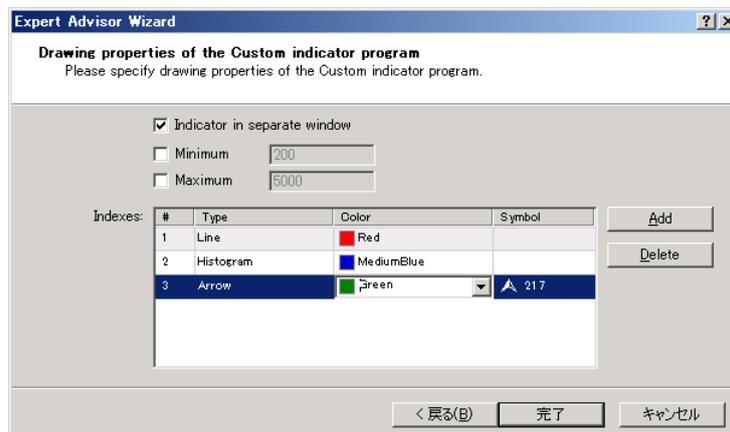


※ 「Name」「Type」「Initial value」はダブルクリックして、値を変更・編集することができます
 ・[次へ>] を選択します

－ 4. [次へ>] を選択して、「Drawing properties」の設定ダイアログに進みます



・ [Add] を「3回」選択して、「Indexe」を設定してみます、
 「Type」「Color」はダブルクリックして、設定を変えることができます。



－ 5. [完了] を選択してウィザードを終了します

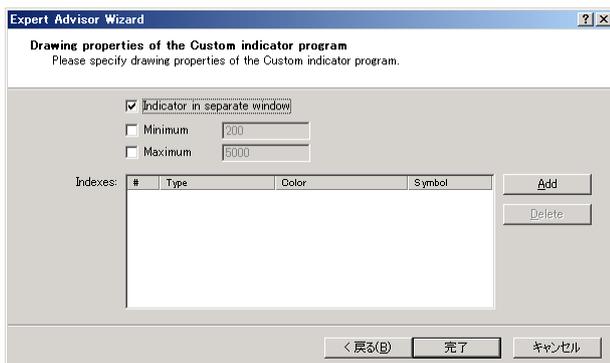
(3) 生成された「MQ4」ファイルを確認します

```

////-----
//// header
//+-----+
//|                                     tag_check_1.mq4 |
//|                                     amenbo       |
//|                                     泉の森の弃財天池 |
//+-----+
//// copyright
amenbo
//// link
泉の森の弃財天池
//// indicator_properties
#property indicator_separate_window
#property indicator_buffers 3
#property indicator_color1 Red
#property indicator_color2 MediumBlue
#property indicator_color3 Green
//// extern_variables
//--- input parameters
extern int      FastEMA=12;
extern int      SlowEMA=26;
extern int      SignalSMA=9;
extern datetime ExtParam1=D'2013.02.24 01:03';
//// mapping_buffers
//--- buffers
double ExtMapBuffer1[];
double ExtMapBuffer2[];
double ExtMapBuffer3[];
//// buffers_used
//--- additional buffers are used for counting
IndicatorBuffers(5);
//// indicators_init
SetIndexStyle(0, DRAW_LINE);
SetIndexBuffer(0, ExtMapBuffer1);
SetIndexStyle(1, DRAW_HISTOGRAM);
SetIndexBuffer(1, ExtMapBuffer2);
SetIndexStyle(2, DRAW_ARROW);
SetIndexArrow(2, 217);
SetIndexBuffer(2, ExtMapBuffer3);
SetIndexEmptyValue(2, 0.0);
////-----

```

(4) 「(3)」コードとの比較のために、「Drawing properties」に手をつけず、そのまま [完了] とした場合に生成される「MQ4」ファイルも確認します。



つまり、この画面で [Add] を実行せずにそのまま [完了] とする場合は、

```

////-----
//// header
//+-----+
//|                                     Tag_check.mq4 |
//|                                     amenbo      |
//|                                     泉の森の弃財天池 |
//+-----+
//// copyright
amenbo
//// link
泉の森の弃財天池
//// indicator_properties
#property indicator_separate_window
//// extern_variables
/-- input parameters
extern int      FastEMA=12;
extern int      SlowEMA=26;
extern int      SignalSMA=9;
//// mapping_buffers
//// buffers_used
/-- additional buffers are used for counting
    IndicatorBuffers(2);
//// indicators_init
////-----

```

※下記の「赤書き」内容が手作業の [Add] で生成される事が判ります。

```

//// indicator_properties
#property indicator_separate_window
#property indicator_buffers 3
#property indicator_color1 Red
#property indicator_color2 MediumBlue
#property indicator_color3 Green
//// extern_variables
/-- input parameters
extern int      FastEMA=12;
extern int      SlowEMA=26;
extern int      SignalSMA=9;
extern datetime ExtParam1=D'2013.02.24 01:03'; ←◆これは手入力したもの
//// mapping_buffers
/-- buffers
double ExtMapBuffer1[];
double ExtMapBuffer2[];
double ExtMapBuffer3[];
//// buffers_used
/-- additional buffers are used for counting
    IndicatorBuffers(5); ←◆ここは「バッファ数」が異なる
//// indicators_init
    SetIndexStyle(0, DRAW_LINE);
    SetIndexBuffer(0, ExtMapBuffer1);
    SetIndexStyle(1, DRAW_HISTOGRAM);
    SetIndexBuffer(1, ExtMapBuffer2);
    SetIndexStyle(2, DRAW_ARROW);
    SetIndexArrow(2, 217);
    SetIndexBuffer(2, ExtMapBuffer3);
    SetIndexEmptyValue(2, 0.0);
////-----

```

3. 調査2；ウィザードに表示される「ファイル名」とタグ記述

(1) 下記のMQTファイルを作成した・・・スクリプトとEAを意識

- ①ファイル名；[Script_amenbo.mqt]；

```
<expert>
type=SCRIPT_ADVISOR
description=Script Check
</expert>
```

- ②ファイル名；[Expert_amenbo.mqt]；

```
<expert>
type=EXPERT_ADVISOR
description=Expert Check
</expert>
```

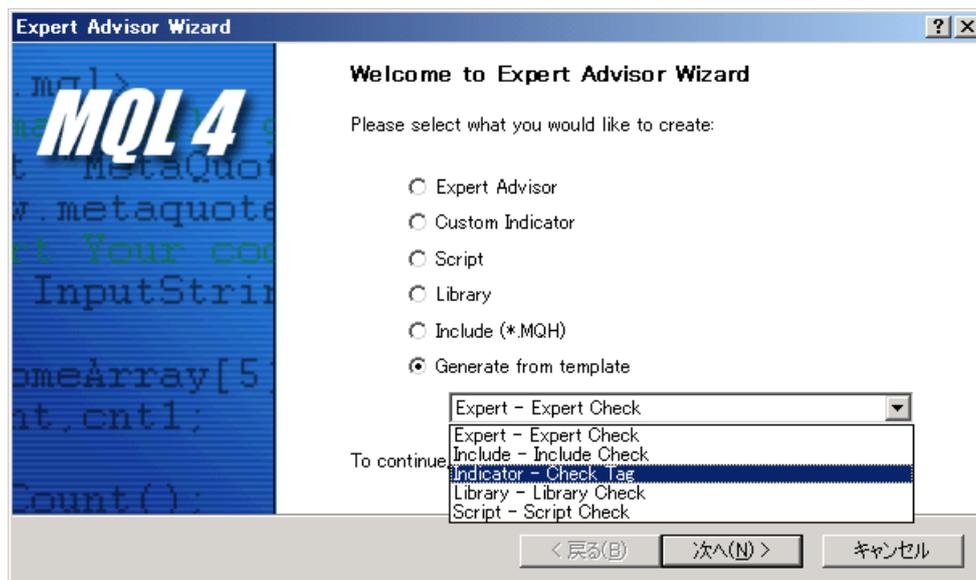
- ③ファイル名；[Include_amenbo.mqt]；

```
<expert>
type=INCLUDE_ADVISOR
description=Include Check
</expert>
```

- ④ファイル名；[Library_amenbo.mqt]；

```
<expert>
type=LIBRARY_ADVISOR
description=Library Check
</expert>
```

(2) ウィザードから見えるファイル名を確認



※ 「Wxpert-Expert Check」、「Script-Script Check」等に注目

4. 判明事項（「MQT」と「MQ4」ファイルの対比より）

(1) 基本

- 1. パラメータの設定値；
 <expert>と</expert>タグで挟んだ間に記述
- 2. ウィザードやパラメータで設定された内容の展開；

- ① 「#・・・#」の様に、「#」で挟まれた場所に展開する
- ② 展開項目は「#」と「#」に記述する

－ 3. イメージ；

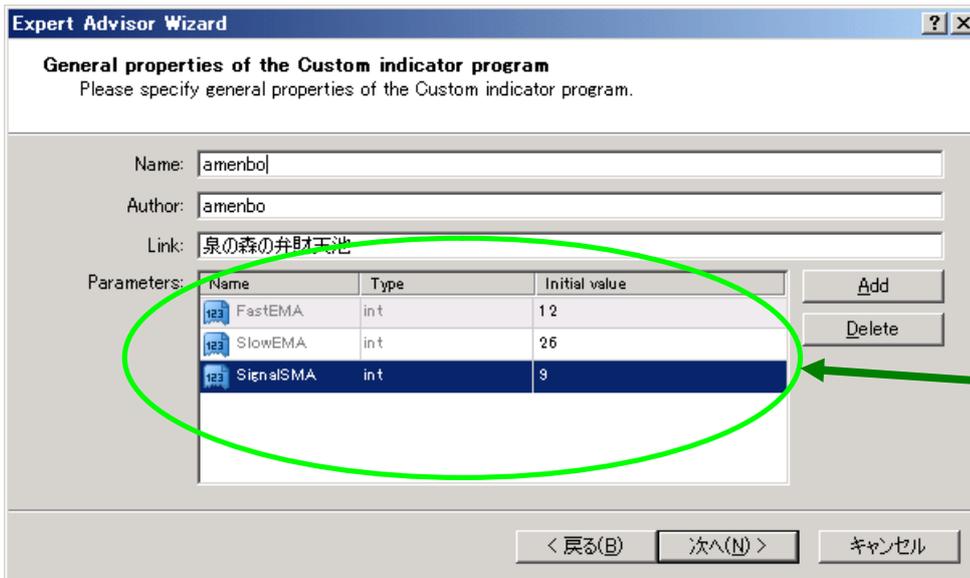
```

<exprt>
 ・・・タイプやバッファ数などを記述・・・
<param>
 ・・・パラメータ値の設定・・・
</param>
</exprt>
.....
#展開項目#
.....
    
```

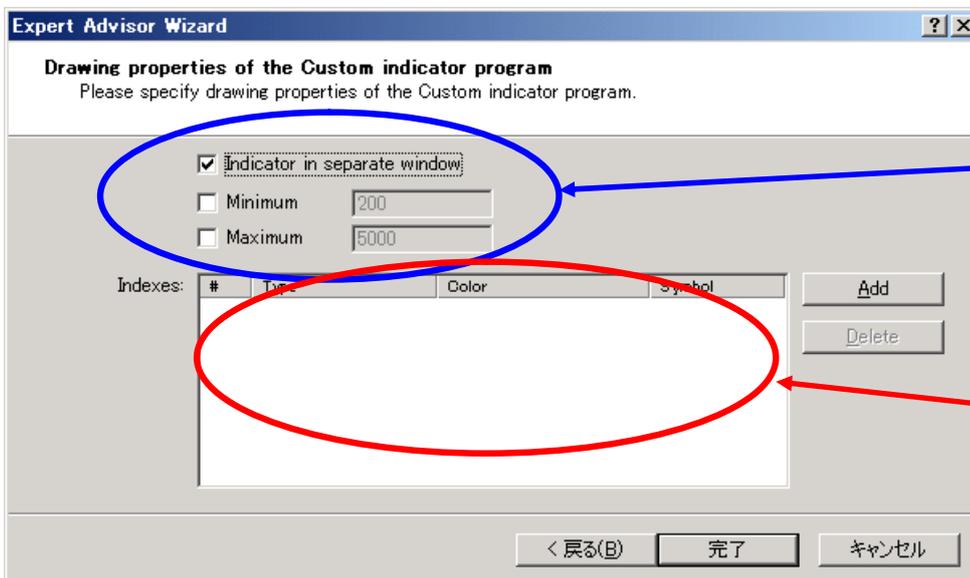
(2) 各種パラメータ・タグと生成コード (及びウィザード項目)

<exprt>		テンプレート・タイプ	使用場面・特記
type=	EXPERT_ADVISOR	EA テンプレート	
	SCRIPT_ADVISOR	スクリプト・テンプレート	
	INDICATOR_ADVISOR	インディケータ・テンプレート	
	LIBRARY_ADVISOR	ライブラリ・テンプレート	
	INCLUDE_ADVISOR	インクルード・テンプレート	
description=	任意名称 (英数字)	ウィザード表示「ファイル名称」	
separate_window=	整数	サブウィンドウ数	インディケータ
used_buffers=	整数	インディケータ・バッファ数	インディケータ
ind_minimum=	整数	インディケータ表示の最小値	インディケータ
ind_maximum=	整数	インディケータ表示の最大値	インディケータ
<param>			
type=	int, double, bool datetime, color, string	extern 入力パラメータ設定	
name=	任意名称 (英数字)	extern 入力パラメータ設定	
value=	type 対応データ値	extern 入力パラメータ設定	
</param>			
<ind>			
type=	Line, Section, Histogram, arrow	インディケータのタイプ	
color=	カラー指定 例 ; Red, Silver など	インディケータの色	
</ind>			
</exprt>			

- ※ 「<ind> </ind>」タグの機能が、どうしても確認できませんでした！ (なぜか機能せず)、何が問題なのか、現状でも不明のままです。(アメンボは一時保留状態、諸兄にて判ったら教えて！)
- ※ 「experts¥templates」中の「MACD.mqt」には「<ind> </ind>」が記述されているのですが！！
- ※ 「<param> </param>」と「<ind> </ind>」のウィザード上での対応は、下記の様になるはずなのですが、「<param> </param>」しか確認できず！、でした。



<param>...</param>
 で指定した extern 変数の
 テンプレートが設定される

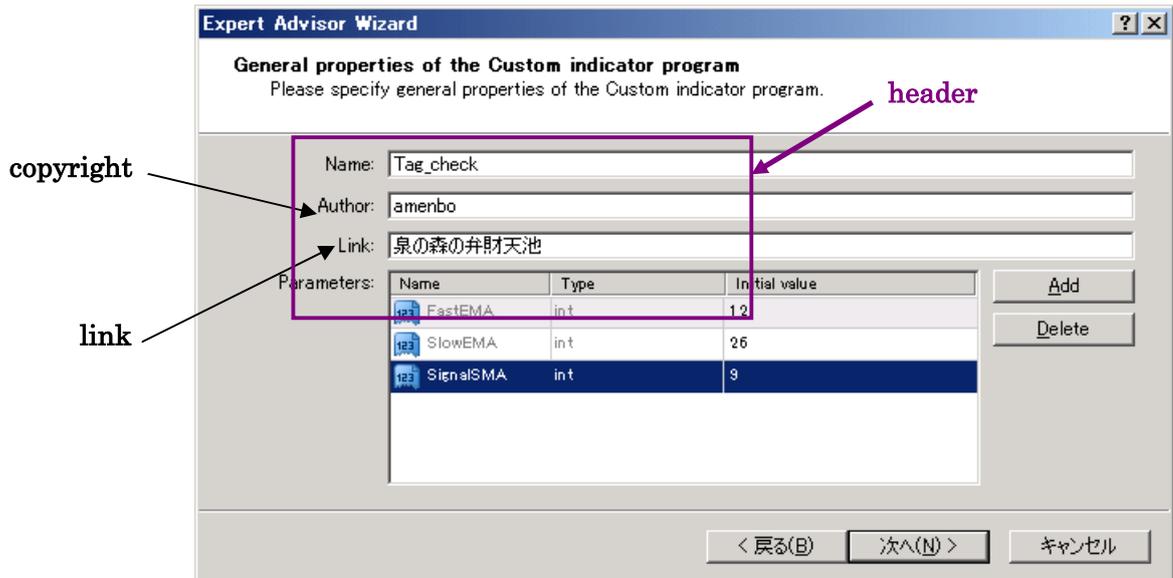


「separate_window=」
 「ind_minimum=」
 「ind_maximum=」で指定した
 テンプレートが設定される

<ind>...</ind>
 で指定したテンプレートが設
 定されるはずなんだけど！
 うまくいかない！のは何故？

(3) 展開記号と内容

#header#	下図と①を参照
#copyright#	下図と②を参照
#link#	下図と③を参照
#indicator_properties#	④を参照
#extern_variables#	⑤を参照
#mapping_buffers#	⑥を参照
#buffers_used#	⑦を参照
#indicators_init#	⑧を参照



- ① [#header#] の展開内容例 ;

```
//+-----+
//|                                     Script_Tag_amenbo.mq4 |
//|                                     amenbo             |
//|                                     泉の森の弁財天池     |
//+-----+
```

- ② [#copyright#] の展開内容例 ;

```
#property copyright "amenbo"
```

- ③ [#link#] の展開内容例 ;

```
#property link      "泉の森の弁財天池"
```

- ④ [#indicator_properties#] の展開内容例 ;

```
#property indicator_separate_window
#property indicator_buffers 3
#property indicator_color1 Red
#property indicator_color2 MediumBlue
#property indicator_color3 Green
```

- ⑤ [#extern_variables#] の展開内容例 ;

```
extern int      FastEMA=12;
extern int      SlowEMA=26;
extern int      SignalSMA=9;
extern datetime ExtParam1=D'2013.02.24 01:03';
```

- ⑥ [#mapping_buffers#] の展開内容例 ;

```
double ExtMapBuffer1[];
double ExtMapBuffer2[];
double ExtMapBuffer3[];
```

- ⑦ [#buffers_used#] の展開内容例 ;

```
IndicatorBuffers(5);
```

⑧ [#indicators_init#] の展開内容例 ;

```
SetIndexStyle(0, DRAW_LINE);
SetIndexBuffer(0, ExtMapBuffer1);
SetIndexStyle(1, DRAW_HISTOGRAM);
SetIndexBuffer(1, ExtMapBuffer2);
SetIndexStyle(2, DRAW_ARROW);
SetIndexArrow(2, 217);
SetIndexBuffer(2, ExtMapBuffer3);
SetIndexEmptyValue(2, 0.0);
```

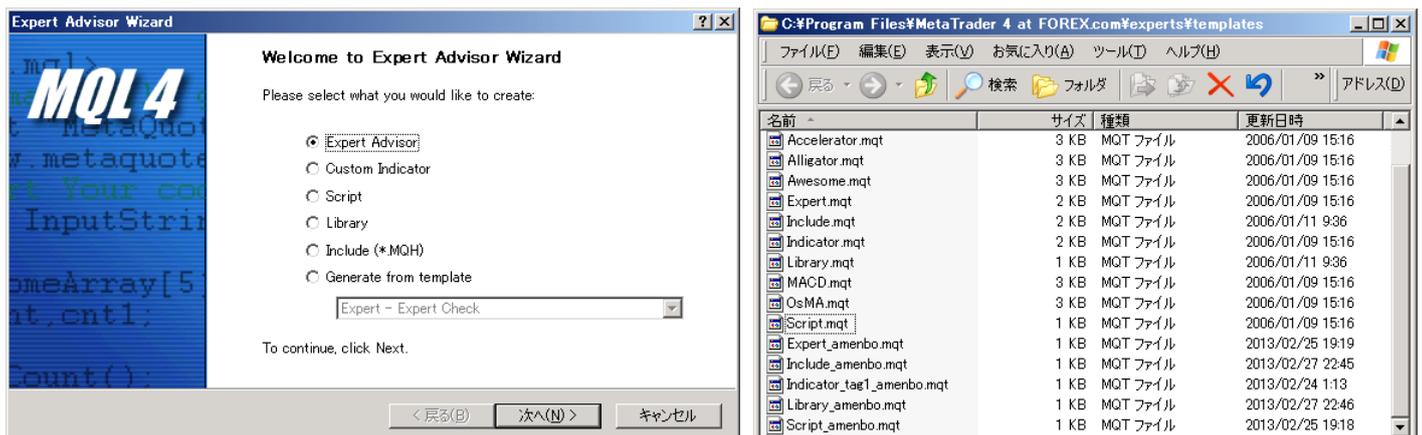
※「赤書き」部は、

- ・テンプレートの「タグ」は機能しなかったが、
- ・ウィザードで指定すれば生成されている。

(4) ウィザードとファイルの関係

・[ウィザード] のチェックに対応しているデフォルトのファイル名

ウィザード上でのチェック	デフォルト・ファイル名
Expert Advisor	Expert.mqt
Custom Indicator	Indicator.mqt
Script	Script.mqt
Library	Library.mqt
Include (*.MQH)	Include.mqt
Generate from tenplate	上記以外のファイル



(5) 「MACD.mqt」内容を参考に記載しておきます

- ・アメンボのメタ・エディタでは、「赤書き」部タグがテンプレートとして機能しません。
? (アメンボのメタ・エディタがおかしいのか?)

```

<expert>
type=INDICATOR_ADVISOR
separate_window=1
used_buffers=2
<param>
type=int
name=FastEMA
value=12
</param>
<param>
type=int
name=SlowEMA
value=26
</param>
<param>
type=int
name=SignalSMA
value=9
</param>
<ind>
color=Silver
type=DRAW_HISTOGRAM
</ind>
<ind>
color=Red
</ind>
</expert>
#header#
#property copyright "#copyright#"
#property link      "#link#"

#indicator_properties#
#extern_variables#
#mapping_buffers#
//---- indicator buffers
double ExtSilverBuffer[];
double ExtRedBuffer[];

//+-----+
//| Custom indicator initialization function |
//+-----+
int init()
{
    #buffers_used#;
//---- drawing settings
    #indicators_init#
//----
    SetIndexDrawBegin(1, SignalSMA);
    IndicatorDigits(5);
//---- indicator buffers mapping
    SetIndexBuffer(0, ExtSilverBuffer);
    SetIndexBuffer(1, ExtRedBuffer);

```

```

//---- name for DataWindow and indicator subwindow label
IndicatorShortName("MACD("+FastEMA+", "+SlowEMA+", "+SignalSMA+")");
//---- initialization done
return(0);
}
//+-----+
//| Moving Averages Convergence/Divergence |
//+-----+
int start()
{
int limit;
int counted_bars=IndicatorCounted();
//---- check for possible errors
if(counted_bars<0) return(-1);
//---- last counted bar will be recounted
if(counted_bars>0) counted_bars--;
limit=Bars-counted_bars;
//---- macd counted in the 1-st buffer
for(int i=0; i<limit; i++)

ExtSilverBuffer[i]=iMA(NULL, 0, FastEMA, 0, MODE_EMA, PRICE_CLOSE, i)-iMA(NULL, 0, SlowEM
A, 0, MODE_EMA, PRICE_CLOSE, i);
//---- signal line counted in the 2-nd buffer
for(i=0; i<limit; i++)
ExtRedBuffer[i]=iMAOnArray(ExtSilverBuffer, Bars, SignalSMA, 0, MODE_SMA, i);
//---- done
return(0);
}
//+-----+

```

以 上