Newbie's Guide to MQL4 Objects

By Steve Fleming

Automated Trading Software

FREE! You Now Own Resell and Giveaway Rights to This Report

Greetings!

By owning resell and giveaway rights, you may freely distribute this report to anyone you wish, resell it for any price and keep 100% of the profits, or just give it to your friends as a great gift. The choice is yours.

The only restriction is that you cannot modify this document in any way without permission from the author.

Enjoy!

Hot Tip: **If you would like to learn how to earn easy money from this report take a look at the last page for full details.**

Table of Contents

Introduction

"What is my motivation?"

After getting numerous emails asking me to explain in simple and straight forward terms what exactly MQL4 objects are and how to use them I decided to write a short eBook on the subject.

As with all my efforts to educate and help people this book will be aimed at complete novices to programming so there will be no complicated explanations or long words that basically don't really mean much anyway. I want EVERYBODY to be able to grasp the simplicity and power of programming in MQL4 so that they might then go on and start creating their own expert advisors and indicators for their own personal strategies.

If I can't explain it to you then I've not done my job properly and that is not something I intend to let happen.

Once you get some more experience and have built a few EA's yourself you can then branch out and start finding clients and coding their strategies too. You can actually make this a business if you just apply yourself and allow yourself to become successful.

For a little bit of motivation be sure to read this interesting article about what you need to be an MQL4 programmer. It's not as much as you might think! Do You Have the Right Mindset to Become an MQL4 Programmer?

My only wish is that you will learn something and be able to apply it in your live Forex trading and start making some money.

Good luck and enjoy the journey!

$$Your.Name$$

What the Heck are Objects?

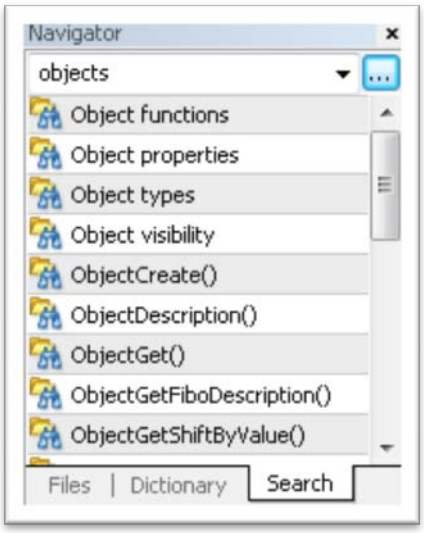
A simple explanation of MQL4 objects

MQL4 objects are nothing more than the graphical elements you see on the screen all the time. You've probably used objects a million times and didn't even realize you had.

When was the last time you drew a horizontal line on your chart? How about drawing a rectangle around a particular area of interest? Ever wrote some text on your chart using a label or text object? Well guess what my friend? You were using objects.

There are 24 different types of objects in MQL4 but don't worry about trying to remember them all because chances are you won't use 99% of them. For those rare occasions you do all you need to know is how to use the search feature in MetaEditor to look through the help file.

It usually sits over to the right of the screen but if it's not there hit 'Ctrl-D'.



Just enter 'objects' in the search box and read the results.

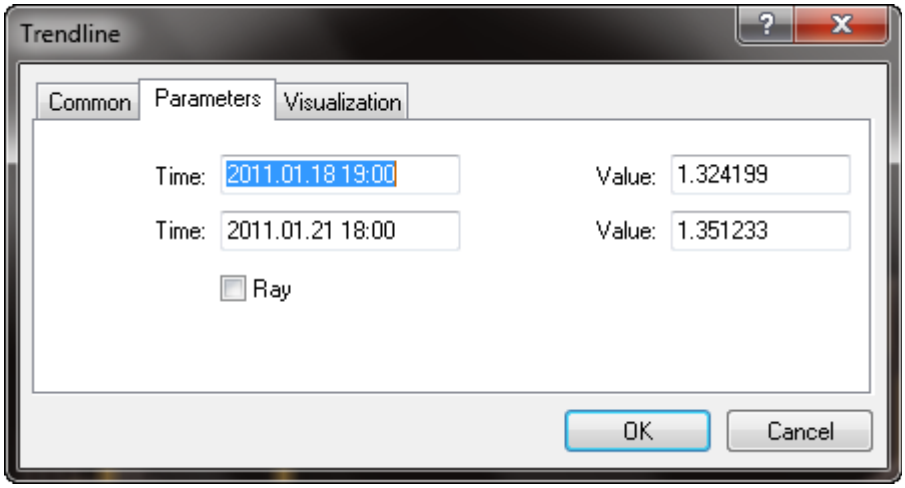
Changing the Appearance of an Object

Not all objects are created equal

If you want to do anything useful with objects you're going to have to know how to manipulate them. This is done by changing their 'properties'. Each object type will have different properties and again you don't have to remember them all. Just use the search help as I've demonstrated above.

You've actually already had experience with properties and didn't even know it. When you put that trend line on your chart and move its start and finish points to define a trend you're changing the time and price properties of the trend line object (OBJ_TREND). Well you're actually changing two time and price properties to be precise.

A really easy way to see this for yourself is to open up a chart and place a trend line on it. Then right click on the trend line object, select 'Trendline Properties' and click the 'Parameter' tab.



When you are changing the properties like this they are referred to as 'parameters' but when you are coding they are 'properties'.

Using Objects to Help Your Trading

A practical implementation of objects

Now you've had a quick taster of what objects actually are we're going to dive straight in and create a really useful indicator that utilizes objects. I actually wrote the first version of this indicator about 3 years ago and I've used it many times since in my trading and find it invaluable.

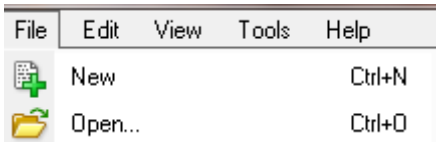
We're going to create a 'Support and Resistance Alert' that will allow you to have two lines on your chart which you can move around to identify an impending level of support and resistance. When either of these levels is breached, either by current price or close of a bar, you will receive an audible alert.

You can set alerts manually using MT4 but as I'm sure you'll agree that's a real pain in the butt. Having this indicator will make it so much easier for you believe me.

Let's Get Started

This step should be simple for you if you've ever looked at MetaEditor but just in case you've never looked at it here's how to do it.

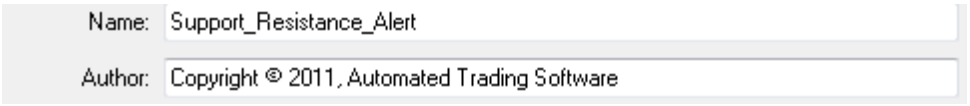
From MT4 hit the 'F4' key and you'll see MetaEditor. Then click File -> New…



You'll then be presented with a wizard that will walk you through the process of creating a blank template for our indicator.

On the first screen select the 'Custom Indicator' button.

Then click 'Next'. Give your indicator a unique name. It can be anything but for this project call it 'Support_Resistance_Alert'.



You can fill out the other fields or leave them as their defaults it doesn't matter. Click 'Next' again and then 'Finish'.

Your indicator template has now been created and we're ready to move on to the next step.

Setting Our External Parameters

Changing Settings Easily

You've no doubt used many EA's and indicators and whenever you want to change their settings you just call up the relevant 'Settings' box and change a few numbers right? Well that's exactly what we're going to do here. We are going to add code that will allow the user of the indicator to change settings easily.

I won't be explaining in too much detail the actual code because I don't want to confuse things and you really don't need to know that right now. If you're interested you can always check out the help file on a particular constant (word) or function.

At the top of your template code type in the following…

```
#property copyright "Steve Fleming"
#property link      "http://www.automatedtradingsoftware.com"

extern bool    AlarmOn                = true; // alarm on or off?
extern bool    SignalOnClose          = false; // if false we will use Bid price to signal piercing of S/R levels
extern string  ResistanceLineName     = "R_Alert"; // name of SR line that will trigger Alarm
extern string  SupportLineName        = "S_Alert"; // name of SR line that will trigger Alarm
extern string  SoundFile              = "alert2.wav"; // name of sound file
extern int     DelayBetweenAlarm      = 10; //
extern color   SRLineClr              = Lime;  // color of SR lines
extern color   SRLineClr2             = Red;  // color of SR lines
extern int     SRLineWidth            = 2;  // width of SR lines
```

It looks like a lot but it's actually not. The words on each line after the '//' are just comments which you don't really need to type but I just left them here so you can see what each parameter is intended for.

Initialize Our Indicator

Getting the Indicator Setup

The 'init' function is called once when the indicator is placed on the chart. It will also be called if you change time frames. It's where we put code that we only need to run once and it's usually where we put code that will basically setup the indicator or EA.

Next we're going to type some code that will tell the indicator to draw our two line objects on the screen. First it checks if they already exist and if not we create them. Again if you want to know more try putting your cursor over a particular function like 'ObjectCreate' and hitting 'F1'.

If you happen to delete one of the lines just change time frames and the init function will be called again which will re-draw them.

```
int init()
  {
//---- indicators
    if(ObjectFind(ResistanceLineName) == -1){
      ObjectCreate(ResistanceLineName, OBJ_HLINE, 0, 0,iHigh(NULL, 0, 0));
    }
   ObjectSet(ResistanceLineName, OBJPROP_COLOR, SRLineClr);
   ObjectSet(ResistanceLineName, OBJPROP_WIDTH, SRLineWidth);

    if(ObjectFind(SupportLineName) == -1){
      ObjectCreate(SupportLineName, OBJ_HLINE, 0, 0,iLow(NULL, 0, 0));
    }
   ObjectSet(SupportLineName, OBJPROP_COLOR, SRLineClr2);
   ObjectSet(SupportLineName, OBJPROP_WIDTH, SRLineWidth);
```

The Start Function

Where it all happens

The start function is called on every single tick that is received from your broker. The best way to think of it is like a switchboard. Every time a tick is received the start function will execute and any code it contains will likewise be executed. Therefore it's where you'll place the 'heart' of your code. Think of the switchboard operator in one of those old movies. A call comes in and she figures out where to route it.

From here any other code is executed via function calls. Again the details are beyond the scope of this eBook but there's plenty of information available if you're interested to learn more.

```
int start()
  {
//----
   if(AlarmOn){
      double PreviousClosePrice = Bid;
      if(SignalOnClose == true){PreviousClosePrice = iClose(NULL, 0, 1);}
      double ResistancePrice = ObjectGet(ResistanceLineName, OBJPROP_PRICE1);
      double SupportPrice = ObjectGet(SupportLineName, OBJPROP_PRICE1);
      if(PreviousClosePrice > ResistancePrice && ResistancePrice != 0)Alerts(1);
      if(PreviousClosePrice < SupportPrice && SupportPrice != 0)Alerts(2);
   }
//----
   return(0);
  }
//+------------------------------------------------------------------+
```

In this simple indicator we only have one other function. It's a custom function that we will write to extend the functionality of the MQL4 built in function 'Alert'. Below the closing curly brace of your start function type the following.

```
void Alerts(int Direction)
{
   static datetime AlertLastTime = 0;

   if(TimeCurrent() > AlertLastTime ) {
      AlertLastTime = TimeCurrent()+DelayBetweenAlarm;
      PlaySound(SoundFile);
   }
}
```

Don't Forget to Clean Up!

Being environmentally friendly

Now the last thing to do is to make sure your code cleans itself after the user removes it from the screen. Many so-called 'programmers' forget this simple but necessary step; make sure you're not one of them!

Seriously though it's such a simple thing to do I don't know why they don't do it. I guess it's just lack of consideration or laziness. Try and get into the habit early of always cleaning up after yourself; it's good coding practice.

Clean up code goes in the deinit function. Similar to the init function this is called only once at the end of the indicators run just before it is permanently deleted from the chart. When a user decides to remove an indicator or EA this is the code that is ran.

Type the following…

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

   if(UninitializeReason() != REASON_PARAMETERS && UninitializeReason() != REASON_CHARTCHANGE){
      ObjectDelete(ResistanceLineName);
      ObjectDelete(SupportLineName);
   }
//----
   return(0);
  }
```

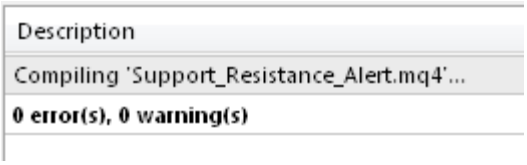
The only thing to note here is that we are going to ensure that no clean up gets done if the user is only changing parameters or time frames. There's nothing more annoying than keep having to reset the lines every time you decide to change time frames.

Enjoy the Fruits of Your Labor

Use your brand new indicator for the 1st time

Once you've got all that code typed in the only thing left to do is compile it and start using your new indicator.

Hit 'F5' and the code will compile. If you've done everything correctly you should see the following message in the 'Errors' tab at the bottom of the MetaEditor screen.



Congratulations! You've created your very first indicator in MQL4. You should be really proud of yourself.

To use the indicator just go back to MetaTrader and drag the new indicator onto your chart. You should see two lines on the screen. Just drag them to where you want to be alerted the next time price rises above (Resistance) or below (Support).

Well I hope you enjoyed that brief introduction to MQL4 objects and I wish you luck on your journey to becoming a competent MQL4 programmer.

Good luck!

Steve Fleming

Pay it forward and Make Some Money along the Way!

I hope you enjoyed this free report and if you're feeling inspired and want to learn all the ins and outs of MQL4 then pick up your copy of my MQL4 Training Video Package

Before you rush off to share this report with your Facebook friends and Twitter followers, let me ask you a question...

Would you like to get paid a commission when the people you share this report with also buy MQL4 Training Video Package

If so, then all you have to do is register as an affiliate of mine and then rebrand this special report with your affiliate ID before sharing it with others. That's it. You can then post it to your blog, share it on Facebook and Twitter, or even use it to build your list.

And here's the best part...

Your readers will also receive "giveaway rights" to this report. Which means, if they give it away ***without*** rebranding it *(and many will)* you'll get paid commissions on ALL of their referred sales as well. Example: Let's say "Stephanie" downloads your report, but ***does not*** rebrand it with her ID. She then shares YOUR branded version with Bill who buys my MQL4 Training Video Package – guess what? **YOU** just got paid... C*ha-ching!*

Here's what to do next...

Step 1: Download FREE rebranding software from this link

Step 2: Using the software follow the instructions to rebrand the file.

Take Your MQL4 to the Next Level by <<CLICKING HERE>>

Step 3: Start sharing your new report with others and **GET PAID**!