

# 1 ชนิดข้อมูลและโอเปอเรเตอร์ต่างๆใน MQL4

MQL4 เองก็เหมือนภาษาคอมพิวเตอร์ทุกภาษาที่ต้องมีชนิดข้อมูลและโอเปอเรเตอร์ต่างๆ

## ชนิดตัวแปรต่างๆใน MQL4

- Integer (int)
- Boolean(bool)
- String(string)
- Floating point number (double)
- Color(color)
- Datatime(datetime)

## รายละเอียดตัวแปรแต่ละชนิด

**-Integer (int)** เป็นตัวแปรที่ใช้เก็บค่าจำนวนเต็มบวกหรือจำนวนเต็มลบ สามารถเก็บค่าได้ระหว่าง -2147483648 2147483848 MQL4 ค่า int ยังใช้เก็บเลขฐาน 8 16 อีกด้วยซึ่งเป็นหนึ่งในข้อที่แตกต่างจากภาษา c

ตัวอย่างการประกาศค่าและใช้งาน

```
int a = 0;
int b = -100;
int c = 1000;
int hex = 00ff00;
```

**-Boolean** เป็นชนิดข้อมูลที่เก็บได้แต่ค่าความจริง (true) หรือค่าความเท็จ (false) ได้เลข 1 แต่ความจริงเลข 0

ตัวอย่างและการประกาศค่าและใช้งาน

```
bool a = true;
bool a=1;
bool b= false ;
bool b=0;
```

**-String** เป็นชนิดข้อมูลที่ใช้เก็บสายอักขระหรือข้อความนั่นเอง

ตัวอย่างและการประกาศค่าและใช้งาน

```
string str1="Hello";
string str2 ="how are you ";
```

**-Floating-point number (double)** เป็นชนิดข้อมูลที่ใช้เก็บเลขทศนิยม

ตัวอย่างและการประกาศค่าและใช้งาน

```
double a = 0.1;
```

```
double b = 0.222;
```

```
double c = 3.14151617;
```

**-Color** เป็นชนิดข้อมูลพิเศษที่มีใน MT4 โดยใช้เก็บค่าสีที่เราต้องการให้แสดงผล โดยมีวิธีเก็บค่า 3 วิธีดังต่อไปนี้

- เก็บค่าสีโดยใช้ชื่อของสี เช่น

```
color a = Red;
```

```
color b = Black ;
```

- เก็บค่าสีโดยใช้ตำแหน่ง RGB หรือเรียกง่ายว่าค่าของแม่สี แต่ในภาษา MQL4 จะใช้สัญญาตัวย่อ

C'ค่าของ R ,ค่าของ G ,ค่าของ B ' ; เช่น

```
color a = C'128,128,128';
```

```
color b = C'100,128,222' ;
```

- เก็บค่าสีโดยใช้เลขฐาน 16 เช่น

```
color a = ffffff;
```

```
color b = ff00aa;
```

สามารถหา ตำแหน่งค่าสีได้จากเว็บนี้ครับ [http://en.wikipedia.org/wiki/Web\\_colors](http://en.wikipedia.org/wiki/Web_colors)

**-Datetime** เป็นชนิดข้อมูลชนิดพิเศษอีกชนิดหนึ่ง MQL4 สร้างขึ้น ชนิดข้อมูลชนิดนี้เราเอาไว้ใช้เก็บค่าของวันที่ครบโดยใช้ ตัวอักษร D' ข้อมูลวันเวลา ' ; เช่น

```
D'2009.01.01 00:00'
```

1

2009

00.00

```
D'1985.05.20 10.10:10'
```

20

1985

10

10

10

```
D'19.07.1980 12'
```

19

1980

12

0

0

```
D'02.02.2009'
```

2 กุมภาพันธ์ 2009

0

0

ตัวอย่างและการประกาศค่าและใช้งาน

```
datetime mybirthday = D'1983.03.20 12.00:00' ;
```

```
datetime now = D'2009.07.30 01.10:10' ;
```

## โอเปอเรเตอร์และนิพจน์

โอเปอเรเตอร์ ( Operator ) คือสัญลักษณ์ที่ใช้กระทำกับข้อมูลเช่น + - \* / เป็นต้น

นิพจน์ ( Expression ) คือ การนำข้อมูลมากระทำกันตั้งแต่ 1 จำนวนขึ้นไปหรืออาจจะเป็นหลายๆข้อมูลก็ได้ โดยข้อมูลนั้นอาจจะเป็น ค่าคงที่ ตัวแปร เมื่อนำข้อมูลมากระทำกันผลลัพธ์ที่ได้ก็จะขึ้นอยู่กับ โอเปอเรเตอร์ที่นำมาใช้

ตัวอย่าง

```
int a=1 ;
```

```
int b=2;
```

```
int c ;
```

c=a+b; ตัวอย่างของ Expression ที่ใช้เครื่องหมายบวก มากระทำกับข้อมูลชุด A

B ค่าที่จะถูกเก็บไว้ที่ C

Expression ในความหมายของผมมันก็คือสมการ ง่ายๆนี่เองแหละครับ ต่อไปนี้

Operator ต่างๆ ซึ่งผมรวบรวมเอามาแสดงเฉพาะ Operator EA ที่มีให้

โหลดในเน็ต ส่วนพวก Operator แปลกๆ ที่ยากจะทำความเข้าใจในระดับผู้หัดเขียน โปรแกรมมัน ผมยังไม่แสดงรายละเอียดและวิธีใช้ทั้งนี้เพื่อไม่ให้เกิดความรู้สึกที่วุ่นวายและสับสนสำหรับผู้ที่ยังเริ่มหัดเขียน โปรแกรมในช่วงแรกๆ

### โอเปอเรเตอร์ทางคณิตศาสตร์ ( Arithmetic Operator )

Opreator		ตัวอย่าง	
+		a = b+c	a เท่ากับผลลัพธ์ของ b c
-		a = b-c	a เท่ากับผลลัพธ์ของ b c
*		a = b*c	a เท่ากับผลลัพธ์ของ b c
/	หารเอาส่วน	a = 5 / 2	a จะมีค่าเท่ากับ 2 2 5 ได้เพียง 2
%		a = 5 % 2	a จะมีค่าเท่ากับ 1 2 5 1
++	เพิ่มค่าทีละ 1	a++	ถ้า a = 1 แล้ว a++ จะมีค่า เท่ากับ 2
--	ลดค่าทีละ 1	a--	ถ้า a = 1 แล้ว a-- จะมีค่า เท่ากับ 0

+ -	เปลี่ยนจากบวกเป็นลบ	$a = -a$	มีความหมายว่าถ้า $a = 1$ $a = -(1)$ แล้ว $a$ เท่ากับ $-1$
-----	---------------------	----------	-----------------------------------------------------------------

Arithmetic Operator เรียงลำดับจากมากไปหาน้อย และจะประมวลผลจากซ้ายไปขวา

**\* มากกว่า / มากกว่า % มากกว่า + มากกว่า -**

ลำดับความสำคัญนี้จะป็นข้อสังเกตเมื่อเราเขียน เครื่องหมาย Arithmetic Operator ในระดับเดียวกัน โดยที่เราไม่ใช้วงเล็บเพื่อบอกลำดับขั้นของการประมวลผล เช่น

เราต้องการผลลัพธ์ของ  $3 + 5 - 2 * 3 / 2$  ซึ่งจะมีค่าเท่ากับ 9 แต่ถ้าเราเขียน นิพจน์ โดยไม่มีวงเล็บเพื่อแสดงลำดับที่จะให้กระทำก่อนหลังแบบนี้

$$a = 3 + 5 - 2 * 3 / 2$$

a ที่ได้จะมีค่าเป็น 5 ซึ่งไม่ใช่คำตอบที่ถูกต้องเหตุที่เป็นเช่นนี้เพราะอะไรมาดูกัน

ผลลัพธ์ 5 ที่ได้เกิดจาก โปรแกรมอ่านค่าจากซ้ายไปขวาและจะประมวลผลจากเครื่องหมายที่มีความสำคัญมากไปน้อย โดยที่ผลลัพธ์ 5 เกิดจาก ค่าที่ 1  $(-2 * 3)$  จะได้ผลลัพธ์คือ -6 ข้อสังเกตตรงนี้นิดหนึ่งครับ ทำไมค่าตัวแรกเป็น -2 ไม่ใช่  $* 3$

จากซ้ายไปขวา โดยที่เริ่มแรกโปรแกรมจะหาลำดับความสำคัญของเครื่องหมายก่อน เมื่อโปรแกรมอ่านมาพบ เครื่องหมาย  $*$  ซึ่งมีความสำคัญมากที่สุด จากนั้นก็เข้ากฎที่ว่าโปรแกรมจะประมวลผลจากซ้ายไปขวา ซ้ายในที่นี้ด้านซ้ายมือของเครื่องหมายซึ่งก็คือค่า -2

\* \* อยู่ หน้าเลข 3 -2 3 เข้าด้วยกันได้ผลลัพธ์คือ -6

การหารเอาส่วนระหว่าง -6 2 ซึ่งค่าที่ได้ก็คือ -3  $a = 3 + 5 - 3$

ต่อไปโปรแกรมก็ทำการบวก 3 5 เข้าด้วยกันได้ 8 จากนั้นค่อยเข้า 8 มาลบออกด้วย 3 จึงได้เท่ากับ 5

\*\*\*มาเขียนเป็นสมการที่ละขั้นตอนให้ดูจะเข้าใจง่ายกว่า\*\*\*

เริ่มต้น  $a = 3 + 5 - 2 * 3 / 2$

1  $a = 3 + 5 (-2 * 3) / 2$  โปรแกรมจะเริ่มประมวลผลในวงเล็บก่อน

2  $a = 3 + 5 (-6 / 2)$  โปรแกรมจะนำผลลัพธ์จากขั้นที่ 1 มาดำเนินการต่อตาม

3  $a = (3 + 5) - 3$  โปรแกรมจะนำผลลัพธ์จากขั้นที่ 2 มาดำเนินการต่อโดย

เครื่องหมายที่เหลือในขั้นนี้มี บวก กับ ลบ และด้วยเครื่องหมายบวกมีค่าความสำคัญมากกว่า

ก่อนที่จะดำเนินการตามเครื่องหมายลบ

ขั้นสุดท้าย  $a = 8 - 3$  ในขั้นนี้จะเหลือเพียงเครื่องหมายเดียวแล้ว คือ ลบ โปรแกรมจึงทำการประมวลผลจึงได้ผลลัพธ์ออกมาเป็น 5 ซึ่งเป็น 5 ที่ผิด จึงขอให้ระวังเรื่องลำดับความสำคัญของข้อมูลให้มากนะครับ โดยการใส่วงเล็บให้เรียงลำดับการกระทำก่อนหลังให้ถูกต้อง โดยโปรแกรมจะประมวลผลจากวงเล็บที่อยู่ในสุดก่อนจึงค่อยออกมาทำการประมวลผลในขั้นต่อไป ถ้าเราต้องการผลลัพธ์ที่ถูกต้อง ของ  $3 \quad 5 \quad 2 \quad 3 \quad 2$  เราก็ต้องใส่วงเล็บให้ถูกต้องดังนี้  $a = (((3+5)-2)*3)/2$  แบบนี้จะได้คำตอบที่ถูกต้องคือ 9

### โอเปอเรเตอร์การเปรียบเทียบ (Relational Operator)

กำหนดให้ตัวแปร a b มีค่าเท่ากับ 5

Operator	หน้าที่	รูปแบบการใช้	ตัวอย่าง	ผลลัพธ์
==	เปรียบเทียบต้องเท่ากัน	$a == b$	$5 == 5$	true คือเป็น
!=	เปรียบเทียบต้องไม่เท่า	$a != b$	$5 != 5$	false คือเป็น
<	เปรียบเทียบต้องน้อยกว่า	$a < b$	$5 < 5$	false คือเป็น
<=	เปรียบเทียบน้อยกว่าหรือเท่ากับ	$a <= b$	$5 <= 5$	true คือเป็น
>	เปรียบเทียบต้องมากกว่า	$a > b$	$5 > 5$	false คือเป็น
>=	เปรียบเทียบต้องมากกว่าหรือเท่ากับ	$a >= b$	$5 >= 5$	true เป็นจริง

### โอเปอเรเตอร์การกำหนดค่า (Assignment Operator)

โอเปอเรเตอร์	ความหมายและการใช้งาน	ผลลัพธ์
=	กำหนดค่าให้ตัวแปร $x=10$	$x = 10$
+=	$x += 1$ $x = x+1$	$x = 10+1 = 11$
-=	$x -= 1$ $x = x-1$	$x = 11-1 = 10$

*=	x *= 2	x = x*2	x = 10*2 = 20
/=	x /= 2	x = x/2	x = 20/2 = 10
%=	x %=3	x = x%3	x = 10%3 = 1

### โอเปอเรเตอร์ตรรกศาสตร์ (Logical Operator)

โอเปอเรเตอร์		รูปแบบการใช้
!	not ( )	!a คือไม่ใช่ a
&&	and ( )	a && b    a    b
	or ( )	a    b    a    b

### ตารางค่าความจริงของ && ( )

	2	ผลลัพธ์
True	True	True
True	False	False
False	True	False
False	False	False

### ตารางค่าความจริงของ || ( )


	2	ผลลัพธ์
True	True	True
True	False	True
False	True	True
False	False	False

### โอเปอเรเตอร์การเพิ่มค่าและลดค่า (Increment & Decrement Operator)

โอเปอเรเตอร์	หน้าที่	ตัวอย่าง	ผลลัพธ์
++a	เพิ่มก่อน	x = 1 ; ++x	x = 2
--a	ลดก่อน	x = 1 ; --x	x=0
a++		x=1 ; x++	x=1
a--		x =1;x--	x=1

**Present by [www.thaiforexea.com](http://www.thaiforexea.com)**

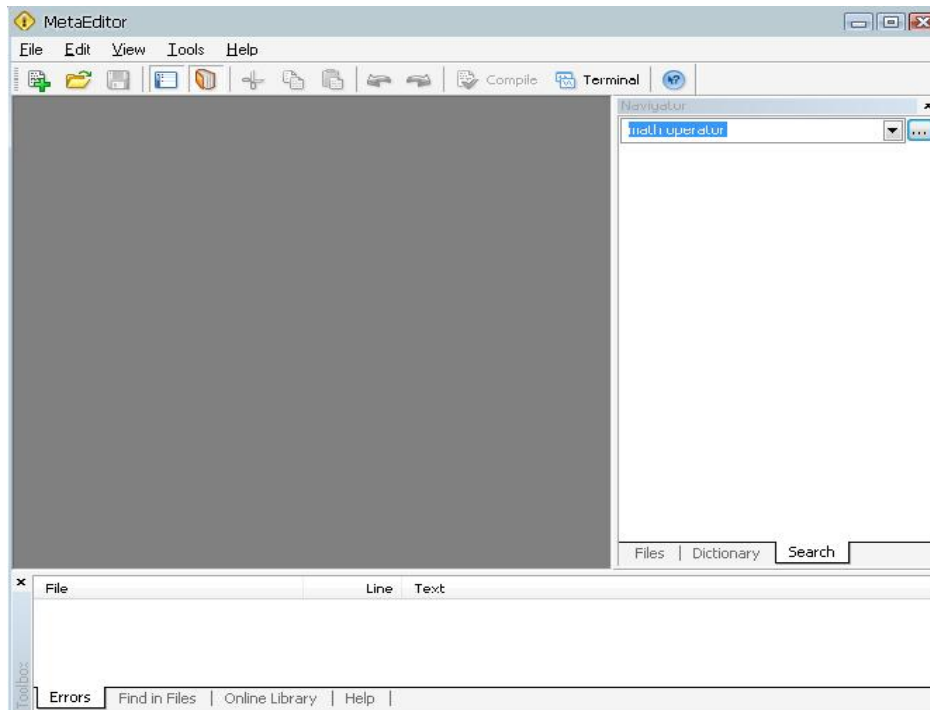
## 2 ทำความรู้จักกับ Meta Editor

เริ่มแรกให้เปิดโปรแกรม MT4 ขึ้นมาของโบรกเกอร์ไหนก็ได้ครับจากนั้นคลิกที่  Meta Editor

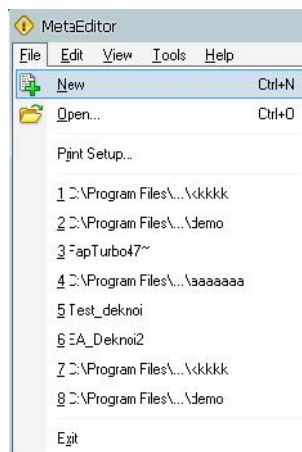


Meta Editor ที่เราเรียกก็ทำงาน ก็จะมีหน้าต่างดังนี้





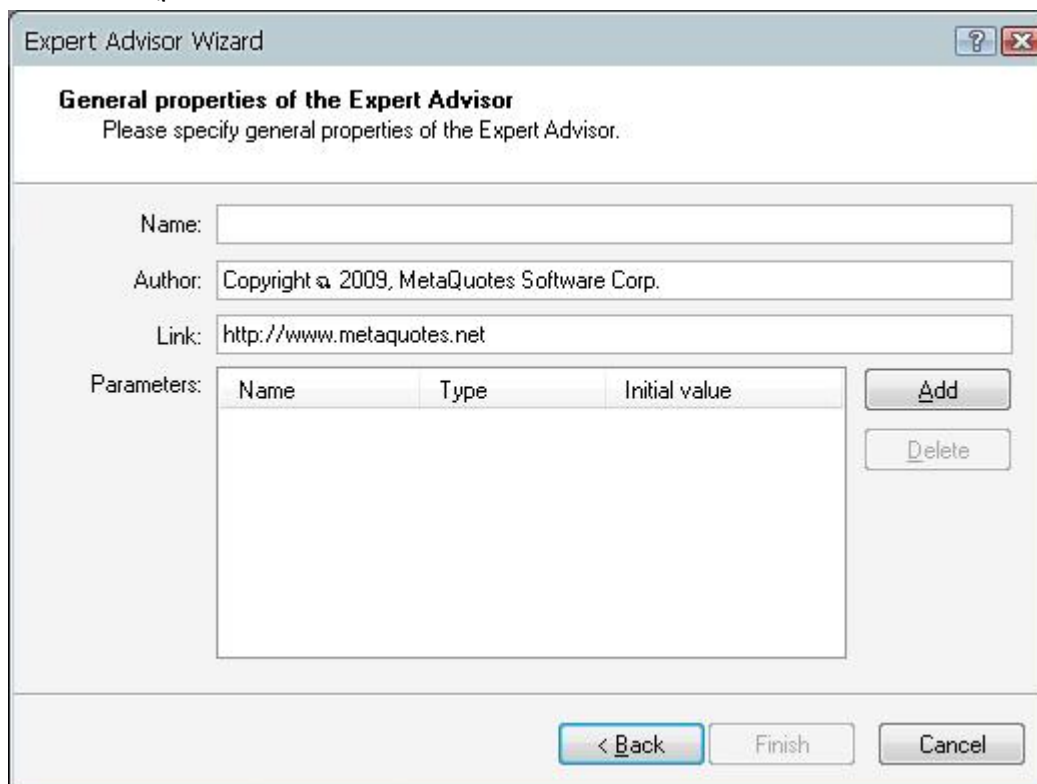
จากนั้นให้คลิกที่ File แล้วเลือก new



จากนั้นจะปรากฏหน้าต่าง Expert Advisor Wizard



ในหน้าต่างนี้ให้เลือกที่ Expert Advisor เพื่อที่เราจะทำการสร้าง EA  
แล้วจะปรากฏหน้าต่างต่อไปนี้



มาถึงหน้านี้ให้เราตั้ง Ea ของเราที่ช่อง Name ครับในที่นี้ผมตั้งไปว่า Hello World จากนั้นให้

finish เลขครับจะได้หน้าจอที่มี code

```
#property link      "http://www.metaquotes.net"

//+-----+
//| expert initialization function |
//+-----+
int init()
{
//----
//----
    return(0);
}
//+-----+
//| expert deinitialization function |
//+-----+
int deinit()
{
//----
//----
    return(0);
}
//+-----+
//| expert start function |
//+-----+
int start()
{
//----
//----
    return(0);
}
//+-----+
```


code เราจะเห็นว่า มีฟังก์ชันอยู่ 3 ฟังก์ชัน คือ init() deinit() และฟังก์ชัน start() ฟังก์ชัน init() เป็นฟังก์ชันที่ใช้กำหนดค่าเริ่มต้นให้ EA Construction โปรแกรมอื่นๆ ซึ่งจะมีฟังก์ชันนี้หรือไม่ก็ได้ ฟังก์ชัน deinit() เป็นฟังก์ชันที่ไวจบการทำงานของฟังก์ชัน init() Destruction ภาษาโปรแกรมอื่นๆ ซึ่งจะมีหรือไม่ก็ได้ ฟังก์ชัน start() เป็นฟังก์ชันที่ไวเริ่มค่าต้นการทำงาน of โปรแกรมจำเป็นต้องมีฟังก์ชันนี้เพราะ หากไม่มี Ea ก็ไม่สามารถทำงานได้

ที่นี้เรามาเริ่มเขียนโปรแกรมแรกกันเลย โดยให้ไปที่ฟังก์ชัน start () ก่อนอื่นของแนะนำ // เป็นเครื่องหมายที่เอาไว้ใช้ comment ข้อความ code ที่เขียนอยู่หลังเครื่องหมายนี้ ภายในบรรทัดเดียวกันจะไม่ถูกทำการประมวลผล และก็จะแนะนำเครื่องหมาย /\* ใส่ข้อความลงไปที่นี่ก็บรรทัดก็ได้ และปิดเครื่องหมายนี้ด้วย สัญลักษณ์แบบนี้ /\* ข้อความที่อยู่ระหว่าง /\* \*/ จะที่บรรทัดก็ตามจะไม่ถูกประมวลผล ที่นี้เราก้มาเริ่ม โปรแกรมแรกกันได้แล้วครับโดยไปที่ ฟังก์ชัน start() แล้วเขียน code

ลงไปให้เหมือนตามภาพด้านล่างนี้

```
//+-----  
int start()  
{  
//----  
    Print("Hello world ");  
//----  
    return(0);  
}  
//+-----
```

Present by [www.thaiforexa.com](http://www.thaiforexa.com)

Code                      Print("Hello world ");                      Print เป็นคำสั่งที่ให้แสดงผล  
ข้อความออกมา เป็นคำสั่งซึ่งมีประโยชน์มากในการเขียน EA เพราะเราสามารถใส่คำสั่ง Print  
เพื่อเช็คความถูกต้องของค่าในตัวแปรต่างๆได้ จากนั้นให้คลิก  Compile หากไม่มีไรผิดพลาด  
จะได้ผลการ compile ตามภาพข้างล่างนี้

```

MetaEditor - [hello world]
File Edit View Tools Window Help
Compile Terminal

//+-----
int init()
{
//----|

//-----
    return(0);
}
//+-----
//| expert deinitialization function
//+-----
int deinit()
{
//--
    Present by www.thaiforeaea.com
//--
    return(0);
}
//+-----
//| expert start function
//+-----
int start()
{
//----
    Print("Hello world ");
//----
    return(0);
}
//+-----

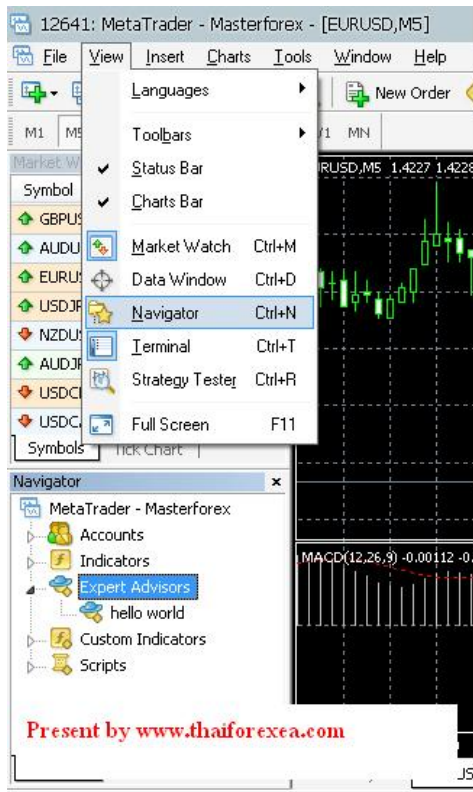
Description
Compiling 'hello world.mq4'...
0 error(s), 0 warning(s)

```

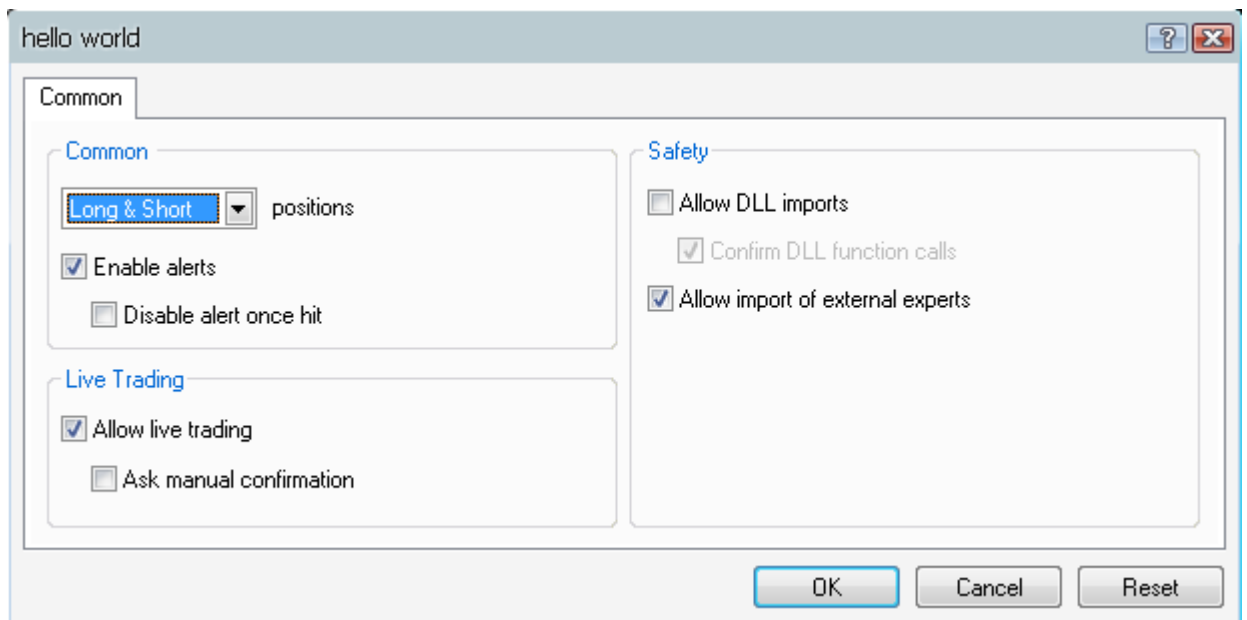
จากภาพด้านล่างแสดงว่า code ที่เราเขียนลงไปมีความถูกต้องเราจึง compile ผ่านไม่มี error

compile ผ่านคือการที่ไม่มี error code ของเราเลย แต่ถ้าหากผลการ  
 compile ไม่มี error เกิดขึ้นแต่มี warning compile นั่นก็ถือว่าผ่านแต่มีบาง  
 function หรือบาง ตัวแปรที่ไม่ได้ถูกนำไปใช้เท่า

จากนั้นให้เปิดโปรแกรม MT4 Broker ไหนก็ได้จากนั้นไปที่หน้าต่าง Navigator



EA ที่เราสร้างขึ้นในหน้าต่าง Navigator หากไม่มีหน้าต่าง Navigator ให้ไปที่ view ด้านบนจากนั้นให้ลาก EA ที่เราสร้างขึ้นมาใส่ในกราฟ จะมี หน้าต่างขึ้นตามภาพด้านล่างนี้



จากนั้นให้เราเลือก Allow live trading แล้วคลิก OK จากนั้นให้ปิด MT4 และเปิดขึ้นมาใหม่อีก

หลังจากที่เปิด  จากนั้นเราจะเห็นสัญลักษณ์รูปยิ้มตามรูป

จากนั้นให้เปิดหน้าต่าง Terminal

Tab Experts

จะเห็นข้อความ Hello

World ที่เราใช้คำสั่ง



Print ให้แสดงผลออกมา

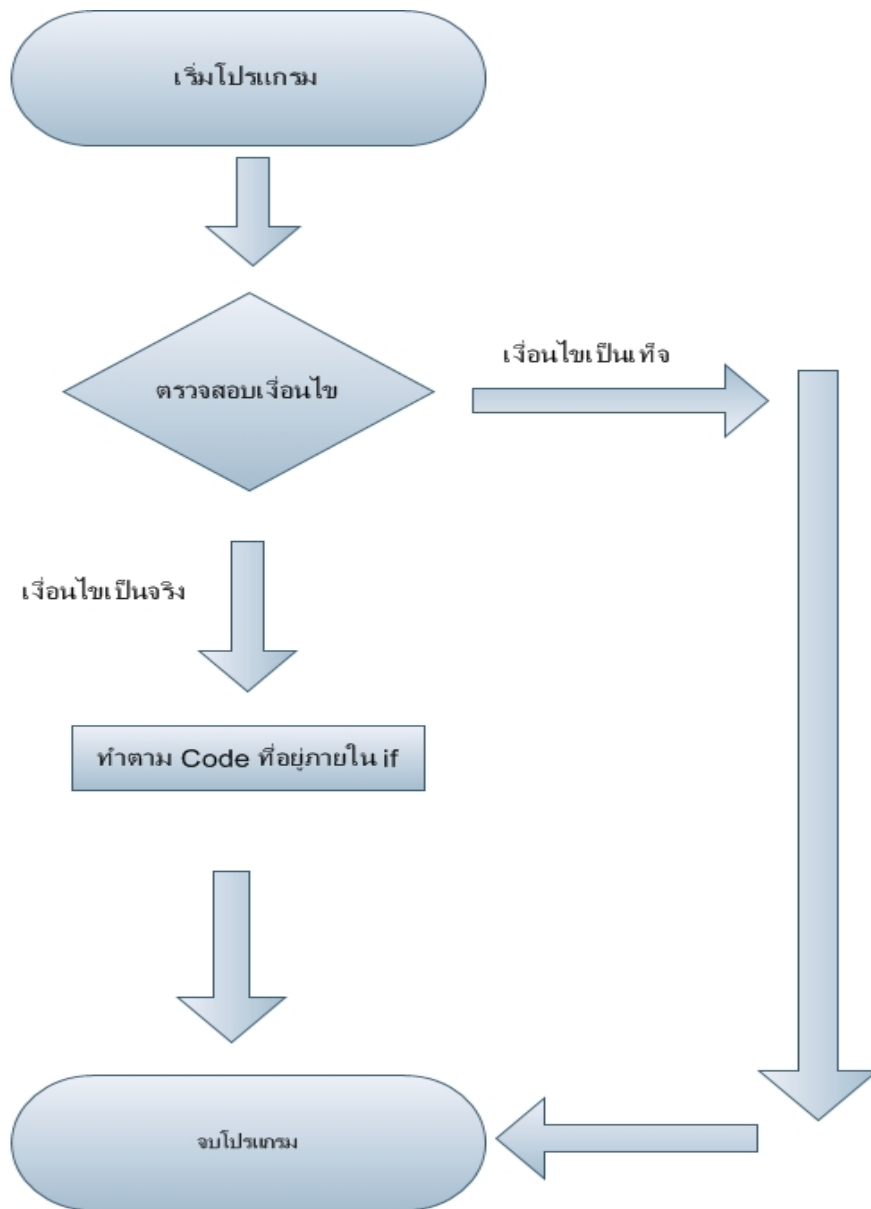
Time	Message
2009.07.28 21:58:29	hello world EURUSD,M5: Hello world
2009.07.28 21:58:29	hello world EURUSD,M5: Hello world
2009.07.28 21:57:47	hello world EURUSD,M5: Hello world
2009.07.28 21:57:23	hello world EURUSD,M5: Hello world
2009.07.28 21:57:04	hello world EURUSD,M5: Hello world
2009.07.28 21:56:56	hello world EURUSD,M5: Hello world

Present by [www.thaiforexea.com](http://www.thaiforexea.com)

### 3 if , if\_else ,switch case

คำสั่งเงื่อนไข เป็นคำสั่งที่เราเอาไว้ให้โปรแกรมทำตามเมื่อเงื่อนไขเข้าเกณฑ์ที่กำหนด เป็นคำสั่งที่เราจะใช้บ่อยที่สุดใน EA เลยครับเพราะเป็นคำสั่งที่เราเอาไว้สร้างจุดเข้าซื้อ เข้าขาย Order ง่ายขึ้น

**If** เป็นคำสั่งแรกที่เราจะเรียนรู้กันเกี่ยวกับคำสั่งเงื่อนไขโดยมีหลักการทำงานตาม Flow Chart ด้านล่างนี้



รูปด้านบนคือ Flow Chart

if ต่อมาเรามาดูตัวอย่าง code



if ง่ายๆกันดีกว่า

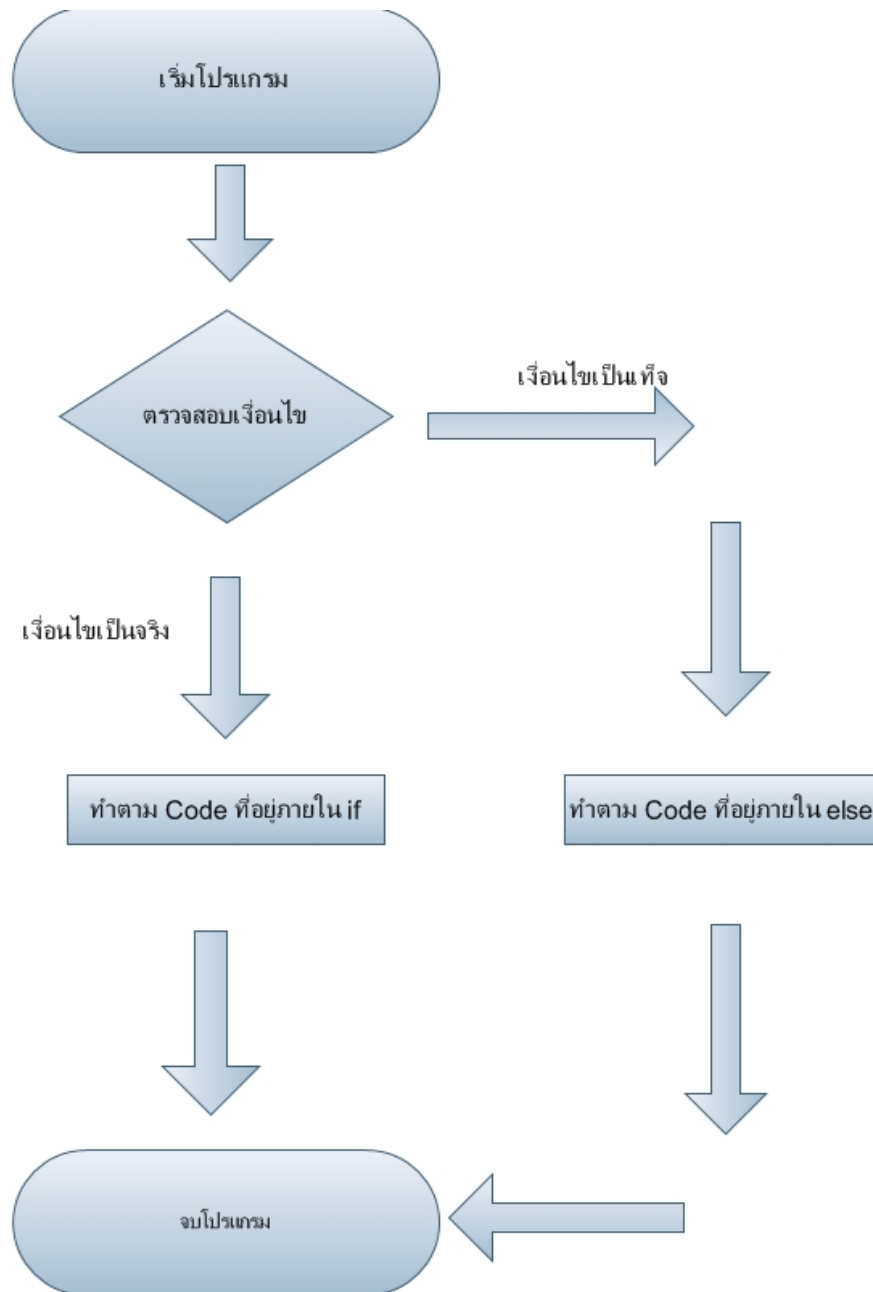
```
int a = 0;

if(a==0)
{
    Print("Hello world ");
}
```

code ที่เห็น เริ่มต้นโปรแกรมด้วยการกำหนดค่า a ให้เก็บค่า 0 เอาไว้และเราใช้คำสั่ง if ให้ if ตรวจสอบค่า a ว่ามีค่าเท่ากับ 0 หรือไหมหากค่า a เท่ากับ 0 ให้ทำการแสดงข้อความ Hello world ขึ้นมา หากไม่เท่ากับ 0 ก็ให้ทำการจบโปรแกรมเลย แต่ถ้าหากเราต้องการให้โปรแกรมทำอย่างอื่นต่อถ้าค่าที่ออกมาเป็นเท็จ อะเราจะใช้คำสั่งอะไร ในกรณีนี้เราสามารถใช่

**if\_else**

Flow Chart ในหน้าต่อไป



รูปด้านบนคือ Flow Chart  
if ง่ายๆกันดีกว่า

if\_else ต่อมาเรามาดูตัวอย่าง code

```

int a = 0;

if(a==0)
{
    Print("Hello world ");
}
else
{
    Print("A is not 0 ");
}

```

code เป็น code else เพิ่มเข้าไปหากเรารันโปรแกรมนี้ ผลลัพธ์ที่ได้ก็จะแสดงคำว่า Hello world เหมือนเดิมแต่หากเราเปลี่ยนค่า a ให้เป็นค่าอื่นที่ไม่ใช่ 0 ผลลัพธ์ที่ได้จะแสดงออกมาเป็น “A is not 0 “

**Switch Case** เป็นคำสั่งที่ใช้ในการตัดเงื่อนไขอีกคำสั่งหนึ่ง ของให้ดูตัวอย่าง code นี้ก่อน

```

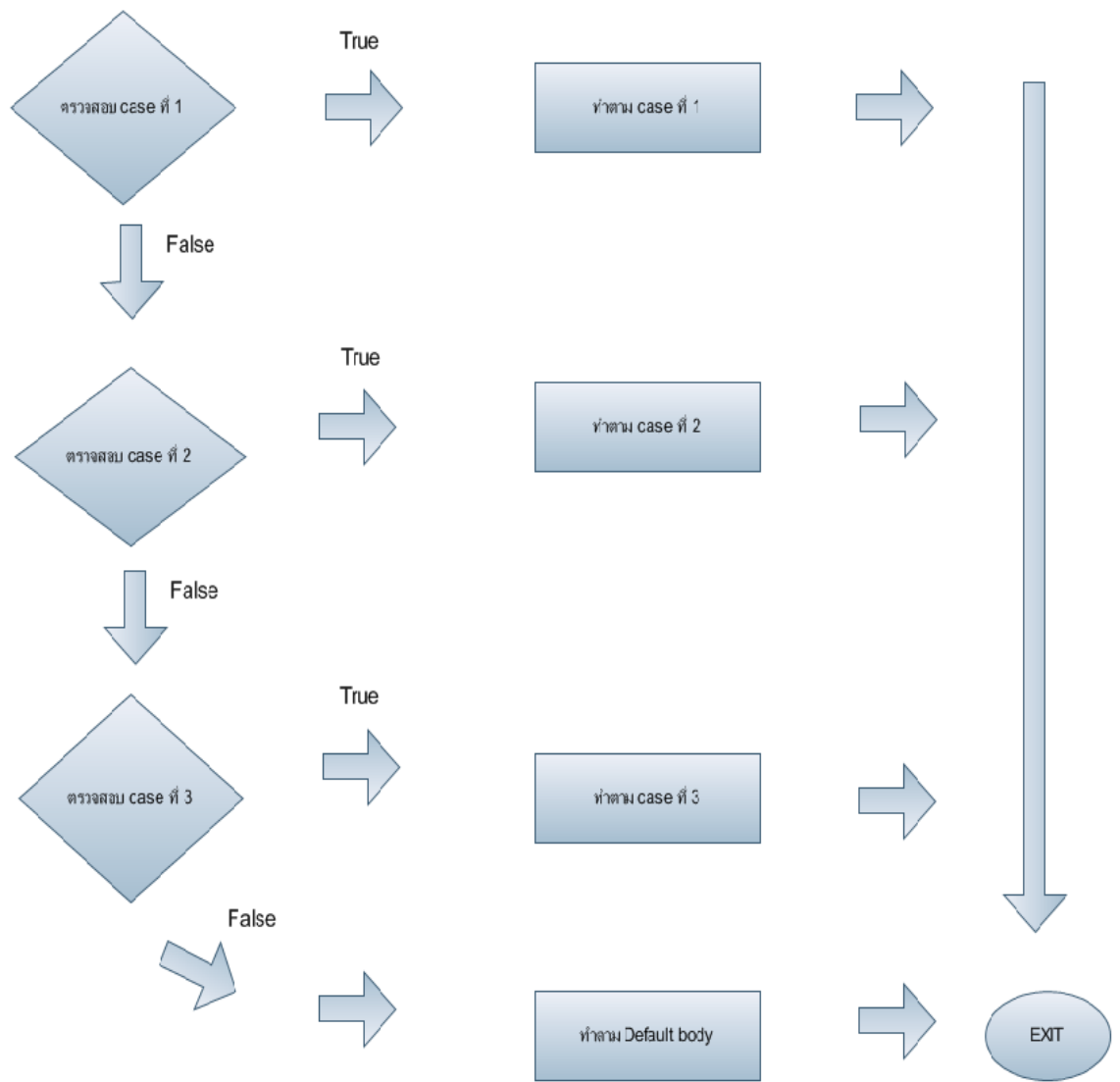
switch(a)
{
    case '0':
        Print("A = 0 ");
        break;

    case '1':
    case '2':
        Print("A = 1 or 2 ");
        break;

    default:
        Print("A != 1 or 2 or 3 ");
        break;
}

```

Flow Chart ด้านล่างนี้



switch จะตรวจสอบค่า a โดยที่ถ้า a มีค่าเท่ากับ 0 “A = 0”  
 switch break; ถ้าค่า a เท่ากับ 1 2  
 “A = 1 or 2 “ break ; อีกเช่นกัน หากค่า a ไม่เท่ากับ  
 0,1,2 default :  
 ให้แสดงผล “ A != 1 or 2 or 3 “ และจะจบด้วยคำสั่ง break; อีกเช่นเดิม

Present by [www.thaiforex.com](http://www.thaiforex.com)

## 4 loop ( for , while ) array

loop เป็นการที่เราต้องการให้คอมพิวเตอร์ทำสิ่งที่เราต้องการทำซ้ำจนกว่าจะได้สิ่งที่เราต้องการ เช่น เราต้องการให้คอมพิวเตอร์ บวกเลข 1 เป็นจำนวน 100 ครั้ง หรือเราต้องการให้คอมพิวเตอร์พิมพ์หรือแสดงข้อความ 100 รอบเป็นต้น คำสั่งที่ใช้ในการวน loop มีด้วยกัน 2 for while

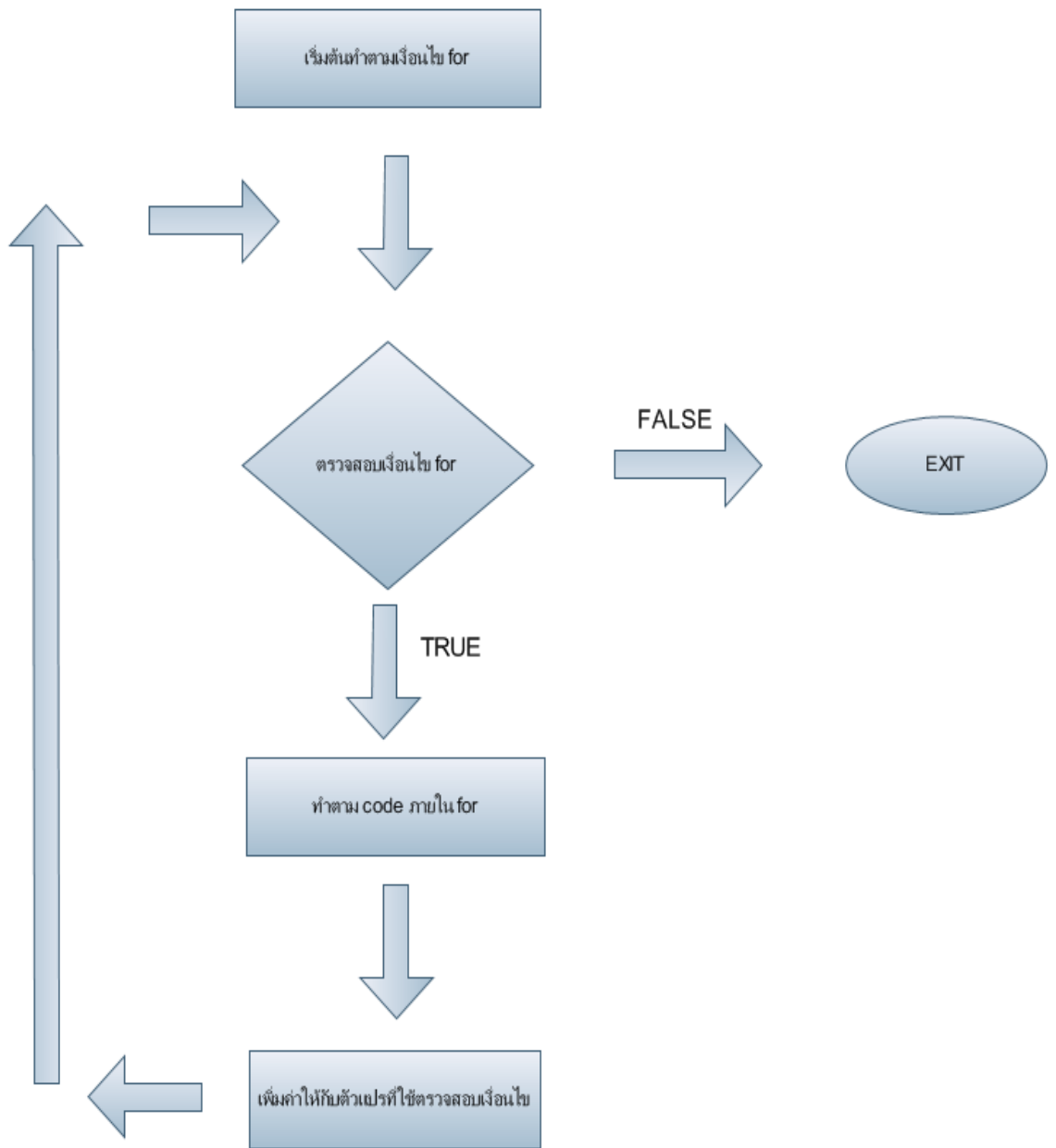
### for

```
for(int i =0;i<10;i++)
{
    Print("Hello ",i+1);
}
```

code ข้างบนเมื่อรัน โปรแกรม จะแสดงผลดังนี้

Hello 1  
Hello 2  
.  
.  
.  
Hello 10

เป็นการสั่งให้คอมพิวเตอร์แสดงผล Hello 10  
for ( ประกาศตัวแปรที่จะใช้เป็นเงื่อนไข ;  
เพิ่มค่าหรือลดให้กับตัวแปรที่ใช้เป็นเงื่อนไข ) code ข้างบนในบรรทัด for  
i ให้มีค่าเริ่มต้นเป็น 0 จากนั้นเราใช้เครื่องหมาย ; ให้คอมพิวเตอร์รู้ว่าสิ้นสุดการ  
ประกาศตัวแปร หลังจากนั้นเราเริ่มต้นประกาศเงื่อนไขว่าถ้า  $i < 10$  เงื่อนไขยังเป็นจริงอยู่ จากนั้น  
เราปิดการประกาศเงื่อนไขด้วยเครื่องหมาย ; และเราประกาศการเพิ่มขึ้นของตัวแปรว่าเพิ่มทีละ 1  
i++ จากนั้นเราจึงปิด ) แล้วขึ้นบรรทัดใหม่เปิด { code ให้คอมพิวเตอร์ทำงานเมื่อ  
เงื่อนไขเป็นจริงจากนั้นเราก็ใช้ } เพื่อให้คอมพิวเตอร์รู้ถึงจุดสิ้นสุดของ code  
for สามารถอธิบายด้วย flow chart ดังต่อไปนี้



## while

**while** มีลักษณะการทำงานเช่นเดียวกับ for แต่มีวิธีเขียนให้ตรวจสอบเงื่อนไขให้แตกต่างกัน  
เท่านั้นเองดูตัวอย่าง code

```
while(i<10)
{
    Print("Hello ",i+1);
    i++;
}
```

code ข้างบนเมื่อรัน โปรแกรม จะแสดงผลดังนี้

Hello 1

Hello 2

.

.

.

Hello 10 ผลลัพธ์ที่ได้ก็เหมือนกับ for ด้านบนแต่ที่แตกต่างก็เป็นวิธีเขียนเงื่อนไข

กล่าวคือ while จะตรวจสอบเงื่อนไขก่อนถ้าจริงก็จะทำตาม code while แล้วค่อยมีการ

เปลี่ยนแปลงค่าของเงื่อนไขภายใน code while ต่างจาก for ที่เราต้องกำหนดทุกอย่างก่อนถึง

while จะมีประโยชน์มากถ้าเงื่อนไขที่ต้องให้ตรวจสอบเกิดจาก การ

เปลี่ยนแปลงค่าของตัวแปรที่ใช้ตรวจสอบมีความไม่แน่นอนเช่น ในโปรแกรม EA

ต้องการให้โปรแกรมพิมพ์ข้อความว่า Sell Order ในขณะที่เราเริ่มทำการปิด order

จนกว่า เราจะปิด order EA ของเราเช็คเงื่อนไขว่าต้องทำการปิด order

ทั้งหมดก็จะเริ่มแสดงชื่อว่า Sell Order ไปเรื่อยๆจนกว่าจะปิด Order หหมด ปัญหาที่เราต้องใช้

while แทนที่จะใช้ for ในการสั่งให้พิมพ์ข้อความในครั้งนี้ก็คือเราไม่รู้ว่าจะใช้เวลาานเท่าไร

จนกว่า ea จะปิด Order ได้หมดหากใช้ for เราต้องกำหนดจำนวนรอบที่จะใช้พิมพ์ลงไปเลย แล้ว

for จะสั่งเพิ่มหรือลดค่าตัวแปรที่ใช้ตรวจสอบเงื่อนไขเองอัตโนมัติด้วย แต่คำสั่ง for มีหน้าที่

แค่เช็คเงื่อนไขอย่างเดียวไม่ได้มีหน้าที่เพิ่มหรือลดค่าตัวแปรที่ใช้ตรวจสอบเงื่อนไขด้วย ซึ่งเราต้อง

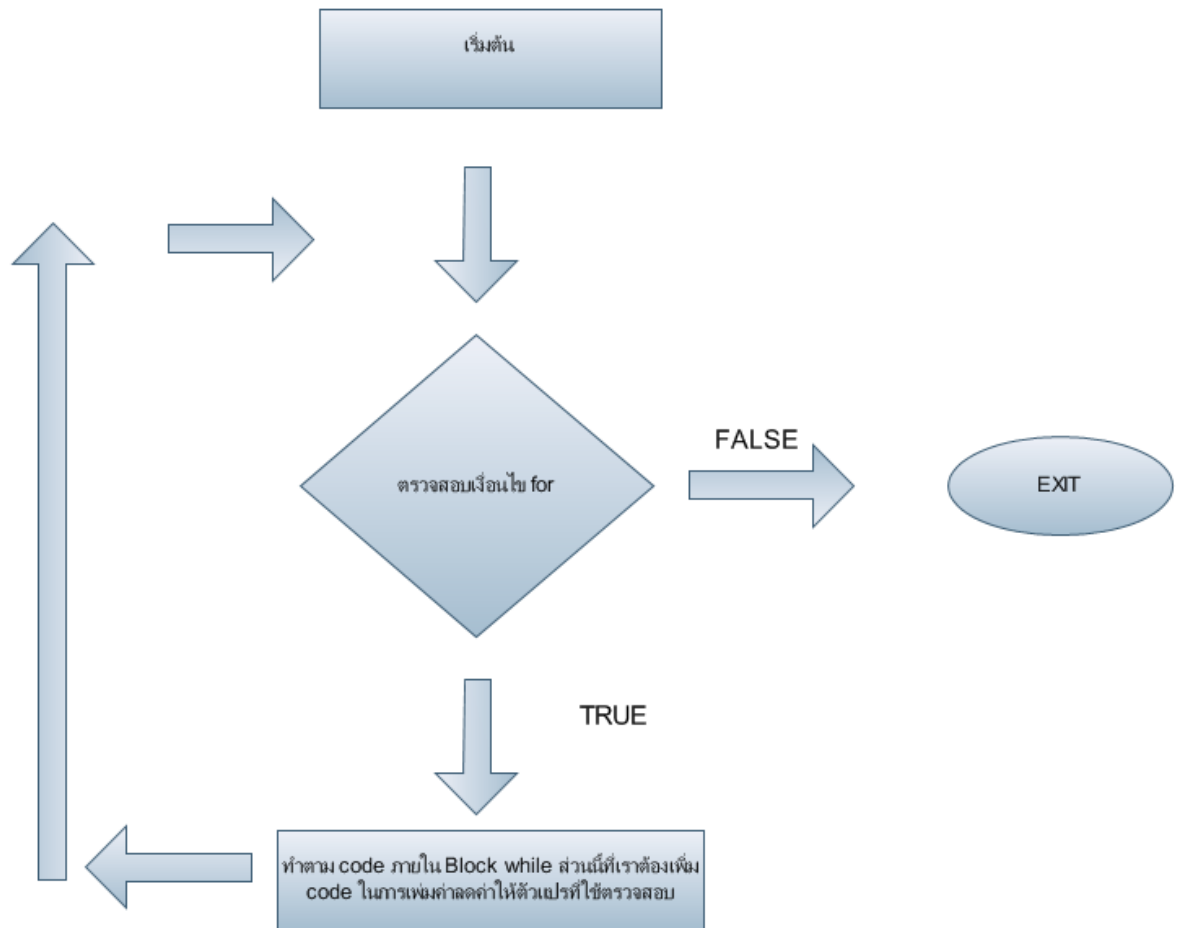
เป็นคนเขียน code ให้สั่งเพิ่มหรือลดค่าตัวแปรที่ใช้ตรวจสอบเงื่อนไขเอง ในกรณีนี้ใน block

while code ลงไปให้ตรวจสอบจำนวน order ที่เราเปิดอยู่ว่ายังมากกว่า

0 หรือไม่ถ้ายังมากกว่า 0 loop while ก็ยังทำงานต่อไปหาเมื่อตรวจสอบพบว่ามีค่าเท่ากับ 0

loop while EA ของเราก็จะเลิกพิมพ์ข้อความ Sell Order สังเกตของแต่ต่าง

loop for flow chart ด้านล่างนี้



## Array

**Array** เป็นโครงสร้างหนึ่งที่สามารถเก็บชนิดข้อมูลได้หลายๆ จำนวนภายในตัวแปรเดียว หรือเราเรียกว่าหลายๆ อิลิเมนต์ (Element) นอกจากนี้สามารถเก็บได้หลายข้อมูลในตัวแปรเดียวแล้ว ยังมีหลายมิติอีกด้วยรูปแบบตัวแปร array            type            [ขนาดของอาร์เรย์];  
เช่น

`int a[10];` มีความหมายว่า ตัวแปร array    int            10  
ขนาดอาร์เรย์ มีค่าเริ่มต้นจาก 0

MT4

`int a[5] = {1,2,3,4,5};`

array และก็กำหนดค่าให้ array ทุกตำแหน่ง  
array แต่ละตัวมีค่าดังนี้



```
a[0] = 1
a[1] = 2
a[2] = 3
a[3] = 4
a[4] = 5
```

ในตัวอย่างนี้เรายังมีที่ว่างของ a[5] อีกหนึ่งตำแหน่ง

```
int a[5];
```

วิธีนี้เราประกาศแต่ชื่อ array

array เท่านั้น ส่วนข้อมูลที่เก็บเราค่อยมา

```
int a[]
```

วิธีการนี้เราประกาศแค่ชื่อ array โดยที่ขนาดและข้อมูลนั้นยังไม่ได้ประกาศซึ่งเราสามารถกำหนดขนาดและ ข้อภายหลังได้

```
int a[5]={1,2,3,4,5}

for(i =0;i<5;i++)
{
    Print("Number ",a[i]);
}
```

ผลลัพธ์จาก code

Number 1

...

...

Number 5

เป็น code ง่าย

สามารถเก็บได้หลายค่าในตัวแปรเดียวของ array

**Present by [www.thaiforex.com](http://www.thaiforex.com)**

## 5 function

**function** เป็น code ส่วนย่อยของโปรแกรมที่มีความสามารถในการทำงานอย่างใดอย่างหนึ่ง โดยเฉพาะ โดยส่วนใหญ่เวลาที่เราเรียกใช้ฟังก์ชันจะต้องมีการส่งค่าให้ฟังก์ชัน ค่าที่ส่งไปนี้ ศัพท์ทางโปรแกรมเมอร์เรียกว่า การส่งค่า อากิวเมนต์ให้กับฟังก์ชัน ซึ่งไม่จำเป็นต้องมีการส่งค่าให้ฟังก์ชัน ขึ้นอยู่กับลักษณะการทำงานของฟังก์ชันเอง ว่าสามารถทำงานอัตโนมัติได้เลยไหมหรือว่าจะจำเป็นต้องรับค่า อากิวเมนต์ ที่ถูกส่งมาตอนเรียกใช้ฟังก์ชัน แต่ในฟังก์ชันเราไม่เรียกค่านั้นว่าอากิวเมนต์แต่กลับเรียกว่า ค่าพารามิเตอร์ เรียกสั้นๆง่ายๆก็เรียกว่าค่าที่ส่งมา ก็พอครับ ที่นี้ ทุกคนที่ไม่เคยเรียนวิชาโปรแกรมมิ่ง มาก่อนอาจจะงง ผมจะขอยกตัวอย่างง่ายๆครับ โดยผมจะเอาเรื่องในชีวิตประจำวันมาเปรียบเทียบให้ฟังครับ ตัวอย่าง ในชีวิตประจำวันเช่นการโทรศัพท์โทรออกไปหาให้บางคน เวลาที่เราขงโทรโทรศัพท์ขึ้นก็เทียบได้กับการที่เราประกาศว่าจะใช้ฟังก์ชันในการโทรออก เมื่อเรากดหมายเลขโทรศัพท์จนครบนั้นก็เป็นการส่งค่าให้กับฟังก์ชัน ที่บริษัทโทรศัพท์เปรียบได้

code ของฟังก์ชัน จะรับเบอร์โทรศัพท์ที่คุณกดเข้ามา และทำการโทรออกให้คุณ การโทรออกหรือไม่ติดก็คือค่าหรือสิ่งที่ฟังก์ชันตอบสนองการเรียกใช้ของเราไงครับ อ่านดูแล้วเข้าใจไม่ครับถ้าไม่เข้าใจก็โพสถามที่ board ได้ครับ

ที่นี้เรามาดูประเภทของ ฟังก์ชันกันครับ ฟังก์ชันมี 2 ประเภทคือ ฟังก์ชันที่ส่งค่าคืนกลับมา กับ ฟังก์ชันที่ไม่ส่งค่าคืนกลับ ฟังก์ชันที่ส่งค่าคืนกลับมาเป็นฟังก์ชันที่ทำงานเฉพาะอย่างเมื่อมีการรับค่าแล้ว ทำงานแล้วก็ต้องมีการคืนค่า ตัวอย่างฟังก์ชันประเภทนี้ได้แก่ฟังก์ชัน `int start()` ที่ก่อน code ต้องมีการ return ค่าออกมา ส่วนฟังก์ชันที่ไม่คืนค่าคือมันทำงานจบได้ผ่านในตัวมันเอง เช่น ฟังก์ชันของคำสั่ง `Print()`

### วิธีการประกาศฟังก์ชัน

ฟังก์ชันที่มีการ return ค่าต้องขึ้นด้วย type ของข้อมูลที่จะมีการ return ค่าด้วยเช่น ต้องการให้มีการคืนตัวเลขก็ต้องประกาศด้วย `int` ชื่อฟังก์ชัน() เช่น `int sum()` เป็นต้น

ฟังก์ชันที่ไม่ต้องการให้มีการคืนค่าให้ประกาศหน้าฟังก์ชันด้วย `void` ชื่อฟังก์ชัน() เช่น `void show()` เป็นต้น

### ตัวอย่างการประกาศ function

```
int sum(int a) //ตัวอย่างฟังก์ชันที่มีการรับค่าและคืนค่า
{
    code ของฟังก์ชัน
    return 0 ;
}
```

```
void show() //ตัวอย่างฟังก์ชันที่ไม่มีการรับค่าและคืนค่า
{
    Print("Show ");
}

```

ตำแหน่งที่จะประกาศฟังก์ชันให้ประกาศก่อนฟังก์ชัน int start() เสมอ ดูตัวอย่างตาม code ด้านล่างนี้

```
void show()
{
    Print("Show function ");
}

int sumnumber(int a)
{
    return (a+1);
}

int start ()
{
    show();
    Print("Result ",sumnumber(1));

    return(0);
}

```

code เห็นตำแหน่งที่ประกาศฟังก์ชันจะต้องประกาศก่อนฟังก์ชัน start EA  
 ที่นี้ โดยตัวอย่างมี 2 ฟังก์ชันคือ show() sumnumber() โดยที่ฟังก์ชัน show() เป็นฟังก์ชันที่  
 ไม่มีการรับค่าและไม่มีการ return ค่ากลับ ส่วนฟังก์ชัน sumnumber() เป็นฟังก์ชันที่มีการรับค่า  
 และมีการคืนค่าออกมาด้วย

ผลลัพธ์ที่ได้จากการรันนี้คือ

Show function  
 Result 2

ฟังก์ชันมีประโยชน์มากใน EA ให้มีความสามารถเพิ่มขึ้นนอกเนื่องจากคำสั่ง  
 MT4 ที่เกี่ยวกับอินดิเคเตอร์พื้นฐาน เช่นการพัฒนาฟังก์ชันให้รู้จัก  
 การจัดสรรเงินลงทุน หรือใช้พัฒนาให้ EA สร้างเงื่อนไขในการเข้าซื้อขายด้วยกลยุทธ์ ที่คิดขึ้นเอง

**Present by [www.thaiforex.com](http://www.thaiforex.com)**

## 6

## Trade

เป็นการรวมคำสั่งต่างๆที่ทำให้ EA สามารถเบื้องต้น นะครับผมจะพยายามรวมเท่าที่สำคัญๆออกมาเท่านั้นเพราะเราสามารถศึกษาเพิ่มเติม ได้ที่ MQL4 Reference ได้เลยนะครับ โดยผมขอ order ชนิดต่างๆ และก็ฟังก์ชันที่เกี่ยวกับอินดิเคเตอร์พื้นฐาน เท่านั้นนะครับ เพราะมันก็เพียงพอสำหรับเริ่มสร้าง EA ขึ้นมาสักตัวแล้วครับเมื่อเรามีความชำนาญก็จะประยุกต์ทฤษฎีต่างๆ เข้ามาเพื่อสร้าง EA ได้เองหละครับ

**OrdersSelect ()** เป็นคำสั่งที่สำคัญที่สุดครับเพราะใช้ในการเลือก Order ที่เราเปิดแล้วเราต้องใช้คำสั่ง **order**

รูปแบบการใช้งาน

```
bool OrderSelect(int index, int select, int pool=MODE_TRADES)
```

Parameters      OrderSelect

index	-	Order index or order ticket depending on the second parameter.
select	-	Selecting flags. It can be any of the following values: SELECT_BY_POS – อ้างอิงโดยใช้ตำแหน่งของ Order วิธีนี้จะใช้กับการเปิด Order ไม่ได้ครับ SELECT_BY_TICKET – ใช้อ้างอิง Order ด้วย หมายเลข id Order
pool	-	Optional order pool index. Used when the selected parameter is SELECT_BY_POS. It can be any of the following values: MODE_TRADES - ใช้เพื่อที่จะให้ระบุถึง Order ที่กำลังจะเปิดหรือ Order ที่เปิดอยู่แล้วครับ MODE_HISTORY – ใช้เพื่อที่จะระบุถึง Order ที่เราทำการปิดไปแล้วครับ

การวิธีการใช้ OrderSelect() นั้นมีวิธีการใช้อยู่ 2 วิธีคือใช้คำสั่งนี้ก่อนคำสั่งที่เกี่ยวกับ Order อื่น กับ ใช้

Order

Order ที่ไม่ต้องใช้ควบคู่ OrderSelect() ได้แก่

OrdersTotal()      OrdersHistoryTotal()

Order ที่ต้องประกาศก่อน OrderSelect() ได้แก่

OrderSend()

Order ที่จะต้องประกาศ OrderSelect() ก่อนจะประกาศใช้คำสั่ง Order นั้นได้แก่

OrderMagicNumber() , OrderType(), OrderTicket()  
,OrderTakeProfit(),OrderSymbol(),OrderSwap(),OrderStopLoss(),OrderProfit()  
OrderOpenTime(),OrderClosePrice(),OrderCloseTime(),OrderOpenPrice(),  
OrderLots(),OrderExpiration(),OrderCommission(),OrderComment(),  
OrderPrint(),OrderProfit(),OrderStopLoss(),OrderSwap() น่าจะหมดแล้วนะ

เพื่อเป็นการไม่สับสน ต่อไปนี้ผมจะมีสัญลักษณ์ # ต่อท้ายทุกฟังก์ชันที่นำมาแสดงโดยที่

# สีเขียวแสดงว่าไม่ต้องมีการใช้ **OrderSelect()** ร่วมด้วย

# สีแดงแสดงว่าต้องมีการใช้ **OrderSelect()** ก่อนถึงใช้งานได้

# สีน้ำเงินแสดงว่าใช้ **OrderSelect** ต่อท้ายหลังจากใช้ฟังก์ชันนี้แล้ว

**OrdersTotal** เป็นคำสั่งที่ใช้ตรวจสอบจำนวน **Order** ของเราที่เปิดอยู่และรวมจำนวน **Pending Order** ด้วยครับ

รูปแบบการใช้งาน

```
int OrdersTotal() #  
เช่น int a = OrdersTotal();  
แบบนี้ค่า a          Order ที่เปิดอยู่ทั้งหมดแล้วครับ
```

**OrdersHistoryTotal()** เป็นคำสั่งให้นับจำนวน **Order** ที่เราปิด **Position** ไปแล้ว จะแสดงจำนวนเท่าไรขึ้นอยู่กับว่าใน **Tab Account History** เราตั้งค่าให้แสดง **History** ในช่วงเวลาใดมั่ง

รูปแบบการใช้งาน

```
int OrdersHistoryTotal ()#  
เช่น int a = OrdersHistoryTotal();  
แบบนี้ค่า a          Order History แล้วครับ
```

**OrderSend** ใช้การสั่งให้โปรแกรมทำการเปิด **Order**

```
int OrderSend ( string symbol, int cmd, double volume, double price, int slippage, double stoploss, double takeprofit, string comment=NULL, int magic=0, datetime expiration=0, color arrow_color=CLR_NONE ) #
```

Parameters:

symbol	- สัญลักษณ์ของคู่ค่าเงิน เช่น GBPUSD แต่หากเราใช้ symbol() จะเป็นการบอกให้ใช้คู่ที่เราเปิด EA อยู่
cmd	- ชนิดของการเปิด Order                   ศึกษาได้ที่ตารางข้างล่าง
volume	- lot ที่จะเปิด อันนี้ขึ้นอยู่กับแต่ละ Broker ครับว่ากำหนดขั้นต่ำเท่าไร
price	- ราคาที่จะใช้ในการเปิด Order
slippage	- จำนวนจุดที่มากที่สุดที่ให้เปิด order ถ้า EA ไม่สามารถเปิด order price ได้
stoploss	- ราคาที่ให้ทำการ stop loss
takeprofit	- ราคาที่ให้ทำการ Takeprofit
comment	- Order comment text. Last part of the comment may be changed by server.
magic	- ตัวนี้เหมือนหมายเลขพิเศษที่เราสร้างขึ้นมาครับ
expiration	- เวลาที่จะให้ Pending Order
arrow_color	- EA

ตารางค่า cmd ชนิดของการเปิด order ชนิดต่างๆ

Constant	Value	Description
OP_BUY	0	Buying position.
OP_SELL	1	Selling position.
OP_BUYLIMIT	2	Buy limit pending position.
OP_SELLLIMIT	3	Sell limit pending position.
OP_BUYSTOP	4	Buy stop pending position.
OP_SELLSTOP	5	Sell stop pending position.

## ตัวอย่างการใช้คำสั่ง OrderSend

```

if (kway==1)
{
    ticket=OrderSend(Symbol(),OP_BUY,Lots,Ask,3,Bid-Point*20,Ask+Point*10,"smaa",0,0,Green);
    OrderSelect(ticket,SELECT_BY_TICKET,MODE_TRADES);
}

if (kway==2)
{
    ticket=OrderSend(Symbol(),OP_SELL,Lots,Bid,3,Ask+Point*20,Bid-Point*10,"smaa",555,0,Red);
    OrderSelect(ticket,SELECT_BY_TICKET,MODE_TRADES);
}

```

## OrderClose () ใช้ในการปิด Order

**bool OrderClose(int ticket, double lots, double price, int slippage, color Color=CLR\_NONE)** # ตัวนี้ถ้าใช้ในรูปแบบนี้ไม่ต้องใช้ OrderSelect แต่หากเราเรียกใช้

OrderLots OrderTicket ก็ต้องประกาศใช้ OrderSelect ก่อนครับ

Parameters:

ticket	-	Id	Order
lots	-	Number of lots.	
price	-	Preferred closing price.	
slippage	-	จำนวนจุดที่ตลาดเคลื่อนไหวมากที่สุดในการปิด Order	
Color	-	สีของลูกศรในกราฟที่จะให้ EA แสดงตำแหน่งที่ปิด Order	

## OrderClosePrice() ใช้ในการแสดงราคาปิด

**double OrderClosePrice()** # ต้องใช้คู่กับ OrderSelect Mode\_History

## OrderCloseTime () ใช้เพื่อแสดงว่าปิด Order

**datetime OrderCloseTime()** # ต้องใช้คู่กับ OrderSelect Mode\_History

**OrderCommnet ()** ใช้เพื่อแสดง comment    **Order**

```
string OrderComment () #
```

**OrderCommission()** ใช้แสดงค่า Commission

```
double OrderCommission() #
```

**OrderDelete ()**                      **Pending Order**

```
bool OrderDelete (int ticket, color Color=CLR_NONE) #
```

Parameters:

ticket	-	Id	Order
Color	-	สีของลูกศรในกราฟที่จะให้ EA	

**OrderExpiration()** ใช้ในการกำหนดอายุของ **Pending Order**

```
datetime OrderExpiration () #
```

**OrderLots** ใช้การเรียกดูจำนวน **Lots**    **Order**

```
double OrderLots()#
```

**OrderMagicNumber()** ใช้ในการเรียกดูหมายเลข **MagicNumber** ที่เรากำหนดให้ **Order**

```
int OrderMagicNumber() #
```

**OrderModify()** ใช้ในการแก้ไขค่า **Stoploss** ค่า **Takeprofit**

```
bool OrderModify(int ticket, double price, double stoploss, double takeprofit, datetime expiration, color arrow_color=CLR_NONE) #
```

Parameters:

ticket	-	Id	Order
price	-	New open price of the pending order.	
stoploss	-	New StopLoss level.	
takeprofit	-	New TakeProfit level.	
expiration	-	Pending order expiration time.	
arrow_color	-	สีของลูกศรในกราฟที่จะให้ EA	

**OrderOpenPrice()** ใช้ในการเรียกดูราคาเปิดของ Order

`double OrderOpenPrice()` #

**OrderOpenTime()** ใช้เรียกดูเวลาที่เรทำการเปิด Order

`datetime OrderOpenTime()`#

**OrderPrint()** ใช้ในการสั่งให้ Ea Order

`void OrderPrint()`#

แสดงจะมีดังต่อไปนี้ครับ

ticket number; open time; trade operation; amount of lots; open price; Stop Loss; Take Profit;  
close time; close price; commission; swap; profit; comment; magic number; pending order  
expiration date.

**OrderProfit()** ใช้แสดง Profit ของแต่ละ order ครับค่าที่ได้เป็นค่า Profit อย่างเดียวครับ  
ไม่รวมค่า swap และค่าคอมมิชชั่น

`double OrderProfit()`#

**OrderStopLoss()** ใช้ในการแสดงค่า stoploss Order

`double OrderStopLoss ()`#

**Orderswap()** ใช้แสดงค่า swap

`double OrderSwap()`#

**OrderSymbol()** ใช้แสดง Pair ของ ค่าเงินครับเช่น GBPUSD

`string OrderSymbol()`#

**OrderTakeProfit ()** คล้ายกับ OrderProfit ครับแต่ตัวนี้รวมทุกอย่างแล้ว

`double OrderTakeProfit()`#

**OrderTicket()** ใช้แสดง ID Order

`int OrderTicket()`#

**OrderType** ใช้แสดงประเภทของ Order ที่เรทำการเปิดครับ

`int OrderType`

Order จะมีดังต่อไปนี้ครับ



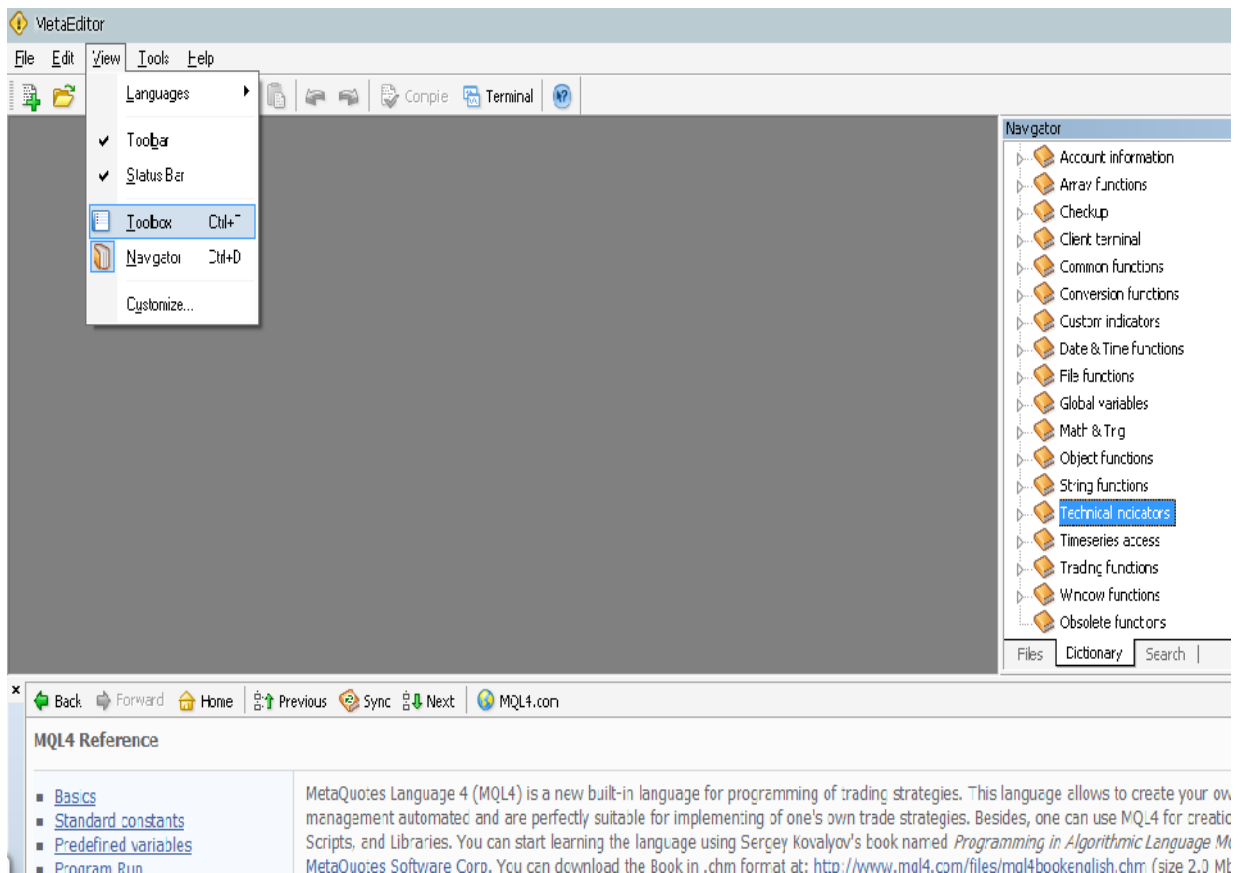
OP\_BUY - buying position,  
OP\_SELL - selling position,  
OP\_BUYLIMIT - buy limit pending position,  
OP\_BUYSTOP - buy stop pending position,  
OP\_SELLLIMIT - sell limit pending position,  
OP\_SELLSTOP - sell stop pending position.

6 นี้รู้สึกเหมือนแปล Text ใจไม่รู้ครับ หากอ่านตรงไหนแล้วไม่เข้าใจหรือมีข้อผิดพลาดประการใดขออภัย  
ล่วงหน้าครับ และสามารถศึกษาเพิ่มเติมได้ที่ [reference](#) MT4

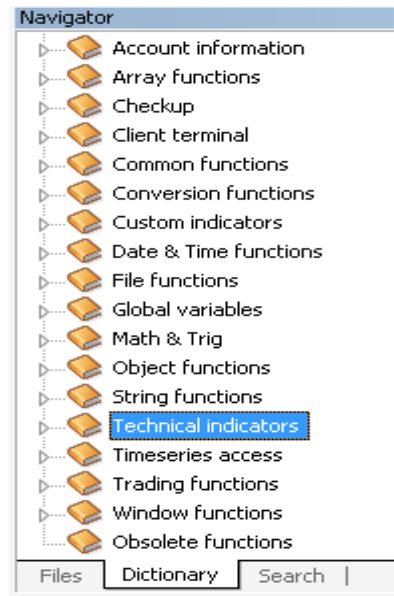
**Present by [www.thaiforex.com](http://www.thaiforex.com)**

## 7 การนำค่าของอินดิเคเตอร์มาใช้งาน

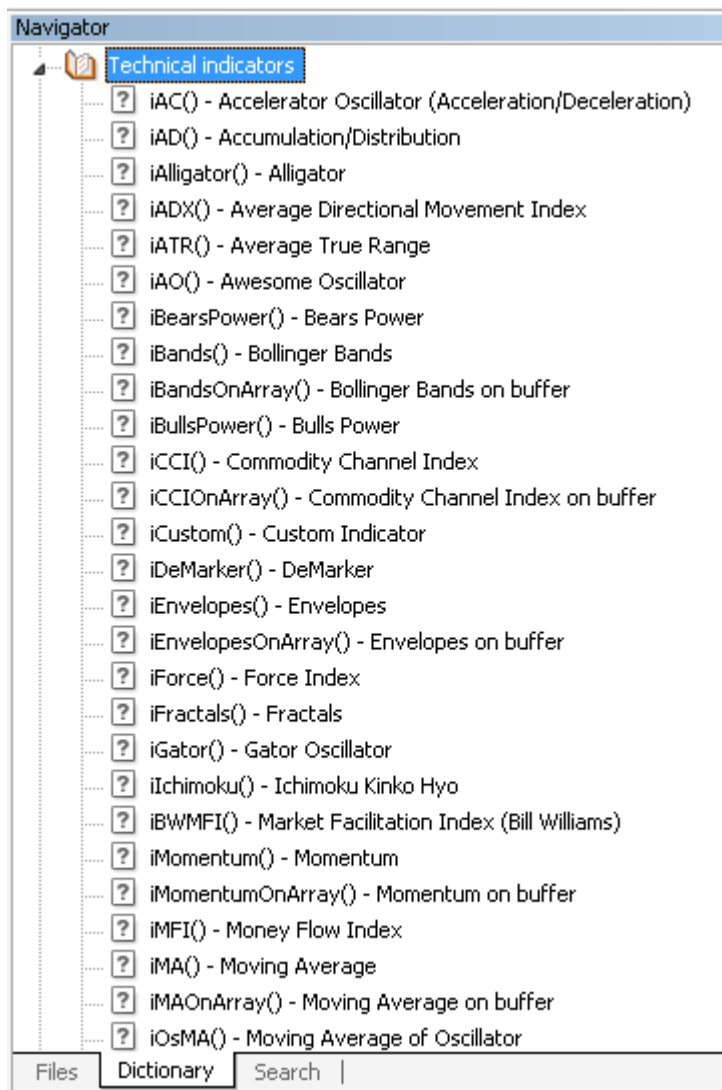
การนำค่าของอินดิเคเตอร์ออกมาใช้ใน EA นะครับจะไม่กล่าวรายละเอียดของฟังก์ชันในอินดิเคเตอร์แต่ละตัวนะครับเพราะสามารถศึกษาเองได้  
ก่อนอื่นเลยผมขอแนะนำให้ทุกคนรู้จัก **Reference Meta Editor** หลายคนที่ไม่เคยศึกษาการโปรแกรมมิ่งมาก่อนจะถามว่าอะไรคือ Reference Reference ก็คือคู่มือที่ใช้อ้างอิงวิธีการเขียนโปรแกรมครับ ทุกๆภาษาที่ต้องมี Reference ตัวอย่างเช่นชุดพัฒนาภาษาของ Microsoft visual studio Reference MSDN Java Java Reference Reference รายละเอียดของฟังก์ชันมาตรฐานทุกตัวที่ถูกพัฒนาขึ้นมาในชุดเขียนโปรแกรมคอมพิวเตอร์นั้นๆ เรียกใช้งานด้วยครับ เริ่มเลยนะครับให้เปิด Meta Editor



Tab view                      **Toolbox**    **Navigator**                      Tab Toolbox ด้านล่าง  
Navigator ทางด้านขวามือครับ โดยที่ Tab Toolbox จะเป็นตัวแสดงรายละเอียด Navigator  
สารบัญที่ให้เราเอาไว้เลือกดูรายละเอียดต่างๆครับ ต่อจากนี้ก็ขอให้ไปที่ Navigator แล้วที่ Tab ด้านล่างให้  
Dictionary                      นั่นก็ให้เลือก Technical indicators ครับตามภาพด้านล่างนี้เลยครับ



Technical indicators ก็จะแสดงรายชื่อของ อินดิเคเตอร์ที่ MQL4



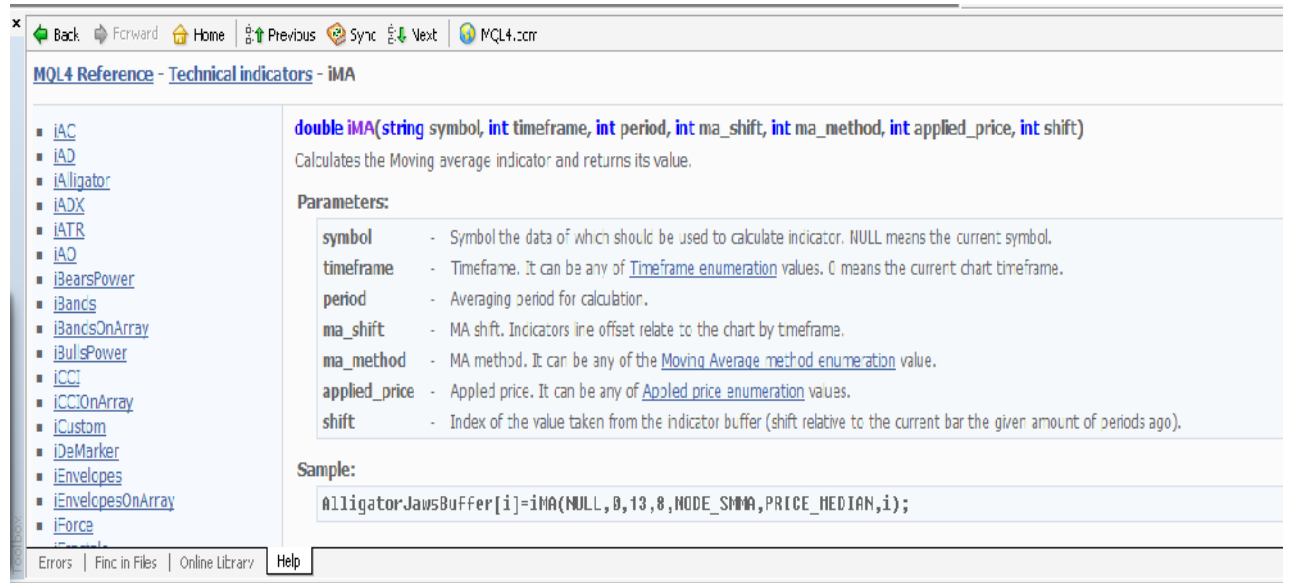
เราจะเห็นว่ามียาละเอียดของฟังก์ชันเยอะแยะเลยที่นำมาทำความเข้าใจกับสิ่งที่เห็นกันก่อนนะครับ

### ตัวอย่างที่เห็น iMA() – Moving Average หมายความว่าอย่างไร

iMA() อันนี้เป็นคำสั่งเรียกใช้งานฟังก์ชันอินดิเคเตอร์ Moving Average Navigator

ชื่อฟังก์ชันที่เรียกใช้งาน - จะบอกว่าฟังก์ชันนี้จะเรียกใช้งานอินดิเคเตอร์ตัวไหนนะครับ โดยทุกฟังก์ชันที่เห็นต้องมีการส่งค่าให้ฟังก์ชันทั้งนั้นเลยนะครับ ตรงนี้ผมขออนุญาตไม่ลงรายละเอียดนะครับ เพราะเพื่อนๆสามารถคลิกดูรายละเอียดที่ชื่อฟังก์ชันใน Navigator แล้วรายละเอียดทั้งหมดพร้อมตัวอย่าง

### Tab ToolBox



มีหมดทุกอย่างเลยครบทั้งรูปแบบ ฟังก์ชัน คำอธิบายค่าพารามิเตอร์ พร้อมทั้งตัวอย่างการนำไปใช้งานแต่ เป็นภาษาอังกฤษเท่านั้นเองครับ ตรงนี้อาจเป็นปัญหาของหลายคนแต่ถ้าไม่เข้าใจอะไรก็โพสต์ไว้ที่เวบบอร์ดก็ได้ครับ ผมเชื่อว่าต้องมีคนที่เค้าช่วยแปลให้ได้แน่ๆครับ ต่อไปจะเป็นตัวอย่างการสร้าง EA ที่นำค่าอินดิเคเตอร์มาทำการซื้อขายแล้วนะครับ ก่อนที่จะสร้าง EA ทุกตัวเราต้องมี idea ก่อนว่าจะให้มันทำงานยังไง ในตัวอย่างนี้ผมนำค่า MA มาใช้นะครับเริ่มจากง่ายก่อนเลย ขอออกตัวก่อนนะครับ Idea ผมอันนี้คิดขึ้นแค่เพื่อตัวอย่างการเขียนโปรแกรมเท่านั้นนะครับไม่รับรองว่าถ้าเอาไป Trade จริงๆแล้วจะทำกำไรได้นะครับ Idea นี้เพื่อฝึก

EA เท่านั้นนะครับ IDEA ก็คือ ถ้าราคาปัจจุบันมากกว่าเส้น ma 7 วันและเส้น ma 7 ากกว่าเส้น ma 14 วัน และเส้น ma 14 วันอยู่สูงกว่า เส้น MA 30 วันให้เปิด BUY Position  
ราคาอยู่ต่ำกว่าเส้น ma 7 ma 7 วันอยู่ต่ำกว่า 14 วัน และเส้น 14 วันอยู่ต่ำกว่าเส้น 30 วันให้ทำการเปิด SELL Position เขียนให้ดูเป็นสมการง่ายๆดังนี้ครับ

ราคาปัจจุบัน > ma 7 day > ma 14 day > ma 30 day หากเป็นจริงให้ทำการ Buy Position

ma 30 day > ma 14 day > ma 7 day > ราคาปัจจุบัน หากเป็นจริงให้ทำการ Sell Position

โดยมีเงื่อนไขเพิ่มเติมว่า จะ Take profit 50 Stop loss 500 และเปิดได้ทีละ Order

มาวิเคราะห์เงื่อนไขกันครับ จาก IDEA นี้เราต้องใช้อินดิเคเตอร์ 1 ชนิด แต่ต้องการอินดิเคเตอร์ 3 ค่า เพื่อนำมาเปรียบเทียบกับราคาปัจจุบัน เราทราบว่า EA นี้ต้องมีการตั้งค่า Stop loss Take Profit ต่  
เริ่มต้น มีการเปิดทั้ง buy sell Position และเปิดได้ครั้งละ 1 order เท่านั้น

EA กันเปิด Meta Editor new แล้วเลือก Expert Advisor  
ชื่อตามที่ต้องการลงไป แต่ผมให้ชื่อมันว่า firstea finish แกรมจะสร้าง ให้เราขึ้นมาให้ทุกคน  
ไปที่ฟังก์ชัน start() แล้วใส่ code ให้ copy

```
double ma7 , ma14 ,ma30 ; // ประกาศตัวแปรไว้รับค่าของ ma 3 ค่า
int total; //ประกาศตัวแปรเอาไว้รับค่าจำนวน order
int ticket; //ประกาศตัวแปรไว้เก็บค่า Id order
ma7=iMA(NULL,0,7,8,MODE_SMMA,PRICE_MEDIAN,0); //ให้ตัวแปรเก็บค่า ma 7
ma14=iMA(NULL,0,14,8,MODE_SMMA,PRICE_MEDIAN,0); //ให้ตัวแปรเก็บค่า ma14
ma30=iMA(NULL,0,30,8,MODE_SMMA,PRICE_MEDIAN,0); //ให้ตัวแปรเก็บค่า ma30

total=OrdersTotal(); // ใช้ OrdersTotal order ทั้งหมดแล้วเก็บไว้ในตัวแปร total

if(total<1) // order ว่ามี order เปิดอยู่หรือแล้วหรือเปล่าเพราะเราต้องการเปิดทีละ 1 เท่านั้น
{

    if(Ask>ma7>ma14>ma30) // BUY Position
    {

        ticket=OrderSend(Symbol(),OP_BUY,0.1,Ask,3,Bid-
Point*50,Ask+Point*50,"firstea",0,0,Green);
        OrderSelect(ticket,SELECT_BY_TICKET,MODE_TRADES);

// ทำการเปิด buy order stoplose 500 takeprofit 50

    }

    if(Bid<ma7<ma14<ma30)
    {

        ticket=OrderSend(Symbol(),OP_SELL,0.1,Bid,3,Ask+Point*50,Bid-
Point*50,"firstea",0,0,Red);
        OrderSelect(ticket,SELECT_BY_TICKET,MODE_TRADES);
// ทำการเปิด buy order stoplose 500 takeprofit 50
    }

}
```

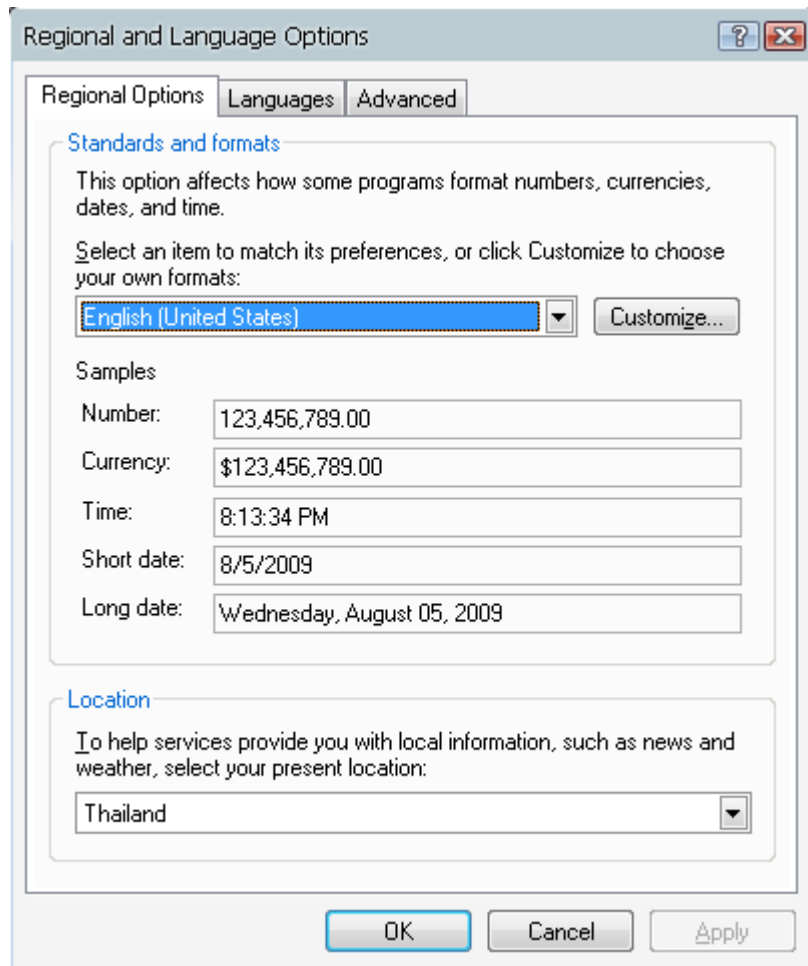
จากนั้นให้ทำการ Compile หากไม่มีไรผิดพลาดเราก็ Compile ผ่านไม่พบ error

EA ที่ได้ไปทดสอบ Backtest ก่อนหรือจะทำการทดสอบด้วย DEMO ก็แล้วแต่ เนื้อหาบทที่ 7 ก็จบลงแค่นี้ 8 จะเป็นการนำ EA backtest

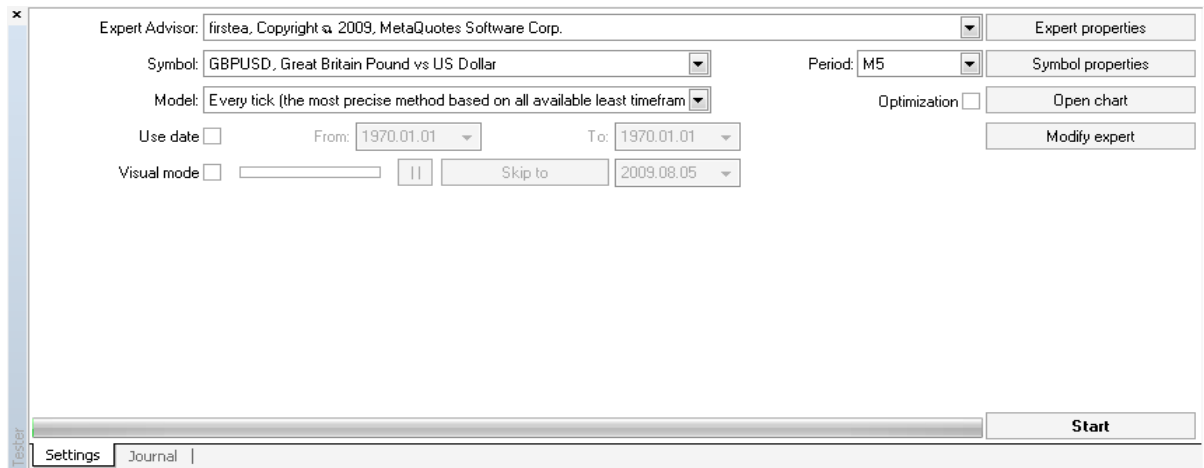
**Present by [www.thaiforexea.com](http://www.thaiforexea.com)**

## 8 BackTest EA

BackTest เป็นการทดสอบ EA ที่เราเขียนขึ้นมาด้วยข้อมูลราคาของ ค่าเงิน ย้อนหลัง ก่อนที่เราจะทำ BackTest ได้ขอให้ตรวจสอบคอมพิวเตอร์ของเราก่อนว่า รูปแบบการแสดงผลวันที่ของเราเป็น ไทยหรือไหม รูปแบบของไทยคือแสดงวันที่ เป็นปี พศ เนื่องจากระบบ BackTest MT4 ไม่เข้าใจปี พศ เราจึงจำเป็นต้องปรับเปลี่ยนรูปแบบการแสดงผลวันที่กลับไปปี คศ ก่อน โดยไปที่ Start -> Setting -> Control Panel -> Regional and Language Option จากนั้นเลือกให้เหมือนรูปด้านล่างแล้วคลิก Apply Ok



Restart เครื่องใหม่เพื่อให้ระบบแสดงวันที่เป็นปี พศ      restart เครื่องเรียบร้อยแล้วให้  
ทำการเปิด MT4      จากนั้นจากนั้นให้ทำการติดตั้ง EA ลงบนกราฟที่ต้องการจะทำการ BackTest  
F6      ปรากฏหน้าต่างดังรูปด้านล่างนี้



### รายละเอียดแต่ละช่อง

**Expert Advisor** -> ช่องนี้ให้เลือกชื่อย่อ EA

**Symbol** -> ช่องนี้ให้เลือก Pair ของค่าเงินที่จะทำการทดสอบ

**Period** -> ช่องระยะเวลาของกราฟ

**Model** -> รูปแบบที่ใช้ทดสอบ

**Use date** -> ใช้เลือกช่วง

**Visual mode** -> ใช้แสดงการทำงานในรูปแบบกราฟฟิคคือ โปรแกรมจะวาดกราฟแสดงในช่วงเวลาที่ทำการ

**Expert properties** -> อันนี้เอาไปปรับแต่งรูปแบบการทดสอบครับ สามารถปรับเปลี่ยนจำนวนเงินที่ใช้ทดสอบ รูปแบบการเปิด order และใช้ปรับเปลี่ยนค่าของ EA สามารถกำหนดค่าเริ่มต้นได้ครับ

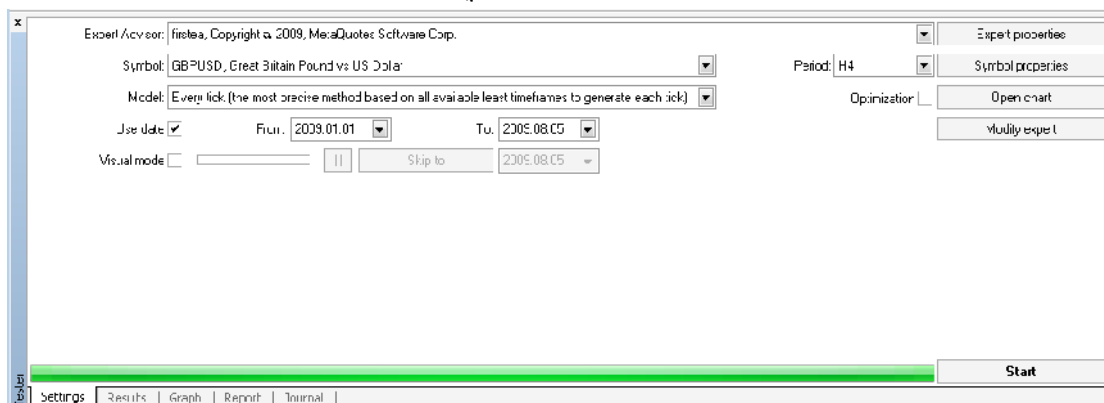
**Symbol properties** -> อันนี้จะแสดงข้อมูลของค่าเงินที่เราเลือกให้ทดสอบ โดยที่แต่ละ โบรเกอร์จะแสดงไม่

**Open chart** -> อันนี้ให้แสดงกราฟ

**Modify expert** -> อันนี้จะเอาไปเปิด EA เพื่อทำการแก้ไขครับ

**Start** -> .ใช้เริ่มทำ

### Run การทดสอบเสร็จจะปรากฏภาพแบบด้านล่างนี้



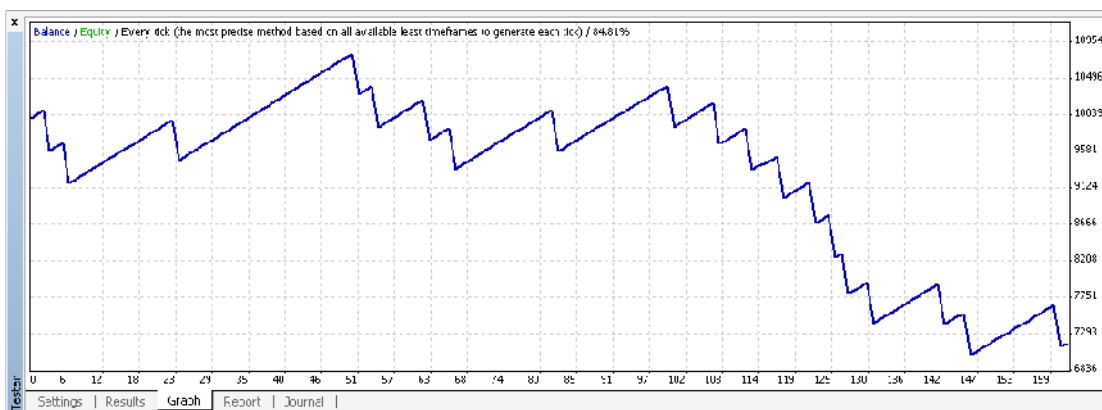
Tab 3 Tab Result , Graph , Report



#	Time	Type	Order	Size	Price	S/L	T/P	Profit
1	2009.01.05 00:05	sel	1	0.10	1.4534	1.5037	1.4484	
2	2009.01.05 02:59	buy	1	0.10	1.4484	1.5037	1.4484	50.00
3	2009.01.05 02:59	sel	2	0.10	1.4481	1.4984	1.4431	
4	2009.01.05 04:12	buy	2	0.10	1.4431	1.4984	1.4431	50.00
5	2009.01.05 04:12	sel	3	0.10	1.4427	1.4930	1.4377	
6	2009.01.06 19:59	buy	3	0.10	1.4930	1.4930	1.4377	-504.82
7	2009.01.06 19:59	sel	4	0.10	1.4927	1.5430	1.4377	
8	2009.01.06 20:50	buy	4	0.10	1.4877	1.5430	1.4377	50.00
9	2009.01.06 20:50	sel	5	0.10	1.4874	1.5377	1.4324	
10	2009.01.07 11:37	buy	5	0.10	1.4824	1.5377	1.4324	48.00
11	2009.01.07 11:37	sel	6	0.10	1.4821	1.5324	1.4771	
12	2009.01.08 15:33	buy	6	0.10	1.5324	1.5324	1.4771	-505.44
13	2009.01.08 15:30	sel	7	0.10	1.5322	1.5825	1.5272	
14	2009.01.08 16:51	buy	7	0.10	1.5272	1.5825	1.5272	50.00
15	2009.01.08 16:51	sel	8	0.10	1.5260	1.5771	1.5272	

Tab Results จะแสดงการเข้าเปิดและปิด Order

Account History



Tab Graph

Account ในช่วงที่ทำการทดสอบว่าจำนวนเงินทุน

เปลี่ยนแปลงอย่างไร

Items in test	1050	Ticks modeled	2134595	Modelling quality	04.01%
Microarchitect chaotic errors	4				
Initial deposit	10000.00				
Total net profit	-7834.54	Gross profit	6695.36	Gross loss	-3729.60
Profit factor	0.71	Expected payoff	-17.01		
Absolute drawdown	3029.32	Maximal drawdown	3855.32 (35.67%)	Relative drawdown	35.67% (3855.32)
Total trades	161	Winning positions (count %)	15 (86.20%)	Winning positions (count %)	0 (0.00%)
		Profit trades (% of total)	142 (88.20%)	Loss trades (% of total)	19 (11.80%)
	Largest profit trade		50.00	Loss trade	-524.64
	Average profit trade		43.55	Loss trade	-512.10
	Maximum consecutive wins (profit in money)		27 (1337.26)	consecutive losses (loss in money)	1 (-524.64)
	Maximal consecutive profit (count of wins)		1337.26 (27)	consecutive loss (count of losses)	-524.64 (-)
	Average consecutive wins		7	consecutive losses	1

Tab report จะเป็นการสรุปรวมการทำงานของ EA ที่เราใช้ทดสอบ

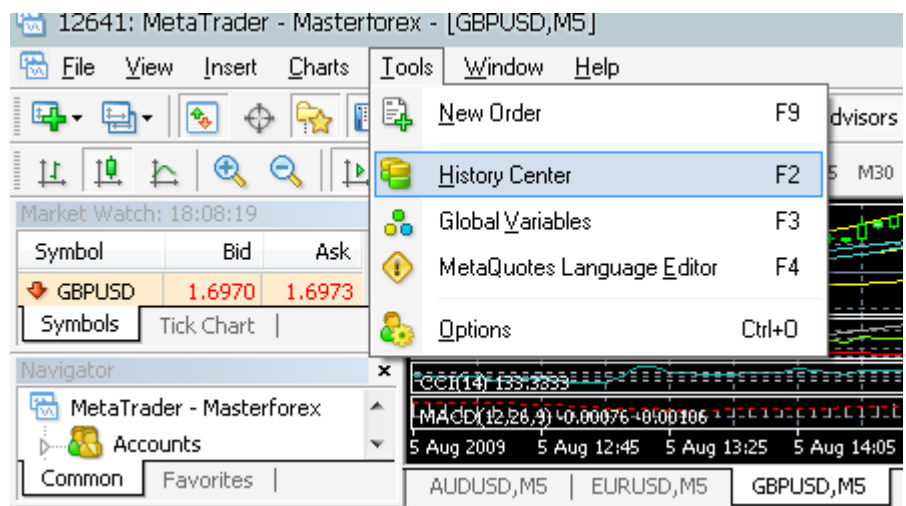
จากข้อมูลที่เราได้จากการ run

EA เราก็ต้องมา debug EA

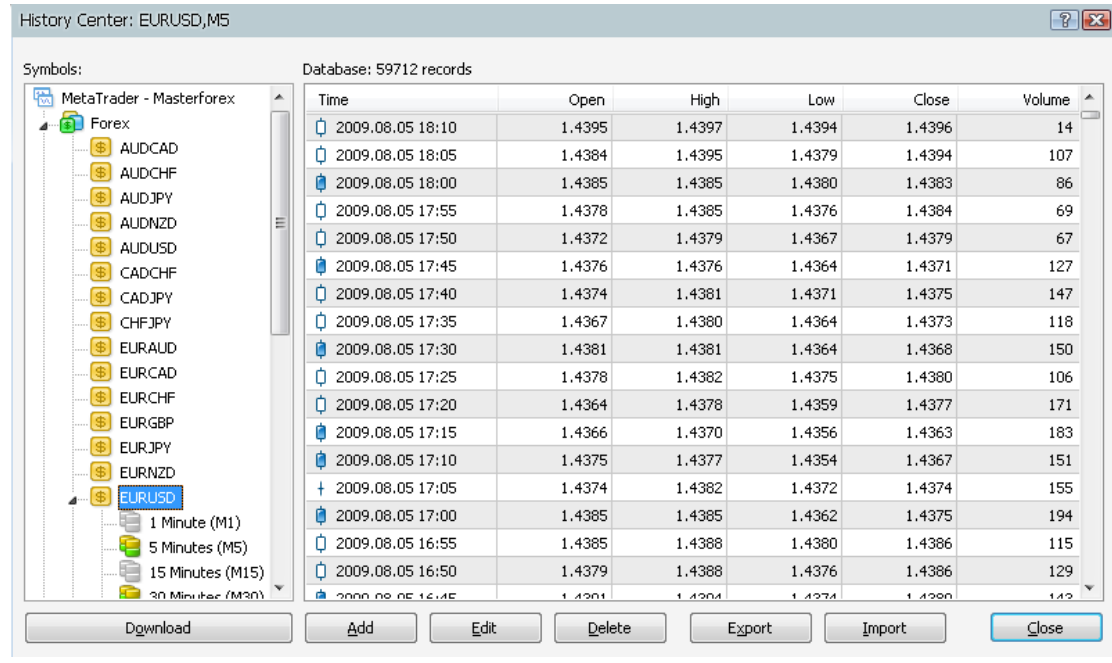
ทำงานที่ช่อง Results ว่าการเข้าเปิดและปิด Order ถูกต้องตรงตามตรรกะที่เราคิดเอาไว้หรือไม่ ถ้าไม่ ก็ต้องสังเกตดูว่ามีการทำงานผิดพลาดต้องไหนถ้าตรวจสอบพบ ก็ต้องกลับไปแก้ไข code จนกว่าจะทำงานได้ตามที่

เราคิดไว้ หากการทำงานทุกอย่างถูกต้องแต่ผลการทำกำไรไม่ดีเราก็ต้องกลับไปทบทวนถึง กลยุทธ์ในการทำการ  
 เข้าเปิดเข้าปิด order ใหม่จากนั้นก็ทำการเขียน code แล้วตรวจสอบการทำงานใหม่ทำอย่างนี้ไปเรื่อยๆ  
 จนกว่าค้นพบกลยุทธ์ที่เหมาะสมกับตัวเราเองครับ

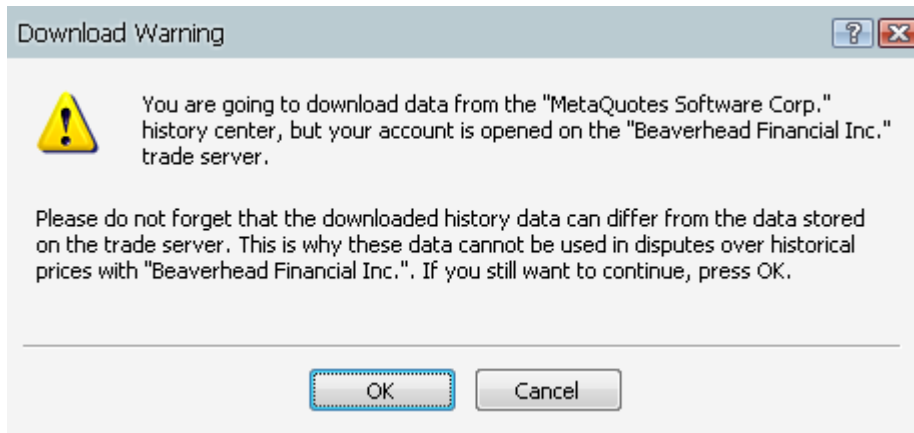
Download ข้อมูลกราฟเพื่อนใช้ในการทดสอบ EA MT4 หากเราไม่ทำ download  
 ข้อมูลย้อนหลังจะไม่สามารถ backtest ย้อนหลังได้ไกลเพราะข้อมูลของกราฟไม่มีเราต้องการทำการ  
 ทดสอบย้อนหลังหลายปีเราก็ต้องทำการ download ข้อมูลย้อนหลังก่อนโดยเริ่มที่ Tab tool  
 History Center ตามภาพด้านล่าง



จากนั้นจะปรากฏหน้าต่าง History Center ขึ้นมาดังรูปด้านล่าง



จากนั้นให้ทำการเลือก Pair ของค่าเงินและเลือชนิดของกราฟ จากนั้นคลิก Download  
 ด้านล่างนี้



ให้คลิก OK MT4 จะเริ่มทำการดาวน์โหลดข้อมูลมาจากบริษัทผู้ออกแบบโปรแกรม MT4 download  
ข้อมูลเสร็จให้คลิก close restart MT4 ขึ้นมาใหม่เราก็สามารถทดสอบ EA ช้อยหลังได้หลายปี  
เท่าที่ผมลองดู บางคู่ของค่าเงินได้ยาวสุด 11 ปีครับ ผมขอจบบทที่ 8 ลงแค่นี้ครับ

**Present by [www.thaiforexea.com](http://www.thaiforexea.com)**