و – معالجة ما قبل التنفيذ

(تحويل القيم)

تحويل القيم هو نظام فرعى داخل لغة MQL4 والتى تهدف الى تهيئة وتحضير واعداد الاكواد البرمجية قبل تنفيذ البرنامج

يتيح لنا تحويل القيم مراجعة سريعة وادخال تعديلات على الاكواد ، فامكانية تحديد أسماء محدده لبعض الثوابت يساعد فى سهولة قراءة وفهم الكود البرمجى

اذا تم استخدام الرمز # فى السطر الأول من البرنامج ، يعنى هذا ان هذا السطر خاص بتحويل القيم

وسنتناول الأتى :

* إعلان الثوابت
* جمع المتحكمات
* تضمين الملفات
* استيراد الدوال

© 2000-2012, MetaQuotes Software Corp.

1 إعلان الثوابت

بإستخدام الرمز البرمجى # يمكنك تعريف الاسم أو الرمز الذى يشير الى سلسلتة الخاصه ومن ثم يتم استبدال هذه القيم فى مرحلة تحويل القيم أو ما نسميها بمرحلة معالجة ما قبل التنفيذ

ويمكن استبدال هذا الاسم بأى رمز أو اسم اخر وليس فقط أرقام

لاحظ :

```
#define identifier value
```

تأخذ القيمة اى شكل واى نوع طالما تتفق مع نفس قواعد المتغيرات

مثال :

```
#define ABC          100
#define PI           0.314
#define COMPANY_NAME "MetaQuotes Software Corp."

...

void ShowCopyright()
  {
   Print("Copyright © 2001-2007, ",COMPANY_NAME);
   Print("http://www.metaquotes.net");
  }
```

© 2000-2012, MetaQuotes Software Corp.

2 جمع المتحكمات

كل برنامج تمت برمجتة بطريقة لغة MQL4 يسمح لك باضافة بعض الميزات والخيارات الخاصة

#property

والتى من شأنها مساعدة العميل على تقديم خدمة مناسبة لـه حـسب احتياجاتـة الخاصة

ويتبادر اولاً اننا نشير للمؤشرات بالمقام الأول

لاحظ :

```
#property identifier value
```

والجدول التالى يوضح الثوابت وانواعها والوصف الخاص بها

Constant	Type	Description
link	string	a link to the company website
copyright	string	the company name
stacksize	int	stack size
library		a library; no start function is assigned, non-referenced functions are not removed
indicator_chart_window	void	show the indicator in the chart window
indicator_separate_window	void	show the indicator in a separate window
indicator_buffers	int	the number of buffers for calculation, up to 8
indicator_minimum	double	the bottom scaling limit for a separate indicator window

© 2000-2012, MetaQuotes Software Corp.

تابع : جدول الثوابت وتعريفها

Constant	Type	Description
indicator_maximum	double	the top scaling limit for a separate indicator window
indicator_colorN	color	the color for displaying line N, where N lies between 1 and 8
indicator_widthN	int	width of the line N, where N lies between 1 and 8
indicator_styleN	int	style of the line N, where N lies between 1 and 8
indicator_levelN	double	predefined level N for separate window custom indicator, where N lies between 1 and 8
indicator_levelcolor	color	level line color
indicator_levelwidth	int	level line width
indicator_levelstyle	int	level line style
show_confirm	void	before script run message box with confirmation appears
show_inputs	void	before script run its property sheet appears; disables show_confirm property

أمثلة :

FXArabia

```
#property link      "http://www.metaquotes.net"
#property copyright "MetaQuotes Software Corp."
#property library
#property stacksize 1024
```

© 2000-2012, MetaQuotes Software Corp.

3 تضمين الملفات

سطر الأوامر التالى :

#include

يمكن وضعه فى اى مكان بالبرنامج ، ولكن عادة ما يتم وضعه فى البداية الكود البرمجى

لاحظ :

```
#include <file_name>
#include "file_name";
```

أمثلة :

```
#include <WinUser32.mqh>
#include "mylib.mqh"
```

يقوم محول القيم (معالجة ما قبل التنفيذ) باستبدال هذا السطر مع مضمون ومحتوى الملف التالى

WinUser32.mqh

والذى غالباً ما يوضع فى المسار التالى

terminal_directory\experts\include

© 2000-2012, MetaQuotes Software Corp.

4 تضمين الملفات

يتم استيراد الدوال من ملفات :

(ex4 files.)*

ويتم تحديده بالضبط قبل الاستيراد :

#import

لكى يتم تمرير المعلومات والاكواد بطريقة سليمة ومنع حدوث تضارب فى البيانات التى يتم استدعائها فى وقت واحد

لاحظ :

```
#import "file_name"
   func1 define;
   func2 define;
   ...
   funcN define;
#import
```

الدوال المستدعاه يجب ان تكون فريدة ووحيدة فى أسمائها اى ان اسمائها لا تكون مكررة ولم تستخدم من قبل فى اى جزء من اجزاء البرنامج

لتجنب الاخطاء عند التشغيل ، لابد من الاتزام باستدعاء ملفات من نفس الصيغة كما اشرنا من قبل

© 2000-2012, MetaQuotes Software Corp.

أمثلة :

```
#import "user32.dll"
   int    MessageBoxA(int hWnd, string lpText, string lpCaption, int uType);

#import "stdlib.ex4"
   string ErrorDescription(int error_code);
   int    RGB(int red_value, int green_value, int blue_value);
   bool   CompareDoubles(double number1, double number2);
   string DoubleToStrMorePrecision(double number, int precision);
   string IntegerToHexString(int integer_number);

#import "ExpertSample.dll"
   int    GetIntValue(int);
   double GetDoubleValue(double);
   string GetStringValue(string);
   double GetArrayItemValue(double arr[], int, int);
   bool   SetArrayItemValue(double& arr[], int,int, double);
   double GetRatesItemValue(double rates[][6], int, int, int);
   int    SortStringArray(string& arr[], int);
   int    ProcessStringArray(string& arr[], int);
#import
```

© 2000-2012, MetaQuotes Software Corp.

الى هنا ينتهى الفصل الاول والذى تناولنا فيه الأساسيات الخاصة بلغة البرمجة

MQL4

* القواعد (البناء الهيكلى)
* أنواع البيانات
* العمليات والتعبيرات
* المتحكمات
* الدوال
*المتغيرات
* معالجة ما قبل التنفيذ (تحويل القيم)

فى الفصل القادم سنتناول بالتفصيل الثوابت القياسية فى هذة اللغة

يُتْبَعْ

© 2000-2012, MetaQuotes Software Corp.