MQL4 COURSE

By Coders’ guru

-9-

Preprocessors

Welcome to my last **theoretical** lesson in this series.
In the next series of lessons we will start to build our first Customer Indicator,
So I recommend you to read all the nine lessons carefully before the real work starts.

Now, let’s enjoy the Preprocessors:

What are the Preprocessors mean?

Preprocessors are the instructions you give to the compiler to carry them out before starting (processing) your code.

For example if you used the preprocessor directive **#include <win32.h>** that’s mean you telling the compiler to include the content of the file “**win32.h**” in the place you wrote the include keyword before processing your code.

In MQL4 there are four of preprocessors directives:

1- define directive:

define directive used to generate a constant.

The constant is very like the variable with only one different, you set its value only once and you can not change its value in your code like the variable.

For example:

```
#define my_constant        100
```

As you can notice in the above example there’s no assignment symbol (=) but only space between the constant name (**my_constant**) and its value (**100**).

And you can notice too that the line didn’t end with semi-colon but it ended with a carriage-return character (new line).

The name of constant obeys the same rules you had learnt about choosing the identifier names (lesson 2 SYNTAX), for example you can’t start the constant name with a number or exceeds 31 characters.

The value of the content can be any type you want.

The compiler will replace each occurrence of constant name in your source code with the corresponding value.

So you can use the above constant in your code like that:

```
sum = constant1 * 10;
```

2- property directive:

There are predefined constants called “**Controlling Compilation**” included in the MQL4 language, which you can set them in your program.

They are the properties of your program which you can set them using the compiler directive “**property**” and the compiler will write them in the settings of your executable program (ex4 file).

For example:

```
#property link        "http://www.forex-tsd.com"
#property copyright   "Anyone wants to use"
```

This is the list of the MQL4 predefined constants:

Constant	Type	Description
link	string	a link to the company website
copyright	string	the company name

stacksize	int	stack size
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	int	the bottom border for the chart
indicator_maximum	int	the top border for the chart
indicator_colorN	color	the color for displaying 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
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

3- include directive:

When you asking the compiler to include a file name with the “**include**” directive, it’s very like when you copy the entire file content and paste it in the place of the line you write include.

For example:

```
#include <win32.h>
```

In the above example you telling the compiler to open the file “**win32.h**” and reads all of its content and copy them in the same place of the include statement.

Note: in the above example you enclosed the file name with Angle brackets (<>) and that’s mean you telling the compiler to use the default directory (usually, terminal_directory\experts\include) to search for the file **win32.h** and don’t search the current directory.

If the file you want to include located at the same path of your code, you have to use quotes instead of angle brackets like this:

```
#include “mylib.h”
```

In the both cases if the file can’t be found you will get an error message.

You can use include at anywhere you want but it usually used at the beginning of the source code.

Tip: *It’s a good programming practice to write the frequently used code in a separate file and use include directive to put it in your code when you need (just an advice).*

4- import directive:

It’s like include directive in the aspect of using outside file in your program.

But there are differences between them.

You use **import** only with MQL4 executables files (.ex4) or library files (.dll) to import their functions to your program.

For example:

```
#import "user32.dll"
  int MessageBoxA(int hWnd,string lpText,string lpCaption,
                        int uType);
  int MessageBoxExA(int hWnd,string lpText,string lpCaption,
                          int uType,int wLanguageId);
#import "melib.ex4"
#import "gdi32.dll"
  int      GetDC(int hWnd);
  int      ReleaseDC(int hWnd,int hDC);
#import
```

When you import functions from “ex4” file you haven’t to declare their functions to be ready for use.

While importing the functions from a “.dll” file requires you to declare the functions you want to use like this:

```
int MessageBoxA(int hWnd,string lpText,string lpCaption,
                        int uType);
```

And only the functions you has declared you can use in your code.

You must end the import directives with a blank import line **#import** (without parameters).

I hope you enjoyed the lesson. And I hope you are ready now for your first Custom Indicator.

I welcome very much the questions and the suggestions.

See you
Coders' Guru
30-10-2005