

**Название советника:** QLT

**Год выпуска:** 2016

**Версия:** 1.51

**Терминал:** MT4 build 1010

**Сайт:** <http://forum.tradelikeapro.ru/index.php?topic=14413.0>

**Валютные пары:** EURUSD, GBPUSD, EURJPY, AUDUSD, EURGBP, AUDJPY, USDCAD

**Таймфрейм:** M1 (рекомендуется 5-тизнак, хотя может работать и на 4-х знаке)

**Время торговли:** Круглосуточно

**Использование:** Поместить советник на график M1 нужной пары и загрузить соответствующий пресет. **Настройки по умолчанию не использовать!** Наличие индикатора Quantum не требуется.

**ММ:** \$5000 на начальные 0.01 лота.

Советник основан на индикаторе Quantum. Индикатор показывает минимумы и максимумы за определенное количество свечей. Торговая стратегия по данному индикатору подразумевает открытие ордеров против текущего движения, в расчете на откат/разворот.

Советник строит сетки buy/sell независимо друг от друга. Советник имеет собственный счетчик ордеров. Он не зависит от того, удалось открыть реальный ордер или нет, сработала отложка или нет, и отражает то количество, которое должно было быть по стандартному алгоритму. Все расчеты, связанные с количеством ордеров, используют именно этот счетчик. Значение счетчика выводится в окно графика вместе с другими данными. Количество реально открытых ордеров ни в каких расчетах не участвует.

## FAQ.

**Q.** Как увеличить объем открываемых ордеров?

**A.** Если коэффициент умножения лота (Klots) задать равным 2, то объем открываемых ордеров увеличится в 2 раза, если Klots=3, то в 3 раза и т.д.

**Q.** Как сделать, чтобы объем открываемых ордеров увеличивался по мере роста депозита?

**A.** Для этого нужно установить Klots=0 и задать размер свободной маржи на 0.01 лота (MarginForLot). Например, если MarginForLot=3000, то, когда размер депозита увеличится до 6000, объем ордеров увеличится в 2 раза.

**Q.** Что за строчки в логах «Count (SELL) reset (time)» и т.п.?

**A.** Это сброс счетчиков.

Count (SELL) reset (time). Настройка NumPriceTime - от начала отсчета сигналов прошло больше положенного времени.

Count (SELL) reset (size). Настройки NumBoxes и NumPoints - цена прошла небольшое расстояние, но уже выдала кучу сигналов.

Count (SELL) reset (size2). Настройка NumBoxes2 - сигнал на каждой новой свече и таких свечей уже больше положенного.

Count (SELL) reset (numqdc2). Настройка Numqdc2 - сигнал от quantum(qdc2).

▼ EURUSD,M1 1.12814 1.12814 1.12790 1.12795

BUY Orders: 0 OtlLots: 0.00 w: 0 (8) virt: 0 K: 10

SELL Orders: 0 OtlLots: 0.00 w: 0 (8) virt: 0 K: 10

ProfitPoints: 2 PointLoss: 0 (21)

Коэффициент  
умножения лота

Количество  
виртуальных  
ордеров

Объем не  
сработавших  
отпожек

Количество  
пунктов профита,  
требующихся для  
закрытия сетки в  
плюс

Количество  
пропущенных сигналов  
(в скобках значение из  
настроек)

Количество пунктов  
допустимого убытка при  
закрытии сетки (в скобках  
количество ордеров при  
превышении которого  
действует данный  
параметр)

Количество сигналов по  
которым должны были  
открыться ордера. Из-за  
использования отпожек не  
всегда равно количеству  
реально открытых ордеров

16 Aug 2016 16 Aug 09:26 16 Aug 09:58 16 Aug 10:30 16 Aug 11:02 16 Aug 11:34 16 Aug 12:06 16 Aug 12:38 16 Aug 13:10 16 Aug 13:42 16 Aug 14:14 16 Aug 14:46 16 Aug 15:18

AUDUSD,M1 0.72197 0.72197 0.72197 0.72197

BUY Orders: 4 OtiLots: 0.00 w: 10 (10) virt: 0 K: 1

SELL Orders: 0 OtiLots: 0.00 w: 0 (10) virt: 0 K: 1

ProfitPoints: 1 PointLoss: 0 (18)

Start new grid: ON

#1 buy 0.01

#2 buy 0.01

#3 buy 0.01

4 Jan 02:16 4 Jan 02:24 4 Jan 02:32 4 Jan 02:40 4 Jan 02:48 4 Jan 02:56 4 Jan 03:04 4 Jan 03:12 4 Jan 03:20 4 Jan 03:28 4 Jan 03:36 4 Jan 03:44

Визуальный уровень, при пересечении которого ценой, появится сигнал на закрытие сетки

До какой свечи рассчитывается уровень закрытия (QDC)

Уровень, на котором установлен отложенный ордер

```

enum tframe
{
    M1   = PERIOD_M1,
    M5   = PERIOD_M5,
    M15  = PERIOD_M15,
    M30  = PERIOD_M30,
    H1   = PERIOD_H1,
    H4   = PERIOD_H4,
    D1   = PERIOD_D1
};

input int      mn          = 2016; // Magic Number
extern string  cmnt        = "";   // Комментарий
extern int     Slip        = 5;    // Проскальзывание
extern tframe  tf          = M1;   // Используемый таймфрейм

extern bool    NoNewGrid   = false; // Не начинать новую сетку
extern bool    OneGrid     = false; // Строить одну сетку за день (время брокера)

input string   s00         = "";    // -----
input string   s01         = "";    // Начинать сетки только в указанный период
extern bool    UseTimeSession = false; // Использовать период1 (время брокера)
extern int     StartHour     = 1;    // Старт1 (часы)
extern int     StartMinute   = 0;    // Старт1 (минуты)
extern int     StopHour      = 14;   // Стоп1 (часы)
extern int     StopMinute    = 0;    // Стоп1 (минуты)
extern bool    UseTimeSession2 = false; // Использовать период2 (время брокера)
extern int     StartHour2    = 17;   // Старт2 (часы)
extern int     StartMinute2  = 0;    // Старт2 (минуты)
extern int     StopHour2     = 21;   // Стоп2 (часы)
extern int     StopMinute2   = 0;    // Стоп2 (минуты)
input string   s02         = "";    // -----

extern int     Wait         = 0;    // Пропустить N боксов перед открытием сетки

input string   s0          = "";    // Обнулить счетчики если...
extern int     NumBoxes     = 8;    // 1a ...количество сигналов больше и...
extern int     NumPoints    = 8;    // 1b ...цена прошла меньше (пункты)
extern int     NumPriceTime = 180;  // 2 ...или прошло больше времени от начала отсчета (минуты)
extern int     NumBoxes2    = 5;    // 3 ...или количество сигналов равно количеству свечей (>)
extern int     Numqdc2      = 600;  // 4 ...есть сигнал QDC2

input string   s1          = "";    // -----
extern bool    OnlyLowerHigher = true; // Открывать только выше/ниже (для sell/buy) чем предыдущий ордер
extern int     NumOrders     = 8;    // Количество ордеров для которых применяется параметр
extern int     HighLowPoints = 1;    // Количество пунктов между ордерами
extern int     NumOrders2    = 8;    // Количество ордеров для которых применяется предыдущий параметр
extern bool    HL12         = false; // Только в тестере. Верхнее количество ордеров равно нижнему
input string   s2          = "";    // -----

extern int     qde          = 240;   // Quantum eintDepth3 for Entry (QDE)
extern bool    qdcqdc      = false;  // Только в тестере. QDC равно QDE
extern int     qdc         = 240;   // Quantum eintDepth3 for Close (QDC)

input string   s3          = "";    // -----
extern bool    CloseOnlyPlus = true;  // Закрывать сетку только в плюс
extern int     ProfitPoints  = 1;    // Количество пунктов профита
extern int     PointLoss     = 12;   // Допустимый убыток при закрытии (пункты), менее
extern int     NumOrdPointLoss = 50;  // Кол-во ордеров (>), с которых применяется предыдущий параметр
extern int     DistanceForPL = 1000;  // Расстояние от противоположного сигнала до начала сетки (пункты)
input string   s4          = "";    // -----
extern int     ProfitPoints3 = 20;   // Количество пунктов профита (закрытие без QDC)
input string   s41         = "";    // -----

```

```

extern bool   UseTrStop      = false; // Использовать TrailingStop для профита
extern int    TrailingStop0  = 2;     // Начальный уровень (пункты)
extern int    TrailingStop   = 2;     // TrailingStop (пункты)
extern int    TrailingStopStep = 1;    // Шаг TrailingStop (пункты)
extern int    ProfitPoints2  = 4;     // Размер профита, с которого включается трал (пункты)
input string  s5             = "";    // -----

extern bool   UseLaguerre    = false; // Использовать индикатор Laguerre
extern double L_Gamma        = 0.7;   // Gamma
extern int    L_CountBars    = 100;   // Количество свечей для расчета
extern double L2_Gamma       = 0.5;   // Gamma2
extern int    L2_CountBars   = 100;   // Количество свечей2 для расчета
extern int    L_Profit       = 14;    // Не использовать индикатор, если профит больше (пункты)
extern int    L_Orders       = 16;    // Не использовать индикатор, если кол-во ордеров больше
input string  s51            = "";    // -----

extern int    Delta          = 1;     // Расстояние от текущей цены до отложенного ордера (пункты)
extern int    TimeSec        = 210;   // Время жизни отложенного ордера (сек.)
extern int    NumOtlAgr      = 0;     // Количество ордеров (>), с которых начнется накопление объема
extern bool   UseOtlTrailing = false; // Использовать трал для отложенных ордеров
extern int    DeltaTr        = 3;     // Расстояние трала отложенных ордеров (пункты)
extern int    NumOtl         = 10;    // Количество ордеров (>), с которых включать трал
input string  s6             = "";    // -----

extern double Klots          = 1;     // Коэффициент умножения лота (0 - автомат)
extern int    MarginForLot   = 10000; // Размер свободной маржи на 0.01 лота (при Klots = 0)
extern int    CloseAllOrdersDrowdown = 10000; // Только в тестере. Размер просадки для закрытия всех ордеров
input string  s7             = "";    // -----
input double  T1             = 0.01;  // Размер лота1
extern int    kol1           = 12;    // Количество ордеров1
input double  T2             = 0.02;  // Размер лота2
extern int    kol2           = 9;     // Количество ордеров2
input double  T3             = 0.04;  // Размер лота3
extern int    kol3           = 8;     // Количество ордеров3
input double  T4             = 0.08;  // Размер лота4
extern int    kol4           = 7;     // Количество ордеров4
input double  T5             = 0.16;  // Размер лота5
extern int    kol5           = 3;     // Количество ордеров5
input double  T6             = 0.25;  // Размер лота6
extern int    kol6           = 70;    // Количество ордеров6

input string  s72            = "";    // -----
extern bool   UseTestStat    = false; // Только в тестере. Выводить в лог некоторую статистику
extern bool   UseTestStat2   = false; // Только в тестере. Выводить в лог дополнительную статистику

input string  s8             = "";    // -----
extern color  font_color     = clrSilver; // Цвет шрифта
extern int    FontSize       = 10;     // Размер шрифта
enum ugol
{
    topleft    = 0, // верхний левый
    topright   = 1, // верхний правый
    bottomleft  = 2, // нижний левый
    bottomright = 3  // нижний правый
};
extern ugol   corner         = 0;     // Угол окна для вывода информации
extern int    xx             = 10;    // Отступ по горизонтали
extern int    yy             = 20;    // Отступ по вертикали

```

## Описание логики работы и настроек советника.

При появлении сигнала от Quantum начинается отсчет. Если не было сброса счетчика и было пропущено нужное количество сигналов (все в соответствии с настройками), то берется или рассчитывается коэф. умножения лота и открывается рыночный ордер. При появлении новых сигналов выставляются отложенные ордера. Если у новой отложки цена лучше чем у имеющихся, то они закрываются. Новый ордер выставляется с учетом объема не сработавших.

Получив сигнал индикатора (**qde**), советник начинает отсчет этих сигналов. Он пропускает количество сигналов, указанных в настройках (**Wait**), и открывает рыночный ордер. Если в настройках указано пропустить 7 сигналов, то ордер откроется после появления 8-го сигнала.

Счетчик обнуляется в следующих случаях:

1. Было уже **NumBoxes** сигналов, а цена прошла меньше **NumPoints** пунктов. Например, в настройках установлено 6 и 10. Значит при появлении 7-го сигнала, если цена прошла меньше 10 пунктов, счетчик будет обнулен.

Бывает вялотекущее движение цены в одну сторону. При этом прошли не много, а счетчик уже накрутили. Для таких ситуаций и сделана эта настройка.

2. Если с начала отсчета прошло больше **NumPriceTime** минут.

При длительном флэте цена может выдавать новые максимумы/минимумы (скажем 2-3 сигнала) и опять уходить в границы флэта. Если такие ситуации не фильтровать, то к моменту, когда начнется движение, счетчик уже почти израсходуется и сетка начнет строиться раньше, чем надо, т.е. в самом начале активного движения, а не когда движение уже будет затихать.

3. Когда количество сигналов равно количеству свечей (**NumBoxes2**). Если в настройках указано 5, то при появлении 6-го сигнала, счетчик будет обнулен.

Когда начинается активное движение, то сигналы могут появляться на каждой новой свече. Если на этих сигналах открывать сделки, то они сразу пойдут в минус. Лучше такое движение пропустить.

4. При появлении противоположного сигнала.

Например, мы считаем сигналы для открытия сетки в sell. Нужного количества так и не набрали, цена пошла вниз и появился сигнал на buy. Очевидно, что ситуация на рынке изменилась и лучше начать отсчет заново.

5. При появлении сигнала QDC2 (**Numqdc2**).

Цена может и не дойти до появления противоположного сигнала и в то же время пройти приличное расстояние. Лучше тоже сбросить счетчик.

6. При изменении таймфрейма в настройках советника.

При появлении новых сигналов, советник будет выставлять отложенные ордера (стоповые). Отложки виртуальные, т.к. не все брокеры позволяют поставить отложку на оптимальном расстоянии. Если цена безоткатно идет дальше, то отложка не срабатывает, а значит не будет увеличивать просадку. Хотя отложки и съедают часть прибыли, но не редко позволяют избежать слива.

Ставятся отложки на определенном расстоянии (**Delta**) от текущей цены: для buystop - от цены ask, для sellstop - от цены bid.

По прошествии определенного времени (**TimeSec**), не сработавшая отложка удаляется, а ее объем запоминается. Следующая отложка будет выставлена с учетом этого объема.

Если есть активные (не удаленные) отложки, то при выставлении новой проверяется, не будет ли цена этой отложки лучше, чем у активных. Если да, то они закрываются, а новая выставляется с учетом их объемов.

Имеется возможность отключить накапливание объема. Для этого надо указать количество ордеров, с которых начинать накопление (**NumOtlAgr**). Для полного отключения накопления нужно просто поставить большое число. Данная настройка положительно влияет на большие

сетки, но для средних и маленьких больше вреда, чем пользы, т.к. режет прибыль. По умолчанию 0, т.е. накопление работает всегда.

Отложки можно тралить (**UseOtlTrailing**). Правда для маленькой сетки пользы от этого нет. Поэтому в настройках указывается количество ордеров (**NumOtl**), при превышении которого включается трал. Также указывается расстояние трала до цены в пунктах (**DeltaTr**).

Когда начинает строиться сетка, то может случиться ситуация, при которой появляется много сигналов, а цена при этом проходит не так много и с небольшими откатами. Когда же начинается основное движение, то остается целая пачка ордеров, собранных в кучу. Чтобы такого не происходило, есть следующие настройки.

Открывать только выше/ниже (для sell/buy) чем предыдущий ордер (**OnlyLowerHigher**). При включении данной настройки, сигналы, не удовлетворяющие условиям, будут проигнорированы. Т.к. эффективно это только для начала сетки, то указывается количество ордеров, до которого используется этот параметр (**NumOrders**).

В самом начале сетки имеет смысл не только открывать ордера выше/ниже, но и указать побольше расстояние между ними (**HighLowPoints**). Соответственно, указывается и количество ордеров, для которых это действует (**NumOrders2**).

Например, в настройках указаны параметры 19, 2, 5. Значит, первые 5 ордеров будут выставляться на расстоянии не менее 2 пунктов между собой, далее просто выше (или ниже) друг друга, пока количество ордеров в сетке не превысит 19, а дальше уже просто по сигналам, не зависимо от того где в данный момент оказалась цена.

Рано или поздно цена начинает двигаться в нужную сторону и появляется сигнал от индикатора (**qdc**), что надо закрывать сетку. Можно закрывать и в минус, но, на мой взгляд, лучше все-таки еще подождать и закрыть в плюс (пусть даже и маленький). Для этого есть соответствующая настройка (**CloseOnlyPlus**). Также указывается минимальное количество пунктов профита (**ProfitPoints**) при котором будет закрытие сетки, т.е. если на момент появления сигнала, размер профита не менее указанного, то сетка закроется.

Когда сетка вырастает выше среднего размера, лучше ее закрыть с меньшим профитом или даже с некоторым убытком, чем дожидаться когда она вырастет и сожрет весь депозит. Для этого можно указать допустимый убыток (**PointLoss**) и количество ордеров (**NumOrdPointLoss**), при превышении которого и будет учтена настройка. Если убыток указать со знаком минус, получится соответствующий профит. Таким образом, маленькие сетки можно закрывать, например, в +3 пункта, а большие в +1.

Если от противоположного сигнала индикатора до старта сетки цена прошла большое расстояние, то вероятность заметного отката возрастает. В этом случае можно выключить использование **PointLoss**. Для этого нужно задать расстояние в параметре **DistanceForPL**. Если цена прошла больше указанного здесь значения, то **PointLoss** не будет использоваться для текущей сетки.

Бывают такие движения цены (в основном на новостях), когда она сходила в одну сторону, выдала сигналы на открытие сетки, а затем быстренько пошла в другую сторону. Сетка при этом может выйти в приличный плюс, но сигнала на закрытие еще нет. Когда же сигнал все-таки появится, то профит может быть уже не таким большим. Также бывает, что на откате сетка вышла уже в приличный плюс, но до появления сигнала так и не дошла. Чтобы не упускать такие варианты, можно указать размер профита (**ProfitPoints3**), при котором будет закрыта сетка даже без появления сигнала. Проверка осуществляется на каждой новой свече.

Вместо немедленного закрытия сетки, можно попытаться тралить профит, включив соответствующий блок (**UseTrStop**). Здесь есть следующие настройки.

Начальный уровень (**TrailingStop0**). На этом расстоянии от текущей цены будет выставлен уровень стоплоса (виртуальный).

**TrailingStop**. На это расстояние от текущей цены будет подтягиваться стоплос.



Шаг TrailingStop (**TrailingStopStep**). Стоплос будет переставлен, если цена пройдет больше указанного здесь количества пунктов.

Если сетка имеет маленький плюс, то использование трала может привести к закрытию в минус. Чтобы такого не было, есть настройка (**ProfitPoints2**), в которой указывается минимальный размер профита, при котором будет использоваться трал. Т.е., если профит был меньше, то трал не включится и произойдет обычное закрытие сетки.

Кроме трала профита, вместо немедленного закрытия сетки можно использовать индикатор Laguerre (**UseLaguerre**). Используются два индикатора: первый основной с настройками **L\_Gamma** и **L\_CountBars** (количество свечей для индикатора), второй вспомогательный - **L2\_Gamma** и **L2\_CountBars**. Индикатор не будет использоваться, если профит больше указанного **L\_Profit** (пункты) или количество ордеров больше **L\_Orders**.

**Предупреждение!** Использование индикатора в советнике иногда приводит к тому, что некоторые сетки увеличиваются в размерах.

Рассмотрим работу на примере закрытия сетки BUY.

На каждой новой свече рассчитывается и сохраняется значение индикатора.

Если первый индикатор падает или размер сетки превышает количество указанное в настройках, то активация использования индикатора не происходит.

Работа с индикатором разбита на несколько шагов:

0. Активация.

1. Ждем когда дойдем до максимума.

2. Ждем нужного профита.

3. Закрытие сетки.

На каждом этапе есть определенные фильтры, срабатывание которых приводит к переходу на следующий шаг или перескакиванию на другой шаг.

Иногда возникали ситуации, при которых маленькая незакрытая вовремя сетка вырастала в большую. Размер маленькой сетки не имеет значение, т.к. это могли быть и сетки из одного ордера. Все что у нас есть для предотвращения роста - это значения индикатора, размер профита и размер сетки. Отсюда и появились используемые фильтры.

L1 - текущее значение индикатора

L2 - предыдущее значение

L3 - более раннее значение

То же самое для второго индикатора L2\_1, L2\_2, L2\_3.

**Шаг 0.**

При активации запоминается размер профита (в пунктах) и количество ордеров сетки.

Переход на шаг 3, т.е. закрытие сетки произойдет, если выполнится любое из следующих условий:

Rule 1 ( $L3 \leq 1$   $L2 = 1$   $L1 = 1$ )

Rule 2 ( $L2\_3 \leq 1$   $L2\_2 = 1$   $L2\_1 = 1$ )

Rule 3 ( $L2\_3 > L2\_2 > L2\_1$ )

Rule 4 ( $L2\_3 = 1$   $L2\_2 = 1$   $L2\_1 < 1$ )

Rule 5 ( $L3 > L2 < L1$   $L2\_3 > L2\_2 < L2\_1$  ( $L2\_2 = 0$  or ( $L2\_3 = 1$   $L2\_1 = 1$ )))

Переход на шаг 2 будет при выполнении любого из условий:

Rule 1 ( $L1 = 1$ )

Rule 2 ( $L2\_3 < L2\_2 \leq 1$   $L2\_1 = 1$ )

В остальных случаях переход на шаг 1.

**Шаг 1.**

Переход на шаг 2 будет при выполнении любого из условий:

Rule 1 ( $L1 = 1$ )

Rule 2 ( $L2\_1 = 1$ )

Переход на шаг 3 будет при выполнении условия:



$L1 < L2$  StartProfit < CurrentProfit индикатор стал падать, но текущий профит больше начального

В остальных случаях ничего не делаем.

### Шаг 2.

Переход на шаг 3 будет при выполнении любого из условий:

Rule 1 (StartProfit <= CurrentProfit)

Rule 2 ( $2 * \text{StartProfit} / \text{CurrentOrders} * \text{StartOrders} \leq \text{CurrentProfit}$ ) если сетка выросла в размерах, то даже меньшее количество пунктов принесет больше прибыли

В остальных случаях ничего не делаем.

### Шаг 3.

Здесь фильтров нет. Просто закрываем сетку.

Т.к. советник строит сетку ордеров, то в настройках можно указать объем ордеров (**T1, T2, T3, T4, T5, T6**) и их количество (**kol1, kol2, kol3, kol4, kol5, kol6**). Если количество указать ноль, то ордера соответствующего объема открываться не будут. Рекомендуется объем первого ордера (**T1**) указать 0.01, а увеличение делать с помощью коэффициента умножения лота (**Klots**). Таким образом, у вас будет минимально возможный размер ордеров (с соответствующей просадкой и доходностью), который можно легко масштабировать на любой размер депозита всего одним коэффициентом.

Если **Klots=0**, то будет использоваться автоматический расчет коэффициента. При этом используется размер свободной маржи на 0.01 лота (**MarginForLot**). Коэффициент округляется до целого числа в меньшую сторону.

Работу советника можно ограничить по времени, указав один или два интервала, в течение которых, можно стартовать сетки (время брокера). Для этого надо включить **UseTimeSession** и указать время начала (**StartHour, StartMinute**) и завершения (**StopHour, StopMinute**). То же самое для второго: **UseTimeSession2, StartHour2, StartMinute2, StopHour2, StopMinute2**. Советник будет начинать сетки только в указанное время. По окончании интервала, новые сетки открываться не будут, открытые будут продолжать строиться и закрываться в соответствии с настройками.

Можно строить только одну сетку за день (**OneGrid**). Учитывается по дате закрытия, поэтому возможна ситуация, что при открытой сетке может открыться противоположная.

Если вы хотите, чтобы советник вообще не открывал новые сетки, то включите настройку **NoNewGrid**. До ее выключения новых сеток не будет.

При закрытии сетки, ордера сортируются по объему и закрываются с самого большого. Профит/убыток на сортировку не влияют, поэтому ордера одинакового объема могут закрываться в произвольном порядке.

## Настройки, используемые только в тестере.

**HL12.** При включении настройки, NumOrders = NumOrders2.

**qdcqde.** При включении настройки, qdc = qde.

**CloseAllOrdersDrowdown.** Если текущий минус и маржа превысят указанное в настройке значение, то все ордера будут закрыты. Проверка происходит на новой свече. Следует иметь ввиду, что CloseAllOrdersDrowdown умножается на коэффициент умножения лота.

**UseTestStat.** (При проведении оптимизации, рекомендуется выключать для экономии времени). Позволяет посмотреть нагрузку на депо для каждой отдельной сетки, а также посмотреть максимальную нагрузку за весь период. При закрытии сетки, выводится строка вида:

GridStat1;SELL;2011.05.02 01:26:25;2;0.01;2.97;337082.31;-1.06

Здесь: идентификатор статистики; тип сетки; дата и время максимальной просадки; количество ордеров; объем открытых ордеров; максимальный размер использованной маржи; минимальный уровень маржи (в процентах); максимальная просадка

В конце лога выводится общая статистика за весь период:

MaxBuyOrdres: 45 DateTime: 2012.12.07 15:42

MaxBuyDrawdown: -1094.63 DateTime: 2012.12.07 15:41

MaxSellOrdres: 47 DateTime: 2011.10.27 21:07

MaxSellDrawdown: -1094.64 DateTime: 2011.10.27 21:06

MaxMargin: 1040.70 DateTime: 2011.10.27 21:07

MinMarginFree: 9829.97 DateTime: 2011.10.27 21:08

MinMarginLevel: 1044.55% DateTime: 2011.10.27 21:08

При одинаковых значениях, информация будет только о первом событии.

Статистика собирается с интервалом в 2 секунды и не претендует на абсолютную точность, но вполне достаточна для оценки размера задействованных средств. Размер и уровень маржи берутся из данных аккаунта, поэтому при наличии противоположных сеток, здесь будет общий результат. Для извлечения данных статистики из лога, прилагается готовый командный файл gridstat1.bat. Запускать так: gridstat1.bat 20161028.log. В результате будет сформирован файл gridstat1.csv, который можно открыть в Excel.

**UseTestStat2.** (При проведении оптимизации, рекомендуется выключать для экономии времени). Позволяет посмотреть в динамике нагрузку на депо. На каждой новой свече выводится информация вида:

GridStat2;2011.05.02 01:32;2.97;-0.68

GridStat2;2011.05.02 01:33;2.97;-0.43

GridStat2;2011.05.02 01:34;2.97;0.13

GridStat2;2011.05.02 01:35;2.97;0.77

Здесь: идентификатор статистики; дата и время свечи; текущая задействованная маржа; текущий профит

Время округляется до минуты, маржа и профит являются общими для противоположных сеток. Если маржа и профит равны нулю, то данные не выводятся. Для извлечения данных статистики из лога, прилагается готовый командный файл gridstat2.bat. Запускать так: gridstat2.bat 20161028.log. В результате будет сформирован файл gridstat2.csv, который можно открыть в Excel.

По событию OnTester() выводится фактор восстановления, который рассчитывается по формуле PROFIT/ EQUITY.