

## Перенос программ с MQL II на MQL 4.



Александр Иванов (AKA HORN)  
<http://fxtrade.tomsk.ru>

В форумах по программированию для MetaTrader'a иногда появляются недовольные реплики о несовместимости нового языка MQL 4 с языком MQL II, используемым в ранних версиях этой популярной торговой платформы. Многие из пользующихся MetaTrader'ом имеют набор любимых пользовательских индикаторов написанных на MQL II и, естественно, могут быть огорчены тем фактом, что эти индикаторы не будут работать в новой версии торгово-информационной системы MetaTrader 4. Но, оказывается, не всё так драматично, как может показаться. Если под рукой имеется исходный код индикатора на MQL II, то перенос индикаторов на новую платформу не займёт слишком много времени, и может быть выполнен даже не искушённым в программировании человеком.

В этой статье мы постараемся дать советы, которые могут облегчить процесс переноса "старых" индикаторов на новую платформу.

Первое, что хотелось бы отметить - это изменившийся набор функций языка MQL 4.

Естественно, что те изменения, которые претерпел язык MQL, направлены, на создание более удобного инструментария и среды для написания программ. В таком случае было бы логично ожидать от разработчиков только расширения набора функций и других возможностей языка, а не наоборот. Это значит, что любая функция языка MQL II дублируется в версии языка MQL 4.

Новая версия языка MQL поддерживает больший набор функций и, чтобы не запутаться в этом изобилии, названия функций, за редким исключением, сами помогают понять, к какой группе относится та или иная функция. Например, названия всех функций для работы с массивами начинаются с префикса `Array`, а названия математических функций - с префикса `Math`. В связи с этим не все функции в новой версии языка имеют в точности то же название, что и в старых программах, но почти с полной уверенностью можно сказать, что их "двойник" существует и в новой версии.

Второй момент, на который необходимо обратить внимание - это то, что программы советников и индикаторов теперь разделены на три обязательные функции: `init()`, `start()` и `deinit()`. Мы уже писали об этом в одном из номеров Forex Magazine (Forex Magazine, 7 Июня 2004, №20 "Программирование на MQL 4: Введение.").

Большинство индикаторов содержит цикл, в котором на каждом баре вычисляется значение индикатора. Собственно этот цикл как раз и является тем местом, в котором считается индикатор, а то, что идёт перед этим циклом и иногда после него - является кодом, "обслуживающим" его корректную работу. При переносе программы на MQL 4, весь этот цикл, скорее всего, без дополнительных изменений нужно поместить в тело функции `start()`.

Инициализация переменных, которая происходит перед выполнением цикла, может быть помещена в функцию `init()`.

Входные параметры индикатора, которые объявлялись в MQL II с ключевым словом `inputs`, должны быть превращены в параметры функции `init()`.

Функция `deinit()`, скорее всего, останется пустой, так как программы на MQL II зачастую не содержат никаких действий по устранению последствий своей работы.

Третье, что требуется сделать при переносе программы на новую версию языка MQL, - это привести её синтаксис в соответствие с требованиями MQL 4. Этот процесс попробуем описать пошагово:

### Шаг 1:

Все операторные скобки `"begin"` и `"end"` заменить на `"{"` и `"}"`, соответственно. Все, кто и без того использовал при написании программ на MQL II фигурные скобки в качестве операторных скобок, сами уже избавили себя от лишних усилий на этом этапе переноса программы на новый язык.

Далее нужно удалить все вхождения в будущую программу на MQL 4 служебного слова языка MQL II - слова "then".

## Шаг 2:

Меняем все логические операторы "and", "or" и "not" на "&&", "||" и "!" соответственно. Если кто-то использовал в программах на MQL II операторы "&" и "|", то не забудьте и их тоже превратить в "&&" и "||", соответственно.

## Шаг 3:

Особенность оператора "=" в MQL II заключалась в том, что в тех местах, где результатом его работы ожидалось логическое значение, выполнялось сравнение значений, а не присваивание переменной стоящей в левой части значения из правой части. То есть в таких ситуациях оператор "=" играл роль оператора "==". Это часто встречалось в операторах "If" и "While". В MQL 4, дабы исключить неоднозначность, разработчики чётко разделили эти два оператора, и теперь, если требуется сравнить два значения, следует использовать оператор "==".

Это замечание очень важно, и его не следует игнорировать ни в коем случае. Иначе новый вариант индикатора будет преподносить вам неожиданные "сюрпризы" всякий раз, когда встретит вместо сравнения двух переменных их присваивание.

## Шаг 4:

Раньше при написании программ на MQL II можно было не задумываться о том, как писать название переменной или оператора - используя заглавные буквы или строчные? Компилятору было всё равно, будет ли в программе написано "While" или "while". Программисты сами решали в соответствии со своими предпочтениями оформления кода, как им писать не только названия переменных, но и служебные слова языка.

В новой версии языка MQL эта ситуация изменилась, и теперь, если написать "If", то компилятор попытается интерпретировать это как название переменной, а не как начало оператора выбора, и, не найдя объявления такой переменной, выдаст ошибку. Поэтому,

следует обратить на это внимание, и при переносе программ на MQL 4 проследить, чтобы в написании операторов "if", "while", "for" и прочих не использовались заглавные буквы. Так же следует обратить внимание на то, что регистр букв играет роль и в объявлениях переменных и функций. То есть, если вы объявили переменную, как "Var1", то к ней нельзя будет обратиться по имени "var1" или "VAR1". Компилятор будет воспринимать каждое из трёх имён, как имена разных переменных.

Учитывая всё вышесказанное, следующий, корректный с точки зрения MQL II, код:

```
Variable: Var1(0);
Variables: Var2(0.), Var3(0.);

// ...
// выполнять какие-то действия
// ...

If((Var1 < shift)And
   ((VAR2 = 0)Or (VAR2 = 0)))
then Begin
    // выполнять какие-то действия
End;
```

должен будет выглядеть так:

```
int Var1 = 0;
double Var2 = 0;
double Var3 = 0;

// ...
// выполнять какие-то действия
// ...

if((Var1 < shift) &&
   ((Var2 == 0)|| (Var2 == 0)))
{
    // выполнять какие-то действия
};
```

## Шаг 5:

Пятым шагом можно было бы предложить "разнести" переменные по тем блокам, где они используются, и тем самым разделить их объявления на глобальные и локальные, относительно области видимости функций. Но это уже один из шагов, который не придаст программе новой функциональности, а лишь улучшит её читаемость и, быть может, увеличит скорость её работы. На первых же этапах можно обойтись и без этого шага, просто помещая объявления всех переменных в глобальную область видимости, то есть объявляя их сразу перед объявлением первой функции в файле исходного кода программы.

Если все вышеперечисленные действия были аккуратно выполнены, то индикатор должен скомпилироваться и начать работать.

Конечно, всего не предугадать, и при переносе индикаторов на MQL 4 будут существовать отдельные моменты, которые потребуют индивидуального подхода, но, в целом, процедура переноса индикаторов с MQL II на MQL 4 должна соответствовать процедуре, описанной в статье.

Для примера переноса читателю предлагается сравнить версии индикатора "Медианное сглаживание", написанного на MQL II и на MQL 4, и опубликованного в 15-м и в 20-м выпусках журнала Forex Magazine.

*Александр Иванов  
для Forex Magazine*