

Инструкция по программированию торговых стратегий в WealthLab

 **АВТОР** ДМИТРИЙ ВЛАСОВ  **МАРТА** - 12 - 2012  **КОММЕНТАРИЕВ:** 19

Уважаемые читатели блога "Финансовая лаборатория". Сегодня я хочу начать выкладывать для Вас инструкцию по программированию торговых стратегий в программе WealthLab. На текущий момент мне не удалось найти эту инструкцию на русском языке, поэтому я решил перевести ее самостоятельно. Сразу скажу, для чего это делаю:

Во-первых, в последнее время я усиленно изучаю английский язык. А учить язык всегда легче, если применяешь его для того, что тебе действительно интересно. В моем случае, это без сомнения трейдинг и торговые стратегии.

Во-вторых, я изучаю еще один язык - язык программирования C#. А весь код для программирования стратегий в программе ВелсЛаб как раз приходится писать на этом языке программирования.

В-третьих, надеюсь эта инструкция будет полезной для Вас - ведь многие хотят строить и тестировать собственные торговые стратегии, но не владеют английским языком.

Хочу сразу предупредить, что скорее всего это будет не дословный перевод. Некоторые предложения я буду пытаться построить так, чтобы их можно было читать и понимать проще всего.

Если увидите какие-либо ошибки или Вам что-то покажется не совсем ясно - пишите об этом в комментариях - с удовольствием поправлю.

Введение:

Как выполнить код, приведенный в примерах к инструкции по программированию торговых стратегий на языке WealthScript

Глава 1: Объекты "БАРЫ" в программе Wealth-Lab

1. **Объект Bars в Wealth Lab - что такое бары в Велс Лаб**
2. **OHLC/V Series - последовательности цен при программировании в Wealth Lab 6.3**
3. **Как использовать при программировании стратегий шкалу, интервал и дату**
4. **Symbol Info - подробная информация о торговом инструменте в программе WealthLab**

Глава 2: Объекты "DataSeries" - ряды данных в Wealth-Lab

1. **DataSeries - основная структура данных в Wealth-Lab и математические операции при программировании торговых стратегий**
2. **Как получить доступ к единичному значению DataSeries**
3. **Наполнение созданных самостоятельно DataSeries данными при помощи WealthScript**
4. **Как получить доступ ко второму финансовому инструменту при проектировании торговых стратегий в Велс Лаб.**
5. **Синхронизация данных при построении торговых стратегий с несколькими финансовыми инструментами**

Глава 3: Рисование на графиках Wealth-Lab

1. **Построение графиков в программе Wealth-Lab при помощи WealthScript**
2. **Отображение на графике синтетических финансовых инструментов**
3. **Отображение на графике в программе Велс Лаб рядов данных (индикаторов, осцилляторов)**

RSS товары



RSS новости



Twitter



Последние комментарии

макс: А где взять свечки с открытым интересом? на сайте финама нет



ring10: Добрый день Дмитрий Как сделать, чтобы открывался 1 велф лаб? Если открываешь под пользователем, т [...]



Дмитрий Власов: Буян, рад, что Вам пригодились материалы данной статьи!



Максим: Сам себе отвечу. protected override void Execute() { Bars rts = GetExternalSymbol("RTS", t [...]



Буян: Спасибо Дмитрий за статью! Мне удалось разработать хорошую рабочую торговую систему, используя возмо [...]



Последние посты



Бесплатный вебинар: 7 смертных грехов при оптимизации торговых систем

НОЯБРЯ - 5 - 2012



Отзывы участников курса до результата Wealth-Lab. С нуля до первой торговой системы за 5

дней.
НОЯБРЯ - 3 - 2012



Быстрее, Выше, Сильнее: Обновление Wealth-Lab до новой версии 6.4

ОКТАБРЯ - 23 - 2012



Курс до результата: Wealth-Lab – от нуля до первой торговой системы за 5 дней

ОКТАБРЯ - 23 - 2012



Обзорная экскурсия по достопримечательностям программы Wealth-Lab

ОКТАБРЯ - 20 - 2012

Берите

4. Написание текста на графиках Wealth-Lab программными методами

5. Рисование графических объектов на графике Wealth-Lab

Глава 4: Использование индикаторов в программе Wealth-Lab

1. Проектирование и построение индикаторов в Wealth-Lab

2. Делаем индикаторы в Wealth-Lab своими руками

3. Стабильность индикаторов в программе Wealth-Lab в зависимости от ширины тестируемого окна.

Глава 5: Программирование торговых стратегий в Wealth-Lab

1. Программирование торговых стратегий в программе Wealth-Lab

2. Виды торговых сигналов в Wealth-Lab

3. Организация торгового цикла в программе Wealth-Lab и его переделка в скрининговую систему

4. Программирование однопозиционных стратегий (SP Strategies) в программе Wealth-Lab

5. Использование при программировании торговых стратегий торговых приказов AtStop и AtLimit

6. Программирование выходов, основанных на времени удержания позиции в Wealth-Lab

7. Ловушки при построении торговых систем, связанные с проблемой подглядывания в будущее

8. Использование обоснованного заглядывания в будущее при программировании торговых стратегий в Велс Лаб

9. Создание параметров торговой стратегии в Wealth-Lab

10. Создание многопозиционных стратегий в Wealth-Lab

11. Что такое алерты и как их использовать в Wealth-Lab

12. Анализ данных в Wealth-Lab с учетом разных таймфреймов

Ну чтож, начнем. Сегодня первая часть инструкции по программированию в Велс Лаб:

Введение в инструкцию по программированию торговых стратегий в программе Wealht-Lab.

Всю инструкцию по программированию Вы сможете найти [в этой рубрике](#) . Чтобы не пропустить новые части этой инструкции - не забудьте [подписаться на обновления блога](#) "Финансовая лаборатория".



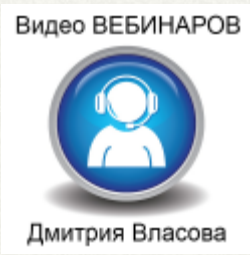
Добро пожаловать в инструкцию по WealthScript программированию!

Назначение инструкции по WealthScript программированию заключается в том, чтобы снабдить Вас базовыми (и некоторыми более сложными) концепциями, которые помогут Вам описать Ваши торговые стратегии, используя язык WealthScript, являющийся частью программы WealthLab.

WealthScript – это библиотека классов для построения графиков и торговых функций в пространстве имен {WealthLab}.NET.

Вы будете удивлены, каких успехов сможете достигнуть программируя торговые системы с помощью WealthScript!

Эта инструкция предполагает, что Вы уже знакомы с .NET программированием, но даже если это не так, следуя приведенным примерам, Вы без труда разберетесь с ее содержанием. Мы настоятельно рекомендуем Вам самостоятельно выполнить те фрагменты кода, которые здесь приведены ([Как выполнить приведенный код](#) Вы узнаете в следующей части).



МТС под микроскопом:
WLMA System

Подробный разбор торговой системы WLMA System
3900 руб.

[В корзину](#)

Рубрики

Выберите рубрику

Обновления:

- Новые посты по Email
- Новые посты по RSS
- Комментарии по RSS
- Следим на Twitter

Метки

C# HowTo: **Wealth-Lab**
Видео Индикаторы МТС Торговые роботы Юмор алготрейдинг мероприятия обучение

Управление

[Регистрация](#)

[Вход](#)

[RSS Записей](#)

[RSS Комментариев](#)

[WordPress.org](#)

Во многих случаях Вы сможете найти здесь уже готовый код для тех ситуаций, которые Вы искали.

В этой инструкции для демонстрации концепций программирования и тестирования торговых систем используются множество наиболее востребованных WealthScript функций. Однако в рамках одной инструкции невозможно рассказать про каждую простейшую WealthScript функцию. Полный список функций, доступных для использования в торговых стратегиях вместе с их синтаксисом, описанием и примерами можно найти в WealthScript QuickRef (в Tools меню, или нажав F11).

Для тех функции, которые существуют в пространстве имен {}WealthLab, но не задокументированны в QuickRef, не дается гарантии корректной работы при кодировании с их помощью торговых стратегий.

Программирование в .NET

Wealth-Lab Developer (версия 6) – это платформа для технического анализа и трейдинга, в которой Вы сможете запрограммировать и протестировать любые торговые стратегии на любом .NET языке. Это действительно так. Вы можете использовать любой из более чем 20-ти языков программирования.NET для корректной работы с этой платформой.

Однако редактор Wealth-Lab стратегий, снабжен только компилятором C#, следовательно, построитель стратегий генерирует код только используя синтаксис языка программирования C#. Поэтому, естественно, все примеры снабжены кодом на языке C#. Конечно же, Вы можете строить и тестировать торговые стратегии на любом .NET языке программирования.

Решаете, какой .NET язык программирования начать изучать?

Мы рекомендуем для новичков посетить курсы по программированию на языке C#. Ведь именно C# это действительно современный, предназначенный для .NET язык программирования, который вобрал в себя все лучшее из других языков программирования. Если Вы уже программировали раньше, то изучение языка C# не составит для Вас труда.

Но я же не программист!

Построитель стратегий программы Wealth-Lab Developer (версии 6) включает в себя средства автоматизации и все известные нам индикаторы, и дает Вам возможность применять их в различных сочетаниях. Благодаря этому возможности построения стратегий, даже вообще без знания языка программирования, практически безграничны.

Тем не менее, возможны такие стратегии, которые Вы будете не в состоянии запрограммировать и протестировать без использования языка программирования. Это сложные стратегии, которые включают в себя работу с разными таймфреймами, или стратегии с ротацией нескольких финансовых инструментов, или что-то подобное.

Если однажды Вы обнаружите стратегию, которую невозможно полностью запрограммировать используя доступные средства автоматизации, Вам все равно придется окунуться в мир программирования с использованием WealthScript.

Как использовать эту инструкцию?

Несмотря на то, что Wealth-Lab Developer включает в себя довольно большое количество drag-and-drop Мастеров для построения графиков и даже для построения торговых стратегий, мощь этой программы смогут оценить только те, кто умеет программировать.

Для того, чтобы сделать эту инструкцию по программированию на языке WealthScript наиболее полезной, мы постарались снабдить ее множеством примеров. Советуем Вам сначала пролистать каждую главу по порядку, т.к. базовые принципы рассмотрены в самом начале.

Совет:

Если Вы ищете определенный пример с кодом, попробуйте поискать выражение "How to" в поисковом указателе (либо тег КАК: в блоге "Финансовая лаборатория").

На сегодня все. В следующий раз опубликую часть, **посвященную барам**.

Нет похожих сообщений.

 РУБРИКИ: **WEALTHSCRIPT**

Комментариев: 19

Александр пишет:

13 марта 2012 в 16:01

Отличное начинание! Готов даже подключиться к переводу. Отправьте мне текст на английском – я переведу и скину вам на почту. Понравится – опубликуете. Может еще кто изъявит желание. Так быстренько и переведем Инструкцию.

Ответить

Дмитрий Власов пишет:

10 мая 2012 в 00:21



Александр, спасибо за предложение. Но хочется самому попереводить, потому что в процессе все понимаешь лучше. Плюс английский заодно учу.

Ответить

Андрей пишет:

18 марта 2012 в 16:45



Добрый день, как подключить WLD6 к квику?

Ответить

Дмитрий Власов пишет:

10 мая 2012 в 00:20



К квику можно подключить написав брокер Адаптер. Вообще, т.к. Wealth-Lab купила Fidelity – было принято решение не документировать такое решение. Однако есть умельцы. К примеру, в Церихе сделали собственный адаптер к Квику. Игорь Чечет сделал брокерадаптер к АльфаБрокеру.

Ответить

Дмитрий Власов пишет:

17 июня 2012 в 23:57



Вот здесь: написал специальный пост по вопросу связки QUIK с Wealth-Lab.

Ответить

rasswet пишет:

18 марта 2012 в 17:21



по велсу 6 есть перевод у цериха. но там по-моему именно дока к программе.

Ответить

Дмитрий Власов пишет:

10 мая 2012 в 00:18



Привет. В Церихе действительно есть только перевод инструкции к самой программе, а не к инструкции по программированию.

Ответить

tradetoforts пишет:

14 апреля 2012 в 22:44



Для российского трейдера лучшего всего использовать ТСЛаб. Примеры робота для фьючерса РТС. Видео:

<http://www.youtube.com/watch?v=Mg2ZZeYe0OY>

Робот на фьючерсе доллара: <http://www.youtube.com/watch?v=YUC4dcbFTbY>

Ответить

Дмитрий Власов пишет:

10 мая 2012 в 00:23



На мой взгляд, решение самое оптимальное должно быть такое: в Велс Лаб проходит тестирование стратегий (там есть к тому же генетический метод подбора оптимальных параметров и метод монте карло, плюс куча других фишечек). А уже выбранную и протестированную стратегию можно торговать на TSLab, т.к. там нет проблем с подключением к российским брокерам и получением котировок и т.п.

Ответить

Gerig пишет:

2 мая 2012 в 09:24



Не согласен с этим утверждением. Я достаточно длительное время использовал TSlab, более года, освоил его достаточно хорошо, готовил скрипты как с использованием “кубиков”, так и C#. Есть опыт написания достаточно успешных скриптов на TSlab. Конечно программа не плохая, но сравнивать ее с Wealth-lab просто не корректно. TSlab не дает тех возможностей какие можно получить от Wealth-lab. Сейчас я полностью отказался от TSlab и использую только Wealth-lab. Испытываю при этом огромное удовольствие от работы с Wealth-lab, строю в ней нейросети (с использованием дополнительных опций “Neuro-lab”).

Вообщем для новичков, для создания представления о рынке и его простейших взаимосвязей, на первом этапе наверное может сгодится и TSlab, но перейдя на Wealth-lab вы поймете, что TSlab – детская песочница, по сравнению со стройплощадкой.

Ответить

Дмитрий Власов пишет:

10 мая 2012 в 00:24



Для новичков (особенно при наличии инструкций на русском) освоить Велс Лаб намного проще. На мой взгляд.

Ответить

Gerig пишет:

11 мая 2012 в 21:34



Да, пожалуй я с вами соглашусь. Хотя сейчас у многих молодых людей (в отличие от моего поколения) достаточно хороший английский и по большому счету необходимости в переводе нет. Быстро осваивают суть, ведь так и должно быть – молодое поколение должно быть лучше, и это приятно.

Ответить

Ответы на вопросы по программированию торговых стратегий в Wealth-Lab | Финансовая лаборатория пишет:

14 мая 2012 в 13:34

[...] 1. Где найти русскую инструкцию по программированию торг... [...]

Ответить

Сергей пишет:

20 мая 2012 в 20:07



А “работают”-ли Ваши инструкции в Wealth-Lab Developer (версия 4)?

Ответить

Дмитрий Власов пишет:

20 мая 2012 в 20:29



Приветствую. Инструкции – не мои, я только перевожу и снабжаю примерами с российского рынка, плюс иногда немножко комментирую в тексте то, что считаю нужным прокомментировать.

Для Wealth-Lab (версия 4) они не подойдут, т.к. в инструкции язык программирования торговых стратегий – C#, а раньше был другой – внутренний язык.

Язык C# намного гибче и удобнее.

Ответить

Сергей пишет:

20 мая 2012 в 21:13



Извините. Я не правильно выразился. Воспринимает 4-я версия синтаксис написания программ 6-й?

Четвёрке почему-то не нравится “using”, да и в конце кода там используется “end” вместо “}”.

Ответить

Сергей пишет:

20 мая 2012 в 21:15

Всё понял. Спасибо.

Ответить



Торговая система HighLowLong - шаблон кода и оптимизация. | Финансовая лаборатория пишет:

26 мая 2012 в 20:31

[...] [...]

Ответить

код стратегии для Wealth Lab | Финансовая лаборатория пишет:

13 июня 2012 в 15:17

[...] Язык, который используется для написания кода, описывающего Вашу торговую систему называется C#.

Подробнее о том, на каком языке пишутся стратегии можно прочитать здесь. [...]

Ответить

Оставить комментарий

Имя (Обязательно)

Почта (Обязательно)

Вэб-сайт

Получать новые комментарии по электронной почте. Вы можете [подписаться](#) без комментирования.

© 2012, Финансовая лаборатория. Все права защищены.

Любое использование материалов допускается только с указанием активной ссылки на источник.

448

240

185

В день мой блог посещает 77 человек, сегодня

121

mail.ru РЕЙТИНГ

757121

506

193

Анонсы

BY FEEDBURNER

Подписка

[RSS](#)

[Twitter](#)

[ВКонтакте](#)

[facebook](#)

Сайт

[Создать блог](#)

[Содержание](#)

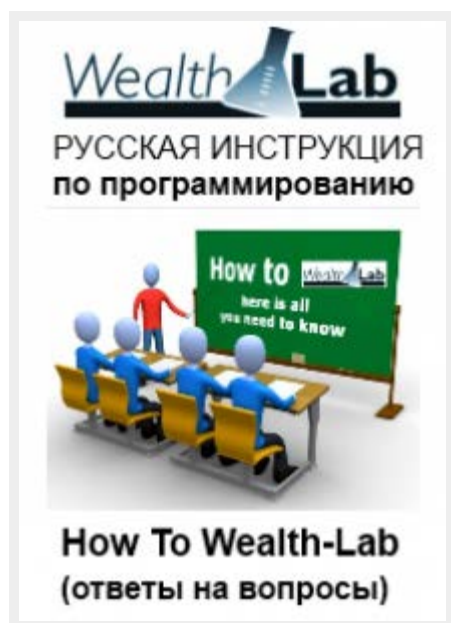
[О блоге](#)

[Карта](#)

Ответы на вопросы по программированию торговых стратегий в Wealth-Lab.

 АВТОР ДМИТРИЙ ВЛАСОВ  МАЯ - 14 - 2012  КОММЕНТАРИЕВ: 25

Здесь будут накапливаться вопросы, которые возникают у тех, кто осваивает построение торговых стратегий с помощью C# и языка WealthScript программы Wealth-Lab. По сути, это будет база данных по ответам, которые находятся в инструкциях Велс Лаб. Если У Вас есть вопрос, на который Вы не нашли ответа здесь - пишите его в комментариях - постараюсь помочь найти на него ответ.



1. Где найти русскую инструкцию по программированию торговых стратегий в программе Wealth-Lab, с подробным описанием языка C#, библиотеки WealthScript и примерами?
2. Какой язык программирования нужно выучить для того, чтобы уметь строить и тестировать торговые стратегии в программе Wealth-Lab?
3. Как выполнить в программе Wealth-Lab код примеров, который приводится в Инструкции по программированию и статьях с описанием языка C# и библиотеки WealthScript?
4. Как с помощью кода посчитать количество баров на графике Wealth-Lab?
5. Как найти номер первого внутридневного бара, который был несколько дней назад?
6. Как найти номер бара, который соответствует конкретной дате на графике Велс Лаб?
7. Как в коде торговой стратегии программы Wealth Lab получить доступ к значениям цен (OHLC) и объема (V)?
8. Как в коде торговой стратегии программы Wealth-Lab отобразить на графике ряды цен (OHLC) и объемов (V)?
9. Как с помощью кода скорректировать первоначальные значения цен и объемов (OHLC/V) в программе Велс Лаб?
10. Как программно узнать текущий таймфрейм или интервал графика?
11. Как получить доступ к дате и времени соответствующего бара на графике в программе Wealth-Lab?
12. Как получить доступ к краткому названию текущего финансового инструмента (тикера) или к полному названию компании при программировании торговых стратегий в Велс Лаб?
13. Как получить доступ к величине минимального шага цены, если торгуемый инструмент является

RSS товары



RSS новости



Twitter



Последние комментарии

макс: А где взять свечки с открытым интересом? на сайте финама нет



ring10: Добрый день Дмитрий Как сделать, чтобы открывался 1 велф лаб? Если открываешь под пользователем, т [...]



Дмитрий Власов: Буян, рад, что Вам пригодились материалы данной статьи!



Максим: Сам себе отвечу. protected override void Execute() { Bars rts = GetExternalSymbol("RTS", t [...]



Буян: Спасибо Дмитрий за статью! Мне удалось разработать хорошую рабочую торговую систему, используя возмо [...]



Последние посты



Бесплатный вебинар: 7 смертных грехов при оптимизации торговых систем

НОЯБРЯ - 5 - 2012



Отзывы участников курса до результата Wealth-Lab. С нуля до первой торговой системы за 5 дней.

НОЯБРЯ - 3 - 2012



Быстрее, Выше, Сильнее: Обновление Wealth-Lab до новой версии 6.4

ОКТАБРЯ - 23 - 2012



Курс до результата: Wealth-Lab – от нуля до первой торговой системы за 5 дней

ОКТАБРЯ - 23 - 2012



Обзорная экскурсия по достопримечательностям программы Wealth-Lab

ОКТАБРЯ - 20 - 2012

Берите

фьючерсом?

14. Как получить доступ к барам второго (внешнего) финансового инструмента в рамках одной торговой стратегии программы Wealth-Lab?

15. Как оперировать (Умножать, Делить, Добавлять или Вычитать) несколькими DataSeries при построении торговых стратегий в Wealth-Lab?

16. Как создать ряд данных, содержащий типичную цену ("Typical Price") с помощью кода при создании торговых стратегий в Велс Лаб?

17. Как сдвинуть значения DataSeries вправо или влево по шкале времени?

18. Как получить массив DataSeries только с положительными значениями?

19. Как получить доступ к информации конкретного бара графика Велс Лаб при построении и тестировании торговой стратегии?

20. Как создать последовательность данных DataSeries и заполнить каждую ее ячейку нулевыми значениями?

21. Как поменять значения в массивах DataSeries?

22. Как получить доступ к DataSeries второго финансового инструмента в рамках одной торговой стратегии, проектируемой в Wealth-Lab?

23. Как организована по умолчанию синхронизация нескольких финансовых инструментов по времени в Велс Лаб?

24. Как поменять способ синхронизации данных в Wealth-Lab?

25. Как создать индикатор используя несинхронизированные данные второго финансового инструмента?

26. Какие области графика выделяются в программе Wealth-Lab для отображения разных данных?

27. Как создать индивидуальную область на графике?

28. Как: скрыть область объема на графике в программе Wealth-Lab?

29. Как в Wealth-Lab отобразить на графике внешний финансовый инструмент?

30. Как отобразить на графике в программе Велс Лаб синтетический финансовый инструмент?

31. Как сравнить основной финансовый инструмент с бенчмарком и отобразить их на одном графике?

32. Как отобразить на графике программы Велс Лаб последовательность данных (DataSeries)?

33. Как отобразить на графике в программе Wealth-Lab область объемов выше чем область цен?

34. Как отобразить на индивидуальной области графика количественные показатели торговой стратегии?

35. Как контролировать вертикальную Y-шкалу на графиках в программе Wealth-Lab?

36. Как написать вертикальный текст на гарфике в программе Wealth-Lab?

37. Как сделать более удобным описание рядов данных (индикаторов, осцилляторов и т.п.) для отображении описания ряда данных на графике?

38. Как в программе Велс Лаб нарисовать линию тренда и продлить ее вправо по шкале времени?

39. Как начертить на графике в программе Велс Лаб скоростные линии сопротивления (1/3 - 2/3 линии)?

40. Как в Wealth-Lab выделить зону на графике в виде многоугольника?

41. Где можно посмотреть полный синтаксис применяемого индикатора?

42. Как получить доступ к полному ряду данных (DataSeries) индикатора?

43. Как рассчитать значение индикатора только для одного бара?

44. Как рассчитать значение индикатора несколькими способами?

45. Как создать "полуофициальный (semi-formal)" индикатор своими руками?

46. Как создать индикатор MACD с самостоятельно определяемым периодом?

47. Как создать индикатор, отражающий взвешенную цену закрытия

48. Как создать индикатор СПРЕДа, отражающий разницу между двумя финансовыми инструментами?

49. Как создать осциллятор силы Элдера (EFO)?

50. Как самостоятельно создать NRTR_WATR индикатор Константина Копыркина?



Видео ВЕБИНАРОВ



Дмитрия Власова



МТС под микроскопом: WLMA System

Подробный разбор торговой
системы WLMA System

3900 руб.

 **В корзину**

Рубрики

Выберите рубрику

Обновления:



Новые посты по Email



Новые посты по RSS



Комментарии по RSS



Следим на Twitter

Метки

C# HowTo: **Wealth-Lab**
Видео Индикаторы МТС Торговые
роботы Юмор алготрейдинг
мероприятия обучение

Управление

Регистрация

Вход

RSS Записей

RSS Комментариев

WordPress.org

51. Как создать индикаторы используя внешние финансовые инструменты?
52. Почему значения некоторых индикаторов нестабильны и зависят от периода доступных данных?
53. Почему опасно использовать нестабильные значения индикаторов?
54. С какими индикаторами нужно быть осторожным?
55. Что представляет собой класс стратегий в программе Велс Лаб?
56. Как выполнить компиляцию кода стратегий в программе Wealth-Lab?
57. Какие ограничения существуют для кода стратегий в программе Wealth-Lab?
58. Какие торговые сигналы поддерживаются в программе Wealth-Lab?
59. Какие параметры имеют торговые сигналы в Wealth-Lab?
60. Как совершить теоретический трейд в программе Велс Лаб?
61. Как произвести сделку в определенную заранее дату применяя метод DateTimeToBar()?
62. Как организовать главный цикл при построении торговых стратегий в Wealth-Lab?
63. Как переделать код торговой стратегии в код скрининговой системы?
64. Где найти шаблон программного кода для однопозиционной стратегии Wealth-Lab?
65. Что представляют из себя позиции (Positions) в программе Wealth-Lab?
66. Как проверить с помощью кода - существует ли активная позиция в текущий момент времени в Wealth-Lab?
67. Как в Wealth-Lab осуществляется закрытие позиции?
68. Как разместить лимитный или стоп приказ на покупку при программировании стратегий в Wealth-Lab?
69. Как создать отложенные (GTC) ордера (приказы, действующие до отмены) при моделировании торговли в Велс Лаб?
70. Как отображать на графике программы Велс Лаб стоп-приказы с помощью программного кода?

Если Вам понравилась эта статья - подпишитесь на **новые СТАТЬИ по RSS** или получайте наши **новые СТАТЬИ на E-mail**:
Рекомендую почитать также:

- Курс для тех, кто хочет создавать собственные торговые стратегии: Wealth-Lab: C# на службе трейдера
- Как установить связь между Wealth-Lab и Visual Studio для отладки торговой стратегии
- Видео вебинара по программе Wealth-Lab в рамках курса “Wealth-Lab: C# на службе трейдера”
- Объект Bars в Wealth Lab – что такое бары в Велс Лаб
- Наполнение созданных самостоятельно DataSeries данными при помощи WealthScript

 РУБРИКИ: [WEALTH-LAB](#), [WEALTHSCRIPT](#), [ПРОГРАММЫ ТРЕЙДЕРУ](#)

Комментариев: 25

Andrew пишет:

21 мая 2012 в 12:18



Добрый день!

Подскажите пожалуйста, к кому можно обратиться за помощью по программе FinLab.Trade. Писал на разные ящики, никто не отвечает

У меня при открытие файла лицензионного ключа, программа пишет:

“К сожалению, ни одной лицензии добавлено не было. Возможна, лицензии истекли или повреждены. Подробности:

Ошибок не обнаружено!”

Помогите пожалуйста!

Ответить

Дмитрий Власов пишет:

21 мая 2012 в 14:04



Андрей, добрый день. Мы сейчас программу FinLabTrade передали компании АЛОП. Если хотите – можете попробовать договориться об условиях ее использования с Воронежским представительством АЛОРа (e-mail: vrnfic@alor.ru или vrnfic@vrnfic.ru).

Ответить

Alexander Panarin пишет:

21 мая 2012 в 19:35



Добрый вечер, Дмитрий. Возможно сделать расчеты индикаторов и нарисовать их с разным Time-Frame, т.е. как в ТСлабе есть минутка а расчет индикаторов на базе часовика. Там это делается путем сжатия и разжатия времени, а как в Велс Лабе?

Ответить

Дмитрий Власов пишет:

21 мая 2012 в 19:43



Да, это все можно сделать без проблем в WealthLab. Об этом чуть позже целый пост будет. Пока скажу, что для этих целей используются следующие методы: SetScaleCompressed(), SetScaleDaily(); SetScaleWeekly(); SetScaleMonthly();

Ответить

Gerig пишет:

25 мая 2012 в 10:06



Добрый день, Александр! В TSlab это можно сделать применив кубик сжатие, но картинка получится ломаная, не наглядная, у TSlab нет того функционала, который есть в Wealthl-lab.

Ответить

Alexander Panarin пишет:

21 мая 2012 в 19:45



Очень интересные выши посты. Подождем.

Ответить

Kartman пишет:

28 мая 2012 в 10:43



Дмитрий, как создать два ряда данных из белых и чёрных отдельно?

Ответить

Дмитрий Власов пишет:

28 мая 2012 в 11:40



Нужно два раза выполнить метод PlotSeries();
При этом в первом методе выбрать свойство Color.Black, а во втором случае – соответственно Color.White (правда, на белом фоне будет не видно, так что лучше какой-либо другой фон выбрать, либо цвет, передающий ряд данных).

```
PlotSeries(myPane, Close, Color.Black, WealthLab.LineStyle.Line, 2);  
PlotSeries(myPane, Close, Color.White, WealthLab.LineStyle.Line, 2);
```

Ответить

Kartman пишет:

28 мая 2012 в 12:03



Спасибо. Я не про то. DataSeries для свечей вверх и свечей вниз. Вот например, для RSI (нууу за 14 дней) среднее вверх надо разделить на среднее вниз. Для начала надо узнать их сумму. как всё это сделать?) Я понимаю, что RSI там вшитый есть – мне нужно кодом.

Ответить

Дмитрий Власов пишет:

29 мая 2012 в 15:40



Т.е. Вы хотите уточнить – как создавать собственные индикаторы? Все это тоже несложно делается – создаете собственную DataSeries и заполняете ее значениями, которые Вам нужны. Примеров куча [вот в этой статье приводится](#)

Ответить

Alexander Panarin пишет:
6 июня 2012 в 11:54



Добрый день, Дмитрий. Подскажите, пожалуйста, встречали Вы проработки кода по определению сильных уровней с помощью объемно-вертикального анализа (т.е. то, чем занимается Volfix на основе теории Стеделмайера)?

Ответить

Дмитрий Власов пишет:
6 июня 2012 в 16:40



Александр, честно говоря такую реализацию для Велса я не встречал.

Ответить

Alexander Panarin пишет:
7 июня 2012 в 09:23



Дмитрий, жаль конечно, но это очень интересная тема особенно, если объемно-вертикальный анализ совместить с элементами технического анализа, то можно продвинуться намного дальше, чем просто с техническим анализом. Я пытаюсь в этом разобраться, если у Вас будут какие-то мысли на эту тему, то было бы интересно продолжить такой опыт.

Ответить

Дмитрий Власов пишет:
7 июня 2012 в 11:34



Если сформулируете почетче условие на вход в позицию – я готов помочь это закодировать и предоставить Вам этот код.
Было бы интересно попробовать.

Ответить

Alexander Panarin пишет:
7 июня 2012 в 21:18



Добрый вечер, Дмитрий. Можете прислать свою почту? Я сброшу картинку из Волфикса со своими комментариями. Если Вам будет интересно, то можно будет продолжить эту тему.

Ответить

Дмитрий Власов пишет:
7 июня 2012 в 21:55



Вот здесь можно мне написать <http://finlabportal.ru/kontakty/> – я отвечу на письмо.

Ответить

Олег пишет:
9 июня 2012 в 13:59



Дмитрий, подскажите пожуйста:
1. как найти High/Low внутри дня (например с 10:15 до 12:00) по мотивам Вашего примера «Торговая система HighLowLong?»
2. Как для WL подключить //using System.Linq, в VS ошибок нет, а в WL пишет ошибку “Имя типа или пространства имен Linq отсутствует в пространстве имен System (пропущена ссылка на борку?)”
Спасибо.

Ответить

Дмитрий Власов пишет:
9 июня 2012 в 15:13



Олег, приветствую.
1) Если Вы просто хотите входить сделки с 10:15 до 12:00 часов – Вы можете установить дополнительное условие (фильтр) в блоке генерации сигналов. Например вот так:

Код примера на C# (кликните чтобы посмотреть)

```
1  //.....
2
3  for (int bar = 1; bar < Bars.Count - 1; bar++)
4  {
5  // ...
6  int hourBar = Date[bar].Hour;
```

```
7      int minuteBar = Date[bar].Minute;
8      int timecount = hourBar * 100 + minuteBar
9
10     //УСЛОВИЕ ПО ВРЕМЕНИ
11     if (timecount >1015 && timecount < 1200)
12         BuyAtMarket(bar + 1, "Buy");
13     //...
14 }
```

2. Эту строчку можете вообще удалить, либо закомментировать, поставив вначале две косые черты. Она здесь не играет вообще никакой роли.

Может быть кто-нибудь предложит другие варианты...

Ответить

Олег пишет:

9 июня 2012 в 16:45



Дмитрий, спасибо за ответ, но это не совсем то, что я ищу. Идея в следующем: находим макс и мин значение образованное ценой в период с 10:15 до 12:00 и входим в сделку на преодоление этого макс или мин уже после 12:00.

Ответить

Дмитрий Власов пишет:

9 июня 2012 в 17:07



Олег, в принципе, это тоже вполне решаемая задача. В общем виде схема будет выглядеть так: Вы заводите две переменных, одна из которых будет содержать значение максимум цен, другая минимум цен (double maxPrice; double minPrice)

После чего внутри основного торгового цикла проверяете – если время соответствует устанавливаемому Вами интервалу – то проверяете – если максимальная цена текущего бара больше чем значение переменной, то переменная maxPrice получает новое значение. Тоже самое для переменной, которая хранит наименьшее значение.

После этого в блоке генерации сигнала формулируете правила входа в позицию с учетом времени и рассчитанных минимальных и максимальных цен за нужный Вам период.

Код писать сейчас не буду, думаю, идея ясна. Если не будет получаться – пишите – помогу.

Ответить

Дмитрий пишет:

6 июля 2012 в 12:13



Здравствуйте.

Пилю стратегию, использующую указатели, компилятор пишет, компилировать нужно с параметром “/unsafe”, но в мануалах по Wealth Lab нет даже намёка на возможность задавать параметры компиляции.

Не подскажете, возможно ли это в действительности и как?

Спасибо.

Ответить

Дмитрий Власов пишет:

6 июля 2012 в 17:09



Дмитрий, я попросил у знакомого специалиста ответить на этот вопрос. Вот что он сказал:

” Лучше не надо использовать указатели в стратегии 😊

Такая опция компилятора есть в студии. Она позволяет компилировать небезопасный код.

Лично я предпочитаю для перехода между safe-unsafe использовать managed c++ проекты. В них можно переход между managed и unmanaged кодом делать, а из c# использовать только managed часть этих библиотек”

Если не совсем понятно – пишите – уточним.

Ответить

SAV пишет:

6 июля 2012 в 16:43



Дмитрий, думаю здесь проблема не в том, что в мануалах нет об этом речи, а суть заключается в том, что .net приложения, использующие unmanaged код должны компилироваться с учетом ключевого слова unsafe. Т.е когда Вы используете метод, тип, еще что-нибудь, то ставьте перед ним unsafe.

Попробуйте откомпилировать в Visual Studio:
Установка параметра компилятора в среде разработки Visual Studio
Откройте страницу Свойства проекта.
Выберите страницу свойств Построение.
Установите флажок Разрешить небезопасный код.
<http://msdn.microsoft.com/ru-ru/library/ct597kb0.aspx>

Ответить

wlrt пишет:
6 июля 2012 в 17:05



Дмитрий, по поводу Вашей стратегии с указателями, возможно это какая-то ошибка, либо вы пытаетесь использовать сторонние компоненты? можете написать подробнее указатели на что вы используете в стратегии. В с# вообще, не только в Wealth-Lab, в этом практически нет необходимости.
PS Параметры можно указать в MS Visual Studio, если вынести туда свой код

Ответить

Дмитрий пишет:
7 июля 2012 в 13:15



Да, пожалуй, обойдусь без указателей, т.к. мне просто нужна функция, возвращающая несколько значений, и я только что вычитал в интернете про ref. 😊

Ответить

Оставить комментарий

Имя (Обязательно)

Почта (Обязательно)

Вэб-сайт

Получать новые комментарии по электронной почте. Вы можете [подписаться](#) без комментирования.

© 2012, Финансовая лаборатория. Все права защищены.

Любое использование материалов допускается только с указанием активной ссылки на источник.

224

79

73

В день мой блог посещает 84 человека,сегодня уже 45

mail.ru

РЕЙТИНГ

757399

263

76

Андрей

BY FEEDBURNER

Подписка

[RSS](#)

[Twitter](#)

[ВКонтакте](#)

[facebook](#)

Сайт

[Создать блог](#)

[Содержание](#)

[О блоге](#)

[Карта](#)



Как выполнить код, приведенный в примерах к инструкции по программированию торговых стратегий на языке WealthScript

 АВТОР ДМИТРИЙ ВЛАСОВ  МАЯ - 1 - 2012  КОММЕНТАРИЕВ: 33

В инструкции по программированию торговых стратегий в Велс Лаб приводится достаточно много примеров. Было бы очень здорово, чтобы Вы, осваивая язык программирования C# и библиотеку WealthScript выполняли каждый пример кода, который приводится в инструкции. Это поможет Вам глубже разобраться со спецификой программы и быстрее научиться самостоятельно создавать и тестировать торговые стратегии.

Как выполнить код примеров в программе Wealth-Lab?

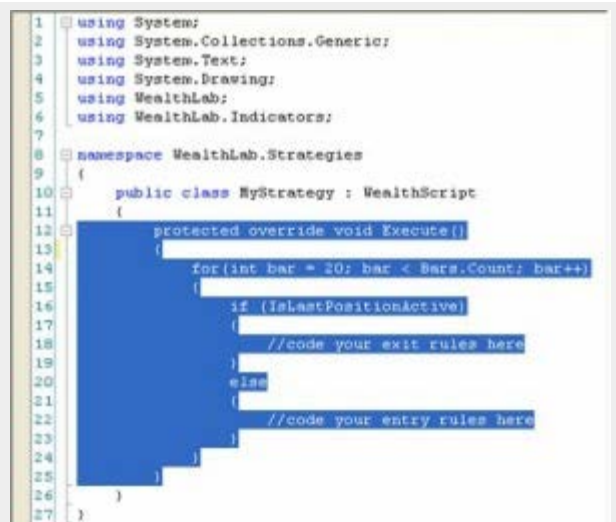
Для того, чтобы выполнить приведенный в приводящихся в инструкции по программированию примерах код, выполни следующие шаги:

1. Откройте новое окно стратегий «new Strategy Window» (Ctrl+Shift+S).
2. Щелкните по вкладке редактора «Editor».
3. Код в инструкции представлен в двух форах: *полной* и *частичной*.

Полная: Код является полным, если Вы видите выражение "using" в самом начале (именно так, как изображено на рисунке внизу). В этом случае нужно сопировать приведенный и полностью заменить весь код, который Вы видите во вкладке «Editor». После того, как Вы произведете вставку кода, переходите к шагу 5.

Частичная: Для того, чтобы сконцентрироваться на сущности примера, чаще всего приводится только метод Execute(). В этом случае, полностью замените только метод Execute(), который выделен на нижней картинке.

4. Обратите внимание на зеленую строчку комментариев в приведенном примере, в которой может содержаться описание начальных условий или инструкция по установке.
5. Щелкните кнопку «Compile» на панели истрменов «Editor» и убедиться, что внизу появится надпись "Strategy compiled successfully!" (Стратегия успешно скомпилирована).
6. Щелкните по кнопке F5 на клавиатуре, или, по Вашему выбору, на финансовый инструмент на панели инструментов для того, чтобы запустить пример на выполнение.



Замена метода ()Execution в редакторе стратегий

Выполнив эти несложные шаги Вы сможете запустить в программе WealthLab 6.3 приведенный в примерах код...

RSS товары



RSS новости



Twitter



Последние комментарии

макс: А где взять свечки с открытым интересом? на сайте финама нет



ring10: Добрый день Дмитрий Как сделать, чтобы открывался 1 велф лаб? Если открываешь под пользователем, т [...]



Дмитрий Власов: Буян, рад, что Вам пригодились материалы данной статьи!



Максим: Сам себе отвечу. protected override void Execute() { Bars rts = GetExternalSymbol("RTS", t [...]



Буян: Спасибо Дмитрий за статью! Мне удалось разработать хорошую рабочую торговую систему, используя возмо [...]



Последние посты



Бесплатный вебинар: 7 смертных грехов при оптимизации торговых систем

НОЯБРЯ - 5 - 2012



Отзывы участников курса до результата Wealth-Lab. С нуля до первой торговой системы за 5

дней.

НОЯБРЯ - 3 - 2012



Быстрее, Выше, Сильнее: Обновление Wealth-Lab до новой версии 6.4

ОКТАБРЯ - 23 - 2012



Курс до результата: Wealth-Lab – от нуля до первой торговой системы за 5 дней

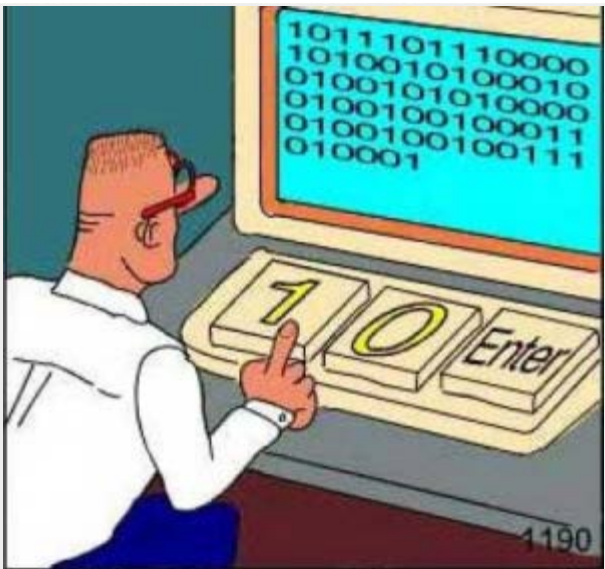
ОКТАБРЯ - 23 - 2012



Обзорная экскурсия по достопримечательностям программы Wealth-Lab

ОКТАБРЯ - 20 - 2012

Берите



Как выполнить пример кода в Wealth-Lab

Нет похожих сообщений.

 РУБРИКИ: [WEALTHSCRIPT](#)

Комментариев: 33

Как использовать при программировании стратегий шкалу, интервал и дату | Финансовая лаборатория пишет:

4 мая 2012 в 13:30

[...] Пример (как выполнить пример приведенного кода)... [...]

Ответить

Инструкция по программированию торговых стратегий в WealthLab | Финансовая лаборатория пишет:

4 мая 2012 в 14:08

[...] Как выполнить код, приведенный в примерах к инструкции... Главная Что такое уровни пивот и для чего они [...]

Ответить

OHLC/V Series - последовательности цен при программировании в Wealth Lab 6.3 | Финансовая лаборатория пишет:

5 мая 2012 в 14:01

[...] Пример (как выполнить пример приведенного кода)... [...]

Ответить

Symbol Info - подробная информация о торговом инструменте в программе WealthLab | Финансовая лаборатория пишет:

5 мая 2012 в 15:20

[...] Пример (как выполнить пример приведенного кода)... [...]

Ответить

DataSeries - основная структура данных в Wealth-Lab и математические операции при программировании торговых стратегий | Финансовая лаборатория пишет:

7 мая 2012 в 16:14

[...] Пример (как выполнить пример приведенного кода)... [...]

Ответить

Как получить доступ к единичному значению DataSeries | Финансовая лаборатория пишет:

8 мая 2012 в 13:24

[...] Пример (как выполнить пример приведенного кода)... [...]

Ответить



Видео ВЕБИНАРОВ



Дмитрия Власова



**МТС под микроскопом:
WLMA System**

Подробный разбор торговой
системы WLMA System

3900 руб.

 **В корзину**

Рубрики

Выберите рубрику

Обновления:

-  Новые посты по Email
-  Новые посты по RSS
-  Комментарии по RSS
-  Следим на Twitter

Метки

С# HowTo: **Wealth-Lab**
Видео **Индикаторы** **МТС** **Торговые**
роботы **Юмор** **алготрейдинг**
мероприятия **обучение**

Управление

Регистрация

Вход

RSS Записей

RSS Комментариев

WordPress.org

Как получить доступ ко второму финансовому инструменту при проектировании торговых стратегий в Велс Лаб. | Финансовая лаборатория пишет:

10 мая 2012 в 11:25

[...] Пример (как выполнить пример приведенного кода)... [...]

Ответить

Наполнение созданных самостоятельно DataSeries данными при помощи WealthScript | Финансовая лаборатория пишет:

10 мая 2012 в 13:49

[...] Пример (как выполнить пример приведенного кода)... [...]

Ответить

Синхронизация данных при построении торговых стратегий с несколькими финансовыми инструментами | Финансовая лаборатория пишет:

11 мая 2012 в 12:29

[...] Пример (как выполнить пример приведенного кода)... [...]

Ответить

Построение графиков в программе Wealth-Lab при помощи WealthScript | Финансовая лаборатория пишет:

12 мая 2012 в 08:57

[...] Пример (как выполнить пример приведенного кода)... [...]

Ответить

Отображение на графике синтетических финансовых инструментов | Финансовая лаборатория пишет:

13 мая 2012 в 08:47

[...] Пример (как выполнить пример приведенного кода)... [...]

Ответить

Объект Bars в Wealth Lab - что такое бары в Велс Лаб | Финансовая лаборатория пишет:

13 мая 2012 в 13:58

[...] цен при программировании в Wealth Lab 6.3 Главная Как выполнить код, приведенный в примерах к инструкции... [...]

Ответить

Отображение на графике в программе Велс Лаб рядов данных | Финансовая лаборатория пишет:

14 мая 2012 в 11:49

[...] Пример (как выполнить пример приведенного кода)... [...]

Ответить

Написание текста на графиках Wealth-Lab программными методами | Финансовая лаборатория пишет:

15 мая 2012 в 15:08

[...] Пример (как выполнить пример приведенного кода)... [...]

Ответить

Ответы на вопросы по программированию торговых стратегий в Wealth-Lab | Финансовая лаборатория пишет:

15 мая 2012 в 15:32

[...] [...]

Ответить

Рисование графических объектов на графике Wealth-Lab | Финансовая лаборатория пишет:

16 мая 2012 в 12:45

[...] Пример (как выполнить пример приведенного кода)... [...]

Ответить

Проектирование и построение индикаторов в Wealth-Lab | Финансовая лаборатория пишет:

17 мая 2012 в 13:29

[...] Пример (как выполнить пример приведенного кода)... [...]

Ответить

Стабильность индикаторов в программе Wealth-Lab в зависимости от ширины тестируемого окна. | Финансовая лаборатория пишет:

22 мая 2012 в 15:09

[...] Пример (как выполнить пример приведенного кода)... [...]

Ответить

Программирование торговых стратегий в программе Wealth-Lab | Финансовая лаборатория пишет:

22 мая 2012 в 15:11

[...] Пример (как выполнить пример приведенного кода)... [...]

Ответить

Делаем индикаторы в Wealth-Lab своими руками | Финансовая лаборатория пишет:

22 мая 2012 в 15:21

[...] Пример (как выполнить пример приведенного кода)... [...]

Ответить

Виды торговых сигналов в Wealth-Lab | Финансовая лаборатория пишет:

24 мая 2012 в 02:01

[...] Пример (как выполнить пример приведенного кода)... [...]

Ответить

Организация торгового цикла в программе Wealth-Lab и его переделка в скрининговую систему | Финансовая лаборатория пишет:

24 мая 2012 в 08:42

[...] Пример (как выполнить пример приведенного кода)... [...]

Ответить

Программирование однопозиционных стратегий (SP Strategies) в программе Wealth-Lab | Финансовая лаборатория пишет:

25 мая 2012 в 14:10

[...] Пример (как выполнить пример приведенного кода)... [...]

Ответить

Использование при программировании торговых стратегий торговых приказов AtStop и AtLimit | Финансовая лаборатория пишет:

26 мая 2012 в 08:17

[...] Пример (как выполнить пример приведенного кода)... [...]

Ответить

Торговая система HighLowLong - шаблон кода и оптимизация. | Финансовая лаборатория пишет:

26 мая 2012 в 13:03

[...] Пример (как выполнить пример приведенного кода)... [...]

Ответить

Программирование выходов, основанных на времени удержания позиции в Wealth-Lab | Финансовая лаборатория пишет:

28 мая 2012 в 10:26

[...] Пример (как выполнить пример приведенного кода)... [...]

Ответить

Ловушки при построении торговых систем, связанные с проблемой "подглядывания" в будущее | Финансовая лаборатория пишет:

30 мая 2012 в 01:00

[...] Пример (как выполнить пример приведенного кода)... [...]

Ответить

Использование обоснованного заглядывания в будущее при программировании торговых стратегий в Велс Лаб | Финансовая лаборатория пишет:

30 мая 2012 в 21:01

[...] Пример (как выполнить пример приведенного кода)... [...]

[Ответить](#)

Создание параметров торговой стратегии в Wealth-Lab | Финансовая лаборатория пишет:

31 мая 2012 в 21:40

[...] Пример (как выполнить пример приведенного кода)... [...]

Ответить

Создание многопозиционных стратегий в Wealth-Lab | Финансовая лаборатория пишет:

2 июня 2012 в 20:28

[...] Пример (как выполнить пример приведенного кода)... [...]

Ответить

Что такое алерты и как их использовать в Wealth-Lab | Финансовая лаборатория пишет:

2 июня 2012 в 20:29

[...] Пример (как выполнить пример приведенного кода)... [...]

[Ответить](#)

Анализ данных в Wealth-Lab с учетом разных таймфреймов | Финансовая лаборатория пишет:

3 июня 2012 в 15:06

[...] Пример (как выполнить пример приведенного кода)... [...]

Ответить

Схема генератора механических торговых систем и создание новой торговой техники | Финансовая лаборатория пишет:

4 июня 2012 в 20:20

[...] Пример (как выполнить пример приведенного кода)... [...]

Ответить

Оставить комментарий

Имя (Обязательно)

Почта (Обязательно)

Получать новые комментарии по электронной почте. Вы можете [подписаться](#) без комментирования.

© 2012, Финансовая лаборатория. Все права защищены.
Любое использование материалов допускается только с указанием активной ссылки на источник.



448
240
185



В день мой блог
посещает **77**
человек,
сегодня



121



рейтинг
757123 508
193



BY FEEDBURNER

Подписка

- [RSS](#)
- [Twitter](#)
- [ВКонтакте](#)
- [facebook](#)

Сайт

- [Создать блог](#)
- [Содержание](#)
- [О блоге](#)
- [Карта](#)

Объект Bars в Wealth Lab – что такое бары в Велс Лаб

 **АВТОР ДМИТРИЙ ВЛАСОВ**  **МАЯ - 2 - 2012**  **КОММЕНТАРИЕВ: 3**

Баром является любой промежуток времени, в котором можно определить цену открытия (open), закрытия (close), а также наибольшие (high) и наименьшие (low) цены. Такие бары в литературе обычно называются (OHLC) бары.



Цены бара могут быть и различными и одинаковыми. Бар может состоять из любого количества минут (например 1, 2, 5, 20, 30-минутные бары), или дней, недель, месяцев, кварталов или лет. Также в Wealth-Lab поддерживаются бары, кратные тикам и секундам.

Объект **Bars** представляет собой исторический ряд данных о ценах (OHLC) и объемах (V) соответствующих баров. Также **Bars** могут содержать дополнительные именованные ряды данных, такие как открытый интерес для фьючерсов.

В принципе, Вы можете думать о барах как о названии компании или торговом инструменте. Хотя на самом деле бары содержат списки значений цен и объемов с соответствующими им датами (историю цен торгуемого инструмента). Таким образом, бары используются для доступа во время тестов к торгуемому инструменту (символу), его тикеру и прочей информации.

Как: посчитать количество баров на графике

Используйте свойство **Count** объекта **Bars**.

Обратите внимание на то, что свойство Count дает Вам представление о суммарном количестве баров. Однако поскольку Bars является массивом, который содержит в себе объекты Bar, то первый номер бара в массиве обозначается как нулевой элемент. Соответственно, последний номер бара на графике будет равен (Bars.Count - 1). Зная это, Вы не будете удивлены когда обнаружите, что на графике Wealth-Lab последний бар имеет номер 999, хотя Вы точно знаете что имеется ровно 1000 баров.

Пример №00: ([как выполнить пример приведенного кода](#))...

Код примера на C# (кликните чтобы посмотреть) 

```
1 | PrintDebug("График имеет " + Bars.Count + " баров.");
2 | PrintDebug("Номер первого бара = 0, а номер последнего бара = " + (Bars.Count - 1));
```

Как: найти номер первого внутридневного бара, который был несколько дней назад

Так как номера баров на графике всегда обозначаются от нуля до (Bars.Count - 1), часто полезно знать номер внутридневного бара, где 0 - это номер первого бара внутри дня. Для того, чтобы найти внутридневной номер конкретного бара используйте метод Bars.IntradayBarNumber().

Пример использует метод Bars.IntradayBarNumber() наоборот для того, чтобы найти номер первого бара конкретного дня.

[RSS товары](#)



[RSS новости](#)



[Twitter](#)



Последние комментарии

макс: А где взять свечки с открытым интересом? на сайте финама нет



ring10: Добрый день Дмитрий Как сделать, чтобы открывался 1 велф лаб? Если открываешь под пользователем, т [...]



Дмитрий Власов: Буян, рад, что Вам пригодились материалы данной статьи!



Максим: Сам себе отвечу. protected override void Execute() { Bars rts = GetExternalSymbol("RTS", t [...]



Буян: Спасибо Дмитрий за статью! Мне удалось разработать хорошую рабочую торговую систему, используя возмо [...]



Последние посты



Бесплатный вебинар: 7 смертных грехов при оптимизации торговых систем

НОЯБРЯ - 5 - 2012



Отзывы участников курса до результата Wealth-Lab. С нуля до первой торговой системы за 5 дней.

НОЯБРЯ - 3 - 2012



Быстрее, Выше, Сильнее: Обновление Wealth-Lab до новой версии 6.4

ОКТАБРЯ - 23 - 2012



Курс до результата: Wealth-Lab – от нуля до первой торговой системы за 5 дней

ОКТАБРЯ - 23 - 2012



Обзорная экскурсия по достопримечательностям программы Wealth-Lab

ОКТАБРЯ - 20 - 2012

Берите

Пример №01: (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть)

```
1 using System;
2 using System.Collections.Generic;
3 using System.Text;
4 using System.Drawing;
5 using WealthLab;
6 using WealthLab.Indicators;
7
8 namespace WealthLab.Strategies
9 {
10     public class MyStrategy : WealthScript
11     {
12         //Метод возвращает номер первого бара "n дней назад" для внутридневного
13         графика
14
15         public int FirstBarXDaysAgo(int DaysAgo) //вводя в скобки "дней назад"
16         узнаем номер первого бара указанного дня
17         {
18             int startBar = Bars.Count;
19
20             for (int day = 1; day <= DaysAgo; day++)
21
22                 startBar = startBar - 1 - Bars.IntradayBarNumber(startBar - 1);
23
24             return startBar;
25         }
26
27         protected override void Execute()
28         {
29             ClearDebug();
30             PrintDebug("Первый номер бара текущего дня = " + FirstBarXDaysAgo(1));
31             PrintDebug("Первый номер бара вчерашнего дня = " + FirstBarXDaysAgo(2));
32         }
33     }
34 }
```

В результате выполнения такого кода Вы получите окно, в котором отобразится информация о номере первого бара текущего дня и первого бара вчерашнего дня. Вот как это будет выглядеть:



Иллюстрация работы метода IntradayBarNumber()

Теперь рассмотрим следующий вопрос:

Как: Найти номер бара, который соответствует конкретной дате

Используйте метод ConvertDateToBar() объекта Bars. До тех пор, пока Вы уверены в том, что конкретная дата будет найдена на графике, удобно установить значение для параметра exactMatch = false. К примеру, для того, чтобы найти первый бар внутри дня для конкретной даты (в том случае, если параметр exactMatch равен false) Вам нужно всего лишь указать дату, не указывая при этом время.

Пример №02: (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть)

```
1 protected override void Execute()
2 {
3     ClearDebug();
4
5     int year = 2012;
6     int month = 02;
7     int day = 16;
8
9 }
```



Видео ВЕБИНАРОВ



Дмитрия Власова

E-mail курсы



ХВАТИТ ЖАЛЕТЬ СВОЮ МТС





МТС под микроскопом:
WLMA System
Подробный разбор торговой системы WLMA System
3900 руб.
[В корзину](#)

Рубрики

Выберите рубрику

Обновления:

 Новые посты по Email
 Новые посты по RSS
 Комментарии по RSS
 Следим на Twitter

Метки

C#

HowTo: Wealth-Lab

Видео

Индикаторы

МТС

Торговые роботы

Юмор

алготрейдинг

мероприятия

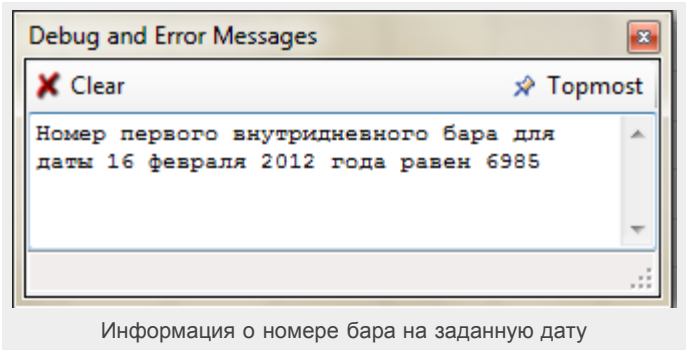
обучение

Управление

- Регистрация
- Вход
- RSS Записей
- RSS Комментариев
- WordPress.org

```
10     DateTime dt = new DateTime(year, month, day);
11
12     PrintDebug("Номер первого внутридневного бара для даты 16 февраля 2012 года равен
13 " + Bars.ConvertDateToBar(dt, false));
}
```

Как видите, выскакивает информационное окошко, которое и сообщает нам номер бара, соответствующий заданной дате.



Вы можете более подробно посмотреть описание метода `DateTimeToBar()` в QuickRef (нажмите F11 в редакторе кода WealthLab).

На сегодня всё. Следующая часть инструкции будет посвящена **последовательностям OHLC/V** и объекту **DataSeries**.

Если Вам понравилась эта статья - подпишитесь на **новые СТАТЬИ по RSS** или получайте наши **новые СТАТЬИ на E-mail**:

Рекомендую почитать также:

- Синхронизация данных при построении торговых стратегий с несколькими финансовыми инструментами
- Проектирование и построение индикаторов в Wealth-Lab
- Использование при программировании торговых стратегий торговых приказов AtStop и AtLimit
- Создание многопозиционных стратегий в Wealth-Lab
- Что такое алерты и как их использовать в Wealth-Lab

 РУБРИКИ: **THE BARS OBJECT**, **WEALTHSCRIPT**

Комментариев: 3

Инструкция по программированию торговых стратегий в WealthLab | Финансовая лаборатория пишет:

3 мая 2012 в 13:58

[...] Объект Bars в Wealth Lab – что такое бары в Велс Лаб Главная Что такое уровни пивот и для чего они [...]

Ответить

Ответы на вопросы по программированию торговых стратегий в Wealth-Lab | Финансовая лаборатория пишет:

16 мая 2012 в 12:57

[...] 4. Как с помощью кода посчитать количество баров на графи... [...]

Ответить

макс пишет:

9 декабря 2012 в 13:55

А где взять свечки с открытым интересом? на сайте финама нет

Ответить



Оставить комментарий

Имя (Обязательно)

Почта (Обязательно)

Вэб-сайт

Получать новые комментарии по электронной почте. Вы можете [подписаться](#) без комментирования.

© 2012, Финансовая лаборатория. Все права защищены.
Любое использование материалов допускается только с указанием активной ссылки на источник.



448
240
185



В день мой блог
посещает **77**
человек,
сегодня — **121**



рейтинг
757124
509
193



Анонсы
BY FEEDBURNER

Подписка

- [RSS](#)
- [Twitter](#)
- [ВКонтакте](#)
- [facebook](#)

Сайт

- [Создать блог](#)
- [Содержание](#)
- [О блоге](#)
- [Карта](#)

OHLC/V Series – последовательности цен при программировании в Wealth Lab 6.3

 **АВТОР ДМИТРИЙ ВЛАСОВ**  **МАЯ - 3 - 2012**  **КОММЕНТАРИЕВ: 6**

Сегодня речь пойдет о том, как при создании торговой стратегии с помощью WealthScript в программе Велс Лаб получить доступ к ценам исследуемого инструмента (OHLC), к его объемам. Также будет дано понятие серий (последовательностей) и класса DataSeries.

Серии цен открытия, закрытия, максимумов и минимумов, а также объемов (OHLC/V) образуют первоначальный график финансового инструмента. Доступ к этим данным можно получить с помощью одноименных свойств класса Bars.



Сложно говорить о сериях OHLC/V в барах без предварительного представления о типе данных класса DataSeries. Эта тема обсуждается далее, ну а сейчас достаточно будет упомянуть о типе значений класса DataSeries. В DataSeries применяется тип double - с плавающей точкой двойной точности.

Как: получить доступ к значениям OHLC/V

Для того, чтобы получить доступ к значениям конкретного бара, как и обычно в классе DataSeries, нужно использовать квадратные скобки [].

Пример кода показывает подробное описание метода нахождения средних цен закрытия последних пяти баров на графике (позднее мы изучим как можно выполнить ту же самую задачу с помощью индикаторов).

Пример ([как выполнить пример приведенного кода](#))...

```
Код примера на C# (кликните чтобы посмотреть)
1  protected override void Execute()
2
3  {
4      const int period = 5;
5      double sum = 0d;
6
7      for(int bar = Bars.Count - 1; bar >= Bars.Count - period; bar--)
8
9          sum = sum + Bars.Close[bar];
10
11      double avg = sum / period;
12
13      PrintDebug("Среднее значение " + period + "-ти последних цен закрытия равняется:
14      " + avg);
15  }
```

В результате выполнения данного кода выскочит окошко, которое сообщит о рассчитанном среднем значении цен закрытия 5-ти последних баров.

[RSS товары](#)



[RSS новости](#)



[Twitter](#)



Последние комментарии

макс: А где взять свечки с открытым интересом? на сайте финама нет



ring10: Добрый день Дмитрий Как сделать, чтобы открывался 1 велф лаб? Если открываешь под пользователем, т [...]



Дмитрий Власов: Буян, рад, что Вам пригодились материалы данной статьи!



Максим: Сам себе отвечу. protected override void Execute() { Bars rts = GetExternalSymbol("RTS", t [...]



Буян: Спасибо Дмитрий за статью! Мне удалось разработать хорошую рабочую торговую систему, используя возмо [...]



Последние посты



Бесплатный вебинар: 7 смертных грехов при оптимизации торговых систем

НОЯБРЯ - 5 - 2012



Отзывы участников курса до результата Wealth-Lab. С нуля до первой торговой системы за 5

дней.
НОЯБРЯ - 3 - 2012



Быстрее, Выше, Сильнее: Обновление Wealth-Lab до новой версии 6.4

ОКТАБРЯ - 23 - 2012



Курс до результата: Wealth-Lab – от нуля до первой торговой системы за 5 дней

ОКТАБРЯ - 23 - 2012



Обзорная экскурсия по достопримечательностям программы Wealth-Lab

ОКТАБРЯ - 20 - 2012

[Берите](#)



Серии Open, High, Low, Close и Volume уже существуют в базовом классе WealthScript, поэтому если Вы хотите сослаться на одну из этих 5-ти последовательностей, то Вы не обязательно должны уточнять, что они являются свойствами класса Bars. Другими словами, написание Close будет совершенно равнозначно написанию Bars.Close, High эквивалентно Bars.High и т.д.

Как: отобразить на графике серии OHLC/V

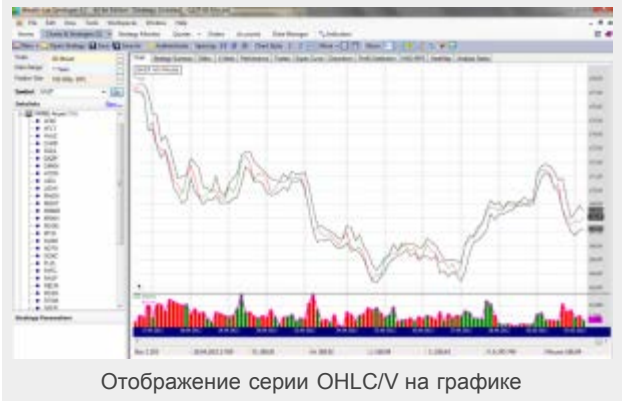
Обычно нет необходимости специально отображать на графике последовательности OHLC/V, так как программа уже делает это автоматически, когда Вы выбираете стиль графика. Однако, предположим, что Вы захотели отобразить вокруг линейного графика, построенного по ценам закрытия верхние и нижние "полосы" максимальных и минимальных цен. Здесь представлен пример того, как с помощью функций WealthScript, используя доступ к значениям серий High, Low и Volume реализовать эту задачу.

Пример (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть)

```
1 protected override void Execute()
2
3 {
4     // Отображаем на графике цен "конверт" по ценам High и Low
5
6     PlotSeries(PricePane, High, Color.Gray, WealthLab.LineStyle.Solid, 2);
7     PlotSeries(PricePane, Low, Color.Gray, WealthLab.LineStyle.Solid, 2);
8
9     // Заключаем в "конверт" гистограмму объема
10
11     PlotSeries(VolumePane, Volume, Color.Fuchsia, WealthLab.LineStyle.Solid, 2);
12 }
```

В итоге получится вот такая картина:



Для того, чтобы больше узнать о построении графика смотри здесь...

Как: Поменять значения серий OHLC/V

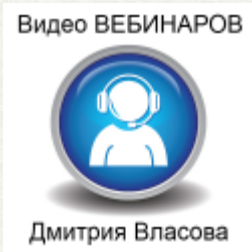
В некоторых случаях Вам может понадобиться скорректировать значения первоначальных последовательностей OHLC/V "на лету", без необходимости сохранять результат такой коррекции. К примеру, это может понадобиться для того, чтобы сделать первоначально отрицательные значения положительными.

Пример, приведенный ниже проверяет самое маленькое значение и в случае, если это число оказывается отрицательным, прибавляет ко всем значениям всех последовательностей некоторое целое число, которое превратит все значения серий OHLC/V в неотрицательные числа.

Пример (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть)

```
1 protected override void Execute()
2
3 {
4
5     //
6     Если самое маленькое число является отрицательным, нужно увеличить его,
7     сделав положительным, одновременно увеличив все прочие серии OHLC/V
8
9     double low = Lowest.Value(Bars.Count - 1, Low, Bars.Count);
10
11     if (low < 0)
12 }
```



МТС под микроскопом: WLMA System

Подробный разбор торговой системы WLMA System

3900 руб.

В корзину

Рубрики

Выберите рубрику

Обновления:

- Новые посты по Email
- Новые посты по RSS
- Комментарии по RSS
- Следим на Twitter

Метки

C# HowTo: Wealth-Lab
Видео Индикаторы МТС Торговые
роботы Юмор алготрейдинг
мероприятия обучение

Управление

Регистрация

Вход

RSS Записей

RSS Комментарии

WordPress.org

```
13 {
14     low = Math.Ceiling(Math.Abs(low));
15
16     for (int bar = 0; bar < Bars.Count; bar++)
17
18     {
19         Open[bar] = Open[bar] + low;
20         High[bar] = High[bar] + low;
21         Low[bar] = Low[bar] + low;
22         Close[bar] = Close[bar] + low;
23     }
24 }
```

Как результат, все значения серий станут неотрицательными.

На сегодня все. В следующей статье будет показано, как при программировании стратегий в WealthScript можно **получить доступ к шкале, интервалам а также к дате и времени.**

Если Вам понравилась эта статья - подпишитесь на **новые СТАТЬИ по RSS** или получайте наши **новые СТАТЬИ на E-mail:**

Рекомендую почитать также:

- **Объект Bars в Wealth Lab – что такое бары в Велс Лаб**
- **Symbol Info – подробная информация о торговом инструменте в программе WealthLab**

 РУБРИКИ: **THE BARS OBJECT**, **WEALTHSCRIPT**

Комментариев: 6

Объект Bars в Wealth Lab - что такое бары в Велс Лаб | Финансовая лаборатория пишет:

6 мая 2012 в 21:01

[...] [...]

Ответить

DataSeries - основная структура данных в Wealth-Lab и математические операции при программировании торговых стратегий | Финансовая лаборатория пишет:

7 мая 2012 в 16:13

[...] же ссылаться на другой массив DataSeries или на объект Bars. Последовательности OHLC/V и все индикаторы являются экземплярами класса [...]

Ответить

Ответы на вопросы по программированию торговых стратегий в Wealth-Lab | Финансовая лаборатория пишет:

14 мая 2012 в 13:34

[...] [...]

Ответить

Kartman пишет:

24 мая 2012 в 16:01

А, что это (" , >) работает в WL?

Ответить



Дмитрий Власов пишет:

24 мая 2012 в 16:54

Глюк какой-то в коде был. Поправил. Так вроде все нормально.

Ответить



Инструкция по программированию торговых стратегий в WealthLab | Финансовая лаборатория пишет:

20 июня 2012 в 12:19

[...] 2. OHLC/V Series – последовательности цен при программировании ... [...]

Ответить

Оставить комментарий

Имя (Обязательно)

Почта (Обязательно)

Вэб-сайт

Получать новые комментарии по электронной почте. Вы можете **подписаться** без комментирования.

© 2012, Финансовая лаборатория. Все права защищены.
Любое использование материалов допускается только с указанием активной ссылки на источник.



448 👁
240 📄
185 👤



В день мой блог
посещает **77**
человек,
сегодня **121**



РЕЙТИНГ
757125 510
193



BY FEEDBURNER

Подписка

- [RSS](#)
- [Twitter](#)
- [ВКонтакте](#)
- [facebook](#)

Сайт

- [Создать блог](#)
- [Содержание](#)
- [О блоге](#)
- [Карта](#)

Как использовать при программировании стратегий шкалу, интервал и дату

 АВТОР ДМИТРИЙ ВЛАСОВ  МАЯ - 4 - 2012  КОММЕНТАРИЕВ: 7

Сегодня речь пойдет о том, как использовать при программировании стратегий в Wealth Lab таких важных инструментов, как шкала сжатия времени (таймфрейм) а также как программно можно получать доступ к дате, времени, текущей шкале и интервалу времени. Интуитивно все это понятно, но для того, чтобы объяснить это компьютеру при построении кода на языке WealthScript нужно со всем этим предметно разобраться.



Свойство Bars.Scale возвращает значение временного масштаба для активного инструмента. Текущий временной масштаб (таймфрейм) Вы можете временно изменить, применив для этого одну из функций языка WealthScript (смотри раздел мультитаймфреймовый анализ). Значениями временной шкалы (возможными таймфреймами) могут быть тики, секунды, минуты, дни, недели, месяцы, кварталы или годы. Для внутрисуточного масштаба свойство BarInterval возвращает целое числовое значение минут, секунд или тиков.

Как: программно узнать текущий таймфрейм или интервал

По возможности всегда старайтесь написать код торговой системы таким образом, чтобы торговую стратегию можно было протестировать на различных временных масштабах и интервалах без дополнительного редактирования кода.

Пример ([как выполнить пример приведенного кода](#))...

Приведенный пример сначала проверяет, является ли текущая шкала (Bars.Scale) минутной (BarScale.Minute). Если это действительно так, то используя свойство Bars.BarInterval определяется время для выхода из текущей позиции в конце дня (на открытии последнего бара окончания сессии).

Код примера на C# (кликните чтобы посмотреть)

```
1  protected override void Execute()
2
3  {
4      if (Bars.Scale != BarScale.Minute)
5      {
6          //Если выбран не минутный таймфрейм, то появляется следующее предупреждение:
7          DrawLabel(PricePane, "Пожалуйста, выберите минутный масштаб для графика",
8              Color.Red);
9
10         return;
11     }
12
13     /* Предположим для простоты, что выбран 60 минутный интервал (или меньше)
14        определяем время выхода при условии, что сессия заканчивается в 16:00
15        (если заканчивается в 18:40, то нужно писать 1800 + 40) */
16
17     int exitTime = 1500 + 60 - Bars.BarInterval;
18
19     /* Находим период, эквивалентный двум последним дням (сколько свечей нужного
20        интервала составляют 2 дня)
21        при условии, что день длится 6,5 часов (или 780 минут) 6,5 часов (длится
22        сессия) переводим в минуты, затем в дни
23        после чего в зависимости от интервала подсчитываем количество свечей */
24
25     int Period = (int) (6.5 * 60 * 2) / Bars.BarInterval;
```

RSS товары



RSS новости



Twitter



Последние комментарии

макс: А где взять свечки с открытым интересом? на сайте финама нет



ring10: Добрый день Дмитрий Как сделать, чтобы открывался 1 велф лаб? Если открываешь под пользователем, т [...]



Дмитрий Власов: Буян, рад, что Вам пригодились материалы данной статьи!



Максим: Сам себе отвечу. protected override void Execute() { Bars rts = GetExternalSymbol("RTS", t [...]



Буян: Спасибо Дмитрий за статью! Мне удалось разработать хорошую рабочую торговую систему, используя возмо [...]



Последние посты



Бесплатный вебинар: 7 смертных грехов при оптимизации торговых систем

НОЯБРЯ - 5 - 2012



Отзывы участников курса до результата Wealth-Lab. С нуля до первой торговой системы за 5

дней.

НОЯБРЯ - 3 - 2012



Быстрее, Выше, Сильнее: Обновление Wealth-Lab до новой версии 6.4

ОКТАБРЯ - 23 - 2012



Курс до результата: Wealth-Lab – от нуля до первой торговой системы за 5 дней

ОКТАБРЯ - 23 - 2012



Обзорная экскурсия по достопримечательностям программы Wealth-Lab

ОКТАБРЯ - 20 - 2012

Берите

```
28     DataSeries breakoutHigh = Highest.Series(High, Period);
29
30     PlotStops();
31
32     for(int bar = Period; bar < Bars.Count; bar++)
33     {
34         if (IsLastPositionActive)
35         {
36             Position p = LastPosition;
37
38             int currentTime = Date[bar].Hour * 100 + Date[bar].Minute;
39
40             if (currentTime >= exitTime)
41
42                 SellAtMarket(bar + 1, p, "Market Close");
43
44             else
45
46                 SellAtLimit(bar + 1, p, p.EntryPrice * 1.02, "Profit Target");
47         }
48     }
49     else
50     {
51         {
52             // Войти в позицию при прорыве уровня, но не входить на первом баре дня
53
54             if (!Bars.IsLastBarOfDay(bar))
55
56                 BuyAtStop(bar + 1, breakoutHigh[bar]);
57         }
58     }
59 }
```

Как видите, вне зависимости от того, какой интервал времени (таймфрейм) Вы используете - 5 минут, 10 минут, 30 или даже 60 минут - позиция закрывается точно на последнем баре текущего дня.

Как: получить доступ к дате и времени

Свойство Date объекта Bars (Bars.Date) предоставляет возможность доступа к объекту DateTime.

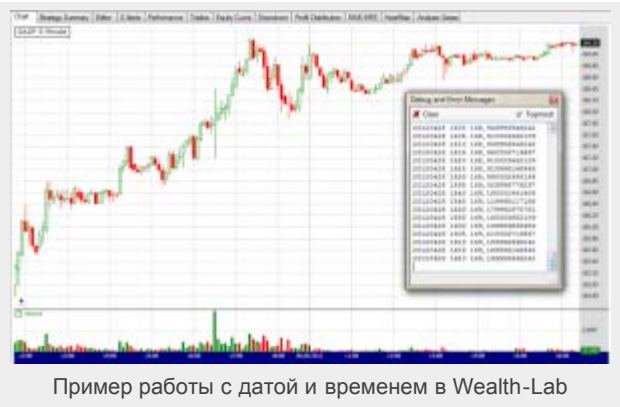
Пример (как выполнить пример приведенного кода)...

Предыдущий пример продемонстрировал, как получить доступ отдельно к часу и отдельно к минуте, которые соответствуют конкретному бару. Этот же пример демонстрирует как отобразить на экране точные дату и время, соответствующие цене закрытия бара в кратком формате.

Код примера на C# (кликните чтобы посмотреть)

```
1     protected override void Execute()
2
3     {
4         ClearDebug();
5
6         for(int bar = 20; bar < Bars.Count; bar++)
7
8         {
9             PrintDebug(String.Format( "{0:yyyyMMdd HHmm }", Date[bar] ) + Close[bar]);
10        }
11    }
```

В результате выполнения данного кода Вы получите текстовую таблицу, состоящую из 3-х столбцов в первом из которых будет отображена дата в указанном формате (год, месяц, число), а затем время (Часы и минуты).



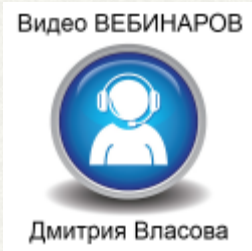
Пример работы с датой и временем в Wealth-Lab

Вы, конечно же сможете выбрать любой формат отображения даты и времени.

На сегодня всё. В следующий раз будем изучать как можно получить доступ к свойствам торгуемого финансового инструмента, т.е. более подробно разберем SymbolInfo.

Нет похожих сообщений.

РУБРИКИ: THE BARS OBJECT, WEALTHSCRIPT



МТС под микроскопом: WLMA System

Подробный разбор торговой системы WLMA System
3900 руб.

В корзину

Рубрики

Выберите рубрику

Обновления:

- Новые посты по Email
- Новые посты по RSS
- Комментарии по RSS
- Следим на Twitter

Метки

C# HowTo: **Wealth-Lab**
Видео Индикаторы МТС Торговые роботы Юмор алготрейдинг мероприятия обучение

Управление

- Регистрация
- Вход
- RSS Записей
- RSS Комментариев
- WordPress.org

Комментариев: 7

ОНLС/V Series - последовательности цен при программировании в Wealth Lab 6.3 | Финансовая лаборатория пишет:

5 мая 2012 в 14:02

[...] [...]

Ответить

Ответы на вопросы по программированию торговых стратегий в Wealth-Lab | Финансовая лаборатория пишет:

17 мая 2012 в 13:41

[...] [...]

Ответить

Сергей пишет:

25 мая 2012 в 20:36



Подскажите. Почему некоторые примеры (не все) при компиляции показывают ошибку? Копирую в WLD 6.3.14.0.

Ответить

Дмитрий Власов пишет:

26 мая 2012 в 10:22



Сергей, все примеры, которые есть здесь я лично опробовал на WLD той же версии, что и у Вас. У меня все работало без сбоев. Можешь показать – при выполнении какого примера появляется ошибка?

Кстати, вот [здесь](#) можно посмотреть как выполнять примеры. Иногда в тексте дан лишь кусок примера. На это нужно обратить внимание.

Ответить

Antrax пишет:

13 июня 2012 в 12:17



```
//int Period = ( (6.5 * 60 * 2) / Bars.BarInterval);
```

Вот в этой строчке кода косяк, явное преобразование типов не может быть выполнено-поэтому и код не компилируется.

Ответить

Дмитрий Власов пишет:

13 июня 2012 в 12:57



Да, эту строчку нужно было заменить на
`int Period = (int)(6.5 * 60 * 2) / Bars.BarInterval;`

Тогда все будет нормально.

В коде внес изменение.

Ответить

Инструкция по программированию торговых стратегий в WealthLab | Финансовая лаборатория пишет:

20 июня 2012 в 12:23

[...] [...]

Ответить

Оставить комментарий

Имя (Обязательно)

Почта (Обязательно)

Вэб-сайт

Получать новые комментарии по электронной почте. Вы можете [подписаться](#) без комментирования.

© 2012, Финансовая лаборатория. Все права защищены.
Любое использование материалов допускается только с указанием активной ссылки на источник.



448
240
185



В день мой блог
посещает **77**
человек,
сегодня — **123**



рейтинг
757127512
193



Анонсы
BY FEEDBURNER

Подписка

- [RSS](#)
- [Twitter](#)
- [ВКонтакте](#)
- [facebook](#)

Сайт

- [Создать блог](#)
- [Содержание](#)
- [О блоге](#)
- [Карта](#)

Symbol Info – подробная информация о торговом инструменте в программе WealthLab

 АВТОР ДМИТРИЙ ВЛАСОВ  МАЯ - 5 - 2012  КОММЕНТАРИЕВ: 2

В блоге "Финансовая лаборатория" уже была статья, в которой мы подробно рассматривали - [как настроить Symbol Manager в Wealth Lab](#). Однако мало настроить и знать, что каждый финансовый инструмент специфичен. Настройка специфики каждого финансового инструмента может производиться вручную в Symbol Info Manager. Однако важно научиться - как делать это с помощью кода при программировании стратегий, а также как получить доступ к уже имеющейся информации программно. Об этом и будет сегодняшняя статья.



Напомню Вам о том, в чем специфика финансовых инструментов:

- один финансовый инструмент является акцией, другой фьючерсом.
- один имеет лот содержащий 1-ну акцию, другой - тысячу акций.
- Некоторые котировки акций предполагают наличие 2-х знаков после запятой, другие 5-ти.
- Отличаются акции также минимальным шагом цены.
- Фьючерсы имеют разное гарантийное обеспечение (ГО)

Без лишних слов перейдем непосредственно к заявленной теме.

Как: получить доступ к краткому названию текущего финансового инструмента (тикера) или к полному названию компании.

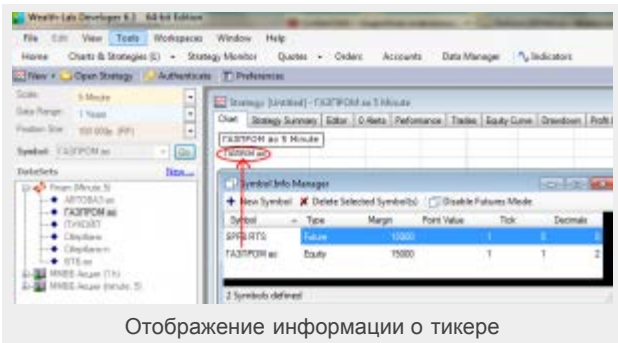
Тикер финансового инструмента (Symbol) также как и полное название компании (SecurityName) являются текстовыми свойствами объекта (Bars).

Пример ([как выполнить пример приведенного кода](#))...

Код примера на C# (кликните чтобы посмотреть)

```
1 protected override void Execute()
2
3 {
4
5     DrawLabel(PricePane, Bars.Symbol + ": " + Bars.SecurityName, Color.Black);
6
7 }
```

В результате исполнения этого кода на графике в верхнем левом углу отобразиться название тикера торгуемого инструмента, которое мы указывали в Simbol Info Manager.



Отображение информации о тикере

RSS товары



RSS новости



Twitter



Последние комментарии

макс: А где взять свечки с открытым интересом? на сайте финама нет



ring10: Добрый день Дмитрий Как сделать, чтобы открывался 1 велф лаб? Если открываешь под пользователем, т [...]



Дмитрий Власов: Буян, рад, что Вам пригодились материалы данной статьи!



Максим: Сам себе отвечу. protected override void Execute() { Bars rts = GetExternalSymbol("RTS", t [...]



Буян: Спасибо Дмитрий за статью! Мне удалось разработать хорошую рабочую торговую систему, используя возмо [...]



Последние посты



Бесплатный вебинар: 7 смертных грехов при оптимизации торговых систем

НОЯБРЯ - 5 - 2012



Отзывы участников курса до результата Wealth-Lab. С нуля до первой торговой системы за 5

дней.
НОЯБРЯ - 3 - 2012



Быстрее, Выше, Сильнее: Обновление Wealth-Lab до новой версии 6.4

ОКТАБРЯ - 23 - 2012



Курс до результата: Wealth-Lab – от нуля до первой торговой системы за 5 дней

ОКТАБРЯ - 23 - 2012



Обзорная экскурсия по достопримечательностям программы Wealth-Lab

ОКТАБРЯ - 20 - 2012

Берите

Примерно таким же способом можно получить доступ и к прочей информации о торуемом инструменте.

Как: получить доступ к величине минимального шага цены, если торгуемый инструмент является фьючерсом

Такие понятия как минимальный шаг цены (Tick), гарантийное обеспечение (Margin), цена пункта (PointValue), обычно ассоциируются с фьючерсами. Информация о всех этих свойствах содержится в классе SymbolInfo, который сам по себе является членом класса Bars. В Wealth-Lab версии 6, для типа финансового инструмента акции (Equity) по умолчанию установлена величина минимального шага цены равная 0,01.

Для того, чтобы определить - какой инструмент в настоящее время тестируется - фьючерс или акция - используйте свойство объекта SymbolInfo под названием SecurityType. Это свойство может принимать три значения: Equity (акция), Future (фьючерс), MutualFund (взаимный фонд).

Пример (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть)

```
1  protected override void Execute()
2
3  {
4
5      SymbolInfo si = Bars.SymbolInfo;
6
7      DrawLabel(PricePane, si.Symbol + " Tick = " + si.Tick, Color.Blue);
8
9      int bar = Bars.Count - 1;
10
11     /*Создать алерт на покупку по цене на 1 минимальный шаг цены больше цены
12 закрытия для фьючерсов
13 и на 5 минимальных шагов цены выше цены закрытия для всех прочих типов*/
14
15     if ( si.SecurityType == WealthLab.SecurityType.Future )
16
17         BuyAtLimit(bar + 1, Close[bar] + si.Tick);
18
19     else
20
21         BuyAtLimit(bar + 1, Close[bar] + si.Tick * 5);
22 }
```

Данный пример показывает, как в зависимости от типа инструмента могут выполняться различные условия работы с ним.

Возможно, в процессе разработки и тестирования различных стратегий Вы решите торговать более сложные стратегии - такие, как арбитраж и парный трейдинг. В этом случае, перед Вами непременно возникнет такой вопрос:

Как: получить доступ к барам второго (неосновного) финансового инструмента

Здесь необходимо пояснить, что первичным инструментом принято называть тот инструмент, по которому Вы кликнули, либо в более общем смысле - тот инструмент, который является рабочим в данной стратегии.

По умолчанию Вы работаете с барами первичного инструмента, однако, Вы можете получить доступ к барам любого другого инструмента, для чего нужно вызвать SetContext. Это позволит получить доступ к объекту Bars второго финансового инструмента, к его свойствам и методам. Но что особенно полезно, Вы получите возможность явно совершать сделки с этим вторым финансовым инструментом.

Для того, чтобы поподробнее узнать об этом - смотрите тему: торговые стратегии с несколькими финансовыми инструментами (Multi-Symbol Strategies).

На сегодня это все. В следующий раз поговорим об объекте **DataSeries и его операторах**. Чтобы не пропустить новую информацию - не забывайте подписаться на RSS поток нашего блога.

Если Вам понравилась эта статья - подпишитесь на **новые СТАТЬИ по RSS** или получайте наши **новые СТАТЬИ на E-mail**:

Рекомендую почитать также:

- Объект Bars в Wealth Lab – что такое бары в Велс Лаб
- OHLC/V Series – последовательности цен при программировании в Wealth Lab 6.3

РУБРИКИ: THE BARS OBJECT, WEALTHSCRIPT



Видео ВЕБИНАРОВ



Дмитрия Власова



МТС под микроскопом: WLMA System

Подробный разбор торговой системы WLMA System

3900 руб.

В корзину

Рубрики

Выберите рубрику

Обновления:



Новые посты по Email



Новые посты по RSS



Комментарии по RSS



Следим на Twitter

Метки

C# HowTo: Wealth-Lab
Видео Индикаторы МТС Торговые
роботы Юмор алготрейдинг
мероприятия обучение

Управление

Регистрация

Вход

RSS Записей

RSS Комментариев

WordPress.org

Комментариев: 2

Как использовать при программировании стратегий шкалу, интервал и дату | Финансовая лаборатория пишет:

5 мая 2012 в 16:34

[...] Symbol Info – подробная информация о торговом инструменте... Главная OHLC/V Series – последовательности цен при [...]

Ответить

Как использовать при программировании стратегий шкалу, интервал и дату | Финансовая лаборатория пишет:

5 мая 2012 в 16:34

[...] Symbol Info – подробная информация о торговом инструменте... Главная OHLC/V Series – последовательности цен при [...]

Ответить

Ответы на вопросы по программированию торговых стратегий в Wealth-Lab | Финансовая лаборатория
 пишет:

15 мая 2012 в 15:32

[...] [...]

Ответить

Оставить комментарий

Имя (Обязательно)

Почта (Обязательно)

Вэб-сайт

Получать новые комментарии по электронной почте. Вы можете [подписаться](#) без комментирования.

© 2012, Финансовая лаборатория. Все права защищены.
 Любое использование материалов допускается только с указанием активной ссылки на источник.



448

240

185



В день мой блог посещает 77 человек, сегодня

123



РЕЙТИНГ

757128

513

193



BY FEEDBURNER

Подписка

[RSS](#)
[Twitter](#)
[ВКонтакте](#)
[facebook](#)

Сайт

[Создать блог](#)
[Содержание](#)
[О блоге](#)
[Карта](#)

DataSeries – основная структура данных в Wealth-Lab и математические операции при программировании торговых стратегий

 **АВТОР** ДМИТРИЙ ВЛАСОВ  **МАЯ - 7 - 2012**  **КОММЕНТАРИЕВ: 5**

DataSeries - является базовой структурой данных, которая работает с историческими рядами данных. По своей внутренней организации DataSeries является числовым массивом (List<double>), ассоциированным с временным массивом (List<DateTime>). Массив DataSeries может управляться в экземпляре другого массива, или же ссылаться на другой массив DataSeries или на объект Bars. **Последовательности OHLC/V** и все индикаторы являются экземплярами класса DataSeries.

Характеристика класса DataSeries:

DataSeries является последовательностью чисел с плавающей точкой типа double.
Все массивы DataSeries содержат точно такое же количество значений, что и бары на графике.

Исключение: Только если Вы специально, работая со вторым финансовым инструментом, вызывая методы SetContext() либо GetExternalSymbol() присвоите параметру синхронизации значение false - возвращаемые этими методами последовательности (Series) могут иметь другое по сравнению с барами на графике количество значений.

Операторы при работе с DataSeries

Как и любыми прочими данными, значениями, содержащимися в экземплярах класса DataSeries можно оперировать. Для этого предназначены специальные операторы. Давайте с ними и познакомимся.



Как: оперировать (Умножать, Делить, Добавлять или Вычитать) несколькими DataSeries

Выполнение математических операций над DataSeries в Wealth-Lab Developer версии 6.* интуитивно очень понятно. Просто выполняйте любые математические операции над DataSeries - как будто они являются самыми обычными числами типа double. Вы можете умножить или разделить DataSeries одну на другую, сложить их между собой, вычесть одну DataSeries из другой DataSeries. При этом как результат таких операций Вы получите новую DataSeries.

На самом деле, когда производят операции над двумя DataSeries, операции производятся над значениями из одинаковых по дате и времени баров в каждой из последовательностей. Когда же происходят операции с DataSeries и каким либо числом (константой), происходит операция в каждом из элементов массива DataSeries.

Когда происходит операция деления одной DataSeries на другую, WealthLab подавляет ошибку деления на ноль. Данная операция переопределена таким образом, что результатом деления на ноль является

RSS товары



RSS новости



Twitter



Последние комментарии

макс: А где взять свечки с открытым интересом? на сайте финама нет



ring10: Добрый день Дмитрий Как сделать, чтобы открывался 1 велф лаб? Если открываешь под пользователем, т [...]



Дмитрий Власов: Буян, рад, что Вам пригодились материалы данной статьи!



Максим: Сам себе отвечу. protected override void Execute() { Bars rts = GetExternalSymbol("RTS", t [...]



Буян: Спасибо Дмитрий за статью! Мне удалось разработать хорошую рабочую торговую систему, используя возмо [...]



Последние посты



Бесплатный вебинар: 7 смертных грехов при оптимизации торговых систем

НОЯБРЯ - 5 - 2012



Отзывы участников курса до результата Wealth-Lab. С нуля до первой торговой системы за 5

дней.
НОЯБРЯ - 3 - 2012



Быстрее, Выше, Сильнее: Обновление Wealth-Lab до новой версии 6.4

ОКТАБРЯ - 23 - 2012



Курс до результата: Wealth-Lab – от нуля до первой торговой системы за 5 дней

ОКТАБРЯ - 23 - 2012



Обзорная экскурсия по достопримечательностям программы Wealth-Lab

ОКТАБРЯ - 20 - 2012

Берите

не ошибка, а нулевое значение (od) для соответствующего бара.

Пример (как выполнить пример приведенного кода)...

Используя этот пример мы продемонстрируем - как можно сложить два DataSeries (High и Low) друг с другом.

Код примера на C# (кликните чтобы посмотреть)

```
1 // Создание массива DataSeries, содержащего значения средних цен (Average Price)
2
3 protected override void Execute()
4
5 {
6     DataSeries avgPrice = (High + Low) / 2;
7     avgPrice.Description = "Average Price";
8     PlotSeries(PricingPane, avgPrice, Color.Blue, WealthLab.LineStyle.Solid, 1);
9 }
```

Этот пример просто показывает, как можно создавать нужный Вам массив DataSeries. В реальной работе при построении торговых систем Вы не должны будете самостоятельно создавать массив со значениями средней цены. Намного проще и правильнее будет воспользоваться уже готовым стандартным индикатором (AveragePrice).

Еще один пример использования математических операция над DataSeries

Как: создать последовательность, содержащую типичную цену ("Typical Price")

Типичная цена ("Typical Price") это ничто иное, как среднее максимальной, минимальной цены и цены закрытия соответствующего бара. Т.е. мы можем сначала сложить 3 соответствующих DataSeries и лишь затем поделить сумму на 3.

Пример (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть)

```
1 protected override void Execute()
2
3 {
4     DataSeries typPrice = (High + Low + Close) / 3;
5     typPrice.Description = "Typical Price";
6     PlotSeries(PricingPane, typPrice, Color.Black, WealthLab.LineStyle.Solid, 1);
7 }
```

Точно также, как и в предыдущем примере, Вы можете воспользоваться стандартным индикатором AveragePriceC, а не изобретать велосипед заново и не создавать самостоятельно данный индикатор.

Теперь изучим очень удобный в практической работе по программированию торговых стратегий в Wealth-Lab пример:

Как: Сдвинуть значения DataSeries вправо или влево по шкале времени

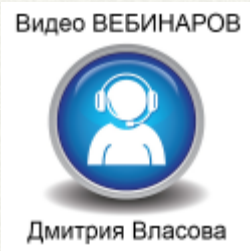
Вы можете "сдвинуть" DataSeries вправо (>>) либо влево (<<) по шкале времени. Для этого просто используйте интуитивно понятные операторы >> и <<, которые сдвигают каждую из "ячеек" массива DataSeries вправо или влево соответственно. При этом одно из значений массива DataSeries с одной из сторон в результате сдвигания "вылезет" за границы графика, при этом на другом конце графика крайней ячейке присваивается нулевое значение.

Сдвиг вправо по шкале времени (>>) является очень полезным приемом, т.к. часто требуется принимать решение на основе анализа цен предыдущего периода. В то время, как сдвиг по шкале времени влево (<<) подразумевает "заглядывание вперед", т.е. использование данных, которые появятся только в будущем. Именно поэтому сдвиг влево не должен использоваться при разработке правил торговых стратегий.

Пример (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть)

```
1 protected override void Execute()
2
3 {
4     // Отобразить на графике на сколько процентов текущее значение отличается от
5     значения 5 баров назад (RoC)
6
7     DataSeries delayedClose = Close >> 5; //сдвигаем значение цен закрытия на 5
8     баров вправо
9
10    DataSeries pctChange = 100 * ( Close / delayedClose - 1); //рассчитываем процент
11    изменения
12
13    pctChange.Description = "5-дневный процент изменения";
14
15    ChartPane changePane = CreatePane(40, true, true);
16
17    PlotSeries(changePane, pctChange, Color.Red, WealthLab.LineStyle.Solid, 1);
18
19    /* Мы будем изучать индикаторы чуть позже, но здесь заметим, что
20    точно такой же результат мы сможем получить если отобразим на графике
21    индикатор ROC (Rate of Change) в виде гистограммы */
22
23    DataSeries roc = ROC.Series(Close, 5);
24
25    PlotSeries(changePane, roc, Color.Black, WealthLab.LineStyle.Histogram, 1);
26 }
```



МТС под микроскопом:
WLMA System
Подробный разбор торговой системы WLMA System
3900 руб.
[В корзину](#)

Рубрики

Выберите рубрику

Обновления:

- Новые посты по Email
- Новые посты по RSS
- Комментарии по RSS
- Следим на Twitter

Метки

C# HowTo: **Wealth-Lab**
Видео Индикаторы МТС Торговые роботы Юмор алготрейдинг мероприятия обучение

Управление

Регистрация

Вход

RSS Записей

RSS Комментариев

WordPress.org

}

Как видите, сверху графика цен получился индикатор, на котором отображается процент изменения цены.



Пример использования сдвига по времени

При этом гистограмма отображает уже готовый индикатор ROC, а красная линия - наши собственные вычисления. Как Вы можете убедиться на рисунке вычисления приводят к совершенно одинаковым результатам.

Как: Получить массив `DataSet` только с положительными значениями

Для того, чтобы преобразовать все данные, содержащиеся в массиве `DataSet` в положительные числа - используйте метод `DataSet.Abs()`.

Пример (как выполнить пример приведенного кода)...

В приведенном примере модуль разницы между двумя `DataSet` добавляется к разнице, для того чтобы в тех ячейках, где значение было отрицательным появились нули. Этот пример демонстрирует работу метода `DataSet.Abs()`.

Код примера на C# (кликните чтобы посмотреть)



```
1 using System;
2 using System.Collections.Generic;
3 using System.Text;
4 using System.Drawing;
5 using WealthLab;
6 using WealthLab.Indicators;
7
8 namespace WealthLab.Strategies
9 {
10     public class MyStrategy : WealthScript
11     {
12         /*
13          * Создаем метод, который возвращает новую DataSet, в которой в
14          * соответствующих барах
15          * подсчитывается - сколько за указанный период было случаев, когда ds2 > ds1
16          * (в качестве аргументов метод принимает две различные DataSet и период)
17          */
18
19         public DataSet LessThanCountSeries(DataSet ds1, DataSet ds2, int
20 Period)
21         {
22             DataSet LTCSeries = ds2 - ds1; //подсчитывается разница
23
24             LTCSeries = LTCSeries + DataSet.Abs(LTCSeries); //если разница
25 отрицательная, то результат дает ноль в соответствующей ячейке
26
27             LTCSeries = LTCSeries / LTCSeries; //Если результат в ячейке не равен
28 нулю, то результат становится равен 1
29
30             return Sum.Series(LTCSeries, Period); //Подсчитывается количество ячеек,
31 в которых была единица
32         }
33
34         protected override void Execute()
35         {
36             //Применяя
37             созданный метод LessThanCountSeries() подсчитываем, сколько цен
38             закрытия находилось внизу 20-ти периодической скользящей средней за
39             предыдущие 20 баров.
40
41             DataSet sma = SMA.Series(Close, 20);
42             DataSet ltc = LessThanCountSeries(Close, sma, 20);
43
44             ChartPane ltcPane = CreatePane(40, true, false);
45
46             PlotSeries(ltcPane, ltc, Color.Blue, WealthLab.LineStyle.Histogram, 1);
47
48             PlotSeries(PricePane, sma, Color.Blue, WealthLab.LineStyle.Solid, 2);
49         }
50     }
51 }
```

В результате выполнения этого кода в программе Wealth-Lab 6.3 получим вот такой график.



Пример использоваия метода `DataSet.ABS()`

Как видите, здесь отображается специальный, созданный нами индикатор, который показывает сколько цен Close было ниже чем 20-ти периодичная скользящая средняя. Возможно это как-то можно использовать в торговых стратегиях в качестве фильтра либо в качестве сигнала на вход в позицию.

На этом обзор операторов при работе с DataSet заканчиваю. В следующий раз поговорим о том, **как получить доступ к значениям массива DataSet на конкретном баре** (в нужные дату и время).

Если Вам понравилась эта статья - подпишитесь на **новые СТАТЬИ по RSS** или получайте наши **новые СТАТЬИ на E-mail**:

Рекомендую почитать также:

- **Как получить доступ к единичному значению DataSet**
- **Наполнение созданных самостоятельно DataSet данными при помощи WealthScript**
- **Как получить доступ ко второму финансовому инструменту при проектировании торговых стратегий в Велс Лаб.**
- **Синхронизация данных при построении торговых стратегий с несколькими финансовыми инструментами**
- **Проектирование и построение индикаторов в Wealth-Lab**

 РУБРИКИ: **DATASERIES**, **WEALTHSCRIPT**

Комментариев: 5

Symbol Info - подробная информация о торговом инструменте в программе WealthLab | Финансовая лаборатория пишет:

7 мая 2012 в 16:29

[...] [...]

Ответить

Наполнение созданных самостоятельно DataSet данными при помощи WealthScript | Финансовая лаборатория пишет:

9 мая 2012 в 09:07

[...] в одной из предыдущих статей, например здесь в последнем [...]

Ответить

Отображение на графике в программе Велс Лаб рядов данных | Финансовая лаборатория пишет:

14 мая 2012 в 10:50

[...] Создать числовую последовательность (DataSet) и, если это необходимо, установить значение для [...]

Ответить

Ответы на вопросы по программированию торговых стратегий в Wealth-Lab | Финансовая лаборатория пишет:

14 мая 2012 в 13:35

[...] [...]

Ответить

Делаем индикаторы в Wealth-Lab своими руками | Финансовая лаборатория пишет:

18 мая 2012 в 22:55

[...] Одним из таких "самодельных индикаторов" является типичная цена (typical price), которую мы уже [...]

Ответить

Оставить комментарий

Имя (Обязательно)

Почта (Обязательно)

Вэб-сайт

Получать новые комментарии по электронной почте. Вы можете [подписаться](#) без комментирования.

© 2012, Финансовая лаборатория. Все права защищены.
Любое использование материалов допускается только с указанием активной ссылки на источник.



448
240
185



В день мой блог
посещает **77**
человек,
сегодня — **123**



РЕЙТИНГ
757129514
193



BY FEEDBURNER

Подписка

- [RSS](#)
- [Twitter](#)
- [ВКонтакте](#)
- [facebook](#)

Сайт

- [Создать блог](#)
- [Содержание](#)
- [О блоге](#)
- [Карта](#)

Как получить доступ к единичному значению DataSet

 АВТОР ДМИТРИЙ ВЛАСОВ  МАЯ - 8 - 2012  КОММЕНТАРИЕВ: 2

После того, как мы изучили основную структуру данных Wealth-Lab в виде DataSet, пришла пора узнать - как можно получить доступ к этим данным в конкретный момент времени. Сделать это, конечно если знать как, очень просто.



Как: получить доступ к информации конкретного бара

Используйте индикатор в виде квадратных скобок [] для доступа к единичному значению массива DataSet. Для этого поставьте в квадратные скобки номер конкретного бара. Мы уже рассматривали такой случай, когда говорили об основных рядах, таких как Close, Volume и т.п.

Пример ([как выполнить пример приведенного кода](#))...

Следующий пример покажет Вам, как получить самое последнее значение ряда средних цен, который мы с Вами уже создавали в предыдущем примере.

Код примера на C# (кликните чтобы посмотреть)

```
1  protected override void Execute()
2  {
3      DataSet avgPrice = (High + Low) / 2;
4      avgPrice.Description = "Средняя цена";
5      PlotSeries(PricePane, avgPrice, Color.Blue, WealthLab.LineStyle.Solid, 1);
6      int lastBarNumber = Bars.Count - 1;
7      double lastAvg = avgPrice[lastBarNumber];
8      DrawLabel(PricePane, "Значение средней цены на последнем баре = " + lastAvg,
9      Color.Black);
10 }
11
12
13
14
15
```

Рассмотрим еще один пример.

Пример ([как выполнить пример приведенного кода](#))...

Изучать технические индикаторы мы будем позже, однако уже сейчас мы укажем на то, что часто удобно получить доступ к значениям индикатора на конкретном баре непосредственно с помощью метода класса DataSet. Именно такой случай рассмотрен в этом примере. Это простейшая проверка, которая определяет не является ли значение 14-ти периодического RSI меньше 30 и если это действительно так, создает алерт на покупку одного контракта.

Код примера на C# (кликните чтобы посмотреть)

```
1  protected override void Execute()
2  {
3      int bar = Bars.Count - 1;
4      if (RSI.Series(Close, 14)[bar] < 30)
5      {
6          BuyAtMarket(bar + 1);
7      }
8  }
```

RSS товары



RSS новости



Twitter



Последние комментарии

макс: А где взять свечки с открытым интересом? на сайте финама нет



ring10: Добрый день Дмитрий Как сделать, чтобы открывался 1 велф лаб? Если открываешь под пользователем, т [...]



Дмитрий Власов: Буян, рад, что Вам пригодились материалы данной статьи!



Максим: Сам себе отвечу. protected override void Execute() { Bars rts = GetExternalSymbol("RTS", t [...]



Буян: Спасибо Дмитрий за статью! Мне удалось разработать хорошую рабочую торговую систему, используя возмо [...]



Последние посты



Бесплатный вебинар: 7 смертных грехов при оптимизации торговых систем

НОЯБРЯ - 5 - 2012



Отзывы участников курса до результата Wealth-Lab. С нуля до первой торговой системы за 5 дней.

НОЯБРЯ - 3 - 2012



Быстрее, Выше, Сильнее: Обновление Wealth-Lab до новой версии 6.4

ОКТАБРЯ - 23 - 2012



Курс до результата: Wealth-Lab – от нуля до первой торговой системы за 5 дней

ОКТАБРЯ - 23 - 2012



Обзорная экскурсия по достопримечательностям программы Wealth-Lab

ОКТАБРЯ - 20 - 2012

Берите

На сегодня все. В следующий раз более подробно рассмотрим, **как заполнить значениями самостоятельно созданные ряды данных DataSeries**.

Если Вам понравилась эта статья - подпишитесь на **новые СТАТЬИ по RSS** или получайте наши **новые СТАТЬИ на E-mail**:

Рекомендую почитать также:

- **DataSeries** – основная структура данных в Wealth-Lab и математические операции при программировании торговых стратегий
- **Наполнение созданных самостоятельно DataSeries данными при помощи WealthScript**
- **Как получить доступ ко второму финансовому инструменту при проектировании торговых стратегий в Велс Лаб.**
- **Синхронизация данных при построении торговых стратегий с несколькими финансовыми инструментами**
- **Создание параметров торговой стратегии в Wealth-Lab**

📁 РУБРИКИ: [DATASERIES](#), [WEALTHSCRIPT](#)

Комментариев: 2

DataSeries - основная структура данных в Wealth-Lab и математические операции при программировании торговых стратегий | Финансовая лаборатория пишет:

9 мая 2012 в 18:52

[...] Как получить доступ к единичному значению DataSeries Главная Symbol Info – подробная информация о [...]

Ответить

Ответы на вопросы по программированию торговых стратегий в Wealth-Lab | Финансовая лаборатория пишет:

17 мая 2012 в 13:52

[...] [...]

Ответить

Оставить комментарий

Имя (Обязательно)

Почта (Обязательно)

Вэб-сайт

Получать новые комментарии по электронной почте. Вы можете **подписаться** без комментирования.



Видео ВЕБИНАРОВ



Дмитрия Власова



**МТС под микроскопом:
WLMA System**

Подробный разбор торговой
системы WLMA System

3900 руб.

В корзину

Рубрики

Выберите рубрику

Обновления:



Новые посты по Email



Новые посты по RSS



Комментарии по RSS



Следим на Twitter

Метки

C# HowTo: **Wealth-Lab**

Видео Индикаторы МТС Торговые
роботы Юмор алготрейдинг
мероприятия обучение

Управление

Регистрация

Вход

RSS Записей

RSS Комментариев

WordPress.org

© 2012, Финансовая лаборатория. Все права защищены.

Любое использование материалов допускается только с указанием активной ссылки на источник.



Подписка

[RSS](#)

[Twitter](#)

[ВКонтакте](#)

[facebook](#)

Сайт

[Создать блог](#)

[Содержание](#)

[О блоге](#)

[Карта](#)

Я

В день мой блог
посещает 77
человек,
сегодня — 123

А.Т.Т.Т.Т.

BY FEEDBURNER

Наполнение созданных самостоятельно DataSeries данными при помощи WealthScript

 **АВТОР** ДМИТРИЙ ВЛАСОВ  **МАЯ** - 9 - 2012  **КОММЕНТАРИЕВ:** 2

После того, как мы разобрались с тем, что такое DataSeries - пришло время научиться самостоятельно создавать и наполнять данными это полезное для создания и тестирования стратегий хранилище данных. Этот навык необходимо довести до автоматизма, т.к. создавая торговую стратегию и ее правила мы будем постоянно с этим сталкиваться. Давайте оттачиванием этого навыка и займемся сегодня.



Как: создать последовательность данных DataSeries и заполнить каждую ее ячейку нулевыми значениями

Достаточно часто при написании правил торговой стратегии появляется необходимость создать простейший индикатор "на лету", написав буквально несколько строчек кода. К примеру, такой подход был продемонстрирован в одной из предыдущих статей, например [здесь](#) в последнем примере.

Однако, при построении других индикаторов необходимо буквально на каждом из баров решать задачу - какими данными заполнять ячейки (при этом часто решается логическая задача: если - то). В этом случае на первом этапе построения такой числовой последовательности необходимо каждую ячейку DataSeries заполнить нулями.

Пример ([как выполнить пример приведенного кода](#))...

Одним из самых простых способов решения задачи заполнения всех ячеек последовательности DataSeries нулевыми значениями является просто вычесть одну уже существующую последовательность DataSeries из другой.

Код примера на C# (кликните чтобы посмотреть) 

```
1 protected override void Execute()
2 {
3     DataSeries random = Close - Close;
4 }
```

Пример ([как выполнить пример приведенного кода](#))...

Предпочтительнее всего создавать новый объект типа DataSeries путем последовательного прохождения баров уже существующих последовательностей (к примеру, OHLC/V) и совершения над ними определенных действия. К тому же, этот метод позволяет без проблем создавать понятное описание новой числовой последовательности DataSeries.

Код примера на C# (кликните чтобы посмотреть) 

```
1 protected override void Execute()
2 {
3     // Создание, заполнение и отображение на графике случайной последовательности
4     DataSeries со значениями от 50 до 100, или нулями.
5
6     DataSeries random = new DataSeries(Bars, "Иницированная нулями, заполненная
```

RSS товары



RSS новости



Twitter



Последние комментарии

макс: А где взять свечки с открытым интересом? на сайте финама нет



ring10: Добрый день Дмитрий Как сделать, чтобы открывался 1 велф лаб? Если открываешь под пользователем, т [...]



Дмитрий Власов: Буян, рад, что Вам пригодились материалы данной статьи!



Максим: Сам себе отвечу. protected override void Execute() { Bars rts = GetExternalSymbol("RTS", t [...]



Буян: Спасибо Дмитрий за статью! Мне удалось разработать хорошую рабочую торговую систему, используя возмо [...]



Последние посты



Бесплатный вебинар: 7 смертных грехов при оптимизации торговых систем

НОЯБРЯ - 5 - 2012



Отзывы участников курса до результата Wealth-Lab. С нуля до первой торговой системы за 5

дней.
НОЯБРЯ - 3 - 2012



Быстрее, Выше, Сильнее: Обновление Wealth-Lab до новой версии 6.4

ОКТАБРЯ - 23 - 2012



Курс до результата: Wealth-Lab – от нуля до первой торговой системы за 5 дней

ОКТАБРЯ - 23 - 2012



Обзорная экскурсия по достопримечательностям программы Wealth-Lab

ОКТАБРЯ - 20 - 2012

Берите

```
7 | случайными числами от 50 до 100");
8 |
9 | Random r = new Random();
10 |
11 | for (int bar = 0; bar < Bars.Count; bar++)
12 | {
13 |     double rVal = 100 * r.NextDouble();
14 |     if (rVal > 50)
15 |         random[bar] = rVal;
16 |     }
17 |
18 | ChartPane rPane = CreatePane( 40, true, true );
19 |
20 | PlotSeries(rPane, random, Color.Blue, WealthLab.LineStyle.Histogram, 1);
21 |
22 | }
```

Как видите, здесь с использованием класса Random происходит создание, инициализация нулями (заполнение) и отображение на графике случайной последовательности DataSeries со значениями от 50 до 100, или нулями. Этот пример работы со случайными числами может пригодиться и при написании стратегий - когда нужно работать с вероятностями.

Рассмотрим еще одни пример.

Как: поменять значения в массивах DataSeries

Вы можете без проблем поменять значения любого массива DataSeries - даже ключевых последовательностей OHLC/V именно так, как Вы этого захотите.

Пример (как выполнить пример приведенного кода)...

Этот пример показывает, как можно определить и скорректировать чрезмерные ценовые "выбросы", которые, если их не скорректировать, могут исказить график по шкале Y.

Код примера на C# (кликните чтобы посмотреть)

```
1 | protected override void Execute()
2 |
3 | {
4 |     const double Sensitivity = 2.5; // Устанавливаем максимальный коэффициент к
5 |     предыдущему бару (чувствительность)
6 |
7 |     DataSeries atr = ATR.Series( Bars, 5 );
8 |     DataSeries hi = Highest.Series(High, 3);
9 |     DataSeries lo = Lowest.Series(Low, 3);
10 |
11 |     for (int bar = 10; bar < Bars.Count; bar++){
12 |
13 |         if (High[bar] > High[bar-1] + Sensitivity * atr[bar])
14 |             High[bar] = Math.Max( hi[bar-1], Math.Max(Open[bar], Close[bar]) );
15 |
16 |         if (Low[bar] < Low[bar-1] - Sensitivity * atr[bar])
17 |             Low[bar] = Math.Min( lo[bar-1], Math.Min(Open[bar], Close[bar]) );
18 |
19 |     }
20 | }
```

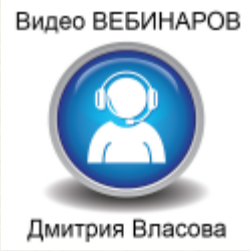
Этот пример демонстрирует, что можно без проблем поменять "на лету" существующие ценовые ряды.

Помимо демонстрации этих возможностей этот небольшой кусочек кода является полезным в практической деятельности по построению и тестированию торговых стратегий. Здесь, как Вы поняли, ищутся "ценовые выбросы" путем последовательной проверки - не превышает ли текущий бар предыдущий на 2,5 ATR, рассчитанного по 5-ти предыдущим барам. И если это действительно так, то корректируется сначала максимальная, а потом и минимальная цена.

На этом примере закончим сегодняшний рассказ. Вы узнали, как заполнять собственноручно созданные DataSeries необходимыми Вам данными. Теперь Вы без труда сможете буквально "на лету" создавать собственные индикаторы, которые являются ничем иным, как привязанной к барам числовой последовательностью, рассчитанной по определенным Вами правилам. В следующий раз дадим первоначальные сведения о том, как в одной стратегии можно **получить доступ не к одному финансовому инструменту**, а к нескольким.

- Если Вам понравилась эта статья - подпишитесь на **новые СТАТЬИ по RSS** или получайте наши **новые СТАТЬИ на E-mail**:
Рекомендую почитать также:
- **DataSeries – основная структура данных в Wealth-Lab и математические операции при программировании торговых стратегий**
 - **Как получить доступ к единичному значению DataSeries**
 - **Как получить доступ ко второму финансовому инструменту при проектировании торговых стратегий в Велс Лаб.**
 - **Синхронизация данных при построении торговых стратегий с несколькими финансовыми инструментами**
 - **Делаем индикаторы в Wealth-Lab своими руками**

РУБРИКИ: DATASERIES, WEALTHSCRIPT



МТС под микроскопом: WLMA System

Подробный разбор торговой системы WLMA System
3900 руб.

В корзину

Рубрики

Выберите рубрику

Обновления:

- Новые посты по Email
- Новые посты по RSS
- Комментарии по RSS
- Следим на Twitter

Метки

С# HowTo: **Wealth-Lab**
Видео Индикаторы МТС Торговые роботы Юмор алготрейдинг мероприятия обучение

Управление

Регистрация

Вход

RSS Записей

RSS Комментариев

WordPress.org

Комментариев: 2

Как получить доступ к единичному значению DataSet | Финансовая лаборатория пишет:

9 мая 2012 в 18:55

[...] [...]

Ответить

Ответы на вопросы по программированию торговых стратегий в Wealth-Lab | Финансовая лаборатория пишет:

22 мая 2012 в 15:28

[...] 20. Как создать последовательность данных DataSet и наполни... [...]

Ответить

Оставить комментарий

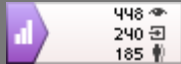
Имя (Обязательно)

Почта (Обязательно)

Вэб-сайт

Получать новые комментарии по электронной почте. Вы можете подписаться без комментирования.

© 2012, Финансовая лаборатория. Все права защищены.
Любое использование материалов допускается только с указанием активной ссылки на источник.



448

240

185



В день мой блог посещает 77 человек, сегодня

123



РЕЙТИНГ

757130515

193

Анонсы

BY FEEDBURNER

Подписка

- [RSS](#)
- [Twitter](#)
- [ВКонтакте](#)
- [facebook](#)

Сайт

- [Создать блог](#)
- [Содержание](#)
- [О блоге](#)
- [Карта](#)

Как получить доступ ко второму финансовому инструменту при проектировании торговых стратегий в Велс Лаб.

 АВТОР [ДМИТРИЙ ВЛАСОВ](#)  МАЯ - 10 - 2012  КОММЕНТАРИЕВ: 7

Зачастую Вы хотите протестировать стратегию, в которой задействован не один финансовый инструмент, а два или более. К таким стратегиям относятся такие стратегии, которые основаны на принципах арбитража, парного трейдинга, ротации финансовых инструментов, сравнения с бенчмарком. К счастью, программа Wealth-Lab может помочь и при тестировании таких неординарных торговых стратегий.

Прежде чем приступить к созданию и тестированию таких стратегий торговли, разберемся с теми стандартными методами, которыми оперирует программа Велс Лаб, а также с основными понятиями, которые относятся к данной сфере торговли.



Вторым, или внешним финансовым инструментом называется финансовый инструмент, который не отображается на основном графике. Если указанного в программе тикера второго финансового инструмента не существует в текущем наборе данных (DataSet), то Wealth-Lab ищет указанный инструмент во всех прочих наборах данных (DataSet) в алфавитном порядке. Если в результате такого поиска искомый тикер финансового инструмента так и не находится, то по необходимости можно будет создать его.

Как: получить доступ к DataSet второго финансового инструмента

Это можно сделать несколькими путями:

Метод 1: Используя метод `GetExternalSymbol()`

Метод `GetExternalSymbol()` возвращает объект `Bars` для выбранного финансового инструмента в текущем таймфрейме. Используйте этот метод только тогда, когда Вы не собираетесь совершать сделки со вторым финансовым инструментом. Этот метод можно использовать только для того, чтобы получить данные второго финансового инструмента для построения индикаторов или отображения их на графике.

В примере мы получили бары финансового инструмента, выступающего как бенчмарк. Используя цены закрытия этого финансового инструмента для вычисления относительного индекса (relative index). При этом использовалась техника под названием Most Anchored Momentum. Очень удобно то, что используя точно такие же свойства баров для бенчмарка, мы можем отобразить на графике финансовый инструмент, используемый в качестве бенчмарка.

Пример ([как выполнить пример приведенного кода](#))...

В данном примере происходит построение специального индекса, используя внешний финансовый инструмент.

RSS товары



RSS новости



Twitter



Последние комментарии

макс: А где взять свечки с открытым интересом? на сайте финама нет



ring10: Добрый день Дмитрий Как сделать, чтобы открывался 1 велф лаб? Если открываешь под пользователем, т [...]



Дмитрий Власов: Буян, рад, что Вам пригодились материалы данной статьи!



Максим: Сам себе отвечу. protected override void Execute() { Bars rts = GetExternalSymbol("RTS", t [...]



Буян: Спасибо Дмитрий за статью! Мне удалось разработать хорошую рабочую торговую систему, используя возмо [...]



Последние посты



Бесплатный вебинар: 7 смертных грехов при оптимизации торговых систем

НОЯБРЯ - 5 - 2012



Отзывы участников курса до результата Wealth-Lab. С нуля до первой торговой системы за 5 дней.

НОЯБРЯ - 3 - 2012



Быстрее, Выше, Сильнее: Обновление Wealth-Lab до новой версии 6.4

ОКТАБРЯ - 23 - 2012



Курс до результата: Wealth-Lab – от нуля до первой торговой системы за 5 дней

ОКТАБРЯ - 23 - 2012



Обзорная экскурсия по достопримечательностям программы Wealth-Lab

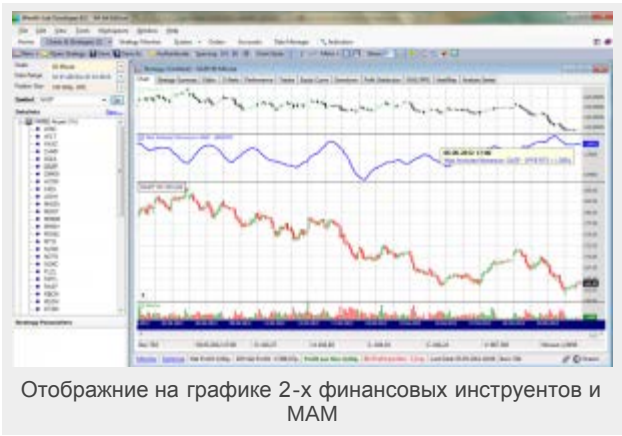
ОКТАБРЯ - 20 - 2012

Берите

Код примера на C# (кликните чтобы посмотреть)

```
1 | protected override void Execute()
2 |
3 | {
4 |     // Создаем Most-Anchored Momentum Index используя второй финансовый инструмент,
5 |     выступающий как бенчмарк
6 |
7 |     Bars bmBars = GetExternalSymbol("SPFB.RTS", true); //получаем бары базы для
8 |     сранения
9 |
10 |    DataSeries mam = Close / bmBars.Close;
11 |
12 |    mam = EMA.Series(mam, 10, EMACalculation.Modern) / SMA.Series(mam, 20);
13 |
14 |    mam.Description = "Most Anchored Momentum: " + Bars.Symbol + " - " +
15 |    bmBars.Symbol;
16 |
17 |    ChartPane mamPane = CreatePane(40, true, true);
18 |
19 |    PlotSeries(mamPane, mam, Color.Blue, LineStyle.Solid, 2);
20 |
21 |    ChartPane bmPane = CreatePane(40, true, true);
22 |
23 |    PlotSymbol(bmPane, bmBars, Color.LightGreen, Color.Black);
24 | }
```

В результате отображается на графике Most Anchored Momentum и инструмент, который выступал у нас базой для сравнения - бенчмарк.



Смысл этого индикатора заключается в том, что мы сравним текущий финансовый инструмент (на нашем графике это акции газпрома) с бенчмарком, т.е. со вторым финансовым инструментом, взятым как мера сравнения (на нашем графике это фьючерс на индекс РТС).

Затем находим отношение цены закрытия основного инстрмента к внешнему инструменту. Усредняем этот полученный СПРЕД двумя скользящими средними и сравниваем эти скользящие средние друг с другом.

Интерпретировать этот МАМ можно так. Если это индикатор растет, то можно сделать вывод, что текущий инструмент растет быстрее (или падает медленнее) чем более широкий рынок. Если же падает, то текущий инструмент ведет себя хуже, чем инструмент, с которым мы проводим сравнение, т.е. растет медленнее (или же падает быстрее).

Вы, конечно же, можете поискать в Интернете более подробные сведения о Most Anchored Momentum и продумать - как можно этот инструмент использовать в своей торговле. Но на сегодня наша цель другая. Продемонстрировать работу одновременно с несколькими финансовыми инструментами.

Как видите, в этом примере продемонстрировано не только как можно использовать второй финансовый инструмент для построения индексов, но и как можно отобразить второй финансовый инструмент на графике.

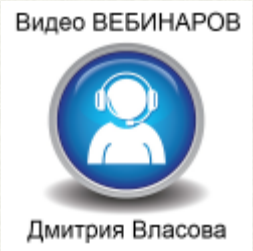
Метод 2: Используя метод SetContext()

Вторым способом доступа к внешнему финансовому инструменту является использование специального метода SetContext().

Метод SetContext() меняет контекст выполняемого кода. При этом основным временно становится не главный финансовый инструмент, который выбирается путем кликанья по нему мышкой в DataSet, а тот финансовый инструмент, который указывается в методе SetContext(). При этом все действия с барами временно начинают выполняться с новым финансовым инструментом. После выполнения метода SetContext() Вы сможете обычным образом совершать сделки с новым финансовым инструментом. Для выяснения подробностей смотри программирование торговых стратегий.

Пример (как выполнить пример приведенного кода)...

```
Код примера на C# (кликните чтобы посмотреть)
1 | protected override void Execute()
2 |
3 | {
4 |
5 |     SetContext("ROSN", true); // Меняем контекст, т.е. начинаем работать со вторым
6 |     (внешним) финансовым инструментом
7 |     Bars rosn = Bars; // Объявляем переменную ROSN типа Bars и сохраняем
8 |     там бары 2-го финансового инструмента
9 |     RestoreContext(); // Возвращаемся к основному финансовому инструменту
10 |
11 |     // Отображаем цены закрытия второго финансового инструмента (MSFT) на графике
12 |     основного (1-го) финансового инструмента
13 |
14 |     PlotSeries(PricePane, rosn.Close, Color.Blue, LineStyle.Solid, 2);
15 |
16 |     // Создаем алерт для текущего (1-го) финансового инструмента (в нашем случае
17 |     газпром)
18 | }
```



МТС под микроскопом:

WLMA System

Подробный разбор торговой системы WLMA System
3900 руб.

[В корзину](#)

Рубрики

Выберите рубрику

Обновления:

- Новые посты по Email
- Новые посты по RSS
- Комментарии по RSS
- Следим на Twitter

Метки

C# HowTo: **Wealth-Lab**
Видео Индикаторы МТС Торговые
роботы Юмор алготрейдинг
мероприятия обучение

Управление

[Регистрация](#)

[Вход](#)

[RSS Записей](#)

[RSS Комментариев](#)

[WordPress.org](#)

```
19     int bar = Bars.Count - 1;
20
21     BuyAtMarket(bar + 1, Bars.Symbol);
22
23     // Создаем алерт для внешнего (2-го) финансового инструмента (в нашем случае
24     роснефть)
25
26     SetContext("ROSN", true);
27
28     BuyAtMarket(bar + 1, Bars.Symbol);
29
30     RestoreContext();
31 }
32
```

Взглянув на получившийся график, мы можем убедиться, что метод SetContext() точно также, как и метод GetExternalSymbol() позволяет строить нужные нам индикаторы, используя информацию о втором финансовом инструменте. Здесь отображается на гафике инструмента "ГАЗПРОМ" линейный график, построенный по ценам закрытия финансового инструмента "РОСНЕФТЬ".



График Газпрома и Роснефти - метод SetContext()

Но кроме того, метод SetContext() позволяет еще и совершать сделки с внешним (2-м) финансовым инструментом. Это доказывает на то, что программа Велс Лаб при выполнении приведенного кода формирует нам полноценные алерты.

Alert Time	Symbol	Account	A...	Qty	Order Type	Price	Scale	Signal Name	Trade Type	Strategy Name
26.05.2012 10:00	ROSN	PaperAccount.01	00000000000000000000	Buy	SOB	Market	60 Minute	SAZP		SAZP
26.05.2012 10:05	ROSN	PaperAccount.01	00000000000000000000	Buy	SOB	Market	60 Minute	ROSN		ROSN

Алерты в результате использования метода SetContext()

Кстати, таких внешних инструментов может использоваться не один, а множество в одной торговой стратегии.

Обычно, как и в предыдущих примерах, Вы должны присвоить параметру синхронизации методов GetExternalSymbol() и SetContext() значение true.

Однако в некоторых случаях Вы захотите присвоить параметру синхронизации этих методов значение false. К примеру, так целесообразно поступать при создании индикаторов, когда сначала используются исходные бары или числовые последовательности внешних финансовых инструментов и лишь потом происходит их синхронизация с основным финансовым инструментом.

Для того, чтобы подробнее разобраться с такими случаями - внимательно изучите следующий пост, который будет посвящен [синхронизации внешних данных](#).

Если Вам понравилась эта статья - подпишитесь на [новые СТАТЬИ по RSS](#) или получайте наши [новые СТАТЬИ на E-mail](#):

Рекомендую почитать также:

- [DataSeries](#) – основная структура данных в Wealth-Lab и математические операции при программировании торговых стратегий
- [Как получить доступ к единичному значению DataSeries](#)
- [Наполнение созданных самостоятельно DataSeries данными при помощи WealthScript](#)
- [Синхронизация данных при построении торговых стратегий с несколькими финансовыми инструментами](#)
- [Создание многопозиционных стратегий в Wealth-Lab](#)

РУБРИКИ: [DATASERIES](#), [WEALTHSCRIPT](#)

Комментариев: 7

Синхронизация данных при построении торговых стратегий с несколькими финансовыми инструментами | Финансовая лаборатория пишет:

11 мая 2012 в 10:22

[...] Как получить доступ ко второму финансовому инструмент... [...]

Ответить

Наполнение созданных самостоятельно DataSeries данными при помощи WealthScript | Финансовая лаборатория пишет:

13 мая 2012 в 21:58

[...] [Как получить доступ ко второму финансовому инструмент...](#) [Главная](#) [Как получить доступ к единичному \[...\]](#)

Ответить

Ответы на вопросы по программированию торговых стратегий в Wealth-Lab | Финансовая лаборатория пишет:

16 мая 2012 в 12:57

[...] [...]

Ответить

Инструкция по программированию торговых стратегий в WealthLab | Финансовая лаборатория пишет:

20 июня 2012 в 12:20

[...] 4. Как получить доступ ко второму финансовому инструмент... [...]

Ответить

Андрей пишет:

9 октября 2012 в 12:48



Дмитрий, спасибо за статью. Она помогла разобраться с подключением второго финансового инструмента. Но у меня остался вопрос.

Я переключаюсь на второй инструмент функцией SetContext(...). Далее хочу изменить таймфрейм второго инструмента функцией SetScaleCompressed(...), но меня перекидывает на первичный инструмент, т.е. я получаю первичный инструмент с только что заданным таймфреймом.

Я пробовал идти на оборот: сначала менял таймфрейм, потом выбирал второй инструмент. В этом случае я получал второй инструмент, но с превичным таймфреймом.

Существует ли возможность изменить и инструмент, и таймфрейм одновременно. Заранее спасибо!

Ответить

Буян пишет:

22 ноября 2012 в 12:15



Спасибо Дмитрий за статью! Мне удалось разработать хорошую рабочую торговую систему, используя возможность анализа второго инструмента!

Ответить

Дмитрий Власов пишет:

27 ноября 2012 в 02:33



Буян, рад, что Вам пригодились материалы данной статьи!

Ответить

Оставить комментарий

Имя (Обязательно)

Почта (Обязательно)

Вэб-сайт

Получать новые комментарии по электронной почте. Вы можете [подписаться](#) без комментирования.

© 2012, Финансовая лаборатория. Все права защищены.
Любое использование материалов допускается только с указанием активной ссылки на источник.



448
240
185



В день мой блог
посещает **77**
человек,
сегодня **123**



РЕЙТИНГ
757132 517
193



BY FEEDBURNER

Подписка

- [RSS](#)
- [Twitter](#)
- [ВКонтакте](#)
- [facebook](#)

Сайт

- [Создать блог](#)
- [Содержание](#)
- [О блоге](#)
- [Карта](#)

Синхронизация данных при построении торговых стратегий с несколькими финансовыми инструментами

АВТОР ДМИТРИЙ ВЛАСОВ МАЯ - 11 - 2012 КОММЕНТАРИЕВ: 6

Всякий раз, когда Вы при построении своей торговой стратегии используете информацию более чем об одном финансовом инструменте, Вам приходится сталкиваться с проблемой синхронизации данных. К примеру, если стратегия предполагает, что Вы должны покупать Газпром только убедившись в том, что индекс ММВБ растет.

Сегодняшний пост поможет Вам понять, как Wealth-Lab синхронизирует данные, когда в одной стратегии задействовано несколько финансовых инструментов. Прочитав этот пост Вы сможете более надежно контролировать процесс этой синхронизации.

Как: получить доступ к дополнительным финансовым инструментам

Вы можете получить [доступ к внешним финансовым инструментам](#) используя в коде торговой стратегии методы `GetExternalSymbol()` или `SetContext()`.

Синхронизация в Велс Лаб по умолчанию

Обычно синхронизации по умолчанию вполне хватает для того, чтобы Вы об этом вообще не беспокоились. Дефолтная синхронизация базируется на концепции первичных баров.



Объект `Bars` первичного финансового инструмента всегда используется при выполнении стратегии.

В программе Wealth-Lab первичным финансовым инструментом называется такой финансовый инструмент, который Вы кликаете на панели данных для того, чтобы запустить торговую стратегию на выполнение, это тикер, который Вы вводите в поле ввода перед тем, как нажать на кнопку "Go", или же это текущий финансовый инструмент при тестировании стратегий со множеством финансовых инструментов.

По умолчанию все внешние данные (бары и `DataSeries`) выравниваются с первичными барами сравнивая дату и время каждого из них на каждом из баров.

Все внешние данные, которые находятся за временными пределами первичных баров (раньше или позже) уничтожаются при синхронизации

Данные, которые существуют в первичных барах, но отсутствуют во внешних данных - добавляются к внешним данным. Значение в добавленные промежутки копируются из предыдущих значений соответствующих данных.

Если во внешних данных имеются данные, дата и время которых отсутствует в первичных барах, то такие внешние данные уничтожаются при синхронизации

В конце концов все внешние данные бар за баром синхронизируются с первичными барами. Это гарантирует, что свойство `Bars.Count` для внешних финансовых инструментов будет в точности соответствовать этому свойству первичных баров, т.е. количество баров первичных и внешних данных будет одинаковым.

[RSS товары](#)[RSS новости](#)[Twitter](#)

Последние комментарии

макс: А где взять свечки с открытым интересом? на сайте финама нет



ring10: Добрый день Дмитрий Как сделать, чтобы открывался 1 велф лаб? Если открываешь под пользователем, т [...]



Дмитрий Власов: Буян, рад, что Вам пригодились материалы данной статьи!



Максим: Сам себе отвечу. protected override void Execute() { Bars rts = GetExternalSymbol("RTS", t [...]



Буян: Спасибо Дмитрий за статью! Мне удалось разработать хорошую рабочую торговую систему, используя возмо [...]



Последние посты



Бесплатный вебинар: 7 смертных грехов при оптимизации торговых систем

НОЯБРЯ - 5 - 2012



Отзывы участников курса до результата Wealth-Lab. С нуля до первой торговой системы за 5

дней.

НОЯБРЯ - 3 - 2012



Быстрее, Выше, Сильнее: Обновление Wealth-Lab до новой версии 6.4

ОКТАБРЯ - 23 - 2012



Курс до результата: Wealth-Lab – от нуля до первой торговой системы за 5 дней

ОКТАБРЯ - 23 - 2012



Обзорная экскурсия по достопримечательностям программы Wealth-Lab

ОКТАБРЯ - 20 - 2012

[Берите](#)

Также Вы можете без проблем создавать и отображать на графике индикаторы, базирующиеся на внешних данных. Следующая схема поясняет процесс синхронизации первичной и внешней DataSeries.



Внешние данные были модифицированы следующим способом:

Бар с датой "4" был удален, т.к. такой даты не существует в первичной последовательности.

Бар "9" был вставлен, при этом его значение было скопировано из предыдущего бара "8" внешней последовательности.

Бар "17" был удален, т.к. он находился вне временных рамок первичной последовательности.

Влияние синхронизации на создание индикаторов

Давайте предположим, что мы хотим создать скользящую среднюю используя 5 последовательных баров внешнего финансового инструмента.

Если эта скользящая средняя будет составлена на основе цен внешнего финансового инструмента после синхронизации, то значение скользящей средней будет больше, чем если бы она рассчитывалась по данным до синхронизации. Это происходит из-за того, что значения в барах 5 и в барах 8 будут на единицу больше, чем если бы расчет проходил без синхронизации.

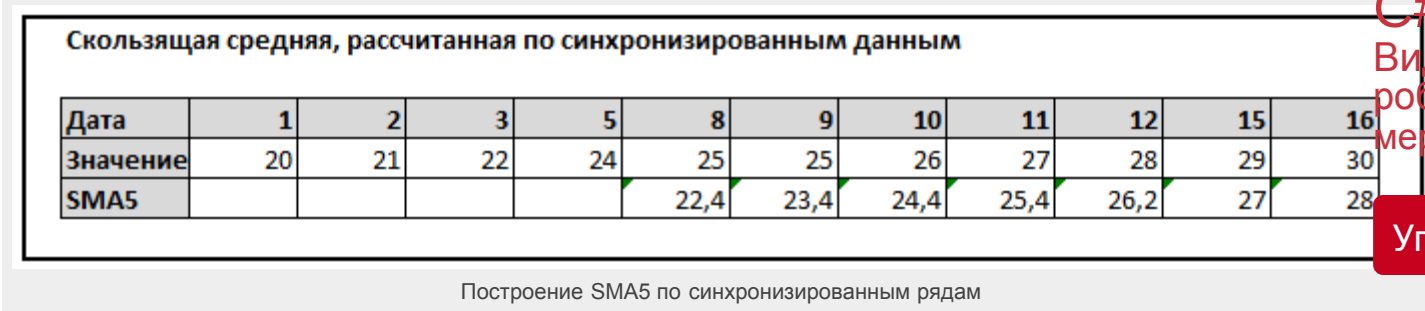
Такая ситуация может устраивать либо не устраивать разработчика торговой системы.

С одной стороны, Вы можете посчитать, что при отсутствии данных или баров наилучшим выходом из ситуации будет копирование значения из предыдущего бара (это быстрее, чем корректировать данные вручную и добавлять недостающие бары).

С другой стороны, внешний финансовый инструмент мог вообще не торговаться по тем или иным причинам в это время.

К счастью, программа Wealth-Lab позволяет Вам справиться с любым развитием ситуации.

Сначала рассмотрим случай, когда скользящая средняя была посчитана после синхронизации, которая применяется по умолчанию:



Смена способа синхронизации по умолчанию

WealthScript дает возможность отключить синхронизацию "по умолчанию" и выполнить синхронизацию уже после расчета индикаторов на первоначальных значениях внешних рядов данных. Это предполагает отключение синхронизации внешних финансовых инструментов при доступе к его барам или рядам данных и последующую синхронизацию с помощью метода Synchronize().

Для того, чтобы отключить синхронизацию, которая используется по умолчанию, нужно присвоить параметру синхронизации в методах GetExternalSymbol() и SetContext() значение false.

Создайте индикаторы на основании несинхронизированных (первоначальных) внешних баров или рядов данных.

Выполните метод Synchronize() для синхронизации внешних рядов для того, чтобы создать синхронизированные бары или ряды данных.

Метод Synchronize() синхронизирует внешние ряды данных точно также, как и метод по умолчанию. Однако здесь мы уже синхронизируем индикаторы, построенные на основе данных внешнего финансового инструмента, а не сами бары внешнего финансового инструмента. Поэтому Вы можете быть уверены, что индикаторы точны, даже если внешний финансовый инструмент или ряд данных не торговался тогда, когда




Видео ВЕБИНАРОВ



Дмитрия Власова

E-mail курсы



ХВАТИТ ЖАЛЕТЬ СВОЮ МТС





МТС под микроскопом:

WLMA System

Подробный разбор торговой системы WLMA System

3900 руб.

В корзину

Рубрики

Выберите рубрику

Обновления:

 Новые посты по Email

 Новые посты по RSS

 Комментарии по RSS

 Следим на Twitter

Метки

C# HowTo: Wealth-Lab

Видео Индикаторы МТС Торговые роботы Юмор алготрейдинг мероприятия обучение

Управление

- Регистрация
- Вход
- RSS Записей
- RSS Комментариев
- WordPress.org

торговался основной финансовый инструмент.

Далее приводится иллюстрация того, как будет выглядеть скользящая средняя если бы мы сначала рассчитали ее на первоначальных данных внешнего финансового инструмента и лишь затем синхронизировали. Обратите внимание на то, что значение скользящей средней для вставленного бара №9 было просто скопировано из значения скользящей средней предыдущего бара.

Скользящая средняя, рассчитанная по несинхронизированным данным (потом синхронизована)											
Дата	1	2	3	5	8	9	10	11	12	15	16
Значение	20	21	22	24	25	25	26	27	28	29	30
SMA5					23	23	24	25	26	27	28

Построение SMA5 и синхронизация

Как: создать индикатор используя несинхронизированные данные второго финансового инструмента

Следующий пример кода получает данные о ценах фьючерса на индекс РТС, создает его 50-периодную скользящую среднюю и лишь затем синхронизирует бары фьючерса на индекс РТС и его скользящую среднюю с выбранным финансовым инструментом и отображает его на графике.

Пример (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть)

```
1  protected override void Execute()
2
3  {
4      // Получаем доступ к внешнему финансовому инструменту, но не синхронизируем его
5
6      Bars rts = GetExternalSymbol("SPFB.RTS", false);
7
8      // Рассчитываем скользящую среднюю по несинхронизированному внешнему финансовому
9      инструменту
10
11     DataSeries rtsSMA = SMA.Series(rts.Close, 50);
12
13     // Теперь синхронизируем бары внешнего инструмента
14     rts = Synchronize(rts);
15
16     //Синхронизируем скользящую среднюю внешнего финансового инструмента
17     rtsSMA = Synchronize(rtsSMA);
18
19     ChartPane rtsPane = CreatePane(75, true, true);
20
21     PlotSeries(rtsPane, rtsSMA, Color.Blue, WealthLab.LineStyle.Solid, 1); //
22     отображаем скользящую среднюю
23
24     PlotSymbol(rtsPane, rts, Color.LightBlue, Color.Black); //отображаем бары внешнего
25     финансового инструмента
26 }
```

Когда Вы выполните в программе Wealth-Lab 6.3 приведенный код, у Вас отобразится вот такой график.



Синхронизированный график

Прошу обратить внимание, что в качестве первичного инструмента здесь используется ГАЗПРОМ, который торгуется на ММВБ и последний час торговли которого начинается в 18-00. В качестве внешнего (второго) инструмента используется фьючерс на индекс РТС, как Вы можете увидеть в коде. Причем, этот фьючерс торгуется и после 19 часов. На графике же, в верхней части Вы можете видеть бары фьючерса на индекс РТС и построенную по ним скользящую среднюю. Причем даже на фьючерсе после 18-ти часового бара отображается сразу первый бар следующей торговой сессии. Именно это демонстрирует, что была проведена синхронизация инструментов.

Замечание о первоначальных барах:

Метод Synchronize() синхронизирует внешние бары или ряды данных по отношению к основным барам. Будьте очень внимательны, т.к. первоначальные бары могут быть также изменены при вызове метода SetContext(). Если Вы хотите синхронизировать первоначальный график баров, убедитесь, что Вы выполнили метод RestoreContext() после использования метода SetContext(). Все это было сделано специально и позволяет Вам при использовании сложных стратегий, например парного трейдинга, полностью контролировать синхронизацию дополнительных (внешних) финансовых инструментов.

На этом сегодняшний пост заканчивается. Вы узнали, как важно уметь управлять синхронизацией данных в стратегиях, где торгуются несколько финансовых инструментов. В следующий раз поговорим о том, **как отображать имеющиеся данные на графиках**, поэтому не забывайте подписываться на RSS обновления нашего блога.

Если Вам понравилась эта статья - подпишитесь на **новые СТАТЬИ по RSS** или получайте наши **новые СТАТЬИ на E-mail**:

Рекомендую почитать также:

- **DataSeries – основная структура данных в Wealth-Lab и математические операции при программировании торговых стратегий**
- **Как получить доступ к единичному значению DataSeries**
- **Наполнение созданных самостоятельно DataSeries данными при помощи WealthScript**
- **Как получить доступ ко второму финансовому инструменту при проектировании торговых стратегий в Велс Лаб.**
- **Организация торгового цикла в программе Wealth-Lab и его переделка в скрининговую систему.**

 РУБРИКИ: **DATASERIES**, **WEALTHSCRIPT**

Комментариев: 6

Как получить доступ ко второму финансовому инструменту при проектировании торговых стратегий в Велс Лаб. | Финансовая лаборатория пишет:

11 мая 2012 в 12:55

[...] [...]

Ответить

Ответы на вопросы по программированию торговых стратегий в Wealth-Lab | Финансовая лаборатория пишет:

14 мая 2012 в 13:35

[...] [...]

Ответить

Antrax пишет:

22 июня 2012 в 11:41



В примере кода , метод GetExternalSymbol() будет работать только в рамках датасета “ММВБ-Акции(1h), чтобы он обращался к к другим датасетам он должен содержать еще один параметр с указанием строкового имени датасета откуда получать символ.

Ответить

Дмитрий Власов пишет:

22 июня 2012 в 23:51



Да нет, тут будет работать в любом наборе данных (DataSet). Просто нужно вот здесь:

```
Bars rts = GetExternalSymbol(“SPFB.RTS”, false);
```

вместо “SPFB.RTS” прописать любой финансовый инструмент для сравнения. (у меня был прописан фьючерс на индекс РТС).

Ответить

Максим пишет:

21 ноября 2012 в 19:59



С одним внешним инструментом понятно, а как создать синтетический инструмент, к примеру SynthSymb=RTS-(4GAZR+4LKOH) и отобразить его на графике? Не осилил 😞

Ответить

Максим пишет:

22 ноября 2012 в 12:24



Сам себе отвечаю.

```
protected override void Execute()
```

```
{
```

```
Bars rts = GetExternalSymbol(“RTS”, true);
```

```
Bars lkoh = GetExternalSymbol(“LKOH”, true);
```

```
Bars gazp = GetExternalSymbol("GAZP", true);
Bars si = GetExternalSymbol("Si", true);

DataSeries rtsRub = rts.Close * 0.02 * si.Close / 1000;
DataSeries synthSymb = rtsRub – ( 2 * lkoh.Close + 3 * gazp.Close );
synthSymb.Description = "rts-2lkoh+3gazp";

ChartPane sPane = CreatePane(40, true, true);
PlotSeries(sPane, synthSymb, Color.Blue, WealthLab.LineStyle.Solid, 2);
}
```

Как-то так. Получились не бары, как хотелось, но для принятия решения достаточно.

Ответить

Оставить комментарий

Имя (Обязательно)

Почта (Обязательно)

Вэб-сайт

Получать новые комментарии по электронной почте. Вы можете [подписаться](#) без комментирования.

© 2012, Финансовая лаборатория. Все права защищены.
Любое использование материалов допускается только с указанием активной ссылки на источник.



448
240
185



В день мой блог
посещает **77**
человек,
сегодня **123**



mail.ru РЕЙТИНГ
757133 518
193



А что?
BY FEEDBURNER

Подписка

- [RSS](#)
- [Twitter](#)
- [ВКонтакте](#)
- [facebook](#)

Сайт

- [Создать блог](#)
- [Содержание](#)
- [О блоге](#)
- [Карта](#)

Построение графиков в программе Wealth-Lab при помощи WealthScript

 **АВТОР ДМИТРИЙ ВЛАСОВ**  **МАЯ - 12 - 2012**  **КОММЕНТАРИЕВ: 3**

При программировании торговых стратегий приходится иметь дело не только с сухими цифрами. Так как работа с торговыми стратегиями тесно связана с человеком и его восприятием, никуда не деться от представления информации в графическом виде.

WealthScript предлагает целый набор методов для того, чтобы контролировать как отображается графическая информация. Вы можете отображать на графике индикаторы, создавать на графике новые области, добавлять на график текст, примечания и графические объекты к барам, включая даже полноценные рисунки.



Как только у Вас появится необходимость выполнить с помощью программного кода графические задачи в окне с графиками, смело открывайте рубрику "Cosmetic Chart" в справочнике "QuickRef" (F11) для того, чтобы отыскать соответствующую функцию.

Области графика (Chart Panes)

На графике располагаются три типа областей, но каждая из них является объектом ChartPane.

Область цены (Price Pane)

Ряды данных (OHLC) баров отображаются в области цен. Это отображение соответствует выбранному стилю графика.

По умолчанию все окна, где на графике отображается торговая стратегия, имеют область цены, которая обозначается в коде стратегии как PricePane.

Область объема (Volume Pane)

Объемы (Bars.Volume) отображаются на графике в виде гистограммы в области объемов. По умолчанию, все окна стратегии с графиком имеют область объемов, которая обозначается в коде стратегии как VolumePane.

Индивидуальные области (Custom Panes)

Все прочие области графика являются индивидуальными областями. Они создаются в коде стратегии с помощью метода CreatePane(), который возвращает объект типа ChartPane.

Как: создать индивидуальную область на графике

Для того, чтобы создать индивидуальную область на графике, вызовите метод CreatePane(). В примере создается три индивидуальных области на графике - две выше области цен и одна ниже области объемов. Должны ли создаваемые индивидуальные области графика отображаться выше или ниже областей цены и объема - указывается с помощью соответствующих параметров метода CreatePane().

Пример ([как выполнить пример приведенного кода](#))...

Код примера на C# (кликните чтобы посмотреть)

```
1 | protected override void Execute()
```

RSS товары



RSS новости



Twitter



Последние комментарии

макс: А где взять свечки с открытым интересом? на сайте финама нет



ring10: Добрый день Дмитрий Как сделать, чтобы открывался 1 велф лаб? Если открываешь под пользователем, т [...]



Дмитрий Власов: Буян, рад, что Вам пригодились материалы данной статьи!



Максим: Сам себе отвечу. protected override void Execute() { Bars rts = GetExternalSymbol("RTS", t [...]



Буян: Спасибо Дмитрий за статью! Мне удалось разработать хорошую рабочую торговую систему, используя возмо [...]



Последние посты



Бесплатный вебинар: 7 смертных грехов при оптимизации торговых систем

НОЯБРЯ - 5 - 2012



Отзывы участников курса до результата Wealth-Lab. С нуля до первой торговой системы за 5 дней.

НОЯБРЯ - 3 - 2012



Быстрее, Выше, Сильнее: Обновление Wealth-Lab до новой версии 6.4

ОКТАБРЯ - 23 - 2012



Курс до результата: Wealth-Lab – от нуля до первой торговой системы за 5 дней

ОКТАБРЯ - 23 - 2012



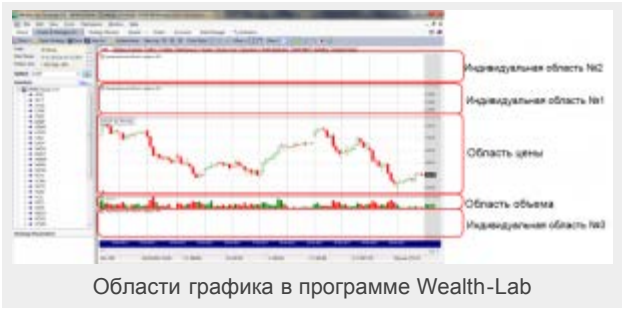
Обзорная экскурсия по достопримечательностям программы Wealth-Lab

ОКТАБРЯ - 20 - 2012

Берите

```
2 | {
3 |     bool abovePricePane = true; //выше области цен? Да!
4 |     bool displayGrid = true; //отображать сетку? Да!
5 |
6 |     //Создаем первую индивидуальную область графика
7 |
8 |     ChartPane myCustomPane1 = CreatePane(40, abovePricePane, displayGrid); //рисуем
9 | первую индивидуальную область на графике выше области цен и с сеткой
10 | DrawLabel(myCustomPane1, "Индивидуальная область графика №1", Color.Black); //
11 | делаем надпись на первой индивидуальной области цен
12 |
13 |     // Создаем вторую индивидуальную область графика
14 |
15 |     displayGrid = false; // отображать сетку? Нет!
16 | ChartPane myCustomPane2 = CreatePane(40, abovePricePane, displayGrid);
17 | DrawLabel(myCustomPane2, "Индивидуальная область графика №2", Color.Black);
18 |
19 |     // Создаем третью индивидуальную область графика
20 |
21 |     abovePricePane = false; //выше области цен? Нет!
22 | ChartPane myCustomPane3 = CreatePane(40, abovePricePane, displayGrid);
23 | DrawLabel(myCustomPane3, "Индивидуальная область графика №3", Color.Black);
24 | }
```

В результате этого примера получится вот такой график:



Обратите внимание на то, что если область цены и область объема существуют на графике в программе Wealth-Lab автоматически по умолчанию, то индивидуальные области один, два и три мы отобразили сами с помощью программного кода. Причем их расположение относительно области цены мы выбрали сами. Также мы определили, что сетка будет отображаться в индивидуальной области №1, а в индивидуальной области №2 и №3 сетка не отображается.

Как: скрыть область объема на графике

Возможно, что Ваша стратегия не предполагает работы с объемами. Поэтому визуальное отображение объемов будет только отвлекать и занимать лишнее место на графике. В программе Велс Лаб существует возможность скрыть область объемов с графика стратегии.

Для того, чтобы скрыть область объемов, нужно просто вызвать метод HideVolume() из любого места кода стратегии.

Пример (как выполнить пример приведенного кода)...

```
Код примера на C# (кликните чтобы посмотреть)
1 | protected override void Execute()
2 | {
3 |     HideVolume(); // метод, позволяющий скрыть область объемов на графике.
4 | }
```

Прочитав сегодняшний пост Вы узнали о том, как можно строить графики в программе Велс Лаб с помощью специальных методов WealthScript. Также сегодня Вы узнали, что график стратегии состоит из нескольких областей, которыми мы можем управлять с помощью специальных команд. В следующий раз Вы узнаете, как на одном графике отобразить цены нескольких финансовых инструментов в различных вариантах.

- Если Вам понравилась эта статья - подпишитесь на **новые СТАТЬИ по RSS** или получайте наши **новые СТАТЬИ на E-mail**:
Рекомендую почитать также:
- Отображение на графике синтетических финансовых инструментов
 - Отображение на графике в программе Велс Лаб рядов данных (индикаторов, осцилляторов).
 - Написание текста на графиках Wealth-Lab программными методами
 - Рисование графических объектов на графике Wealth-Lab
 - Наполнение созданных самостоятельно DataSeries данными при помощи WealthScript

РУБРИКИ: PAINTING THE CHART, WEALTHSCRIPT



Видео ВЕБИНАРОВ



Дмитрия Власова





МТС под микроскопом:
WLMA System
Подробный разбор торговой системы WLMA System
3900 руб.
[В корзину](#)

Рубрики

Выберите рубрику

Обновления:

-  Новые посты по Email
-  Новые посты по RSS
-  Комментарии по RSS
-  Следим на Twitter

Метки

C# HowTo: **Wealth-Lab**
Видео Индикаторы МТС Торговые роботы Юмор алготрейдинг мероприятия обучение

Управление

Регистрация

Вход

RSS Записей

RSS Комментарии

WordPress.org

Комментариев: 3

Синхронизация данных при построении торговых стратегий с несколькими финансовыми инструментами | Финансовая лаборатория пишет:

12 мая 2012 в 10:10

[...] Построение графиков в программе Wealth-Lab при помощи WealthScr... Главная Как получить доступ ко второму [...]

Ответить

Отображение на графике в программе Велс Лаб рядов данных | Финансовая лаборатория пишет:

14 мая 2012 в 11:44

[...] 2. На выбор: Если ряды данных не должны отображаться на графике в области цен либо в области объема, создать индивидуальную область на графике. [...]

Ответить

Ответы на вопросы по программированию торговых стратегий в Wealth-Lab | Финансовая лаборатория пишет:

15 мая 2012 в 15:33

[...] [...]

Ответить

Оставить комментарий

Имя (Обязательно)

Почта (Обязательно)

Вэб-сайт

Получать новые комментарии по электронной почте. Вы можете [подписаться](#) без комментирования.

© 2012, Финансовая лаборатория. Все права защищены.
Любое использование материалов допускается только с указанием активной ссылки на источник.

Подписка

- [RSS](#)
- [Twitter](#)
- [ВКонтакте](#)
- [facebook](#)

Сайт

- [Создать блог](#)
- [Содержание](#)
- [О блоге](#)
- [Карта](#)

Отображение на графике синтетических финансовых инструментов

 **АВТОР** ДМИТРИЙ ВЛАСОВ  **МАЯ** - 13 - 2012  **КОММЕНТАРИЕВ:** 5

О том, как строить графики в программе Wealth-Lab с помощью программного кода мы узнали в предыдущем посте. Сегодня же рассмотрим специфические ситуации, когда нужно отобразить в графической форме не один обыкновенный финансовый инструмент, а синтетический финансовый инструмент, который был получен в результате специальных расчетов либо с применением особых методов.



Первый вопрос, который сегодня рассмотрим будет такой:

Как: отобразить на графике внешний финансовый инструмент

Если Вам не нужен доступ к данным внешнего финансового инструмента, а просто хотите отобразить его на графике, используйте метод PlotSymbol(). Метод PlotSymbol() отображает на графике любые объекты типа Bars. При этом применяется выбранный в настоящее время стиль графика (свечи, бары, и т.д.).

Обращаю особое внимание на то, что метод PlotSymbol() неприменим для трендовых стилей графика, использующих нестандартную шкалу X (таких как Ренко, Каги, крестики и нолики и графики линейного прорыва)

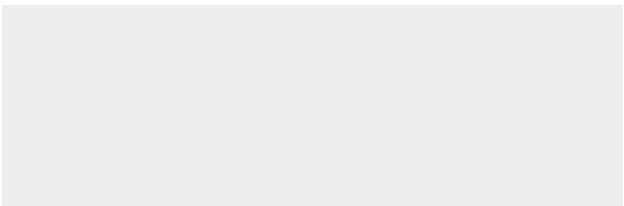
Рассмотрим теперь пример того, как с помощью метода PlotSymbol() отобразить на графике область с внешним финансовым инструментом.

Пример ([как выполнить пример приведенного кода](#))...

Код примера на С# (кликните чтобы посмотреть)

```
1  protected override void Execute()
2
3  {
4      //Отобразить на индивидуальной области графика финансовый инструмент "ГАЗПРОМ"
5
6      Bars gazp = GetExternalSymbol("GAZP", true);
7      ChartPane gazpPane = CreatePane( 100, false, true);
8      PlotSymbol( gazpPane, gazp, Color.DodgerBlue, Color.Red);
9  }
```

В результате выполнения этого кода получается вот такой график:



RSS товары



RSS новости



Twitter



Последние комментарии

макс: А где взять свечки с открытым интересом? на сайте финама нет



ring10: Добрый день Дмитрий Как сделать, чтобы открывался 1 велф лаб? Если открываешь под пользователем, т [...]



Дмитрий Власов: Буян, рад, что Вам пригодились материалы данной статьи!



Максим: Сам себе отвечу. protected override void Execute() { Bars rts = GetExternalSymbol("RTS", t [...]



Буян: Спасибо Дмитрий за статью! Мне удалось разработать хорошую рабочую торговую систему, используя возмо [...]



Последние посты



Бесплатный вебинар: 7 смертных грехов при оптимизации торговых систем

НОЯБРЯ - 5 - 2012



Отзывы участников курса до результата Wealth-Lab. С нуля до первой торговой системы за 5 дней.

НОЯБРЯ - 3 - 2012



Быстрее, Выше, Сильнее: Обновление Wealth-Lab до новой версии 6.4

ОКТАБРЯ - 23 - 2012



Курс до результата: Wealth-Lab – от нуля до первой торговой системы за 5 дней

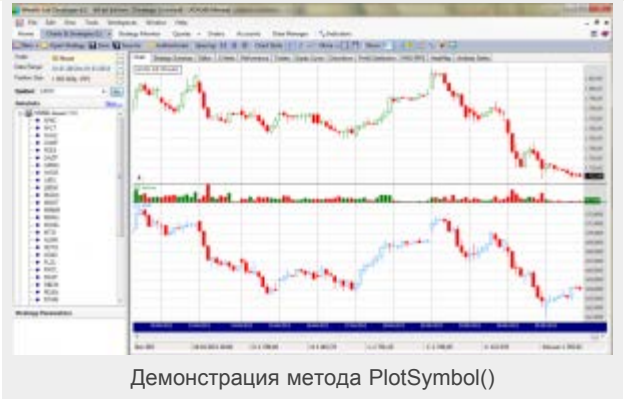
ОКТАБРЯ - 23 - 2012



Обзорная экскурсия по достопримечательностям программы Wealth-Lab

ОКТАБРЯ - 20 - 2012

Берите



Как видите, основной финансовый инструмент (в нашем случае Лукойл) отобразился в области цен сверху графика, а внешний финансовый инструмент (в нашем случае Газпром), график которого построен с помощью метода PlotSymbol() отобразился в нижней области графика.

Существуют такие финансовые инструменты, которые не торгуются на бирже, а рассчитываются с помощью некоторых математических операций. Такие финансовые инструменты называются синтетическими. Давайте теперь посмотрим, какие возможности имеет Велс Лаб для графического отображения синтетических финансовых инструментов.

Как: отобразить на графике синтетический финансовый инструмент

Для отображения на графике синтетического финансового инструмента в программе Wealth-Lab существует специальный метод PlotSyntheticSymbol. При вызове этого метода ему необходимо передать все четыре основные цены OHLC (открытия, максимальную, минимальную и закрытия). Это позволяет сформировать бары правильной формы.

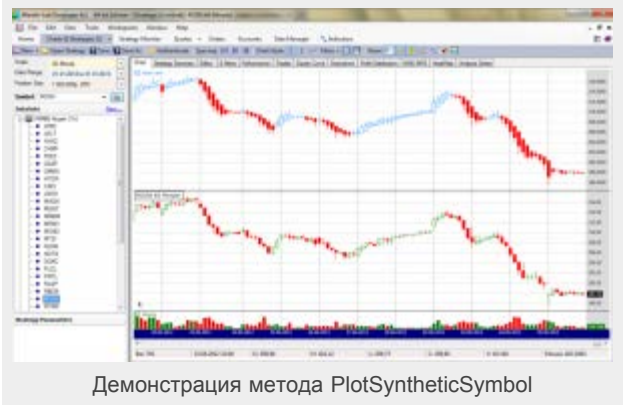
В качестве примера синтетического финансового инструмента может выступать график Heikin-Ashi. Давайте попытаемся построить его.

Пример (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть)

```
1  /* Отобразить на индивидуальной области график Heikin-Ashi выше области цен */
2
3  protected override void Execute()
4
5  {
6      // Добавляем 0 для того, чтобы создать копию ряда данных (DataSeries),
7      // базирующегося на основных последовательностях
8
9      DataSeries HO = Open + 0;
10     DataSeries HH = High + 0;
11     DataSeries HL = Low + 0;
12     DataSeries HC = (Open + High + Low + Close) / 4;
13
14     for (int bar = 1; bar < Bars.Count; bar++)
15     {
16         double o1 = HO[ bar - 1 ];
17         double c1 = HC[ bar - 1 ];
18
19         HO[bar] = ( o1 + c1 ) / 2;
20         HH[bar] = Math.Max( HO[bar], High[bar] );
21         HL[bar] = Math.Min( HO[bar], Low[bar] );
22     }
23
24     ChartPane haPane = CreatePane(100, true, true);
25     PlotSyntheticSymbol(haPane, "Heikin-Ashi", HO, HH, HL, HC, Volume,
26     Color.DodgerBlue, Color.Red); //отображаем на графике синтетический финансовый
    инструмент
    }
```

После запуска полученной стратегии получим такой график:



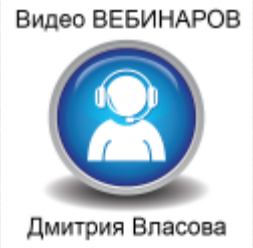
Как видите, на области выше области цен отобразился полноценный Heiki-Ashi график - благодаря методу PlotSyntheticSymbol().

Теперь допустим, что Вам необходимо на одной и той же области графика отобразить и график основного финансового инструмента и график того финансового инструмента, который будет служить ориентиром (так называемый бенчмарк).

Давайте посмотрим, как эту задачу можно выполнить с помощью программы:

Как: сравнить основной финансовый инструмент с бенчмарком

Для того, чтобы выполнить эту нетривиальную задачу нужно привести цены финансового инструмента, который является бенчмарком и основного финансового инструмента к единой шкале, после чего для



МТС под микроскопом: WLMA System

Подробный разбор торговой системы WLMA System
3900 руб.

В корзину

Рубрики

Выберите рубрику

Обновления:

- Новые посты по Email
- Новые посты по RSS
- Комментарии по RSS
- Следим на Twitter

Метки

C# HowTo: Wealth-Lab
Видео Индикаторы МТС Торговые
роботы Юмор алготрейдинг
мероприятия обучение

Управление

Регистрация

Вход

RSS Записей

RSS Комментариев

WordPress.org

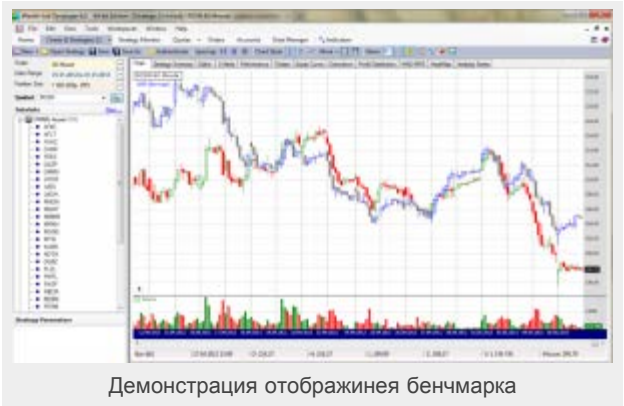
отображения на графике бенчмарка использовать метод PlotSyntheticSymbol().

Пример (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть)

```
1  /*Сначала создадим специальный метод, который позволяет отображать бенчмарк на
2  основном графике*/
3
4  public void PlotBenchMark(string symbol, DateTime fromDate)
5
6  {
7      if (Bars.Symbol != symbol)
8      {
9          int bar = Bars.ConvertDateToBar(fromDate, false);
10
11          double refPrice = Close[bar];
12          Bars bmBars = GetExternalSymbol(symbol, true);
13          double factor = refPrice / bmBars.Close[bar];
14          DataSeries dsO = bmBars.Open * factor;
15          DataSeries dsH = bmBars.High * factor;
16          DataSeries dsL = bmBars.Low * factor;
17          DataSeries dsC = bmBars.Close * factor;
18          PlotSyntheticSymbol(PricePane, symbol + " (relative)", dsO, dsH, dsL, dsC,
19          Volume, Color.Blue, Color.Gray);
20      }
21  }
22
23  //теперь воспользуемся созданным методом
24  protected override void Execute()
25
26  {
27      // Произведем сравнение с бенчмарком (ГАЗПРОМом) с точкой отсчета 01 января 2011
28      PlotBenchMark("GAZP", new DateTime(2010, 1, 1));
29  }
```

Получается такой график:



Демонстрация отображения бенчмарка

Как видите, на одном графике отображается и основной инструмент, и бенчмарк (Газпром) - чего мы и добивались.

Обратите внимание, что в коде сначала мы создаем нужный нам метод, и лишь затем имеем возможность легко и просто воспользоваться этим методом. В принципе, мы можем создавать собственную коллекцию полезных методов для обработки данных и в этом несомненное преимущество программирования на C#.

На сегодня - всё. Вы увидели, как посредством существующих и самостоятельно создаваемых методов можно отображать на графике не только цену и объемы основного финансового инструмента, но и любые данные по более сложным, производным или синтетическим инструментам. В следующий раз рассмотрим все способы отображения на графиках индикаторов, осцилляторов и прочих рядов данных.

Если Вам понравилась эта статья - подпишитесь на **новые СТАТЬИ по RSS** или получайте наши **новые СТАТЬИ на E-mail**:

Рекомендую почитать также:

- Построение графиков в программе Wealth-Lab при помощи WealthScript
- Отображение на графике в программе Велс Лаб рядов данных (индикаторов, осцилляторов).
- Написание текста на графиках Wealth-Lab программными методами
- Рисование графических объектов на графике Wealth-Lab
- Программирование однопозиционных стратегий (SP Strategies) в программе Wealth-Lab

РУБРИКИ: PAINTING THE CHART, WEALTHSCRIPT

Комментариев: 5

Построение графиков в программе Wealth-Lab при помощи WealthScript | Финансовая лаборатория пишет:
13 мая 2012 в 09:29

[...] [...]

Ответить

Ответы на вопросы по программированию торговых стратегий в Wealth-Lab | Финансовая лаборатория пишет:
14 мая 2012 в 13:35

[...] 30. Как отобразить на графике в программе Велс Лаб синтети... [...]

Ответить

Леобакин пишет:

16 мая 2012 в 18:28



Дмитрий, здравствуйте.

А как отобразить (ну и потом применить к нему) график синтетического инструмента, равного допустим “ГАЗ-ЛУК”?

Ответить

Дмитрий Власов пишет:

16 мая 2012 в 19:46



Это сделать очень просто:

Допустим, вы в списке инструментов (DataSet) выбираете основной финансовый инструмент

“Газпром” (GAZP).

Тогда в коде стратегии делаете следующие шаги:

Код примера на C# (кликните чтобы посмотреть) ?

```
1 // 1. с помощью метода GetExternalSymbol() создаете объект Bars второго
2 финансового инструмента (Лукойла)
3
4 Bars lkoh = GetExternalSymbol("LKOH", true); //получаем бары внешнего
5 финансового инструмента
6
7 //
8 2. Создаем ряд данных (DataSeries) в котором для каждого бара будет
9 отображаться значение спреда (разницы) между газпромом и лукойлом
10
11 DataSeries spread = Close - lkoh.Close;
12
13 // 3. Отображаем на графике спред:
14
15 ChartPane spreadPane = CreatePane(40, true, true); //создаем индивидуальную
16 область графика (выше области цены, с сеткой)
17
18 PlotSeries(spreadPane, spread, Color.Black, WealthLab.LineStyle.Line, 2);
19 //рисуем на индивидуальной области спред
```

Вот и все.

Если хотите не только отображать второй финансовый инструмент, но и совершать с ним сделки, используйте метод SetContext()

Более подробное об этом смотрите [здесь: Как получить доступ ко второму инструменту](#) а также [здесь: Как отразить на графике DataSeries](#)

Чуть позже напишу статью с примером торговой стратегии, которая торгует парный трейдинг.

Ответить

Инструкция по программированию торговых стратегий в WealthLab | Финансовая лаборатория пишет:

20 июня 2012 в 12:23

[...] [...]

Ответить

Оставить комментарий

Имя (Обязательно)

Почта (Обязательно)

Вэб-сайт

Получать новые комментарии по электронной почте. Вы можете [подписаться](#) без комментирования.

© 2012, Финансовая лаборатория. Все права защищены.
Любое использование материалов допускается только с указанием активной ссылки на источник.



448
240
185



В день мой блог
посещает **77**
человек,
сегодня **123**



mail.ru РЕЙТИНГ
757135 520
193



BY FEEDBURNER

Подписка

- [RSS](#)
- [Twitter](#)
- [ВКонтакте](#)
- [facebook](#)

Сайт

- [Создать блог](#)
- [Содержание](#)
- [О блоге](#)
- [Карта](#)

Отображение на графике в программе Велс Лаб рядов данных (индикаторов, осцилляторов).

 **АВТОР** ДМИТРИЙ ВЛАСОВ  **МАЯ** - 14 - 2012  **КОММЕНТАРИЕВ:** 3

При разработке торговых стратегий нам не обойтись без построения собственных индикаторов, осцилляторов, числовых фильтров и подобных инструментов. Числовые значения этих инструментов хранятся в программе в виде рядов данных (DataSeries), о которых мы уже говорили ранее. Сегодня рассмотрим, как можно отобразить на графиках в программе Велс Лаб различного рода ряды данных.



Как: отобразить на графике последовательность данных (DataSeries)

Для того, чтобы отобразить ряды данных нужно выполнить следующие шаги:

1. Создать **числовую последовательность (DataSeries)** и, если это необходимо, установить значение для каждого из баров прежде чем переходить к следующим шагам.
2. На выбор: Если ряды данных не должны отображаться на графике в области цен либо в области объема, **создать индивидуальную область на графике**.
3. Выполнить один из следующих методов: PlotSeries(), PlotSeriesFillBand(), PlotSeriesDualFillBand() или PlotSeriesOscillator().

Методы PlotSeries* выполняются только один раз для каждого начертания. Они должны выполняться вне цикла, пробегающего бары.

В предыдущих постах уже приводились различные примеры, которые отображали ряды данных DataSeries с использованием метода PlotSeries(). Для того, чтобы увидеть еще больше примеров для каждого из типов методов PlotSeries - смотрите QuickRef (F11). Для этого выберите там рубрику Cosmetic Chart.*

Как: отобразить на графике область объемов выше чем область цен

Для того, чтобы отобразить область объемов над областью цен на графике нужно:

- скрыть область объемов, выводящуюся по умолчанию;
- создать индивидуальную область на графике;
- отобразить объемы (Bars.Volume) на новой индивидуальной области графика.

Вот как это будет выглядеть в виде кода:

Пример ([как выполнить пример приведенного кода](#))...

RSS товары



RSS новости



Twitter



Последние комментарии

макс: А где взять свечки с открытым интересом? на сайте финама нет



ring10: Добрый день Дмитрий Как сделать, чтобы открывался 1 велф лаб? Если открываешь под пользователем, т [...]



Дмитрий Власов: Буян, рад, что Вам пригодились материалы данной статьи!



Максим: Сам себе отвечу. protected override void Execute() { Bars rts = GetExternalSymbol("RTS", t [...]



Буян: Спасибо Дмитрий за статью! Мне удалось разработать хорошую рабочую торговую систему, используя возмо [...]



Последние посты



Бесплатный вебинар: 7 смертных грехов при оптимизации торговых систем

НОЯБРЯ - 5 - 2012



Отзывы участников курса до результата Wealth-Lab. С нуля до первой торговой системы за 5 дней.

НОЯБРЯ - 3 - 2012



Быстрее, Выше, Сильнее: Обновление Wealth-Lab до новой версии 6.4

ОКТАБРЯ - 23 - 2012



Курс до результата: Wealth-Lab – от нуля до первой торговой системы за 5 дней

ОКТАБРЯ - 23 - 2012



Обзорная экскурсия по достопримечательностям программы Wealth-Lab

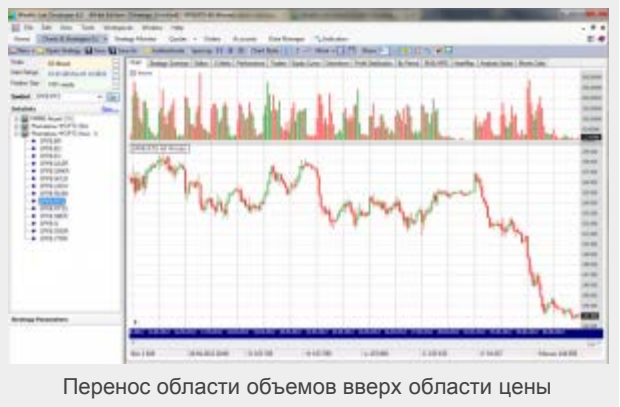
ОКТАБРЯ - 20 - 2012

Берите

Код примера на C# (кликните чтобы посмотреть)

```
1 protected override void Execute()
2 {
3     HideVolume(); //скрываем область объемов
4
5     ChartPane myVolumePane = CreatePane(40, true, true); //создаем индивидуальную
6 область графика (выше области цены, с сеткой)
7
8     PlotSeries(myVolumePane, Bars.Volume, Color.Black, WealthLab.LineStyle.Histogram,
9 5); //рисуем на индивидуальной области объемы
10
11     //раскрашиваем гистограмму объемов в нужный цвет, запуская цикл по всем барам.
12
13     for (int bar = 0; bar < Bars.Count; bar++)
14     {
15         if (Close[bar] > Open[bar])
16         {
17             SetSeriesBarColor(bar, Bars.Volume, Color.Green);
18
19         else
20         {
21             SetSeriesBarColor(bar, Bars.Volume, Color.Red);
22         }
23     }
24 }
```

После выполнения кода мы получим такой график:



Как видите, область объемов здесь отображается выше, а не ниже (как это было по умолчанию) области цен.

Есть возможность отображать прямо на графике не только данных, которые связаны с рядами цен и объемов, но и данные о параметрах торговой стратегии, к примеру, данные о выигрышах и проигрышах.

Как: отобразить на индивидуальной области графика количественные показатели торговой стратегии

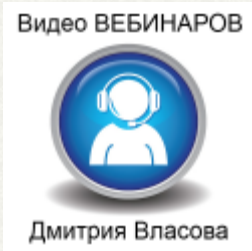
Так как Wealth-Lab обрабатывает стратегии в режиме простого дохода и каждая сделка просто равняется 1-му лоту (или контракту), невозможно получить прямой доступ к количественным показателям стратегии прямо из кода. Это относится к таким количественным показателям стратегии, как величина позиции (Position Sizing), кривая капитала (Equity), наличные (Cash) или просадки (Drowdown). Расчетом таких количественных показателей занимаются задачи в визуализаторе процессов (Performance Visualizer), запускаемые сразу же после расчета основной стратегии (смотри API).

Тем не менее, Вы можете отобразить на графике некоторые текущие количественные характеристики стратегии - такие, как "Среднее время удержания позиции", или "Соотношение прибыльных трейдов к убыточным". Именно такая возможность отображается в следующем чуть ниже примере.

Пример (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть)

```
1 protected override void Execute()
2 {
3     /*
4
5     Здесь Вы описываете правила Вашей стратегии
6
7     */
8
9     PlotWinRate(); //а теперь отображаем количественный показатель Вашей стратегии
10 WinRate (отношение выигрышей к проигрышам)
11 }
12
13 //создаем
14 специальный метод, который позволит нам отобразить прямо на
15 индивидуальной области графика показатель стратегии WinRate
16 public void PlotWinRate()
17 {
18     {
19         DataSeries winRate = Close - Close; // Creates a DataSeries filled with zeroes
20         DataSeries tradeTotal = winRate + 0;
21
22         foreach (Position p in Positions) //пробегаем весь цикл в существующих позициях
23 по отработанной стратегии:
24         {
25             if (p.ExitBar > -1) //для всех позиций, у которых бар выхода из позиции равен
26 или больше 0
27             {
28                 tradeTotal[p.ExitBar] += 1; //считаем сумму всех позиций
29
30                 if (p.NetProfitPercent > 0)
31                 {
32                     winRate[p.ExitBar] += 1; //считаем сумму прибыльных позиций
33                 }
34             }
35         }
36     }
37 }
```



МТС под микроскопом: WLMA System

Подробный разбор торговой системы WLMA System
3900 руб.

В корзину

Рубрики

Выберите рубрику

Обновления:

- Новые посты по Email
- Новые посты по RSS
- Комментарии по RSS
- Следим на Twitter

Метки

C# HowTo: **Wealth-Lab**
Видео Индикаторы МТС Торговые
роботы Юмор алготрейдинг
мероприятия обучение

Управление

Регистрация

Вход

RSS Записей

RSS Комментариев

WordPress.org

```
41     for (int bar = 1; bar < Bars.Count; bar++)
42     {
43
44         tradeTotal[bar] = tradeTotal[bar] + tradeTotal[bar-1];
45         winRate[bar] = winRate[bar] + winRate[bar - 1];
46
47     }
48
49     winRate = 100 * winRate / tradeTotal;
50     winRate.Description = "Win Rate (%)";

```

```

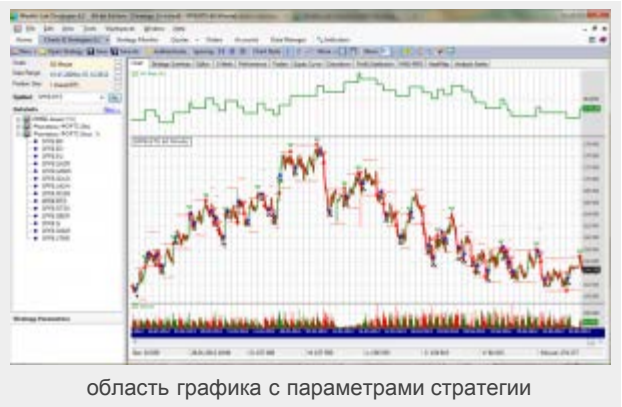
    ChartPane winPane = CreatePane(40, true, true);
    PlotSeries(winPane, winRate, Color.Green, WealthLab.LineStyle.Solid, 1);

}

```

Если Вы подставите в начало этого кода свою торговую стратегию, то на графике отобразится текущее значение показателя, который показывает отношение прибыльных сделок к убыточным.

Для примера, я вставил в начало этого кода простейшую торговую систему, которая входит в позицию и сразу ставит стоп-лосс и тейк-профит. Просто для демонстрации работы панели WinRate. В результате получился вот такой график:



Вы можете увидеть, что сверху на панели добросовестно отображается отношение выигрышных сделок к убыточным. Точно также можно настроить и осуществить вывод той информации, которая Вам нужна.

Еще одна тема, которую хотел бы затронуть сегодня касается контроля над вертикальной шкалой.

Как: контролировать вертикальную Y-шкалу на графиках в программе Wealth-Lab

Велс Лаб автоматически подстраивает масштаб по вертикальной шкале Y для каждой области на графике таким образом, чтобы каждый график цены и индикаторов был виден. Невозможно сжать вертикальную шкалу Y таким образом, чтобы данные были видны на одной части графика, но не видны на другой части этого же графика.

С другой стороны, при использовании метода SetPaneMinMax(), Вы можете "растянуть" вертикальную шкалу соответствующей области таким образом, чтобы она оставалась неизменной на широком диапазоне значений.

К примеру, если осцилятор RSI колеблется между значениями 50 и 90 достаточно долгое время, то Wealth-Lab неминуемо автоматически изменит масштаб вертикальной оси таким образом, чтобы Вы могли видеть только значения между этими уровнями. Однако, если Вы ждете момента, когда RSI опустится ниже уровня 30, Вы можете принудительно отображать участок вертикальной шкалы таким образом, чтобы было видно уровень 30 или даже ниже - чтобы видеть перспективу текущего значения.

Пример (как выполнить пример приведенного кода)...

```

Код примера на C# (кликните чтобы посмотреть)
1  protected override void Execute(){
2      // Обеспечение гарантированной видимости диапазона значений индикатора RSI от 30
3      до 70
4
5      ChartPane RSIPane = CreatePane( 60, true, true );
6      PlotSeries( RSIPane, RSI.Series( Close, 14 ), Color.Navy,
7      WealthLab.LineStyle.Solid, 1 );
      SetPaneMinMax( RSIPane, 30, 70 );
  }

```

На сегодня информации достаточно. В процессе Вы узнали о нескольких методах, которые позволяют Вам достаточно гибко отображать на разных областях графика всевозможные количественные показатели, рассчитанные перед или в процессе выполнения стратегии. Следующий раз мы затронем тему написания текста на графике.

Если Вам понравилась эта статья - подпишитесь на **новые СТАТЬИ по RSS** или получайте наши **новые СТАТЬИ на E-mail**:

Рекомендую почитать также:

- Построение графиков в программе Wealth-Lab при помощи WealthScript
- Отображение на графике синтетических финансовых инструментов
- Написание текста на графиках Wealth-Lab программными методами
- Рисование графических объектов на графике Wealth-Lab
- Программирование однопозиционных стратегий (SP Strategies) в программе Wealth-Lab

Отображение на графике синтетических финансовых инструментов | Финансовая лаборатория пишет:

14 мая 2012 в 10:51

[...] Отображение на графике в программе Велс Лаб рядов данн... Главная Построение графиков в программе Wealth-Lab [...]

Ответить

Ответы на вопросы по программированию торговых стратегий в Wealth-Lab | Финансовая лаборатория пишет:

17 мая 2012 в 13:42

[...] 34. Как отобразить на индивидуальной области графика коли... [...]

Ответить

Юрий пишет:

23 октября 2012 в 21:55



Мне кажется нужно немного поменять два комментария в программе, вычисляющей процент прибыльных сделок. Вместо “считаем сумму всех позиций” следует написать что-то вроде “записываем единицы в серию tradeTotal на тех барах, где были закрыты позиции”. А вместо “считаем сумму прибыльных позиций” написать “записываем единицы в серию winRate на тех барах, где были закрыты прибыльные позиции”.

Ответить

Оставить комментарий

Имя (Обязательно)

Почта (Обязательно)

Вэб-сайт

Получать новые комментарии по электронной почте. Вы можете [подписаться](#) без комментирования.

© 2012, Финансовая лаборатория. Все права защищены.

Любое использование материалов допускается только с указанием активной ссылки на источник.

448

240

185

В день мой блог посещает 77 человек, сегодня

124

mail.ru РЕЙТИНГ

757136

521193

Антиф

BY FEEDBURNER

Подписка

[RSS](#)

[Twitter](#)

[ВКонтакте](#)

[facebook](#)

Сайт

[Создать блог](#)

[Содержание](#)

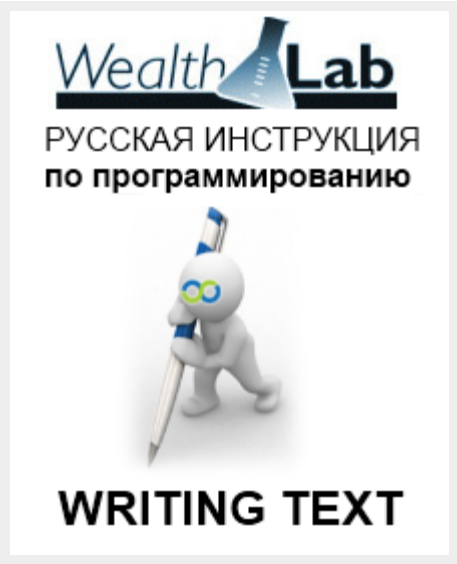
[О блоге](#)

[Карта](#)

Написание текста на графиках Wealth-Lab программными методами

АВТОР ДМИТРИЙ ВЛАСОВ МАЯ - 15 - 2012 КОММЕНТАРИЕВ: 2

После того, как графики инструмента и индикаторов начерчены, появляется необходимость снабдить эти графики некоторыми пояснениями. Иногда нужно прямо на графике написать пометки или пояснения. Конечно, это можно сделать вручную, но если изучить несложные методы работы с текстом в программе Wealth Lab, у Вас появится больше возможностей для работы с информацией.



Для написания текста на графиках с помощью программного кода в программе Велс Лаб доступны следующие методы:

AnnotateBar()

Данный метод необходим для написания текстовых символов выше максимальных цен бара и ниже минимальных цен бара в области цен.

AnnotateChart()

Этот метод подобен методу AnnotateBar(), но он более гибкий, так как Вы можете выбирать область графика, необходимый бар, а также выравнивать размещение текста относительно баров и соответствующих цен.

DrawLabel()

В верхнем левом углу области цен графика, ниже чем название финансового инструмента располагается несколько надписей. Эти надписи автоматически вытягиваются для того чтобы отобразить здесь описание уже нанесенных на график рядов данных (индикаторов, осцилляторов и т.п.).

DrawText()

Этот метод дает возможность написать текстовые символы. Причем расположение этих символов определяется координатами x, y в пикселях, где нулевой точкой отсчета является верхний левый угол графика.

Как: написать вертикальный текст на гарфике в программе Wealth-Lab?

Когда пространство ограничено и Вам необходимо вместить больше примечаний, воспользуйтесь преимуществом метода Annotate Bar() для того, чтобы вставлять по одной букве за один раз - как показано в этом примере:

Пример (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть)

```
1 // Метод написания вертикального текста
2
3 public void VerticalText(int bar, string str, bool abovePrices, Color color)
4 {
5     if (abovePrices)
6         for (int n = str.Length - 1; n >= 0; n--)
7             AnnotateBar(str.Substring(n, 1), bar, abovePrices, color);
8     else
```

RSS товары



RSS новости



Twitter



Последние комментарии

- макс: А где взять свечки с открытым интересом? на сайте финама нет


- ring10: Добрый день Дмитрий Как сделать, чтобы открывался 1 велф лаб? Если открываешь под пользователем, т [...]


- Дмитрий Власов: Буян, рад, что Вам пригодились материалы данной статьи!


- Максим: Сам себе отвечу. protected override void Execute() { Bars rts = GetExternalSymbol("RTS", t [...]


- Буян: Спасибо Дмитрий за статью! Мне удалось разработать хорошую рабочую торговую систему, используя возмо [...]



Последние посты

- 

Бесплатный вебинар: 7 смертных грехов при оптимизации торговых систем

НОЯБРЯ - 5 - 2012
- 

Отзывы участников курса до результата Wealth-Lab. С нуля до первой торговой системы за 5 дней.

НОЯБРЯ - 3 - 2012
- 

Быстрее, Выше, Сильнее: Обновление Wealth-Lab до новой версии 6.4

ОКТАБРЯ - 23 - 2012
- 

Курс до результата: Wealth-Lab – от нуля до первой торговой системы за 5 дней

ОКТАБРЯ - 23 - 2012
- 

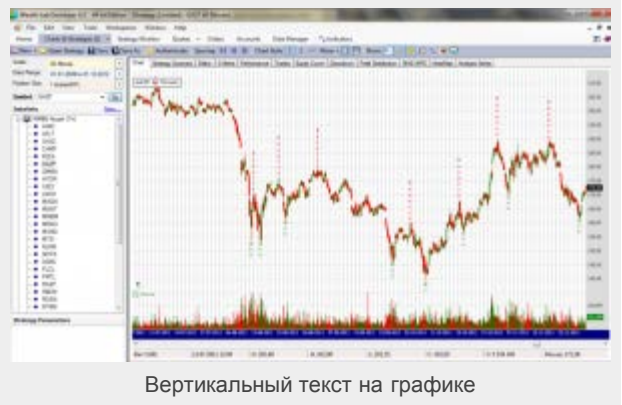
Обзорная экскурсия по достопримечательностям программы Wealth-Lab

ОКТАБРЯ - 20 - 2012

Берите

```
9
10
11     for (int n = 0; n < str.Length; n++)
12         AnnotateBar(str.Substring(n, 1), bar, abovePrices, color);
13 }
14
15 protected override void Execute()
16 {
17     // Создаем дополнительное место для верхнего символа
18
19     HidePanelLines();
20     CreatePane(3, true, false);
21
22     // находим номера баров, на которых формируются пятипроцентные вершины или
23     впадины
24
25     int bar = Bars.Count - 1;
26
27     while (bar > 0)    {
28
29         bar = (int)PeakBar.Series(Close, 5,
30     WealthLab.Indicators.PeakTroughMode.Percent)[bar];
31
32         if (bar > -1) VerticalText(bar, "Вершина", true, Color.Red); //используя
33     созданный метод пишем вертикальный текст
34     }
35
36     bar = Bars.Count - 1;
37
38     while (bar > 0)    {
39
40         bar = (int)TroughBar.Series(Close, 5,
41     WealthLab.Indicators.PeakTroughMode.Percent)[bar];
42
43         if (bar > -1) VerticalText(bar, "Дно", false, Color.Green); //используя
44     созданный метод пишем вертикальный текст
45     }
46 }
47 }
```

Пример с использованием метода AnnotateBar() показывает, как можно сконструировать метод, пишущий текст по одной букве друг над другом вертикально. Потом этот метод применяется и в итоге получается вот такой график:



Как видите, слова "вершина" и "дно" здесь располагаются вертикально над пиками и под впадинами.

Как: сделать более удобным описание рядов данных (индикаторов, осцилляторов и т.п.) для отображении описания ряда данных на графике?

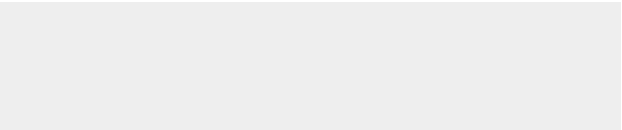
Описание ряда данных на графике зависит от свойства `DataSet.Description`. Wealth-Lab присваивает свойству `DataSet.Description` значение по умолчанию при создании ряда данных. Однако Вы можете самостоятельно поменять это описание на любое, которое Вам нравится, для того чтобы использовать это описание на графике.

Совет: Надпись на графике обуславливается тем цветом, который применяется для отображении на графике рядов данных.

Пример (как выполнить пример приведенного кода)...

```
Код примера на C# (кликните чтобы посмотреть)
1
2 protected override void Execute()
3 {
4     // Создание ряда данных средних цен, сдвинутых на 1 бар вправо по шкале времени.
5
6     DataSet avgPrice = (High + Low) / 2 >> 1;
7
8     string defaultDescription = avgPrice.Description; //запоминаем, кокое было
9     описание по умолчанию
10
11     avgPrice.Description = "Средняя цена (сдвинута на 1 бар вправо)"; //Делаем новое
12     удобное описание:
13
14     PlotSeries(PricePane, avgPrice, Color.Blue, WealthLab.LineStyle.Solid, 1); //
    отображаем ряд данных на графике (описание отображается по умолчанию)
15
16     DrawLabel(PricePane, "Default Description: " + defaultDescription, Color.Red); //
    пишем то описание, которое было по умолчанию...
17 }
18 }
```

Выполнив код мы получим в верхнем левом углу графика вот такой вид:





Видео ВЕБИНАРОВ



Дмитрия Власова



МТС под микроскопом:
WLMA System
Подробный разбор торговой системы WLMA System
3900 руб.
[В корзину](#)

Рубрики

Выберите рубрику

Обновления:

-  Новые посты по Email
-  Новые посты по RSS
-  Комментарии по RSS
-  Следим на Twitter

Метки

C# HowTo: **Wealth-Lab**
Видео **Индикаторы** **МТС** **Торговые**
роботы **Юмор** **алготрейдинг**
мероприятия **обучение**

Управление

- Регистрация
- Вход
- RSS Записей
- RSS Комментариев
- WordPress.org



Написание названия ряда данных на графике в Wealth-Lab

Как Вы можете заметить, синим цветом написано более "дружественное" описание ряда данных, которое мы присвоили свойству `DataSeries.Description`. При этом это описание появляется при отображении ряда данных в верхнем левом углу области цен автоматически, без вызова дополнительных методов. Красным же цветом мы с помощью метода `DrawLabel()` отобразили старое, присваиваемое по умолчанию описание ряда данных.

Итак, сегодня мы вкратце рассмотрели - как можно с помощью программирования отобразить на графике текстовые метки. Это поможет Вам сделать проектируемые Вами стратегии более понятными для Вас и конечных пользователей на этапе их использования. В следующей статье рассмотрим возможности по **отображению на графиках графической информации**, что даст еще больше возможностей. Не забывайте подписываться на наш блог по RSS.

Если Вам понравилась эта статья - подпишитесь на **новые СТАТЬИ по RSS** или получайте наши **новые СТАТЬИ на E-mail**:

Рекомендую почитать также:

- **Построение графиков в программе Wealth-Lab при помощи WealthScript**
- **Отображение на графике синтетических финансовых инструментов**
- **Отображение на графике в программе Велс Лаб рядов данных (индикаторов, осцилляторов).**
- **Рисование графических объектов на графике Wealth-Lab**
- **Анализ данных в Wealth-Lab с учетом разных таймфреймов**

📁 РУБРИКИ: **PAINTING THE CHART, WEALTHSCRIPT**

Комментариев: 2

Ответы на вопросы по программированию торговых стратегий в Wealth-Lab | Финансовая лаборатория пишет:

15 мая 2012 в 15:33

[...] [...]

Ответить

Отображение на графике в программе Велс Лаб рядов данных (индикаторов, осцилляторов). | Финансовая лаборатория пишет:

16 мая 2012 в 12:42

[...] На сегодня информации достаточно. В процессе Вы узнали о нескольких методах, которые позволяют Вам достаточно гибко отображать на разных областях графика всевозможные количественные показатели, рассчитанные перед или в процессе выполнения стратегии. Следующий раз мы затронем тему написания текста на графике. [...]

Ответить

Оставить комментарий

Имя (Обязательно)

Почта (Обязательно)

Вэб-сайт

Получать новые комментарии по электронной почте. Вы можете [подписаться](#) без комментирования.

© 2012, Финансовая лаборатория. Все права защищены.
Любое использование материалов допускается только с указанием активной ссылки на источник.



451
242
187



В день мой блог
посещает **77**
человек,
сегодня — **124**



рейтинг
757137



BY FEEDBURNER

Подписка

- [RSS](#)
- [Twitter](#)
- [ВКонтакте](#)
- [facebook](#)

Сайт

- [Создать блог](#)
- [Содержание](#)
- [О блоге](#)
- [Карта](#)

Рисование графических объектов на графике Wealth-Lab

 **АВТОР ДМИТРИЙ ВЛАСОВ**  **МАЯ - 16 - 2012**  **КОММЕНТАРИЕВ: 7**

При построении торговых стратегий может понадобится выполнить некоторые действия, связанные с рисованием графических объектов. К примеру, нарисовать линии поддержки и сопротивления, зоны перекупленности и перепроданности, отметить цели движения цены, волны и тому подобные вещи.

Для этого в библиотеке WealthScript имеется целый набор необходимых для этого методов. Плюс Вы можете самостоятельно запрограммировать те методы, которые будут Вам помогать при работе с графическими объектами.



При рисовании графических объектов сложнее всего найти те точки, которые будут выступать в качестве основы для рисования. Методы PeakBar() и TroughBar() больше всего подходят для этой цели, однако для нахождения координат на графике для рисования различных формаций, треугольников, прямоугольников могут быть использованы любые методы.

Каждый из следующих методов может использоваться на любой области графика. Для более подробного знакомства и изучения примеров с этими методами спотрите справочник QuikRef (F11).

DrawCircle()

Этот метод рисует маленькие кружочки. Подходит для подсветки ключевых точек или целей на графике.

Код примера на C# (кликните чтобы посмотреть) 

```
1 void DrawCircle(ChatPane pane, int radius, int bar, double value, Color color, Color
2 fillColor, LineStyle style, int width, bool behindBars);
3 void DrawCircle(ChatPane pane, int radius, int bar, double value, Color color,
4 LineStyle style, int width, bool behindBars);
5
6 protected void DrawCircle(ChatPane pane, int bar1, double value1, int bar2, double
7 value2, Color color, Color fillColor, LineStyle style, int width, bool behindBars);
8 protected void DrawCircle(ChatPane pane, int bar1, double value1, int bar2, double
9 value2, Color color, LineStyle style, int width, bool behindBars);
```

С помощью метода DrawCircle() можно нарисовать (и при желании раскрасить) круги на выбранной области графика. Причем сделать это можно двумя различными способами. Каждый из этих двух методов имеет параметр fillColor, с помощью которого можно указать - надо ли раскрашивать нарисованные круги. Последний параметр метода DrawCircle(), который обозначается behindBars - определяет - как должен быть нарисован круг - впереди либо сзади баров на графике. Круг будет нарисован с использованием определенных Вами цвета, стиля и толщины.

Первый метод рисует круг с радиусом, который указывается в пикселях. При этом центром этого круга будут координаты (X) и (Y). В качестве первой координаты выступает номер бара, а в качестве второй координаты - число, которое указываете Вы.

Второй метод рисует круг, радиус которого определяется линией, проведенной между двумя точками.

[RSS товары](#)



[RSS новости](#)



[Twitter](#)



Последние комментарии

макс: А где взять свечки с открытым интересом? на сайте финама нет



ring10: Добрый день Дмитрий Как сделать, чтобы открывался 1 велф лаб? Если открываешь под пользователем, т [...]



Дмитрий Власов: Буян, рад, что Вам пригодились материалы данной статьи!



Максим: Сам себе отвечу. protected override void Execute() { Bars rts = GetExternalSymbol("RTS", t [...]



Буян: Спасибо Дмитрий за статью! Мне удалось разработать хорошую рабочую торговую систему, используя возмо [...]



Последние посты



Бесплатный вебинар: 7 смертных грехов при оптимизации торговых систем

НОЯБРЯ - 5 - 2012



Отзывы участников курса до результата Wealth-Lab. С нуля до первой торговой системы за 5

дней.

НОЯБРЯ - 3 - 2012



Быстрее, Выше, Сильнее: Обновление Wealth-Lab до новой версии 6.4

ОКТАБРЯ - 23 - 2012



Курс до результата: Wealth-Lab – от нуля до первой торговой системы за 5 дней

ОКТАБРЯ - 23 - 2012



Обзорная экскурсия по достопримечательностям программы Wealth-Lab

ОКТАБРЯ - 20 - 2012

Берите

Координаты каждой из этих двух точек также определяются координатами (X) и (Y). При этом первая координата каждой из этих двух точек это номер бара, а вторая координата - это число.

Код примера на C# (кликните чтобы посмотреть)

```
1  protected override void Execute()
2  {
3      // Operator overload
4      for(int bar = 200; bar < Bars.Count; bar++)
5      {
6          // Circle any 200 day High
7          if ( High[bar] == Highest.Series( High, 200 )[bar] )
8              DrawCircle( PricePane, 4, bar, High[bar], Color.Green, Color.DarkGreen,
9  WealthLab.LineStyle.Solid, 1, true );
10         // Circle any 200 day Low
11         if ( Low[bar] == Lowest.Series( Low, 200 )[bar] )
12             DrawCircle( PricePane, bar-1, Low[bar-1], bar, Low[bar-1], Color.Red,
13  Color.DarkRed, WealthLab.LineStyle.Solid, 1, false );
14     }
15 }
```

DrawEllipse()

Для рисования эллипса Вы должны всего лишь определить противоположный угол воображаемого прямоугольника, который устанавливает границы эллипса.

DrawLine()

Этот метод рисует линию, которая соединяет две точки. Линия может быть продлена с тем же уклоном, при нахождении значения координаты Y на следующем баре с помощью метода LineExtendY() или LineExtendYLog() и передавать значение следующей линии, при следующем вызове метода DrawLine().

DrawHorzLine()

Это подходящий метод для того, чтобы нарисовать линии поддержки и сопротивления на фиксированном уровне через весь график.

DrawPolygon()

Этот метод может использоваться для рисования многоугольника, состоящего из n-сторон.

Как: нарисовать линию тренда и продлить ее вправо по шкале времени.

Чтобы нарисовать расширяющуюся линию тренда нужно проделать следующие шаги:

- Первым делом нарисуйте линию между двумя известными точками, использовав для этого метод DrawLine().
- Следующим шагом найдите значение координаты Y на том баре, до которого Вы хотите расширить линию тренда.
- Используйте найденную точку, подставив ее в качестве второго аргумента в метод DrawLine().

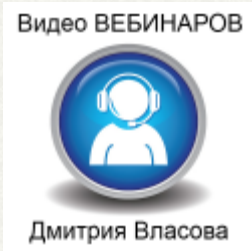
Мы можем упростить эту процедуру, создав специальный метод DrawExtended TrendLine().

Именно создание и применение метода DrawExtended TrendLine() приводится в следующем примере.

Пример (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть)

```
1  protected override void Execute()
2  {
3      // Находим две последник пятипроцентных вершины и чертим линию до самого конца
4      графика.
5
6      int bar = Bars.Count - 1;
7      int pb2 = (int)PeakBar.Value(bar, High, 5,
8  WealthLab.Indicators.PeakTroughMode.Percent);
9      int pb1 = pb2;
10
11      if (pb2 > -1)
12
13          pb1 = (int)PeakBar.Value(pb2, High, 5,
14  WealthLab.Indicators.PeakTroughMode.Percent);
15
16      if ((pb2 == -1) || (pb1 == -1))
17
18          DrawLabel(PricePane, "Не могу найти две 5% вершины", Color.Red);
19
20      else
21          DrawExtendedTrendline(PricePane, pb1, High[pb1], pb2, High[pb2], bar,
22  Color.Blue); //применяем созданный нами метод
23  }
24
25  //Создаем специальный метод, который рисует тренд и расширяет его до указанного бара.
26  public void DrawExtendedTrendline(ChartPane pane, int bar1, double val1, int bar2,
27  double val2, int bar3, Color color)
28  {
29      {
30          // рисуем линию для двух известных нам точек на барах №1 и №2
31
32          DrawLine(pane, bar1, val1, bar2, val2, Color.Blue, WealthLab.LineStyle.Solid, 2);
33
34          // расширяем линию до указанного бара 3
35
36          double val3 = LineExtendY(bar1, val1, bar2, val2, bar3); //Ищем значение
37  координаты Y над баром 3, используя имеющуюся линию
38
39          if (pane.LogScale)
40
41              val3 = LineExtendYLog(bar1, val1, bar2, val2, bar3);
42
43          DrawLine(pane, bar2, val2, bar3, val3, color, WealthLab.LineStyle.Solid,
```



МТС под микроскопом: WLMA System

Подробный разбор торговой системы WLMA System
3900 руб.

В корзину

Рубрики

Выберите рубрику

Обновления:

- Новые посты по Email
- Новые посты по RSS
- Комментарии по RSS
- Следим на Twitter

Метки

C# HowTo: **Wealth-Lab**
Видео Индикаторы МТС Торговые
роботы Юмор алготрейдинг
мероприятия обучение

Управление

Регистрация

Вход

RSS Записей

RSS Комментарии

WordPress.org

```
1); // снова рисуем линию для двух известных нам теперь точек: №2 и №3
}
```

После выполнения данного кода получается такой график:



Как видите, на графике над двумя последними вершинами нарисовалась линия и эта линия расширилась до самого последнего бара графика.

Как: начертить на графике в программе Велс Лаб скоростные линии сопротивления (1/3 - 2/3 линии)

Практически все графические объекты, которые Вы можете начертить вручную могут быть автоматически начерчены на графике с помощью программного кода. При этом исчезает субъективизм подхода. Все будет выполнено согласно строгим наборам правил, которые Вы заранее программируете.

Вот один из примеров использования такого подхода - начертим скоростные линии сопротивления.

Пример (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть)

```
1 using System;
2 using System.Collections.Generic;
3 using System.Text;
4 using System.Drawing;
5 using WealthLab;
6 using WealthLab.Indicators;
7
8 namespace WealthLab.Strategies
9 {
10     public class SpeedResistanceDemo : WealthScript
11     {
12         //создаем специальный методы для черчения скоростных линий сопротивления...
13         public void SpeedResistance(int bar1, double price1, int bar2, double
14 price2)
15         {
16             //
17             Вы можете расширить линию с помощью этой процедуры и используя методы
18             LineExtendY() или LineExtendYLog() при построении на графике semi-log
19
20             double delta = price1 - price2;
21             DrawLine(PricePane, bar1, price1, bar2, price2, Color.Red,
22 WealthLab.LineStyle.Solid, 2);
23             DrawLine(PricePane, bar1, price1, bar2, price2 + delta / 3, Color.Blue,
24 WealthLab.LineStyle.Dots, 1);
25             DrawLine(PricePane, bar1, price1, bar2, price2 + delta * 2 / 3,
26 Color.Blue, WealthLab.LineStyle.Dots, 1);
27         }
28         protected override void Execute()
29         {
30             // Находим последние 8% вершину и дно и чертим скоростные линии
31             сопротивления через эти точки.
32
33             int bar = Bars.Count - 1;
34             int b1 = (int)PeakBar.Value(bar, Close, 8.0,
35 WealthLab.Indicators.PeakTroughMode.Percent);
36             int b2 = (int)TroughBar.Value(bar, Close, 8.0,
37 WealthLab.Indicators.PeakTroughMode.Percent);
38             if (b2 > b1)
39                 SpeedResistance(b1, Close[b1], b2, Close[b2]);
40             else
41                 SpeedResistance(b2, Close[b2], b1, Close[b1]);
42         }
43     }
44 }
```

В результате получается вот такой график скоростных линий сопротивления:



Как: выделить зону на графике в виде многоугольника

И снова используя подходящие вершины и впадины мы можем определить 4 (или более) углов для того, чтобы начертить на графике многоугольник.

Пример (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть) ?

```
1  protected override void Execute()
2  {
3      // Находим две последних пятипроцентных вершины и чертим линию до самого конца
4      графика.
5
6      int bar = Bars.Count - 1;
7      int pb2 = (int)PeakBar.Value(bar, High, 5,
8      WealthLab.Indicators.PeakTroughMode.Percent);
9      int pb1 = pb2;
10
11     if (pb2 > -1)
12
13         pb1 = (int)PeakBar.Value(pb2, High, 5,
14         WealthLab.Indicators.PeakTroughMode.Percent);
15
16     if ((pb2 == -1) || (pb1 == -1)) {
17
18         DrawLabel(PricePane, "Не могу найти две 5% вершины", Color.Red);
19         return;
20     }
21
22     int tb2 = (int)TroughBar.Value(bar, Low, 5,
23     WealthLab.Indicators.PeakTroughMode.Percent);
24     int tb1 = tb2;
25
26     if (tb2 > -1)
27
28         tb1 = (int)TroughBar.Value(tb2, Low, 5,
29         WealthLab.Indicators.PeakTroughMode.Percent);
30
31     if ((tb2 == -1) || (tb1 == -1)){
32
33         DrawLabel(PricePane, "Не могу найти две 5% впадины", Color.Red);
34         return;
35     }
36 }
37
38 //рисуем многоугольник по 4-м найденным точкам:
39
40 DrawPolygon(PricePane, Color.LightBlue, Color.LightBlue,
41 WealthLab.LineStyle.Invisible, 1, true,
42
43     pb1, High[pb1], pb2, High[pb2], tb2, Low[tb2], tb1, Low[tb1]);
44 }
```

Вот что получится в итоге:



Как в Велс Лаб нарисовать многоугольник

Как видите, используя 4 найденных точки мы смогли нарисовать многоугольник. Точно так же можно поступать и для того, чтобы нарисовать многоугольник с большим количеством углов.

Сегодня мы увидели, как используя существующие методы WealthScript и создавая собственные методы можно чертить на графике в программе Wealth-Lab различные графические объекты. В следующий раз перейдем уже к более интересной и близкой к построению стратегий теме - **индикаторы в программе Wealth-Lab**. Чтобы не пропустить новые статьи - не забывайте подписываться по RSS на обновления нашего блога.

Если Вам понравилась эта статья - подпишитесь на **новые СТАТЬИ по RSS** или получайте наши **новые СТАТЬИ на E-mail**:

Рекомендую почитать также:

- Построение графиков в программе Wealth-Lab при помощи WealthScript
- Отображение на графике синтетических финансовых инструментов
- Отображение на графике в программе Велс Лаб рядов данных (индикаторов, осцилляторов).
- Написание текста на графиках Wealth-Lab программными методами
- Создание параметров торговой стратегии в Wealth-Lab

🔖 РУБРИКИ: PAINTING THE CHART, WEALTHSCRIPT

Комментариев: 7

Ответы на вопросы по программированию торговых стратегий в Wealth-Lab | Финансовая лаборатория пишет:

16 мая 2012 в 12:58

[...] [...]

Ответить

Инструкция по программированию торговых стратегий в WealthLab | Финансовая лаборатория пишет:
20 июня 2012 в 12:20

[...] 5. Рисование графических объектов на графике Wealth-Lab [...]

Ответить

Дмитрий пишет:
24 июля 2012 в 13:46



Здравствуйте. А возможно ли изменение/удаление уже нарисованных на графике объектов?
Спасибо.

Ответить

Дмитрий Власов пишет:
24 июля 2012 в 20:49



Вы имеете изменение или удаление нарисованных вручную объектов с помощью кода? Нет, так сделать нельзя. А если изменять или удалять объекты, которые написаны с помощью кода – тогда без проблем.

Ответить

Дмитрий пишет:
25 июля 2012 в 12:50



Да, мне хочется с помощью кода изменять и удалять объекты, нарисованные с помощью кода. В QuickRef не нахожу упоминаний о такой возможности. 😞

Ответить

Юрий пишет:
26 октября 2012 в 19:56



Несколько корректировок к последнему коду.
1)В первом комментарии убрать ” и чертим линию до самого конца графика”, т.к. никакая линия здесь не чертится.
2)По аналогии с первым комментарием вставить вторым комментарием “// Находим две последние пятипроцентные впадины.”
3)В операторе tb1 = (int)TroughBar.Value(pb2, Low, 5, WealthLab.Indicators.PeakTroughMode.Percent);
вместо pb2 надо поставить tb2. Тогда вместо треугольника будет рисоваться необходимый 4-х угольник.
Я об этих исправлениях написал на форум Wealth-Lab (Wealth-Lab 101). Они с ними согласились.

Ответить

Дмитрий Власов пишет:
27 октября 2012 в 11:47



Спасибо, Юрий, за корректировки.

Ответить

Оставить комментарий

Имя (Обязательно)

Почта (Обязательно)

Вэб-сайт

Получать новые комментарии по электронной почте. Вы можете [подписаться](#) без комментирования.

© 2012, Финансовая лаборатория. Все права защищены.
Любое использование материалов допускается только с указанием активной ссылки на источник.



451
242
187



В день мой блог
посещает **77**
человек,
сегодня — **124**



рейтинг
757140525
195



BY FEEDBURNER

Подписка

- [RSS](#)
- [Twitter](#)
- [ВКонтакте](#)
- [facebook](#)

Сайт

- [Создать блог](#)
- [Содержание](#)
- [О блоге](#)
- [Карта](#)

Проектирование и построение индикаторов в Wealth-Lab

АВТОР ДМИТРИЙ ВЛАСОВ МАЯ - 17 - 2012 КОММЕНТАРИЕВ: 2

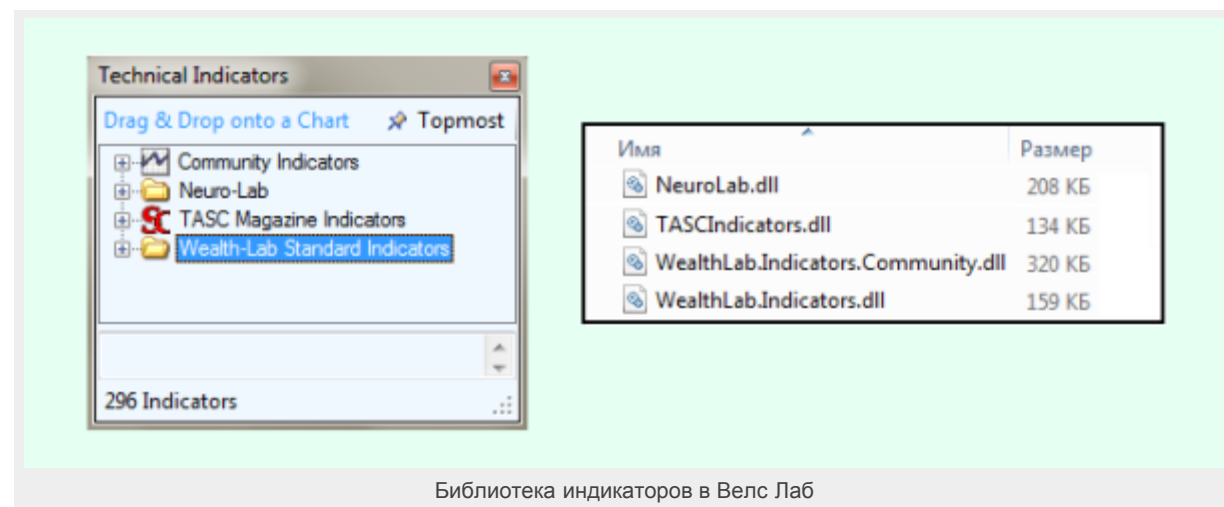
Начиная строить торговую стратегию большинство системных трейдеров сталкиваются с необходимостью использования индикаторов. Индикаторы могут использоваться как при конструировании входа в позицию, так и в качестве фильтров и при сопровождении позиции. Существует огромное множество индикаторов - начиная от стандартных, и заканчивая специфическими, существующие в одном экземпляре и хранимые как зеница ока.

К счастью, у программы WealthLab существует широкий арсенал для работы с индикаторами. Вам не придется заново изобретать велосипед и думать - как запрограммировать RSI, скользящую среднюю или стохастик. За Вас эту работу уже сделали предыдущие поколения трейдеров. И уже готовые индикаторы упаковали в специальные библиотеки. Есть как библиотеки, доступные всем пользователям по умолчанию, так и библиотеки, к которым доступ имеют только официальные, имеющие лицензию пользователи.



Все технические индикаторы в программе Wealth-Lab, доступные в диалоговом окошке, которое открывается при нажатии на кнопку "Indicators" либо из контекстного меню графика "Plot an Indicator on the Chart..." интегрированы в специальных библиотеках индикаторов, которые являются компонентами .NET.

Сразу же после запуска программа Wealth-Lab обнаруживает те библиотеки индикаторов, которые находятся в главной папке, где установлена программа и тут же отображает их в окне доступных индикаторов.



Библиотека индикаторов в Велс Лаб

Для получения более подробной информации смотрите раздел API.

Синтаксис индикаторов

Как и для всех прочих классов, возможность выбора индикаторов и просмотр их синтаксиса интегрирована в редакторе стратегий. Просто нажмите одновременно клавиши Ctrl + Space для того, чтобы открыть контекстное меню завершения кода. Напечатайте несколько букв для того, чтобы внизу отобразился результат поиска. После выбора того, что Вам необходимо, нажмите клавишу Tab или Enter чтобы завершить

RSS товары



RSS новости



Twitter



Последние комментарии

макс: А где взять свечки с открытым интересом? на сайте финама нет



ring10: Добрый день Дмитрий Как сделать, чтобы открывался 1 велф лаб? Если открываешь под пользователем, т [...]



Дмитрий Власов: Буян, рад, что Вам пригодились материалы данной статьи!



Максим: Сам себе отвечу. protected override void Execute() { Bars rts = GetExternalSymbol("RTS", t [...]



Буян: Спасибо Дмитрий за статью! Мне удалось разработать хорошую рабочую торговую систему, используя возмо [...]



Последние посты



Бесплатный вебинар: 7 смертных грехов при оптимизации торговых систем

НОЯБРЯ - 5 - 2012



Отзывы участников курса до результата Wealth-Lab. С нуля до первой торговой системы за 5 дней.

НОЯБРЯ - 3 - 2012



Быстрее, Выше, Сильнее: Обновление Wealth-Lab до новой версии 6.4

ОКТАБРЯ - 23 - 2012



Курс до результата: Wealth-Lab – от нуля до первой торговой системы за 5 дней

ОКТАБРЯ - 23 - 2012

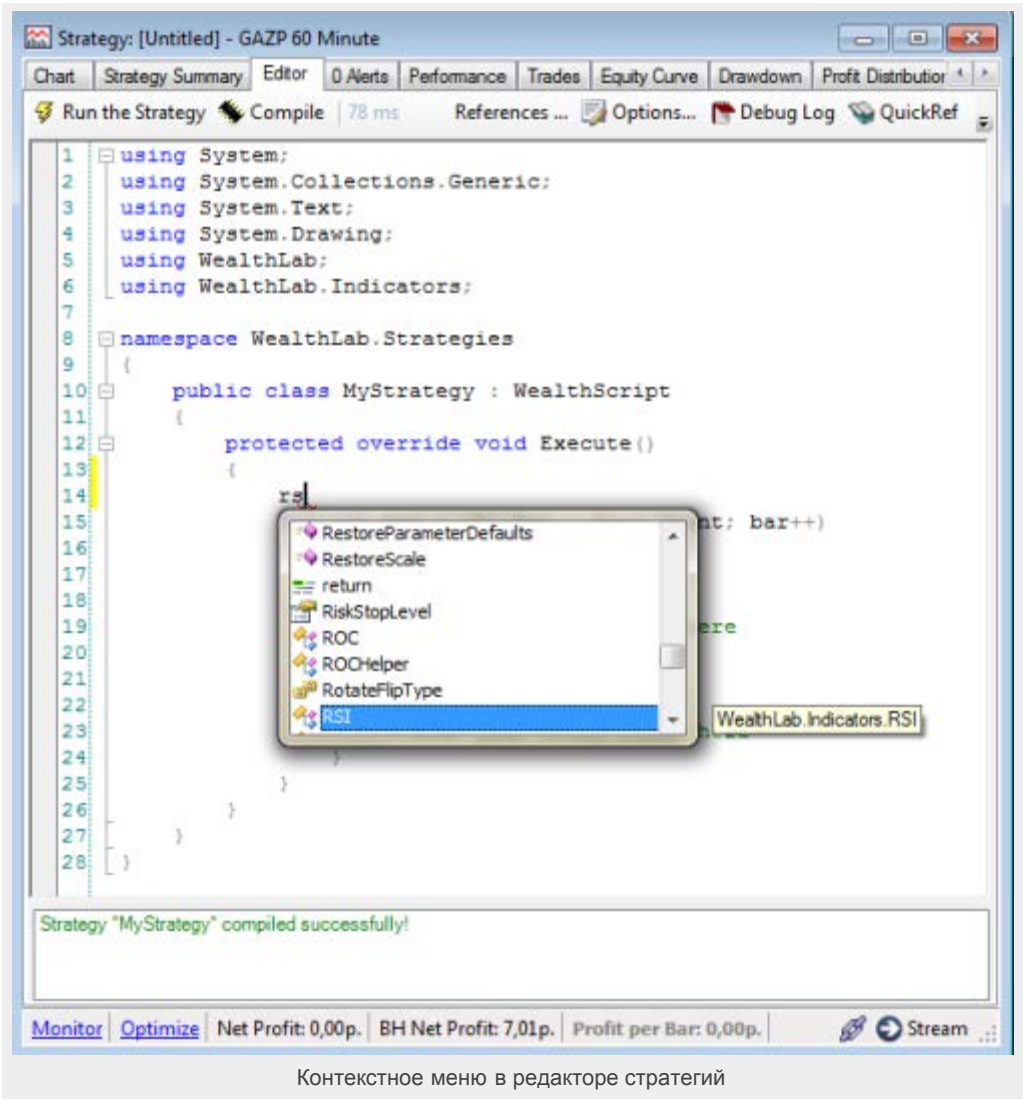


Обзорная экскурсия по достопримечательностям программы Wealth-Lab

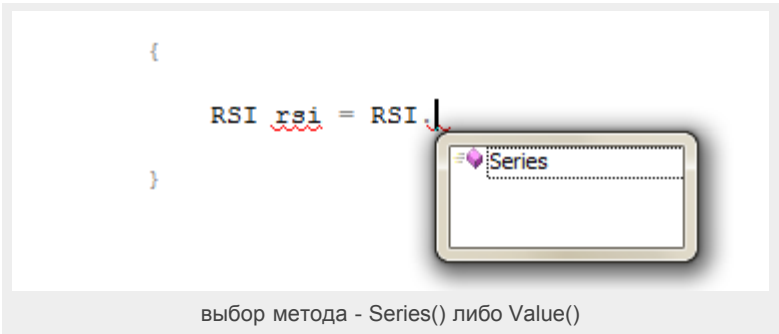
ОКТАБРЯ - 20 - 2012

Берите

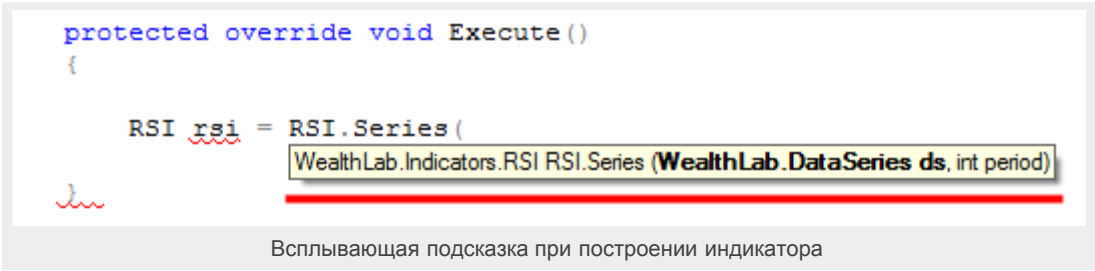
задуманное.



Как и во всех прочих случаях, сразу же после набора разделителя разрядов (точки), происходит автоматическое отображение свойств и методов данного класса. Как Вы вскоре узнаете, все индикаторы имеют метод Series(), а некоторые из индикаторов имеют также и метод Value(). Если по каким-либо причинам список свойств и методов не отобразится автоматически - расположите курсор сразу после разделителя разрядов (точки) и одновременно нажмите две кнопки: Ctrl + Space. В результате список свойств и методов класса появится снова.

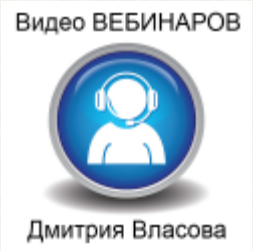
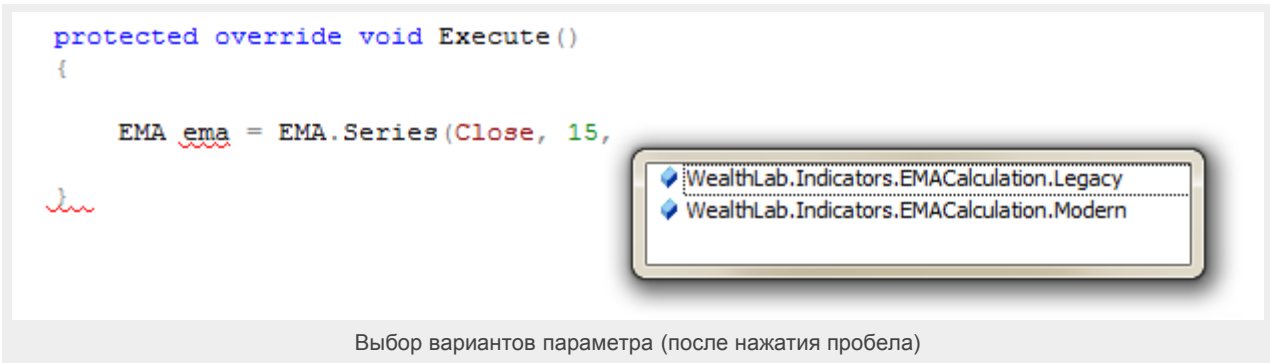


После того, как Вы сделаете выбор из списка параметров и введете открывающую круглую скобку, перед Вами отобразится полный синтаксис индикатора. Если вдруг всплывающая подсказка, информирующая Вас о синтаксисе индикатора исчезнет (это случается, к примеру, при передвижении курсора в сторону от списка параметров), верните курсор на прежнее место - сразу после открывающей круглой скобки и нажмите одновременно три клавиши: Ctrl + Shift + Space для того, чтобы снова увидеть всплывающую подсказку.



Совет:

Возьмите себе за правило обязательно нажимать на пробел после каждой запятой в списке параметров. Это приведет к тому, что у тех параметров, которые являются перечислениями, автоматически появятся варианты, которые Вы сможете выбрать. Иллюстрацию этого можно увидеть на рисунке ниже.



МТС под микроскопом:

WLMA System

Подробный разбор торговой системы WLMA System

3900 руб.

[В корзину](#)

Рубрики

Выберите рубрику

Обновления:

- Новые посты по Email
- Новые посты по RSS
- Комментарии по RSS
- Следим на Twitter

Метки

C# HowTo: **Wealth-Lab**
Видео Индикаторы МТС Торговые роботы Юмор алготрейдинг мероприятия обучение

Управление

Регистрация

Вход

RSS Записей

RSS Комментариев

WordPress.org

Метод Series() - для получения значений индикаторов для всех баров

Выполняя статический метод с названием Series() класс индикаторов возвращает конкретный экземпляр индикатора типа DataSeries. Этот полученный экземпляр типа DataSeries может быть как новым экземпляром, так и экземпляром, который уже существовал ранее и был найден в Кеше. Механизм кеширования сохраняет компьютерные ресурсы следя за тем, чтобы в памяти не создавалось множество идентичных копий одного и того же индикатора.

Как: получить доступ к полному ряду данных индикатора

Для того, чтобы получить ряд данных, созданный индикатором, можно воспользоваться двумя различными методами. Каждый из этих методов можно применять.

Метод 1:

Используйте оператор new для того, чтобы получить экземпляр индикатора типа DataSeries. Помимо того, что такой метод будет казаться "более профессиональным", Вы сможете прямо при вызове этого метода сделать более удобное описание названия получившегося индикатора.

Метод 2:

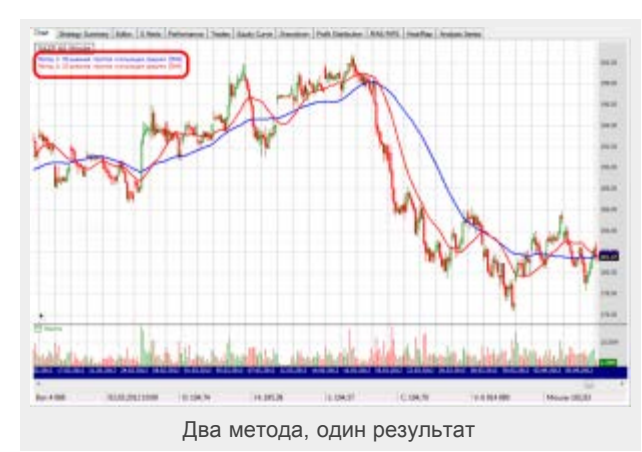
Пользователям, которые стали использовать Wealth-Lab еще до выхода .NET версии, может показаться более комфортным статический метод Series, для того чтобы получить экземпляры индикатора типа DataSeries.

Пример (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть) ?

```
1 | protected override void Execute()
2 | {
3 |
4 |     /* Сопоставление двух методов создания индикаторов */
5 |
6 |     // Метод 1: использование оператора new
7 |
8 |     SMA sma = new SMA(Close, 50, "Метод 1: 50-дневная простая скользящая средняя
9 | (SMA)");
10 | PlotSeries(PricePane, sma, Color.Blue, WealthLab.LineStyle.Solid, 2);
11 |
12 |     // Method 2: использование статистического метода Series()
13 |
14 |     DataSeries sma2 = SMA.Series(Close, 20);
15 |     sma2.Description = "Метод 2: 20-дневная простая скользящая средняя (SMA)";
16 |     PlotSeries(PricePane, sma2, Color.Red, WealthLab.LineStyle.Solid, 2);
17 | }
```

Синяя скользящая средняя сделана при помощи метода 1, а красная скользящая средняя - при помощи метода 2.



Как видите, несмотря на чуть разный синтаксис каждый из этих методов приводит к одному и тому же результату. Вы можете выбрать именно тот метод, который Вам больше нравится.

Статические метод Value() для расчет значения индикатора только для одного бара

Метод Value() - это необязательный для индикаторов метод. Статический метод Value() дает возможность посчитать и возратить значение индикатора "на лету" для конкретного номера бара. При этом нет необходимости полностью создавать экземпляр данного класса. Такой способ соответствует режиму SingleCalcMode, который был доступен для версии Wealth-Lab, которая предшествовала версии .NET. Поскольку здесь не происходит процесса кеширования, метод Value() может пересчитывать индикатор, значение параметров которого могут меняться.

Если индикатор поддерживает метод Value(), то все параметры этого индикатора будут аналогичны параметрам из метода Series() за одним исключением. Здесь появляется новый дополнительный параметр типа int, для указания на номер конкретного бара. Результатом такого метода является число с типом double.

Как: рассчитать значение простого индикатора несколькими способами

Давайте рассмотрим примеры использования обоих методов и метода Series() и метода Value(). Предположим, что Вы хотите определить торгуется ли в настоящее время финансовый инструмент выше своей 50-дневной скользящей средней.

Пример (как выполнить пример приведенного кода)...

Метод Series()

Код примера на C# (кликните чтобы посмотреть) ?

```
1 | //Метод Series()
2 | protected override void Execute()
```

```
3 | {
4 |     int bar = Bars.Count - 1;
5 |     DataSeries sma = SMA.Series(Close, 50);
6 |     if (Close[bar] > sma[bar])
7 |
8 |         DrawLabel(PricePane, "Above 50-day SMA", Color.Blue);
9 |
10 |    else
11 |
12 |        DrawLabel(PricePane, "Below 50-day SMA", Color.Red);
13 | }
```

Метод Value()

Код примера на C# (кликните чтобы посмотреть)



```
1 | //Метод Value()
2 | protected override void Execute()
3 | {
4 |
5 |     int bar = Bars.Count - 1;
6 |     double smaVal = SMA.Value(bar, Close, 50);
7 |
8 |     if (Close[bar] > smaVal)
9 |
10 |        DrawLabel(PricePane, "Above 50-day SMA", Color.Blue);
11 |    else
12 |
13 |        DrawLabel(PricePane, "Below 50-day SMA", Color.Red);
14 | }
```

Как Вы видите, оба метода приводят к совершенно одинаковым результатам. Если мы вспомним, что метод Series() возвращает данные в виде целого ряда значений (тип DataSeries), т.е. индикатор рассчитывается для каждого бара. Поскольку в данном примере нам нужно только одно значение скользящей средней (только на последнем баре), можно прийти к выводу, что в данном конкретном случае нет необходимости нагружать процессор и память, применяя метод Series(). Будет вполне достаточно определить значение скользящей средней на последнем баре с помощью метода Value().

Сегодня мы начали рассматривать обширную тему "Индикаторы в Wealth-Lab". В следующий раз посмотрим, **как можно создавать собственные индикаторы** и убедимся, что в этом нет ничего сложного. Не забывайте подписываться на новые статьи по RSS.

Если Вам понравилась эта статья - подпишитесь на **новые СТАТЬИ по RSS** или получайте наши **новые СТАТЬИ на E-mail**:

Рекомендую почитать также:

- Делаем индикаторы в Wealth-Lab своими руками
- Стабильность индикаторов в программе Wealth-Lab в зависимости от ширины тестируемого окна.
- Программирование торговых стратегий в программе Wealth-Lab
- Ловушки при построении торговых систем, связанные с проблемой подглядывания в будущее
- Видео вебинара по программе Wealth-Lab в рамках курса "Wealth-Lab: C# на службе трейдера"

 РУБРИКИ: [INDICATORS](#), [WEALTHSCRIPT](#)

Комментариев: 2

Рисование графических объектов на графике Wealth-Lab | Финансовая лаборатория пишет:

17 мая 2012 в 13:33

[...] Проектирование и построение индикаторов в Wealth-Lab Главная Написание текста на графиках Wealth-Lab [...]

Ответить

Ответы на вопросы по программированию торговых стратегий в Wealth-Lab | Финансовая лаборатория пишет:

17 мая 2012 в 13:52

[...] [...]

Ответить

Оставить комментарий

Имя (Обязательно)

Почта (Обязательно)

Вэб-сайт

Получать новые комментарии по электронной почте. Вы можете [подписаться](#) без комментирования.

© 2012, Финансовая лаборатория. Все права защищены.
Любое использование материалов допускается только с указанием активной ссылки на источник.



451
242
187



В день мой блог
посещает **77**
человек,
сегодня — **124**



mail.ru РЕЙТИНГ
757141526
195



Подписка

- [RSS](#)
- [Twitter](#)
- [ВКонтакте](#)
- [facebook](#)

Сайт

- [Создать блог](#)
- [Содержание](#)
- [О блоге](#)
- [Карта](#)

Делаем индикаторы в Wealth-Lab своими руками

 **АВТОР ДМИТРИЙ ВЛАСОВ**  **МАЯ - 18 - 2012**  **КОММЕНТАРИЕВ: 5**

В программе Wealth-Lab версии 6 существуют только официальные скомпилированные библиотеки индикаторов. Каждая такая библиотека индикаторов соответствует спецификации для библиотеки индикаторов. Конечно же, Вы можете создать Вашу собственную коллекцию тех индикаторов, которыми пользуетесь чаще всего.

Однако, во многих случаях, нет никакого смысла создавать специальный индикатор и включать его в отдельную библиотеку. Это касается тех индикаторов, которые можно создать буквально "на лету" написав несколько строчек кода и применив простые, легкозапоминающиеся математические операции.



Именно по этой причине мы называем самодельными индикаторами ряды данных (DataSeries), которые создаются "на лету", а не те индикаторы, которые были сделаны Вами самостоятельно заранее и хранятся в скомпилированной библиотеке. Одним из таких "самодельных индикаторов" является **типичная цена (typical price)**, которую мы уже рассматривали.

Как: создать "полуофициальный (semi-formal)" индикатор своими руками

Полуофициальный самодельный индикатор не существует в библиотеке индикаторов "в чистом виде", однако его можно "извлечь" из класса DataSeries точно таким способом, что и полностью интегрированный (библиотечный) индикатор.

Вы можете использовать полуофициальный метод, как это показано в следующем примере, если просто хотите создать пустой ряд данных (DataSeries) и позднее заполнить его элементами в виде чисел. Альтернативой к такому подходу может стать создание ряда данных, инициализированного нулевыми значениями путем вычитанием ряда данных самого из себя.

Оба этих подхода демонстрируются ниже.

Пример (**как выполнить пример приведенного кода**)...

Код примера на C# (кликните чтобы посмотреть)

```
1  /* Моделирование индикатора Sum.Series */
2  using System;
3  using System.Collections.Generic;
4  using System.Text;
5  using System.Drawing;
6  using WealthLab;
7  using WealthLab.Indicators;
8
9  namespace WealthLab.Strategies
10 {
11
12     //создаем "полуофициальный" индикатор (наследник класса DataSeries)
13     public class SumTest : DataSeries
14     {
15
16         public SumTest(DataSeries ds, int period)
17             : base(ds, "Мой ряд данных: Sum.Series")
```

RSS товары



RSS новости



Twitter



Последние комментарии

макс: А где взять свечки с открытым интересом? на сайте финама нет



ring10: Добрый день Дмитрий Как сделать, чтобы открывался 1 велф лаб? Если открываешь под пользователем, т [...]



Дмитрий Власов: Буян, рад, что Вам пригодились материалы данной статьи!



Максим: Сам себе отвечу. protected override void Execute() { Bars rts = GetExternalSymbol("RTS", t [...]



Буян: Спасибо Дмитрий за статью! Мне удалось разработать хорошую рабочую торговую систему, используя возмо [...]



Последние посты



Бесплатный вебинар: 7 смертных грехов при оптимизации торговых систем

НОЯБРЯ - 5 - 2012



Отзывы участников курса до результата Wealth-Lab. С нуля до первой торговой системы за 5

дней.

НОЯБРЯ - 3 - 2012



Быстрее, Выше, Сильнее: Обновление Wealth-Lab до новой версии 6.4

ОКТАБРЯ - 23 - 2012



Курс до результата: Wealth-Lab – от нуля до первой торговой системы за 5 дней

ОКТАБРЯ - 23 - 2012



Обзорная экскурсия по достопримечательностям программы Wealth-Lab

ОКТАБРЯ - 20 - 2012

Берите

```
18         {
19             double sum = 0;
20
21             for (int bar = 0; bar < period; bar++)
22                 sum += ds[bar];
23
24             this[period - 1] = sum;          // Сумма для инициализации периода
25
26             for (int bar = period; bar < ds.Count; bar++)
27                 this[bar] = this[bar - 1] - ds[bar - period] + ds[bar];
28         }
29     }
30
31     public static SumTest Series(DataSeries ds, int period)
32     {
33         SumTest _SumTest = new SumTest(ds, period);
34
35         return _SumTest;
36     }
37 }
38
39 public class MyStrategy : WealthScript
40 {
41     protected override void Execute()
42     {
43
44         int period = 20;
45
46         DataSet sum1 = SumTest.Series(Close, period); // создаем и отображаем
47 на графике полуофициальный индикатор Sum.Series
48
49         ChartPane cp = CreatePane(40, true, true);
50
51         PlotSeries(cp, sum1, Color.Blue, LineStyle.Solid, 1);
52
53         // создаем тот же самый ряд данных "на лету", начиная с ряда данных,
54 наполненного нулями
55
56         DataSet sum2 = Close - Close;
57
58         sum2.Description = "Мой ряд данных: Sum (создан на лету)";
59
60         double sum = 0d;
61
62         for (int bar = 0; bar < period; bar++)
63             sum += Close[bar];
64
65         sum2[period - 1] = sum;          // Сумма для инициализации периода
66
67         for (int bar = period; bar < Bars.Count; bar++)
68             sum2[bar] = sum2[bar - 1] - Close[bar - period] + Close[bar];
69
70         // отображаем на графике в виде гистограммы на той же самой области
71 графика (для сравнения).
72
73         PlotSeries(cp, sum2, Color.Black, LineStyle.Histogram, 1);
74     }
75 }
76
77 }
```

Выполнив этот код мы получим такой график:



Как: создать индикатор MACD с самостоятельно определяемым периодом

Классический индикатор MACD, представленный в программе Wealth-Lab, базируется на разнице между двумя экспоненциальными скользящими средними (EMA) с разными экспонентами. (0,075 и 0,15). Если использовать 26-периодичную EMA с экспонентой 0,074074, 12-периодичную EMA с экспонентой 0,153846, Вы можете сдвинуть значение MACD по сравнению с использованием классического индикатора (можно также взять любую другую комбинацию периодов).

Пример (как выполнить пример приведенного кода)...

```
Код примера на C# (кликните чтобы посмотреть)
1 //
2 // Замечание: Метод калькуляции modern для расчета экспоненциальной
3 скользящей средней (EMA) предполагает, что экспонента равна 2/(1 +
4 Period)
5
6 protected override void Execute()
7 {
8     // Создаем самодельный индикатор MACD
9
10    DataSet macdX = EMA.Series(Close, 12, EMACalculation.Modern) -
11    EMA.Series(Close, 26, EMACalculation.Modern);
12
13 }
```



Видео ВЕБИНАРОВ



Дмитрия Власова



МТС под микроскопом: WLMA System

Подробный разбор торговой
системы WLMA System

3900 руб.

В корзину

Рубрики

Выберите рубрику

Обновления:

- Новые посты по Email
- Новые посты по RSS
- Комментарии по RSS
- Следим на Twitter

Метки

C# HowTo: Wealth-Lab
Видео Индикаторы МТС Торговые
роботы Юмор алготрейдинг
мероприятия обучение

Управление

Регистрация

Вход

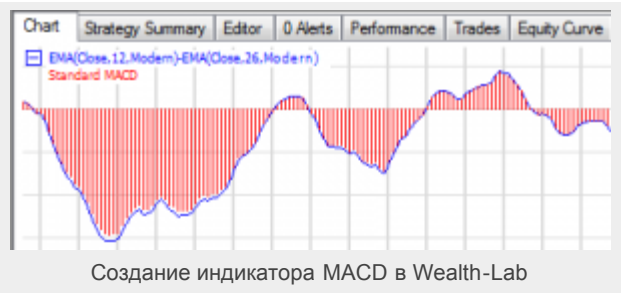
RSS Записей

RSS Комментариев

WordPress.org

```
14 // Отображаем на графике самодельный индикатор MACD и сравниваем его со
15 стандартным (библиотечным) индикатором MACD
16
17 ChartPane macdPane = CreatePane(40, true, true);
18
19 MACD standardMACD = new MACD(Close, "Standard MACD");
20
21 PlotSeries(macdPane, macdX, Color.Blue, LineStyle.Solid, 1);
22
23 PlotSeries(macdPane, standardMACD, Color.Red, LineStyle.Histogram, 1);
24 }
```

Получившийся график MACD демонстрирует нам, что оба подхода в создании индикатора - совершенно однозначны и приводят к одинаковым результатам.



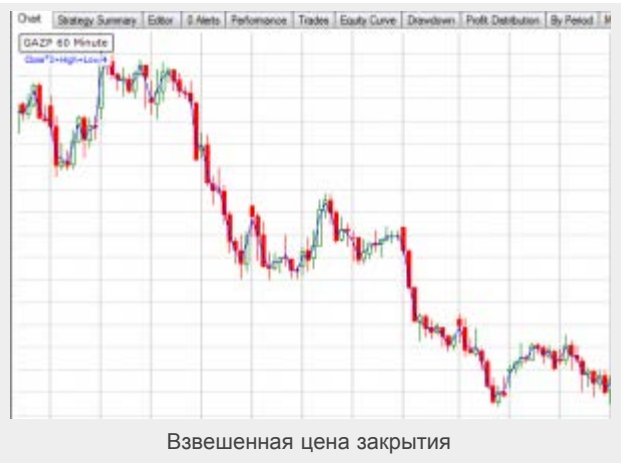
Как: создать индикатор, отражающий взвешенную цену закрытия

Взвешенная цена закрытия сходна с типичной ценой, но отличается тем, что здесь дважды учитывается цена закрытия бара.

Пример (как выполнить пример приведенного кода)...

```
Код примера на C# (кликните чтобы посмотреть)
1 protected override void Execute()
2 {
3     //
4     //
5     //
6 }
```

Получившийся график:



Как: создать индикатор СПРЕДА, отражающий разницу между двумя финансовыми инструментами

Спредом является разница между ценами двух различных финансовых инструментов.

Пример (как выполнить пример приведенного кода)...

Обратите внимание на использование метода GetExternalSymbol() для доступа к барам второго финансового инструмента, который применяется для того, чтобы сразу же получить цену закрытия этого второго инструмента.

```
Код примера на C# (кликните чтобы посмотреть)
1 protected override void Execute()
2 {
3     //
4     //
5     //
6 }
```

Верхняя область графика - это отображение графика Спреда между Роснефтью и Газпромом.



Как: создать осциллятор силы Элдера (EFO)

Доктор Элдер, входя в комнату трейдеров, говорит, что если акция находится в восходящем тренде и 2-х дневная экспоненциальная скользящая средняя построенная по осцилятору силы Элдера (EFO) опускается

ниже нуля, то это является сигналом на покупку. Аналогичным образом, сигнал на продажу появляется тогда, когда акция в нисходящем тренде, а 2-х дневная EMA, построенная по EFO пересекает нулевой уровень снизу вверх.

Пример (как выполнить пример приведенного кода)...

Обратите внимание на то, как используется оператор смещения (>>) для того, чтобы сдвинуть цены закрытия вправо по шкале времени. Именно это позволяет создать ряд данных (DataSeries), в котором отображается разница между сегодняшним и вчерашним закрытием. Эта разница умножается на объем - образуя в результате другой индикатор, создаваемый с помощью одной строчки кода.

Код примера на C# (кликните чтобы посмотреть)

```
1 | protected override void Execute()
2 | {
3 |     // создаем ряд данных, в котором отображается осциллятор силы Элдера
4 |     DataSeries efo = (Close - (Close >> 1)) * Volume;
5 |     efo.Description = "Elder Force";
6 |
7 |     DataSeries efEma = EMA.Series(efo, 2, EMACalculation.Modern);
8 |     ChartPane efPane = CreatePane(40, true, true);
9 |     PlotSeries(efPane, efo, Color.Black, LineStyle.Histogram, 1);
10 |    PlotSeries(efPane, efEma, Color.Green, LineStyle.Solid, 2);
11 | }
```

Верхняя область графика отображает Осциллятор силы Элдера (EFO).



Как: самостоятельно создать NRTR_WATR индикатор Константина Копыркина

Русский трейдер Константин Копыркин создал индикатор под названием "NRTR_WATR": адаптированный вариант переворотной системы. Для измерения текущей волатильности величина истинной волатильности (ATR) усредняется с помощью взвешенной скользящей средней. Таким образом уровень трейлинг стопа (который является и переворотным уровнем) адаптируется к текущим условиям рынка.

Пример (как выполнить пример приведенного кода)...

Пересечение данного уровня NRTR снизу вверх либо сверху вниз может использоваться для входа в позицию по тренду, как это показано в данном примере, либо же в качестве основы для переворотной стратегии (SAR - stop-and-reverse).

Код примера на C# (кликните чтобы посмотреть)

```
1 | protected override void Execute()
2 | {
3 |
4 |     int Period = 20;
5 |     double Multiple = 3.0;
6 |     int Trend = 0;
7 |     double Reverse = 0;
8 |
9 |     DataSeries HPrice = Highest.Series( Close, 20 );
10 |    DataSeries LPrice = Lowest.Series( Close, 20 );
11 |    DataSeries K = WMA.Series( TrueRange.Series( Bars ), Period ) * Multiple;
12 |
13 |    DataSeries NRTR_WATR = new DataSeries( Bars, "NRTR_WATR" );
14 |
15 |    for(int bar = K.FirstValidValue; bar < Bars.Count; bar++)
16 |    {
17 |
18 |        // Рассчитываем ряд данных, где хранится значение NRTR_WATR
19 |
20 |        if( Trend >= 0 )
21 |        {
22 |
23 |            HPrice[bar] = Math.Max( Bars.Close[bar], HPrice[bar] );
24 |
25 |            Reverse = HPrice[bar] - K[bar];
26 |
27 |            if( Bars.Close[bar] <= Reverse )
28 |            {
29 |
30 |                Trend = -1;
31 |
32 |                LPrice[bar] = Bars.Close[bar];
33 |
34 |                Reverse = LPrice[bar] + K[bar];
35 |
36 |            }
37 |
38 |        }
39 |
40 |    }
```

```
41     }
42
43     if( Trend <= 0 )
44     {
45
46         LPrice[bar] = Math.Min( Bars.Close[bar], LPrice[bar] );
47
48         Reverse = LPrice[bar] + K[bar];
49
50         if( Bars.Close[bar] >= Reverse )
51         {
52
53             Trend = 1;
54
55             HPrice[bar] = Bars.Close[bar];
56
57             Reverse = HPrice[bar] - K[bar];
58
59         }
60     }
61
62     }
63
64     NRTR_WATR[bar] = Reverse;
65
66 }
67
68 // Отображаем на графике NRTR_WATR
69
70 PlotSeries( PricePane, NRTR_WATR, Color.Teal, WealthLab.LineStyle.Dotted, 3 );
71
72 //торговая стратегия - вход на следующем баре по тренду после пересечения.
73 for(int bar = NRTR_WATR.FirstValidValue+1; bar < Bars.Count; bar++)
74 {
75
76     {
77
78         if (IsLastPositionActive)
79         {
80
81             {
82
83                 if( CrossUnder( bar, Close, NRTR_WATR ) )
84
85                     SellAtMarket( bar+1, LastPosition, "NRTR Exit" );
86
87             }
88
89             else if( CrossOver( bar, Close, NRTR_WATR ) )
90
91                 BuyAtMarket( bar+1, "NRTR Entry" );
92
93         }
94     }
95 }
```

Именно так выглядит на графике индикатор Константина Копыркина:



Как видите, получилась целая торговая стратегия. Произведя небольшие изменения - можно использовать ее для торговли.

Для того, чтобы более подробно рассмотреть данную тему - можете посмотреть как пример стратегию Glitch Index.

Как: создать индикаторы используя внешние финансовые инструменты

Для построения любых индикаторов требуется доступ к барам либо к рядам данных финансовых инструментов. В зависимости от того, что Вы хотите делать, Вы можете использовать один из двух различных методов для доступа к барам внешних финансовых инструментов.

Пример (как выполнить пример приведенного кода)...

Как только Вы получаете доступ к барам внешнего финансового инструмента, Вы можете использовать информацию как о барах, так и о рядах данных этого внешнего финансового инструмента для построения любого технического индикатора.

```
Код примера на C# (кликните чтобы посмотреть)
1  protected override void Execute()
2  {
3      Bars gazpBars = GetExternalSymbol("GAZP", true);
4
5      // Отображаем график газпрома и его 20-ти периодичной скользящей средней на новой
6      // области графика
7
8      ChartPane gazpPane = CreatePane(60, true, true);
9      DataSeries smaGAZP = SMA.Series(gazpBars.Close, 20);
10     PlotSymbol(gazpPane, gazpBars, Color.LightGreen, Color.Black);
11     PlotSeries(gazpPane, smaGAZP, Color.Blue, LineStyle.Solid, 2);
12
13     // Отображаем 10-ти пириодичный ATR, рассчитанный по данным Газпрома в новой
```

```
14 | области
15 |
16 |     ChartPane atrPane = CreatePane(40, true, true);
17 |     DataSeries atrGAZP = ATR.Series(gazpBars, 10);
    PlotSeries(atrPane, atrGAZP, Color.Red, LineStyle.Solid, 2);
}
```

Как видите, независимо от того, какой инструмент выбран, все индикаторы считаются по данным газпрома, который был отображен в коде примера.



Сегодня мы рассмотрели, как можно самостоятельно, буквально "на лету" делать собственные индикаторы в программе Wealth-Lab. Изучив код сегодняшнего поста Вы получите навыки эффективной работы по созданию собственных индикаторов. В следующий раз продолжим рассматривать работу с индикаторами в программе Wealth-Lab и рассмотрим тему **валидность индикаторов**.

Если Вам понравилась эта статья - подпишитесь на **новые СТАТЬИ по RSS** или получайте наши **новые СТАТЬИ на E-mail**:

Рекомендую почитать также:

- **Проектирование и построение индикаторов в Wealth-Lab**
- **Стабильность индикаторов в программе Wealth-Lab в зависимости от ширины тестируемого окна.**
- **Код торговой системы HighLowLong для wealth lab**
- **Программирование однопозиционных стратегий (SP Strategies) в программе Wealth-Lab**
- **Методика тестирования техники выходов при построении МТС и отбора финансовых инструментов, подходящих для этой техники.**

 РУБРИКИ: [INDICATORS](#), [WEALTHSCRIPT](#), [ГЕНЕРАТОР МТС](#), [ПРИМЕРЫ МТС](#)

Комментариев: 5

Gerig пишет:

19 мая 2012 в 07:38



Доброго времени суток! Я с нового года осваиваю Wealth-lab самостоятельно, изучая инструкцию. Процесс этот шел не быстро, сначала надо перевести, потом осмыслить, что ты перевел, и затем попытаться повторить, то что указано в инструкции. К счастью нашел в интернете Ваш ресурс, где все раскладывается и доводится “практически на молекулярном уровне”. Очередной раз получил удовольствие от прочтения Ваших советов и приемов, с интересом жду новую публикацию. Спасибо за рубрику.

Ответить

Дмитрий Власов пишет:

19 мая 2012 в 10:08



Я тоже осваиваю эту программу. И процесс протекал точно также – перевод, осмысление, повторение. Именно поэтому и решил здесь выкладывать русский перевод Велс Лаб, чтобы как только понадобится что-то вспомнить – все было под руками.

Ответить

Ответы на вопросы по программированию торговых стратегий в Wealth-Lab | Финансовая лаборатория

пишет:

22 мая 2012 в 15:28

[...] [...]

Ответить

Проектирование и построение индикаторов в Wealth-Lab | Финансовая лаборатория пишет:

23 августа 2012 в 10:41

[...] Делаем индикаторы в Wealth-Lab своими руками Главная Рисование графических объектов на [...]

Ответить

Юрий пишет:

29 октября 2012 в 18:26



Дмитрий, здравствуйте!

Мне кажется, что второе предложение в разделе “Как: создать индикатор MACD с самостоятельно определяемым периодом” лучше перевести как то так:”Так как 26-периодная ЕМА имеет экспоненту 0.074074, а 12-периодная 0.153846, то вы можете классический индикатор MACD аппроксимировать разностью 12 и 26-периодных ЕМА (или другой комбинацией периодов).” Здесь речь идет о том, что экспоненты, вычисленные для этих ЕМА по формуле $2/(1+period)$ очень близки к значениям, которые берутся при вычислении классического MACD (0.075 и 0.15). Поэтому и MACD будет не сильно отличаться от классического, о чем и свидетельствует приводимый ниже пример. А в разделе про индикатор Копыркина в предложении “Для измерения текущей волатильности величина истинной волатильности (ATR) усредняется с помощью взвешенной скользящей средней.” в скобках букву А нужно убрать. Вы сами говорите, что с помощью взвешенной скользящей средней усредняется величина истинной волатильности, т.е. именно TrueRange (TR). Это видно и из приводимого ниже кода.

Ответить

Оставить комментарий

Имя (Обязательно)

Почта (Обязательно)

Вэб-сайт

Получать новые комментарии по электронной почте. Вы можете [подписаться](#) без комментирования.

© 2012, Финансовая лаборатория. Все права защищены.
Любое использование материалов допускается только с указанием активной ссылки на источник.



451
242
187



В день мой блог
посещает 77
человек,
сегодня

126



mail.ru РЕЙТИНГ

757144529
196



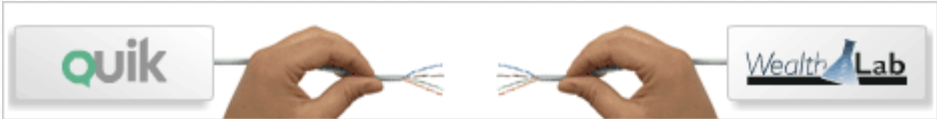
BY FEEDBURNER

Подписка

- [RSS](#)
- [Twitter](#)
- [ВКонтакте](#)
- [facebook](#)

Сайт

- [Создать блог](#)
- [Содержание](#)
- [О блоге](#)
- [Карта](#)



Стабильность индикаторов в программе Wealth-Lab в зависимости от ширины тестируемого окна.

 **АВТОР** ДМИТРИЙ ВЛАСОВ  **МАЯ** - 19 - 2012  **КОММЕНТАРИЕВ:** 5

Трейдеры, которые только начинают разбираться с техническим анализом (а иногда даже и более опытные трейдеры) частенько предполагают, что они могут применять технические индикаторы сразу же, как только у этого технического индикатора появится "валидное" значение.



Для большинства индикаторов, например таких, как простая скользящая средняя, стохастик и т.п. такое предположение обосновано. К примеру, значение 20-периодичной скользящей средней (SMA) всегда будет одинаковым сразу же после одинаковых 20-ти баров. Значения этих индикаторов в конце периода не зависят от данных, которые находятся в начале периода.

Другая большая группа индикаторов (экспоненциальная скользящая средняя, скользящая средняя Вилдера, RSI, MACD, индикатор Кальмана и многие другие) при своем расчете используют свои собственные предыдущие значения.

Скорее всего, Вы заметили, что такие, постепенно рассчитываемые индикаторы, подобные тем, которые находятся во второй группе, быстрее реагируют на движение цены. Однако следует иметь в виду, что значения таких индикаторов в самом начале расчета нестабильны.

Почему значения некоторых индикаторов нестабильны и зависят от периода доступных данных?

Это хороший вопрос. Для ответа на этот вопрос мы подготовили пример, в котором используется 4 индикатора с шириной исследуемого окна 20 баров назад.

Пример ([как выполнить пример приведенного кода](#))...

Код примера на C# (кликните чтобы посмотреть) 

```
1  protected override void Execute()
2  {
3      HideVolume();
4      ChartPane pane1 = CreatePane( 25, true, true );
5      ChartPane pane2 = CreatePane( 25, true, true );
6      ChartPane pane3 = CreatePane( 25, true, true );
7      ChartPane pane4 = CreatePane( 25, true, true );
8      PrintDebug("");
9      PrintDebug("Количество баров = " + Bars.Count);
10
11     DataSeries ema = EMA.Series(Close, 20,
12     WealthLab.Indicators.EMACalculation.Modern);
13     DataSeries macd = MACD.Series(Close);
14     DataSeries rsi = RSI.Series(Close, 20);
15     DataSeries sma = SMA.Series(Close, 20);
16
17     // Print the last value of each indicator to the debug window
18     int bar = Bars.Count - 1;
19     PrintDebug("ema = " + ema[bar]);
20     PrintDebug("macd = " + macd[bar]);
21     PrintDebug("rsi = " + rsi[bar]);
22     PrintDebug("sma = " + sma[bar]);
```

[RSS товары](#)



[RSS новости](#)



[Twitter](#)



Последние комментарии

макс: А где взять свечки с открытым интересом? на сайте финама нет



ring10: Добрый день Дмитрий Как сделать, чтобы открывался 1 велф лаб? Если открываешь под пользователем, т [...]



Дмитрий Власов: Буян, рад, что Вам пригодились материалы данной статьи!



Максим: Сам себе отвечу. protected override void Execute() { Bars rts = GetExternalSymbol("RTS", t [...]



Буян: Спасибо Дмитрий за статью! Мне удалось разработать хорошую рабочую торговую систему, используя возмо [...]



Последние посты



Бесплатный вебинар: 7 смертных грехов при оптимизации торговых систем

НОЯБРЯ - 5 - 2012



Отзывы участников курса до результата Wealth-Lab. С нуля до первой торговой системы за 5

дней.

НОЯБРЯ - 3 - 2012



Быстрее, Выше, Сильнее: Обновление Wealth-Lab до новой версии 6.4

ОКТАБРЯ - 23 - 2012



Курс до результата: Wealth-Lab – от нуля до первой торговой системы за 5 дней

ОКТАБРЯ - 23 - 2012



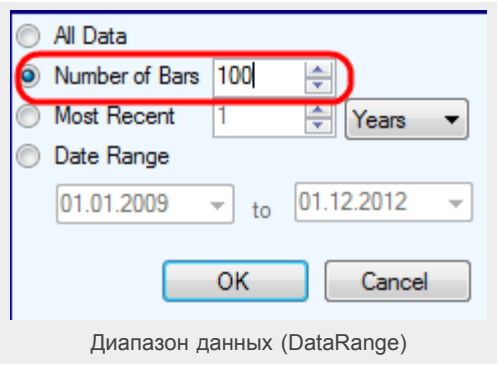
Обзорная экскурсия по достопримечательностям программы Wealth-Lab

ОКТАБРЯ - 20 - 2012

Берите

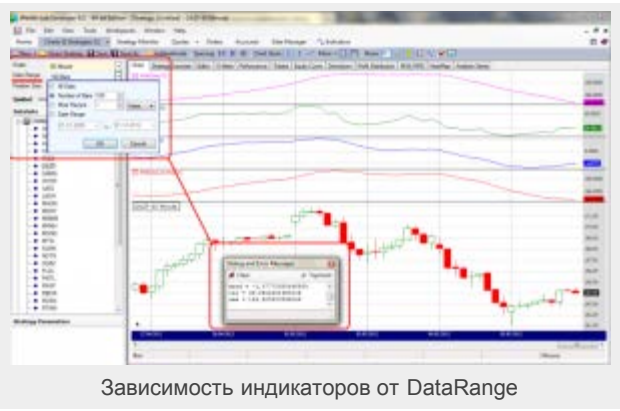
```
23
24     PlotSeries(pane1, ema, Color.Red, WealthLab.LineStyle.Solid, 1);
25     PlotSeries(pane2, macd, Color.Blue, WealthLab.LineStyle.Solid, 1);
26     PlotSeries(pane3, rsi, Color.Green, WealthLab.LineStyle.Solid, 1);
27     PlotSeries(pane4, sma, Color.Fuchsia, WealthLab.LineStyle.Solid, 1);
}
```

Для того, чтобы выполнить этот пример, сначала установите диапазон тестирования (в DataRange) равное 21-му бару и кликните по любому финансовому инструменту.



Диапазон данных (DataRange)

Затем постепенно меняйте ширину диапазона тестирования, увеличивая количество баров до 30, 40, 60, 100 и 120.



Зависимость индикаторов от DataRange

При смене диапазона данных, программа будет рассчитывать и распечатывать новые результаты расчетов для индикаторов на последнем баре.

№ пп	Индикаторы:	30 баров	40 баров		60 баров		100 баров		120 баров	
		Значение	Значение	Именение, %	Значение	Именение, %	Значение	Именение, %	Значение	Именение, %
1.	ema	164,925	164,83009	-0,06	164,75418	-0,05	164,76589	0,01	164,76571	0,00
2.	macd	-1,060081	-1,16022	9,45	-1,07222	-7,58	-1,07773	0,51	-1,07759	-0,01
3.	rsi	36,36748	38,73286	6,50	40,9381	5,69	39,58124	-3,31	39,78702	0,52
4.	sma	164,8205	164,8205	0,00	164,8205	0,00	164,8205	0,00	164,8205	0,00

Изменение значений индикаторов

Вскоре, выполняя эти действия, Вы заметите, что только один из четырех индикаторов (SMA) не меняет своего значения, в то время как значения других индикаторов колеблются пока не стабилизируются на более точном значении.

Почему опасно использовать нестабильные значения индикаторов

Как мы продемонстрировали, значения некоторых индикаторов на коротком отрезке времени могут демонстрировать невероятно разнящиеся результаты. Хитрость здесь заключается в том, чтобы понять те индикаторы, которые Вы используете и игнорировать значения этих индикаторов до тех пор, пока их значения не станут устойчивыми.

Для осмысленных и воспроизводимых результатов торговых стратегий Вы должны использовать индикаторы только с того времени, когда их значения становятся стабильными.

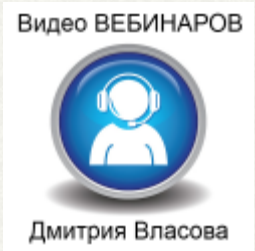
Лучшим способом игнорирования индикаторов, имеющих неустойчивые значения - является запуск торгового цикла после бара, на котором даже индикатор, требующий для своего расчета самое длительное время уже приобрел стабильность.

Для наших 4-х индикаторов, которые используются в примере, Вы можете заметить, что Значение индикатора RSI требует больше всего баров для его стабилизации. Именно поэтому мы должны будем запускать торговый цикл не раньше чем на 60-м баре, а еще лучше - протестировать его значение даже на 120-м баре.

Разумным правилом для того, чтобы не иметь проблем с постепенно рассчитываемыми индикаторами может стать необходимость игнорировать значения этих индикаторов на периоде времени, который в 3 или даже 4 раза превышает тот период, который нужен для их расчета.

Пример (как выполнить пример приведенного кода)...

```
Код примера на C# (кликните чтобы посмотреть)
1  protected override void Execute()
2  {
3      /* начало торгового цикла только после пропуска периода для стабильности
   индикаторов*/
4      for(int bar = 60; bar < Bars.Count; bar++)
5      {
6          /* код стратегии, либо торговые правила */
7      }
8  }
```



МТС под микроскопом: WLMA System

Подробный разбор торговой системы WLMA System
3900 руб.

В корзину

Рубрики

Выберите рубрику

Обновления:

- Новые посты по Email
- Новые посты по RSS
- Комментарии по RSS
- Следим на Twitter

Метки

С# HowTo: Wealth-Lab
Видео Индикаторы МТС Торговые
роботы Юмор алготрейдинг
мероприятия обучение

Управление

Регистрация

Вход

RSS Записей

RSS Комментариев

WordPress.org

С какими индикаторами нужно быть осторожным?

Следующие индикаторы из из библиотеки стандартных индикаторов Wealth-Lab, вычисляются с использованием собственных значений из предыдущих периодов и поэтому могут рассчитывать нестабильные значения в начале ряда данных.

ADX, ADXR, ATR, ATRP, CADO, DIMinus, DIPlus, DSS, EMA, FAMA, Kalman, KAMA, KST, MACD, MAMA, OBV, Parabolic, Parabolic2, RSI, StochRSI, TRIX, UltimateOsc, Vidya, Volatility, WilderMA

Индикатор аккумуляции-дистрибуции является нестабильным в том смысле, что его вычисление "аккумулируется" с определенного бара. Если эта начальная точка меняется, то следовательно тут же меняется и его вычисления для всех прочих баров.

Работая с этими индикаторами не забывайте игнорировать их значения на отрезке времени, который минимум в 4 раза превышает тот отрезок времени, который необходим для их вычисления. Сегодняшняя тема очень важна, так как очень часто трейдеры игнорируют высказанные здесь правила. В следующий раз приступим к самой интересной теме: **программирование торговых стратегий**.

Если Вам понравилась эта статья - подпишитесь на **новые СТАТЬИ по RSS** или получайте наши **новые СТАТЬИ на E-mail**:

Рекомендую почитать также:

- **Проектирование и построение индикаторов в Wealth-Lab**
- **Делаем индикаторы в Wealth-Lab своими руками**
- **Отображение на графике синтетических финансовых инструментов**
- **Виды торговых сигналов в Wealth-Lab**
- **Использование обоснованного заглядывания в будущее при программировании торговых стратегий в Велс Лаб**

 РУБРИКИ: **INDICATORS**, **WEALTHSCRIPT**

Комментариев: 5

budmit пишет:

20 мая 2012 в 23:12



Добрый вечер, Дмитрий.

Как Вы используете WL? Только для тестов систем? а потом готовые в TSLab? Или есть решение для подключения сразу к квику версий выше 4? Мне нужно решение с сишарпом, поэтому 4 версия не подходит.

Вы купили лицензионную версию при отсутствии адаптера к квику? чем оно мотивировано ? Есть же даже бесплатные решения типа стокшарпа...

У меня не спортивный интерес, дело в том, что я много лет сидел на омеге и горя как говориться не знал – ну вот вырос из этих штанишек, посмотрел по сторонам , а софта то великое многообразие появилось, вот и встал вопрос выбора, а это оказалось сделать не легко для человека весьма посредственного в программировании.

Тем не менее я всегда сам кодировал себе стратегии в экселе и в омеге и хочу выбрать развивающийся продукт и чтобы он давал мне много возможностей даже если я их не буду все использовать...

Подскажите может где есть нормальное объективное сравнение софта? Как выбрать себе инструмент оказалось тяжело 😞

Заранее благодарен.

Ответить

Дмитрий Власов пишет:

21 мая 2012 в 11:35



Добрый день.

1. Я используя Wealth-Lab только для тестов торговых систем, т.к.:

а) здесь есть очень удобные инструменты – генетический оптимизатор и монте-карло оптимизатор, что позволяет за очень короткое время находить оптимальные параметры даже для тех стратегий, у которых количество и диапазон параметров велик и обычным способом невозможно за разумное время подобрать нормальные параметры торговой системы.

б) Очень удобная библиотека (не нужно самому заново писать кучу индикаторов, осцилляторов и прочих прикрас для системы.

в) Лицензия дает доступ не только к программе, но и к закрытому для всех разделу сайта wealth-lab.com – где представлено более 150 уже готовых торговых систем + этот раздел постоянно пополняется.

г) Есть доступ к технической поддержке и эта техподдержка работает очень эффективно.

д) Даже непрограммисту (я тоже не программист) довольно легко разобраться с языком написания стратегий C# и с библиотекой WealthScript – тем более есть прекрасное руководство на английском (а теперь, надеюсь, и на русском)

2. Возможность подключить к Квику через брокер Адаптер была буквально месяц назад через брокера Церих, но там у них что-то с руководителем этого направления Арсеном Яковлевым произошло – кажется, ушел куда-то так что теперь можно подключаться только самостоятельно делая брокер-адаптер. Дело это непростое, но умельцы существуют. Смотри, к примеру сайт Игоря Чечета <http://chechet.org> – он торгует через собственный брокер адаптер (кажется, к Альфа брокеру).

3. Если хотите автоматически торговать без заморочек с написанием брокерадаптера и поддержания его в работоспособном состоянии – на мой взгляд разумнее всего поступать так: разрабатывать торговые стратегии в Велсе, а торговать их либо через ТСЛаб, либо используя Stock#

Ответить

Ответы на вопросы по программированию торговых стратегий в Wealth-Lab | Финансовая лаборатория пишет:

22 мая 2012 в 16:24

[...] 52. Почему значения некоторых индикаторов нестабильны и з... [...]

Ответить

Организация торгового цикла в программе Wealth-Lab и его переделка в скрининговую систему | Финансовая лаборатория пишет:

25 мая 2012 в 14:16

[...] В данном примере цикл стартует с 50-го бара данных, однако Вам следует выбирать точку старта торгового цикла исходя из периода тех индикаторов, которые Вы используете в торговой стратегии с учетом их устойчивости. [...]

Ответить

Делаем индикаторы в Wealth-Lab своими руками | Финансовая лаборатория пишет:

23 августа 2012 в 10:41

[...] Стабильность индикаторов в программе Wealth-Lab в зависимо... Главная Проектирование и построение [...]

Ответить

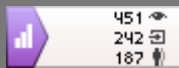
Оставить комментарий

Имя (Обязательно)

Почта (Обязательно)

Вэб-сайт

Получать новые комментарии по электронной почте. Вы можете [подписаться](#) без комментирования.



451
242
187

В день мой блог
посещает **77**
человек,
сегодня

126

А вы?

mail.ru РЕЙТИНГ
757145 530
196

BY FEEDBURNER

[Twitter](#)

[ВКонтакте](#)

[facebook](#)

[Содержание](#)

[О блоге](#)

[Карта](#)

Программирование торговых стратегий в программе Wealth-Lab

АВТОР ДМИТРИЙ ВЛАСОВ МАЯ - 22 - 2012 КОММЕНТАРИЕВ: 10

Главной особенностью Wealth-Lab является то, что используя язык C# и библиотеку WealthScript, Вы получаете возможность запрограммировать любые правила Вашей торговой системы. Всякий раз, когда Вы запускаете запрограммированную стратегию, Wealth-Lab отображает на графике все генерируемые сделки при помощи легко воспринимаемых стрелочек, обозначающих покупки и продажи.

В результате выполнения стратегии рассчитывается целый список показателей, отражающих эффективность стратегии. Также в вашем распоряжении появляется детализированный список всех сделок, которые были бы совершены, придерживаясь Вы всех правил, которые Вы запрограммировали в стратегии.



Посмотрите повнимательнее настройки (доступны при нажатии F12) для того, чтобы увидеть дополнительные возможности отражения результатов выполнения стратегии (дополнительные визуализаторы).

Класс стратегий в программе Велс Лаб

В программе Wealth-Lab классы стратегий являются производными от класса WealthScript. Вы можете добавлять методы, процедуры, переменные, классы и т.п. в класс Вашей стратегии, но при этом класс с Вашей стратегией должен содержать метод Execute(), который является входной точкой для запуска каждой стратегии. И именно здесь программируется основная часть большинства стратегий.

Инсталлированный в Wealth-Lab шаблонный код стратегии показывает типичную структуру кода, применяемого для проектирования торговых стратегий.

Пример (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть)

```
1  <pre>/* Встроенный в Wealth-Lab шаблон кода стратеги */
2  using System;
3  using System.Collections.Generic;
4  using System.Text;
5  using System.Drawing;
6  using WealthLab;
7  using WealthLab.Indicators;
8
9  namespace WealthLab.Strategies
10 {
11     public class MyStrategy : WealthScript
12     {
13         protected override void Execute() //метод, который является "точкой входа"
14         {
15             for (int bar = 20; bar < Bars.Count; bar++) //запускаем цикл,
16             просматривая каждый бар от начального до конечного
17             {
18                 if (IsLastPositionActive) //Если существует активная позиция
19                 {
20                     //Здесь нужно запрограммировать правила выхода из позиции
21                 }
22             }
23             else //если нет активных позиций
24             {
```

RSS товары

RSS новости

Twitter

Последние комментарии

- макс: А где взять свечки с открытым интересом? на сайте финама нет


- ring10: Добрый день Дмитрий Как сделать, чтобы открывался 1 велф лаб? Если открываешь под пользователем, т [...]


- Дмитрий Власов: Буян, рад, что Вам пригодились материалы данной статьи!


- Максим: Сам себе ответчу. protected override void Execute() { Bars rts = GetExternalSymbol("RTS", t [...]


- Буян: Спасибо Дмитрий за статью! Мне удалось разработать хорошую рабочую торговую систему, используя возмо [...]



Последние посты

- 

Бесплатный вебинар: 7 смертных грехов при оптимизации торговых систем
НОЯБРЯ - 5 - 2012
- 

Отзывы участников курса до результата Wealth-Lab. С нуля до первой торговой системы за 5 дней.
НОЯБРЯ - 3 - 2012
- 

Быстрее, Выше, Сильнее: Обновление Wealth-Lab до новой версии 6.4
ОКТАБРЯ - 23 - 2012
- 

Курс до результата: Wealth-Lab – от нуля до первой торговой системы за 5 дней
ОКТАБРЯ - 23 - 2012
- 

Обзорная экскурсия по достопримечательностям программы Wealth-Lab
ОКТАБРЯ - 20 - 2012

Берите

```
25 |                                     //Здесь нужно запрограммировать правила входа в позицию
26 |                                     }
27 |                                 }
28 |                            }
29 |    }
```

Вы можете изменить шаблонный код стратегии - написав свой собственный вариант и сохранив этот шаблон в виде кода стратегий по умолчанию (выбрав Edit > Set as Default Template Code).

Компиляция кода стратегий в программе Wealth-Lab

Класс Strategy, который Вы создаете чтобы описать все правила стратегий компилируется и создается когда Вы:

- Нажимаете на кнопку "Run the Strategy",
- Нажимаете на кнопку "Compile",
- Если Вы открываете нескомпилированную стратегию из проводника стратегий (Strategy Explorer),
- Когда окно стратегии восстанавливается вместе с рабочим пространством.

В результате компиляции стратегии все находящиеся в рамках класса переменные инициализируются только тогда, когда создается экземпляр класса стратегии.

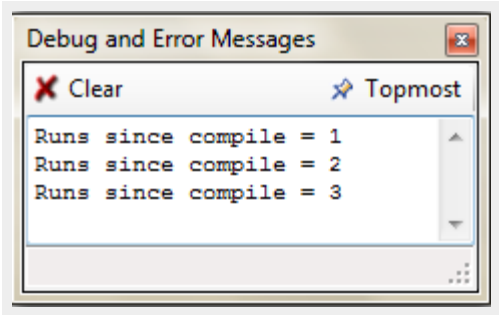
Выполните стратегию, приведенную в примере нажав кнопку "Run the Strategy", нажав клавишу F5, или кликнув по любому финансовому инструменту.

Пример (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть)

```
1 | using System;
2 | using WealthLab;
3 |
4 | namespace WealthLab.Strategies
5 | {
6 |     public class StrategyTest : WealthScript
7 |     {
8 |         private int _scriptRunCountSinceCompile = 0;
9 |
10 |         protected override void Execute()
11 |         {
12 |             _scriptRunCountSinceCompile++;
13 |             PrintDebug("Runs since compile = " + _scriptRunCountSinceCompile);
14 |         }
15 |     }
16 | }
```

Последовательно выполняя указанную в данном примере стратегию (различными способами) Вы увидите, как меняется счетчик количества раз запуска стратегии.



Обращаем Ваше внимание на то, что переменная _scriptRunSinceCompile получает первоначальное значение равное нулю только после компиляции и только после этого сохраняет заданное значение при выполнении метода Execute().

Ограничения для кода стратегий в программе Wealth-Lab

Стратегии, базирующиеся на таких показателях, как Equity, Cash, или Drawdown

Следует запомнить, что архитектура программы Wealth-Lab Developer 6-й версии не позволяет создавать такие торговые стратегии, которые принимают решения о входе в сделку либо о выходе из нее опираясь на показатель Equity, который включает в себя величину Cash (наличных денег) и Drawdown (просадку).

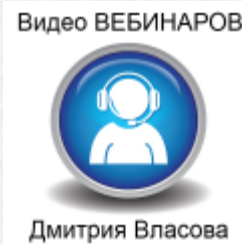
Для того, чтобы получить информацию об определенном методе либо свойстве библиотеки WealthScript, которые используются в коде стратегии - выделите мышкой соответствующий метод либо свойство и нажмите кнопку F11, что позволит Вам обратиться к справочнику QuickRef (F11).

Стратегии с единственной позицией (однопозиционные стратегии), но со множественными условиями на вход

В особых случаях - только для двух режимов тестирования: "Raw Profit" и "Portfolio Simulation" в программе Wealth-Lab возможно не инициировать входной сигнал тогда, когда это ожидается. Для этого должны соблюдаться следующие условия:

1. Код стратегии придерживается однопозиционной логики, что значит в одно и тоже время может существовать только одна позиция для одного финансового инструмента;
2. В коде не прописываются правила управления величиной позиции;
3. Новый сигнал на вход в позицию по условию 1 должен находиться в коде стратегии раньше чем сигнал на выход из позиции по условию 2.,

Wealth-Lab всегда выполняет сделки, прописанные в стратегии в режиме "Raw Profit", поэтому первая сделка



МТС под микроскопом: WLMA System

Подробный разбор торговой системы WLMA System
3900 руб.

В корзину

Рубрики

Выберите рубрику

Обновления:

- Новые посты по Email
- Новые посты по RSS
- Комментарии по RSS
- Следим на Twitter

Метки

C# HowTo: **Wealth-Lab**
Видео Индикаторы МТС Торговые
роботы Юмор алготрейдинг
мероприятия обучение

Управление

Регистрация

Вход

RSS Записей

RSS Комментариев

WordPress.org

в ряду нескольких условий на вход всегда приводит к созданию позиции. Если эта позиция в дальнейшем отвергается (из-за правил управления величиной позиции либо вследствие недостаточности капитала для входа в позицию), то создание новой позиции (даже если выполняется какое-либо другое условие на вход в позицию) невозможно до тех пор, пока не произойдет выход из позиции (это справедливо только для однопозиционная логики). Это объясняется тем, что ветвь входа в позицию не начнет отрабатываться в коде программы пока не произойдет выход из позиции.

В качестве обобщенного примера представьте себе стратегию с однопозиционной логикой кода, которая всегда генерирует сигнал на покупку в каждый вторник или среду, а сигнал на выход из позиции генерируется в пятницу. Выполнение данной стратегии в режиме "Raw Profit" всегда будет приводить ко входу в позицию только во вторник, т.к. логика выхода из позиции предполагает совершение сделки только в пятницу. Соответственно, сделка в среду всегда будет пропускаться. Тестовый вход в позицию в среду никогда не осуществиться, даже если сделка во вторник будет отклонена из-за правил управления величиной позиции либо из-за недостаточности капитала.

Сегодняшняя обзорная часть, посвященная программированию торговых стратегий в программе Wealth-Lab завершена. В следующей статье подробно рассмотрим **виды торговых сигналов**, которые существуют в программе Велс Лаб. Не забывайте подписаться на новые посты по RSS.

Если Вам понравилась эта статья - подпишитесь на **новые СТАТЬИ по RSS** или получайте наши **новые СТАТЬИ на E-mail**:

Рекомендую почитать также:

- **Виды торговых сигналов в Wealth-Lab**
- **Организация торгового цикла в программе Wealth-Lab и его переделка в скрининговую систему.**
- **Ловушки при построении торговых систем, связанные с проблемой подглядывания в будущее**
- **Использование обоснованного заглядывания в будущее при программировании торговых стратегий в Велс Лаб**
- **Создание параметров торговой стратегии в Wealth-Lab**

 РУБРИКИ: **PROGRAMMING TRADING STRATEGIES, WEALTHSCRIPT**

Комментариев: 10

Стабильность индикаторов в программе Wealth-Lab в зависимости от ширины тестируемого окна. | Финансовая лаборатория пишет:

22 мая 2012 в 17:12

[...] Программирование торговых стратегий в программе Wealth-La... Главная Делаем индикаторы в Wealth-Lab своими [...]

Ответить

Ответы на вопросы по программированию торговых стратегий в Wealth-Lab | Финансовая лаборатория пишет:

22 мая 2012 в 17:19

[...] [...]

Ответить

Artem_trader пишет:

3 августа 2012 в 00:28



Начал изучать бй WealthLab и C# заодно. Не в последнюю очередь из-за того, что наткнулся на этот сайт. До этого 4 года работал с WLD 3.0

Наткнувшись на это, начинаю сомневаться в адекватности ТЕСТИРУЮЩЕГО софта:

“Следует запомнить, что архитектура программы Wealth-Lab Developer 6-й версии не позволяет создавать такие торговые стратегии, которые принимают решения о входе в сделку либо о выходе из нее опираясь на показатель Equity, который включает в себя величину Cash (наличных денег) и Drawdown (просадку).”

И это несмотря на то, что:

“Главной особенностью Wealth-Lab является то, что используя язык C# и библиотеку WealthScript, Вы получаете возможность запрограммировать любые правила Вашей торговой системы”

Ну ладно, вопрос такой:

Каким образом можно реализовать установку размера позиции с учётом размера капитала? Я до сих пор не верю, что такие элементарные вещи не могут быть реализованы в WLD 6.x. Я хотел получить новые возможности, а тут “вот оно чо” (с).

У меня к размеру капитала привязана величина стопа, которая каждый раз разная, т.е. для каждой следующей сделки величина стопа рассчитывается индивидуально. От величины стопа рассчитывается кол-во участвующих в сделке контрактов.

PosSizer хочет фиксированной величины стопа, т.е. не подходит. Внутри кода, получается, формулу для расчёта

кол-ва контрактов с учётом указанных мною входных данных, создать не получится?

Может, это вопрос более написания дополнительных библиотек или чего-нибудь в этом духе? Не верю я в такой фейл 😞

Ответить

Дмитрий Власов пишет:

3 августа 2012 в 19:44



Арте́м, приветствую.

Конечно же в WEalth-Lab 6 можно эффективно управлять размером позиции. Особенностью этой программы является то, что размером позиции можно управлять как прямо внутри кода с помощью метода SetShareSize() (почитайе, к примеру здесь: <http://finlab.copiny.com/question/details/id/50224>) так и с помощью специальных модулей управления размером позици (PosSizer).

В Вашем распоряжении уже целая куча таких модлей для управления размером позиции в дефолтной версии. Кроме того, лицензионным пользователям можно скачать уже готовые модули управления размером позиции **вот отсюда**:

И конечно же, Вы можете написать свой собственный модуль управления размером позици, обладаю минимальными навыками работы в C#. Обо всем этом я буду рассказывать на своем авторском курсе про Wealth-Lab. И после этого вкратце выложу здесь на блоге.

А для того, чтобы рассчитать количество лотов (контрактов) для покупки (или продажи) в зависимости от риска на одну сделку, размера капитала и отдаленности Stop Loss – будет достаточно метода управения позицией, который называется “Max Percent Risk” и устанавливает максимальный риск за одну сделку (выражный в процентах от наличного капитала). Затем в зависимости от удаленности Стопа от цены входа этот модуль рассчитывает количество контрактов, котрым можно торговать без риска превысить максимальный уровень риска.

Ответить

Artem_trader пишет:

3 августа 2012 в 00:52



Пропал мой комментарий с решением проблемы...

Ответить

Дмитрий Власов пишет:

3 августа 2012 в 19:46



Арте́м, Ваш комментарий не нашел.. Хотя весь “Спам” перерыл – туда он не попадал. Так что,если не трудно, напишите, плз., еще раз.

Ответить

Artem_trader пишет:

4 августа 2012 в 00:23



Сразу же после того, как написал первый комментарий, немного порылся на коммьюнити (<http://www.wealth-lab.com/Community/Forum/Default.aspx?id=vFcanP5D6YwLLKGFY5FyfA==>). Сразу вопрос не в тему, но вдруг кто знает – как получить доступ в коммьюнити? Примеры кодов скрыты для незарегистрированных юзеров. Я регистрировался, получил подтверждение по почте, активировал аккаунт, но каждый раз при вводе логина получаю надпись “Username is logged”. Фрагменты обсуждений в коммьюнити, таким образом недоступны. Не знаю, с какого конца туда подступиться, пока забил на это, вернёмся к теме.

SetShareSize, естественно, мне знаком, пользуюсь на WLD 3, тут проблем нет.

Возможности PozSizer тоже понятны – они не дают динамически из кода управлять размером позиции, если, например, от сделки к сделке меняется % риска на капитал (типа, в одних рыночных условиях мы можем пойти на бОльший риск, чем в других). Кроме того, неплохо бы вынести этот параметр для оптимизации.

Так вот для этих целей и нужен механизм определения денежной позиции для произвольного бара в ходе тестирования.

Проблема решается путём самостоятельного симулирования кривой эквити. Т.е. вводим переменную, в которой храним сумму капитала, к которой, если у нас открыта позиция, по окончании каждого бара прибавляется чистая прибыль от удержания позиции по итогам этого отдельного бара. Для этого используем метод .NetProfitAsOfBar() объекта Position (Я могу путаться в терминах, заранее сорри, только начинаю изучать ООП и C#).

Технически выглядит так:

```
if (!IsLastPositionActive)
{
    EquitySize = EquitySize + LastActivePosition.NetProfitAsOfBar( bar );
}
```

Правда, справедливости ради, надо сказать, что у меня всё ещё не заработал этот механизм 😞 Выдает ошибку, но

не на этапе компиляции, а Runtime:
“Runtume error: В экземпляре объекта не задана ссылка на объект.
в WealthLabCompile.GSV.Execute()”. Но, думаю, я в ближайшее время с этим разберусь, и вся схема в принципе, должна работать.
Дмитрий, спасибо за ресурс, так бы ещё долго не собрался учиться 😊
P.S. В этот раз убрал все ссылки, посмотрим, пролезет ли коммент на этот раз.

Ответить

Artem_trader пишет:
4 августа 2012 в 00:32



Как говорится, сформулировать проблему – значит, наполовину решить её. Кусок кода, который я привёл, надо было размещать в блоке “если есть активная позиция”, а то в моём примере объект Position не существует (не знаю, правильно ли я сказал, но смысл ясен).
Перенёс в блок “если есть активная позиция”.
Вообще, кстати говоря, мне кажется странным этот шаблон кода, когда сначала проверяются условия для закрытия позиции, а следом – условия для открытия. ИМХО, это изврат 😊
В этом случае, нам не получится открыть и закрыть позицию в течение одного бара, потому что после открытия позиции, счётчик баров увеличиться и мы будем иметь дело уже со следующим баром.

Ответить

Artem_trader пишет:
17 августа 2012 в 15:46



Видимо, то, о чём я написал, мало кто использует, раз никто не высказался по описанной мною проблеме.
Напомню как она звучит – “как управлять размером позиции из кода в зависимости от предполагаемого риска на капитал в КАЖДОЙ конкретной сделке”.
Т.е. внимание! В одной сделке мы согласны рискнуть каким-то процентом от размера капитала, а в следующей сделке – уже другим процентом. Это определяется в зависимости от рыночных условий, не суть.
PozSizer требует, чтобы мы заранее знали риск на капитал в любой сделке, которая может быть сгенерирована системой и просит указывать нас размер риска в %тах.
Решить проблему можно было бы легко и просто, если бы в Wealth-Lab 6.0 присутствовала возможность определения размера капитала в каждом произвольном баре, и эту цифру можно было сохранить в переменной:
ActualEquity = Equity(bar)
В 6й версии такой возможности не оказалось, а PozSizer несмотря на его гибкость, не покрывает все потребности по управлению размером позиции.
Таким образом, я пришёл к тому, что на основе данных по каждой сделке можно симулировать приращение капитала, сохраняя в произвольной переменной эти данные.
В коде это выглядит так:
if (IsLastPositionActive)
EquitySize = EquitySize+LastActivePosition.NetProfitAsOfBar(bar);
Способ вполне работает, однако, сейчас конечная цифра эквити у Тестера и у меня внутри кода различаются 😊
Не существенно, конечно, но неприятно 😊 Комиссии убраны, в чём может быть дело?

Ответить

Graf пишет:
3 октября 2012 в 23:25



Artem_trader
Привет!
Дружище, расскажи, получилось сделать то, что ты задумал?
Я тоже хочу сделать стратегию, в которой риск на сделку (размер позиции) зависит от состояния equity.
Шестая версия дает возможность сделать это стандартными средствами на портфеле или нет?
А придуманный тобой способ сработал?

Ответить

Оставить комментарий

Имя (Обязательно)

Почта (Обязательно)

Получать новые комментарии по электронной почте. Вы можете [подписаться](#) без комментирования.

© 2012, Финансовая лаборатория. Все права защищены.
Любое использование материалов допускается только с указанием активной ссылки на источник.



451
242
187



В день мой блог
посещает **77**
человек,
сегодня **126**



mail.ru РЕЙТИНГ
757146 531
196



BY FEEDBURNER

Подписка

- [RSS](#)
- [Twitter](#)
- [ВКонтакте](#)
- [facebook](#)

Сайт

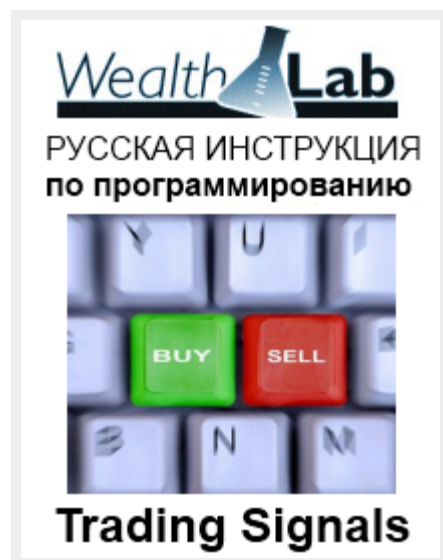
- [Создать блог](#)
- [Содержание](#)
- [О блоге](#)
- [Карта](#)

Виды торговых сигналов в Wealth-Lab

АВТОР [ДМИТРИЙ ВЛАСОВ](#) ⌚ МАЯ - 23 - 2012 💬 КОММЕНТАРИЕВ: 4

Ну вот мы и подобрались к самой интересной теме. Согласитесь, программ, которые красиво строят графики достаточно много. А вот программ, которые позволяют протестировать любые идеи, которые только могут возникнуть у Вас в голове - можно по пальцам пересчитать.

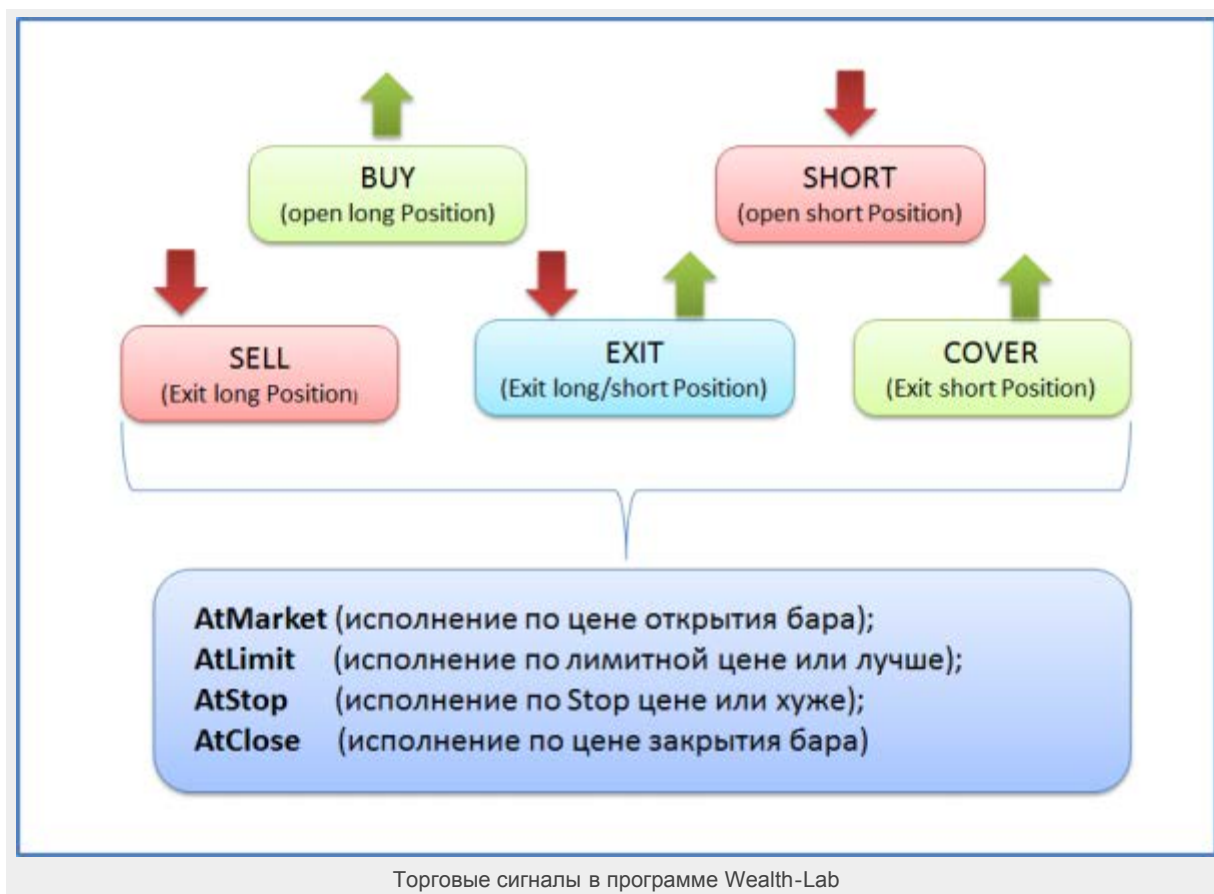
Велс Лаб в умелых руках является той программой, которая позволит Вам не только спроектировать и протестировать любые торговые идеи, но и детально изучить все показатели протестированных систем. А если Вам чего-либо не хватает, Вы сможете, применив C#, доделать недостающие компоненты.



Бэктестинг стратегий в программе Wealth-Lab обозначает, что программа может создавать теоретические сделки в соответствии с определенными критериями.

Какие торговые сигналы поддерживаются в программе Wealth-Lab

Торговые сигналы в программе Велс Лаб это по-сути те распоряжения, которые Вы даете Вашему брокеру. Взгляните на следующую схему:



В Wealth-Lab возможны все комбинации верхней и нижней частей этой схемы. Такое комбинирование приводит к 16 типам приказов (если учитывать Exit, то к 20-ти типам). Т.е. для входа в длинную позицию

RSS товары



RSS новости



Twitter



Последние комментарии

макс: А где взять свечки с открытым интересом? на сайте финама нет



ring10: Добрый день Дмитрий Как сделать, чтобы открывался 1 велф лаб? Если открываешь под пользователем, т [...]



Дмитрий Власов: Буян, рад, что Вам пригодились материалы данной статьи!



Максим: Сам себе отвечу. protected override void Execute() { Bars rts = GetExternalSymbol("RTS", t [...]



Буян: Спасибо Дмитрий за статью! Мне удалось разработать хорошую рабочую торговую систему, используя возмо [...]



Последние посты



Бесплатный вебинар: 7 смертных грехов при оптимизации торговых систем

НОЯБРЯ - 5 - 2012



Отзывы участников курса до результата Wealth-Lab. С нуля до первой торговой системы за 5 дней.

НОЯБРЯ - 3 - 2012



Быстрее, Выше, Сильнее: Обновление Wealth-Lab до новой версии 6.4

ОКТАБРЯ - 23 - 2012



Курс до результата: Wealth-Lab – от нуля до первой торговой системы за 5 дней

ОКТАБРЯ - 23 - 2012



Обзорная экскурсия по достопримечательностям программы Wealth-Lab

ОКТАБРЯ - 20 - 2012

Берите

можно использовать, к примеру, такой тип приказа: BuyAtMarket, а для выхода из длинной позиции применяется, к примеру, тип приказа: SellAtLimit. При этом и BuyAtMarket() и SellAtLimit() - являются методами WealthScript.

Какие параметры имеют торговые сигналы в Wealth-Lab

Указанные методы требуют параметры. Рассмотрим эти параметры торговых сигналов более подробно.

bar - все торговые сигналы в обязательном порядке имеют параметр bar. Этот параметр определяет - на каком баре будет сделана сделка. В общем случае, приказы AtMarket и AtClose располагаются на баре. Исключением могут быть только сделки с недостаточной позицией либо недостатком эквити.

pos - для всех сигналов на выход из позиции (Sell, Cover или Exit) Вы должны определить позицию, к которой будет относиться сигнал на выход. В большинстве случаев в коде однопозиционных стратегий Вы будете чаще всего использовать LastPosition. Для того, чтобы без проблем выходить из любых позиций, присваивайте параметру pos значение Position.AllPositions.

LimitPrice или StopPrice - Вы должны использовать цену stop / limit для ордеров типа AtLimit / AtStop только если Вы собираетесь торговать "в живую". В программе Wealth-Lab приказы AtStop действуют по аналогии с приказами StopLoss. Другими словами, stopPrice представляет собой цену активации приказа.

signalName - данный параметр является не обязательным для всех сигналов и отражается после выполнения стратегии в списке всех сделок в столбце "Signal Name". Наличие записей в этом столбце делает возможным сортировку всех сделок по однотипным сигналам.

Как: совершить теоретический трейд в программе Велс Лаб

Для того, чтобы проиллюстрировать как происходит торговля в программе Wealth-Lab, в качестве примера запустите следующий код стратегии на любом графике с любым таймфреймом.

Пример (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть)

```
1  protected override void Execute()
2  {
3      /* Только тест для иллюстрации */
4      ClearDebug();
5      PrintDebug("Positions.Count = " + Positions.Count);
6
7      // присвоить переменной bar значение номера 10-го с конца бара на графике
8
9      int bar = Bars.Count - 10;
10
11     BuyAtMarket(bar + 1); //заявка на покупку по рынку на открытии следующего бара.
12
13     PrintDebug("Positions.Count = " + Positions.Count);
14     PrintDebug("LastPosition.Active = " + LastPosition.Active);
15
16     // переходим на следующий бар и подаем торговый сигнал на выход из позиции типа
17     AtClose
18
19     bar++; //делаем текущим следующий бар
20
21     SellAtClose(bar + 1, LastPosition); //торговый сигнал на выход из длинной позиции
22     по цене закрытия следующего за текущим бара
23
24     PrintDebug("Positions.Count = " + Positions.Count);
25     PrintDebug("LastPosition.Active = " + LastPosition.Active);
26 }

```

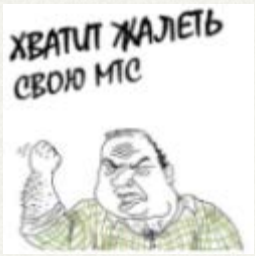
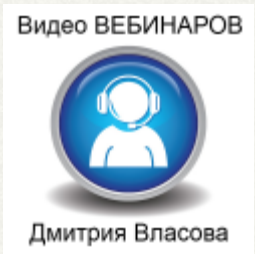
Код программы устанавливает в качестве текущего 10-й с конца бар и выполняет приказ BuyAtMarket на следующем от текущего баре.

Как Вы увидите дальше, использование для торгового сигнала следующего за текущим бара (bar + 1) является традиционным, так как именно такой подход наиболее точным образом воспроизводит способ торговли в реальных торгах. Как в случае с данным примером, так и вообще при бэктестинге стратегий сделки, совершаемые на исторических данных, конечно же являются теоретическими.



Пример торгового сигнала в Wealth-Lab

В результате выполнение кода осуществилась сделка. При этом вход в сделку здесь обозначен серой стрелочкой вверх, а выход из сделки обозначен зеленой стрелочкой вниз. Эта сделка найдет свое отражение не



МТС под микроскопом: WLMA System

Подробный разбор торговой системы WLMA System
3900 руб.

В корзину

Рубрики

Выберите рубрику

Обновления:

- Новые посты по Email
- Новые посты по RSS
- Комментарии по RSS
- Следим на Twitter

Метки

C# HowTo: Wealth-Lab
Видео Индикаторы МТС Торговые
роботы Юмор алготрейдинг
мероприятия обучение

Управление

Регистрация

Вход

RSS Записей

RSS Комментариев

WordPress.org

только на графике, но и на вкладке "Trades". Причем здесь Вы сможете увидеть все числовые характеристики сделки:

Chart Strategy Summary Editor Alerts Performance Trades Equity Curve Drawdown Profit Distribution MAE/MFE HeatMap Analysis Series														
Position	Symbol	Qty	Entry Date	Entry Price	Exit Date	Exit Price	Profit %	Profit \$	Bars Held	Profit per Bar	Entry Name	Exit Name	MAE %	MFE %
Long	GAZP	1	05.05.2012 10:00	162.30	05.05.2012 11:00	162.86	0.24	0.40p	2	0.20p			-0.78	0.65

Вкладка "Trades" в Wealth-Lab

Обратите внимание в примере на то, что свойство Positions.Count (количество позиций) увеличилось сразу же после создания позиции, которая сразу же становится активной (Last.Position.Active == true). После же закрытия позиции, эта позиция остается в списке позиций, но статус активности данной позиции меняется и принимает значение "false".

Данный пример в тонкостях объясняет, что программа Wealth-Lab является ориентированной на позиции.

Обратите особое внимание также на то, что в коде стратегии вообще не упоминается о размере позиции. Дело в том, что величиной позиции управляет не код торговой стратегии. Величина позиции определяется с помощью PosSizer (как установленных по умолчанию, так и написанных Вами самостоятельно). Управлять величиной позиции можно на панели данных. Для более полной информации об этом почитайте инструкцию к программе Велс Лаб.

Как: произвести сделку в определенную заранее дату применяя метод DateTimeToBar()

Сделки могут проводиться только на тех барах, которые существуют на графике. К счастью, функция DateTimeToBar() делает легкой задачу совершения сделки в определенную дату. Эта функция конвертирует значение DateTime в значение, содержащее номер бара. В результате на узнанном таким образом баре мы и проводим сделку.

Пример (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть) ?

```
1  protected override void Execute()
2  {
3      // Если возможно - проведите сделку 29 апреля 2012 года или сразу после (на
4      первом баре после) этой даты
5
6      DateTime dt = new DateTime(2012, 04, 29);
7
8      int bar = DateTimeToBar(dt, false);
9
10     if (bar > -1)
11     {
12         BuyAtMarket(bar, "Дата сделки: " + dt.ToString());
13     }
14     else
15     {
16         DrawLabel(PricePane, "Не существует баров на дату " + dt.ToString() + " или после
этой даты", Color.Red);
17     }
18 }
```

Обратите внимание, что даты 29 апреля на графике не существует, однако сигнал все равно произошел. Посмотрите повнимательнее код данного примера, чтобы понять - почему так случилось.



Сегодня мы с Вами только издалека посмотрели на то, какие возможности дает программа Велс Лаб для тестирования собственных торговых идей, для построения торговых стратегий. Внимательно изучив примеры сегодняшнего поста Вы сможете детально разобраться с тем, какие торговые сигналы существуют в программе Велс Лаб.

В следующий раз продолжим разбираться с программированием торговых стратегий в Wealth-Lab и рассмотрим важнейшую тему - **торговый цикл**. Не забывайте подписываться на новые сообщения по RSS.

Если Вам понравилась эта статья - подпишитесь на **новые СТАТЬИ по RSS** или получайте наши **новые СТАТЬИ на E-mail**:

Рекомендую почитать также:

- Программирование торговых стратегий в программе Wealth-Lab
- Программирование выходов, основанных на времени удержания позиции в Wealth-Lab
- Ловушки при построении торговых систем, связанные с проблемой подглядывания в будущее
- Использование обоснованного заглядывания в будущее при программировании торговых стратегий в Велс Лаб
- Анализ данных в Wealth-Lab с учетом разных таймфреймов



РУБРИКИ: PROGRAMMING TRADING STRATEGIES, WEALTHSCRIPT

Комментариев: 4

Программирование торговых стратегий в программе Wealth-Lab | Финансовая лаборатория пишет:

23 мая 2012 в 15:01

[...] Виды торговых сигналов в Wealth-Lab Главная Стабильность индикаторов в программе [...]

Ответить

Ответы на вопросы по программированию торговых стратегий в Wealth-Lab | Финансовая лаборатория пишет:

23 мая 2012 в 16:05

[...] 58. Какие параметры имеют торговые сигналы в Wealth-Lab? [...]

Ответить

Инструкция по программированию торговых стратегий в WealthLab | Финансовая лаборатория пишет:

20 июня 2012 в 12:20

[...] 2. Виды торговых сигналов в Wealth-Lab [...]

Ответить

Юрий пишет:

1 октября 2012 в 23:04



Дмитрий, здравствуйте!

Не могли бы Вы ответить мне на такой вопрос. На графике к разделу “Как произвести сделку в определенную заранее дату ...” последняя часовая свеча Газпрома за 28.04.2012 имеет время 18:00. И на других графиках и таблицах сделок у Вас первые часовые свечи с нашего рынка имеют время 10:00, а последние – 18:00, т.е. маркируются временем начала формирования часовой свечи. А у меня при построении графиков в WLD 6.3 все свечи маркируются временем окончания формирования свечи, т.е. первая часовая свеча нашего рынка имеет время 11:00, а последняя – 19:00. Подскажите, может в WLD есть где-то настройка, где можно устанавливать маркировку времени на свечах.

С уважением, Юрий.

Ответить

Оставить комментарий

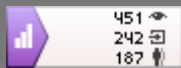
Имя (Обязательно)

Почта (Обязательно)

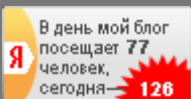
Вэб-сайт

Получать новые комментарии по электронной почте. Вы можете [подписаться](#) без комментирования.

© 2012, Финансовая лаборатория. Все права защищены.
Любое использование материалов допускается только с указанием активной ссылки на источник.



451 👁
242 📄
187 🗣



В день мой блог
посещает **77**
человек,
сегодня **126**



РЕЙТИНГ
757147 **532**
196



Подписка

- [RSS](#)
- [Twitter](#)
- [ВКонтакте](#)
- [facebook](#)

Сайт

- [Создать блог](#)
- [Содержание](#)
- [О блоге](#)
- [Карта](#)

Организация торгового цикла в программе Wealth-Lab и его переделка в скрининговую систему.

 **АВТОР** ДМИТРИЙ ВЛАСОВ  **МАЯ** - 24 - 2012  **КОММЕНТАРИЕВ:** 5

Торговая стратегия, которую мы с Вами создаем может выполнять две разные задачи. Она может использоваться для проведения бектестинга и оптимизации торговых параметров. При этом исследуется каждый бар имеющихся данных. Второе предназначение торговой стратегии - применение ее в торговле. При этом на основе уже сделанного выбора правил и параметров мы создаем программу, которая следит за тем, не наступило ли условия по которому мы должны совершить сделку, т.е. работает в режиме скрининга.

В момент, когда программа Wealth-Lab работает в режиме скрининга, мы заинтересованы только в том, чтобы выявить те финансовые инструменты, которые в текущий момент соответствуют определенным критериям. Обычно именно это является причиной того, что мы интересуемся сложившейся ситуацией только на последнем баре (Bars.Count-1) для отбора данного финансового инструмента из массы других.



Совсем другая ситуация складывается тогда, когда мы занимаемся бэктестингом стратегий. В этом случае мы хотим протестировать - какая ситуация складывается на каждом баре, как только значения наших индикаторов становятся валидными.

Как: организовать главный цикл при построении торговых стратегий в Wealth-Lab

Каждая торговая стратегия обязательно должна иметь главный цикл, который позволяет "пробежаться" по каждому бару на графике. Этот цикл создается специальным оператором и позволяет Вам "просмотреть" данные в режиме "бар за баром", точно также, как это происходит в реальной торговле. Часто этот главный цикл называют торговым циклом.

Пример ([как выполнить пример приведенного кода](#))...

Код примера на C# (кликните чтобы посмотреть)

```
1 | protected override void Execute()  
2 | {  
3 |     for(int bar = 50; bar < Bars.Count; bar++)  
4 |     {  
5 |         // Здесь программируются правила торговой стратегии  
6 |     }  
7 | }
```

В данном примере цикл стартует с 50-го бара данных, однако Вам следует выбирать точку старта торгового цикла исходя из периода тех индикаторов, которые Вы используете в торговой стратегии **с учетом их устойчивости**.

Для примера, если Вы используете при построении торговой стратегии 50-ти периодичную простую скользящую среднюю и RSI с периодом 14 бар, выбирайте самый длинный период среди двух

RSS товары



RSS новости



Twitter



Последние комментарии

макс: А где взять свечки с открытым интересом? на сайте финама нет



ring10: Добрый день Дмитрий Как сделать, чтобы открывался 1 велф лаб? Если открываешь под пользователем, т [...]



Дмитрий Власов: Буян, рад, что Вам пригодились материалы данной статьи!



Максим: Сам себе отвечу. protected override void Execute() { Bars rts = GetExternalSymbol("RTS", t [...]



Буян: Спасибо Дмитрий за статью! Мне удалось разработать хорошую рабочую торговую систему, используя возмо [...]



Последние посты



Бесплатный вебинар: 7 смертных грехов при оптимизации торговых систем

НОЯБРЯ - 5 - 2012



Отзывы участников курса до результата Wealth-Lab. С нуля до первой торговой системы за 5 дней.

НОЯБРЯ - 3 - 2012



Быстрее, Выше, Сильнее: Обновление Wealth-Lab до новой версии 6.4

ОКТАБРЯ - 23 - 2012



Курс до результата: Wealth-Lab – от нуля до первой торговой системы за 5 дней

ОКТАБРЯ - 23 - 2012



Обзорная экскурсия по достопримечательностям программы Wealth-Lab

ОКТАБРЯ - 20 - 2012

Берите

применяющихся индикаторов и выбирайте стартовый бар для запуска торгового цикла с номером 50 (а с учетом устойчивости индикаторов, еще больше). Главный цикл должен завершиться при обработке последнего бара на графике, т.е. при обработке бара с номером (Bars.Count - 1). Не забывайте учитывать при определении стартового пункта для главного цикла учесть **Стабильность индикаторов**.

Как: переделать код торговой стратегии в код скрининговой системы?

Быстрый ответ: Удалите оператор главного цикла и присвойте переменной с номером бара значение последнего бара на графике.

Фактически, Вы можете использовать построитель стратегий для создания простейших скрининговых систем. Для этого нужно сделать следующее:

Во-первых, запустите функцию "Convert to Code-Based Strategy" в обзоре правил. По умолчанию, построитель стратегий использует конструкцию для построения циклов с использованием оператора "for", как это показано в нижеприведенном примере.

Во-вторых, для того, чтобы превратить получившуюся стратегию в Скрининговую систему, Вам всего лишь потребуется удалить цикл, построенный с помощью оператора "for".

В-третьих, присвойте переменной, которая хранит номер бара значение последнего бара на графики.

Пример (**как выполнить пример приведенного кода**)...

Процесс переделки стратегии, использующейся для бектестинга в скрин показан в следующем примере.

Код примера на C# (кликните чтобы посмотреть)

```
1  /* Стратегия с использованием цикла с оператором "for" */
2  protected override void Execute()
3  {
4      for(int bar = 50; bar < Bars.Count; bar++) /* Поменяйте эту строку */
5      {
6          // Здесь торговые правила
7      }
8  }
9
10 /* Скрин: присвойте переменной bar номер последнего бара графика */
11 protected override void Execute()
12 {
13     int bar = Bars.Count - 1; /* TO THIS */
14     {
15         // Здесь торговые правила
16     }
17 }
```

Как видите переделать систему, которая помогает Вам тестировать торговые стратегии и подбирать оптимальные параметры в систему, которая отслеживает состояние рынка и помогает Вам вовремя входить в сделку очень легко. В следующий раз мы подробно рассмотрим **однопозиционные стратегии**. Не забывайте подписаться на обновления блога по RSS.

Если Вам понравилась эта статья - подпишитесь на **новые СТАТЬИ по RSS** или получайте наши **новые СТАТЬИ на E-mail**:

Рекомендую почитать также:

- Программирование выходов, основанных на времени удержания позиции в Wealth-Lab
- Ловушки при построении торговых систем, связанные с проблемой подглядывания в будущее
- Использование обоснованного заглядывания в будущее при программировании торговых стратегий в Велс Лаб
- Что такое алерты и как их использовать в Wealth-Lab
- Анализ данных в Wealth-Lab с учетом разных таймфреймов

 РУБРИКИ: PROGRAMMING TRADING STRATEGIES, WEALTHSCRIPT

Комментариев: 5

Программирование однопозиционных стратегий в программе Wealth-Lab | Финансовая лаборатория пишет:

25 мая 2012 в 13:26

[...] Организация торгового цикла в программе Wealth-Lab и его пе... [...]

Ответить

Виды торговых сигналов в Wealth-Lab | Финансовая лаборатория пишет:

25 мая 2012 в 14:13

[...] Организация торгового цикла в программе Wealth-Lab и его пе... Главная Программирование торговых стратегий в [...]

Ответить



Видео ВЕБИНАРОВ



Дмитрия Власова



МТС под микроскопом: WLMA System

Подробный разбор торговой системы WLMA System

3900 руб.

 В корзину

Рубрики

Выберите рубрику

Обновления:



Новые посты по Email



Новые посты по RSS



Комментарии по RSS



Следим на Twitter

Метки

C# HowTo: **Wealth-Lab**
Видео Индикаторы МТС Торговые роботы Юмор алготрейдинг мероприятия обучение

Управление

Регистрация

Вход

RSS Записей

RSS Комментариев

WordPress.org

Ответы на вопросы по программированию торговых стратегий в Wealth-Lab | Финансовая лаборатория

пишет:

25 мая 2012 в 14:18

[...] 62. Как организовать главный цикл при построении торговых... [...]

Ответить

Что такое алерты и как их использовать в Wealth-Lab | Финансовая лаборатория

пишет:

2 июня 2012 в 08:11

[...] [...]

Ответить

Инструкция по программированию торговых стратегий в WealthLab | Финансовая лаборатория

пишет:

20 июня 2012 в 12:23

[...] 3. Организация торгового цикла в программе Wealth-Lab и его пе... [...]

Ответить

Оставить комментарий

Имя (Обязательно)

Почта (Обязательно)

Вэб-сайт

Получать новые комментарии по электронной почте. Вы можете [подписаться](#) без комментирования.

© 2012, Финансовая лаборатория. Все права защищены.
 Любое использование материалов допускается только с указанием активной ссылки на источник.

451

242

187

В день мой блог посещает 77 человек, сегодня

126

Андрей

BY FEEDBURNER

mail.ru РЕЙТИНГ

757148

533196

Подписка

[RSS](#)
[Twitter](#)
[ВКонтакте](#)
[facebook](#)

Сайт

[Создать блог](#)
[Содержание](#)
[О блоге](#)
[Карта](#)

Программирование однопозиционных стратегий (SP Strategies) в программе Wealth-Lab

АВТОР ДМИТРИЙ ВЛАСОВ МАЯ - 25 - 2012 КОММЕНТАРИЕВ: 4

Когда мы приступаем к созданию и тестированию торговых стратегий сразу хочется создавать стратегии, в которых мы постепенно входим в позицию, постепенно выходим из нее и при этом полностью управляем каждой из набранных позиций. Скажу сразу, это вполне возможно осуществить в программе Wealth-Lab используя библиотеку WealthScript и язык программирования C#. Но лучше двигаться постепенно. Ведь не зря говорят - тише едешь - дальше будешь.

Предлагаю сегодня научиться создавать такие торговые стратегии, которые управляют только одной позицией. Такие стратегии в терминах Wealth-Lab называются SP Strategies (Single-Positions Strategies). Подобрать красиво звучащий перевод на русском языке довольно сложно. Давайте будет называть такие стратегии однопозиционными, либо описательно - стратегиями, управляющими одной позицией.



Стратегии, в которых торгуется только одна позиция в единицу времени называются однопозиционными стратегиями. Это обозначает, что стратегия может или удерживать только одну позицию (длинную или короткую) или же вообще не иметь позиций любого финансового инструмента.

Когда Вы проводите бэктестинг сразу нескольких финансовых инструментов, используя однопозиционную стратегию, общий результат такого исследования может содержать множество позиций одновременно, но только одну активную позицию для одного финансового инструмента.

Шаблон программного кода для однопозиционной стратегии

Шаблон кода стратегий, который встроен в редактор стратегий программы Wealth-Lab является шаблоном именно для однопозиционных стратегий. Этот шаблон кода состоит из **главного цикла** и логики, в соответствии с которой стратегия осуществляет вход в позицию и выход из позиции строго по одной транзакции на одном и том же баре. Другими словами, используя данный шаблон Вы автоматически пишете код, который открывает позицию либо закрывает позицию на определенном баре.

Пример (**как выполнить пример приведенного кода**)...

```
Код примера на C# (кликните чтобы посмотреть)
1 // Шаблон для однопозиционной стратегии
2 protected override void Execute()
3 {
4     for(int bar = 20; bar < Bars.Count; bar++)
5     {
6         if (IsLastPositionActive)
7         {
8             //Запрограммируйте здесь Ваши правила выхода из позиции
9         }
10     }
11     else
12 }
```

RSS товары

RSS новости

Twitter

Последние комментарии

- макс:** А где взять свечки с открытым интересом? на сайте финама нет
- ring10:** Добрый день Дмитрий Как сделать, чтобы открывался 1 велф лаб? Если открываешь под пользователем, т [...]
- Дмитрий Власов:** Буян, рад, что Вам пригодились материалы данной статьи!
- Максим:** Сам себе ответчу. protected override void Execute() { Bars rts = GetExternalSymbol("RTS", t [...]
- Буян:** Спасибо Дмитрий за статью! Мне удалось разработать хорошую рабочую торговую систему, используя возмо [...]

Последние посты

- Бесплатный вебинар: 7 смертных грехов при оптимизации торговых систем**
НОЯБРЯ - 5 - 2012
- Отзывы участников курса до результата Wealth-Lab. С нуля до первой торговой системы за 5 дней.**
НОЯБРЯ - 3 - 2012
- Быстрее, Выше, Сильнее: Обновление Wealth-Lab до новой версии 6.4**
ОКТАБРЯ - 23 - 2012
- Курс до результата: Wealth-Lab – от нуля до первой торговой системы за 5 дней**
ОКТАБРЯ - 23 - 2012
- Обзорная экскурсия по достопримечательностям программы Wealth-Lab**
ОКТАБРЯ - 20 - 2012

Берите

```
13 |         {  
14 |             //Запрограммируйте здесь Ваши правила входа в позицию  
15 |         }  
16 |     }  
17 | }
```

Что представляют из себя позиции (Positions) в программе Wealth-Lab

Ключевой идеей при программирования стратегий и обработки результатов тестирования является концепция списка позиций.

Когда код создает сделки, эти сделки добавляются в список позиций. Вы можете получить доступ к объектам Position из этого списка в любой момент времени во время исполнения стратегии. Именно это частенько приходится делать именно так для того, чтобы определить, какие позиции необходимо передать в качестве параметров в торговый сигнал на выход.

Свойство LastPosition удобно для доступа к самой последней позиции, которая была добавлена к позициям. И именно поэтому свойство LastPosition наиболее удобно для использования в однопозиционных стратегиях.

Для того, чтобы узнать подробности, обратитесь к разделу "Position Management functions" инструкции QuickRef (F11).

Как: проверить существует ли активная позиция

Поскольку в однопозиционных стратегиях мы работаем только с одной открытой (активной) позицией, мы можем с целью сокращения кода использовать логику LastPosition. Если позиция является активной, то мы можем быть совершенно уверены в том, что именно эта позиция является последней в списке всех позиций.

В шаблоне кода, который находится ниже, если последняя позиция (LastPosition) является активной, функция IsLastPositionActive() возвратит значение "true" и мы сразу же перейдем к той ветке кода, которая описывает правила выхода из позиции.

Если же функция IsLastPositionActive() возвратит значение "false", будем проверять - не должны ли мы открыть новую позицию в соответствии с нашими правилами входа в позицию.

Пример (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть) ?

```
1 | /* Данная стратегия демонстрирует идею построения однопозиционной стратегии*/  
2 | protected override void Execute()  
3 | {  
4 |     const int Period = 14; //период скользящей средней  
5 |  
6 |     PlotSeries(PricePane, SMA.Series(Close, Period), Color.Blue, LineStyle.Solid, 1);  
7 |     //отображаем на графике простую скользящую среднюю.  
8 |  
9 |     for(int bar = Period; bar < Bars.Count; bar++)  
10 |     {  
11 |         if (IsLastPositionActive) //если существует активная позиция (правила выхода  
12 | из позиции)  
13 |         {  
14 |             if( CrossUnder(bar, Close, SMA.Series(Close, Period)) ) //если цена  
15 | закрытия пересекает скользящую среднюю сверху вниз  
16 |  
17 |                 SellAtMarket(bar + 1, LastPosition); //продать по рынку на открытии  
18 | следующего бара  
19 |             }  
20 |             else //если нет открытой позиции (правила входа в позицию)  
21 |             {  
22 |                 if( CrossOver(bar, Close, SMA.Series(Close, Period)) ) //если цена закрытия  
23 | пересекает среднюю снизу вверх  
24 |  
25 |                     BuyAtLimit(bar + 1, Close[bar]); //купить на следующем баре с помощью  
26 | лимитного ордера по цене закрытия текущего.  
27 |                 }  
28 |             }  
29 |         }  
30 |     }  
31 | }
```

В результате выполнения такой стратегии на графике отобразится скользящая средняя, входы и выходы из позиции в виде стрелок. Если при выходе из позиции мы получили доход, то стрелочка обозначается зеленым цветом. Если получен убыток, то стрелка будет красной.



В этом примере Вы видите готовую однопозиционную стратегию, которая поможет продемонстрировать Вам идею такого подхода.

Как в Wealth-Lab осуществляется закрытие позиции

Как Вы можете увидеть в предыдущем примере, одним из способов выхода из открытой длинной позиции является функция SellAtMarket(). В этой функции имеется два параметра.



Видео ВЕБИНАРОВ



Дмитрия Власова





МТС под микроскопом:
WLMA System

Подробный разбор торговой системы WLMA System

3900 руб.

 В корзину

Рубрики

Выберите рубрику

Обновления:

-  Новые посты по Email
-  Новые посты по RSS
-  Комментарии по RSS
-  Следим на Twitter

Метки

C# HowTo: Wealth-Lab
Видео Индикаторы МТС Торговые роботы Юмор алготрейдинг мероприятия обучение

Управление

- Регистрация
- Вход
- RSS Записей
- RSS Комментарии
- WordPress.org

первый параметр - это номер бара, на котором позиция должна быть закрыта.
второй параметр - это ссылка на позицию, из которой мы хотим выйти.

Поскольку библиотека WealthScript может работать с торговые стратегиями, которые управляют множеством позиций, мы должны иметь возможность указать в правилах выхода из позиции какую именно позицию мы хотим закрыть. В тех торговых системах, которые управляют единственной позицией в единицу времени мы можем использовать LastPosition чтобы передать методу SellAtMarket() информацию о том, какую позицию мы хотим закрыть.

Сегодняшний пост вкратце рассказал Вам о том, что в программе Велс Лаб тестирование торговых стратегий и исследование результатов такого тестирования базируется на работе с позициями. Все позиции по мере их появления и закрытия группируются в списке позиций. Каждая позиция имеет свой статус, показывающий - активна ли она в текущий момент либо нет. Разобравшись с данным материалом Вы не будете испытывать трудностей при построении как простых, так и сложных торговых стратегий.

Обратите внимание также на то, что все примеры, которые приведены в сегодняшней статье полностью укладываются в шаблон кода однопозиционной стратегии. Эта схема очень проста для восприятия и внимательно посмотрев сегодняшние примеры Вы сможете самостоятельно (даже если ранее не имели опыта программирования) создавать на языке C# с использованием библиотеки WealthScript торговые стратегии.

В следующий раз разберем особенности **применения при программировании торговых стратегий ордеров AtStop и AtLimit**, при этом демонстрации будет приведен код торговой системы на прорыв уровней, которую создал Ван Тарп. Чтобы не пропустить новые статьи - подписывайтесь на обновления по RSS.

Если Вам понравилась эта статья - подпишитесь на **новые СТАТЬИ по RSS** или получайте наши **новые СТАТЬИ на E-mail**:

Рекомендую почитать также:

- **Программирование выходов, основанных на времени удержания позиции в Wealth-Lab**
- **Создание многопозиционных стратегий в Wealth-Lab**
- **Анализ данных в Wealth-Lab с учетом разных таймфреймов**
- **Использование при программировании торговых стратегий торговых приказов AtStop и AtLimit**
- **Что такое алерты и как их использовать в Wealth-Lab**

 РУБРИКИ: **PROGRAMMING TRADING STRATEGIES, WEALTHSCRIPT, ПРИМЕРЫ MTC**

Комментариев: 4

Организация торгового цикла в программе Wealth-Lab и его переделка в скрининговую систему | Финансовая лаборатория пишет:

25 мая 2012 в 14:23

[...] [...]

Ответить

Ответы на вопросы по программированию торговых стратегий в Wealth-Lab | Финансовая лаборатория пишет:

25 мая 2012 в 14:49

[...] 64. Где найти шаблон программного кода для однопозиционно... [...]

Ответить

Торговая система HighLowLong - шаблон кода и оптимизация. | Финансовая лаборатория пишет:

26 мая 2012 в 13:02

[...] иметь не более одной позиции. О том, как выглядит шаблон однопозиционной торговой стратегия я уже [...]

Ответить

Создание многопозиционных стратегий в Wealth-Lab | Финансовая лаборатория пишет:

1 июня 2012 в 18:17

[...] стратегию – ему не всегда хватает возможностей шаблона однопозиционной стратегии. Ведь действительно иногда хочется использовать [...]

Ответить

Имя (Обязательно)

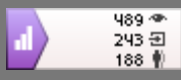
Почта (Обязательно)

Вэб-сайт

Получать новые комментарии по электронной почте. Вы можете **подписаться** без комментирования.

© 2012, Финансовая лаборатория. Все права защищены.

Любое использование материалов допускается только с указанием активной ссылки на источник.



489
243
188



В день мой блог
посещает **77**
человек,
сегодня **126**



РЕЙТИНГ
757149534
196



BY FEEDBURNER

Подписка

- [RSS](#)
- [Twitter](#)
- [ВКонтакте](#)
- [facebook](#)

Сайт

- [Создать блог](#)
- [Содержание](#)
- [О блоге](#)
- [Карта](#)

Использование при программировании торговых стратегий торговых приказов AtStop и AtLimit

 **АВТОР ДМИТРИЙ ВЛАСОВ**  **МАЯ - 26 - 2012**  **КОММЕНТАРИЕВ: 2**

При создании торговых стратегий мы должны входить в рынок либо выходить из позиции используя не только простейшие торговые приказы "по рынку", либо "в момент закрытия", но и более продвинутые торговые ордера, которые выполняются либо не выполняются в зависимости от того, каких уровней достигла цена. В программе Wealth-Lab, как и в реальной торговле такие приказы могут моделировать AtStop и AtLimit ордера.



Подумайте о том, как происходит процесс размещения стоп и лимитных ордеров когда Вы торгуете на реальном счете. После того, как произошло обновление данных на Ваших графиках, Вы определяете ту цену, по которой Вы захотите войти в позицию. И только после этого, на следующем баре Вы отдадите приказ о размещении ордеров на рынке.

Когда Вы программируете стратегии в программе Wealth-Lab процесс размещения ордеров протекает точно также, как и в реальной жизни:

```
получаем данные на текущем баре;
рассчитываем цены стоп и лимитных ордеров;
размещаем соответствующие ордера на следующем баре, который по отношению к текущему бару имеет номер (bar + 1)
```

Как: разместить лимитный или стоп приказ на покупку при программировании стратегий в Wealth-Lab

Для того, чтобы смоделировать стоп или лимитные приказы на покупку, используйте специальные методы BuyAtStop() и BuyAtLimit(). Аналогично Вы сможете размещать такие же ордера для моделирования процесса закрытия длинной позиции, короткой продажи или выхода из короткой позиции.

Точно также, как это происходит и в реальной торговле, существует вероятность того, что эти ордера могут вообще не исполниться на указанном баре. Поэтому методы BuyAt и ShortAt возвращают одно из двух значений:

```
Position (ссылку на позицию) - извещая нас о том, что ордер был исполнен и у нас появилась новая позиция
null - который показывает нам, что сделка по данному ордеру не состоялась.
```

Следует отметить, что методы SellAt и CoverAt возвращают значения булевого типа, которые отвечают на вопрос - "удалось ли закрыть позицию?":

```
true - если удалось закрыть позицию
false - если позицию закрыть не удалось.
```

RSS товары



RSS новости



Twitter



Последние комментарии

макс: А где взять свечки с открытым интересом? на сайте финама нет



ring10: Добрый день Дмитрий Как сделать, чтобы открывался 1 велф лаб? Если открываешь под пользователем, т [...]



Дмитрий Власов: Буян, рад, что Вам пригодились материалы данной статьи!



Максим: Сам себе отвечу. protected override void Execute() { Bars rts = GetExternalSymbol("RTS", t [...]



Буян: Спасибо Дмитрий за статью! Мне удалось разработать хорошую рабочую торговую систему, используя возмо [...]



Последние посты



Бесплатный вебинар: 7 смертных грехов при оптимизации торговых систем

НОЯБРЯ - 5 - 2012



Отзывы участников курса до результата Wealth-Lab. С нуля до первой торговой системы за 5 дней.

НОЯБРЯ - 3 - 2012



Быстрее, Выше, Сильнее: Обновление Wealth-Lab до новой версии 6.4

ОКТАБРЯ - 23 - 2012



Курс до результата: Wealth-Lab – от нуля до первой торговой системы за 5 дней

ОКТАБРЯ - 23 - 2012



Обзорная экскурсия по достопримечательностям программы Wealth-Lab

ОКТАБРЯ - 20 - 2012

Берите

Если возвратиться немного назад - к примеру, который тестирует является ли позиция открытой, при описании правил входа в позицию используется приказ BuyAtLimit(), который располагается на следующем баре с номером (bar + 1). Однако такой приказ размещается только в том случае, если цена закрытия пересекает снизу вверх скользящую среднюю по ценам закрытия.

Конечно же, в данной стратегии можно было бы использовать приказ BuyAtMarket() для гарантированного входа в позицию после того, как произойдет пересечения цены закрытия и скользящей средней. Однако лимитный ордер точно указывает максимальную цену, которую Вы готовы заплатить. Эта цена определяется ценой закрытия текущего бара.

Пример (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть) ?

```
1 else
2 {
3     if( CrossOver(bar, Close, SMA.Series(Close, Period)) )
4
5         BuyAtLimit(bar + 1, Close[bar]);
6
7 }
```

Как: создать отложенные (GTC) ордера (приказы, действующие до отмены) при моделировании торговли в Велс Лаб

В реальности отложенные ордера создаются путем программного манипулирования стоп и лимитными ордерами. Wealth-Lab размещает, меняет, или же продолжает удерживать ордера на открытие позиции на выбранных Вами барах. В результате отложенные ордера в торговых стратегиях являются стоп и лимитными приказами, которые постоянно повторяются до тех пор, пока:

- Приказ исполнится;
- Приказ будет логически передвинут;

Другими словами, отложенные (GTC) ордера отменяются без вызова торгового сигнала.

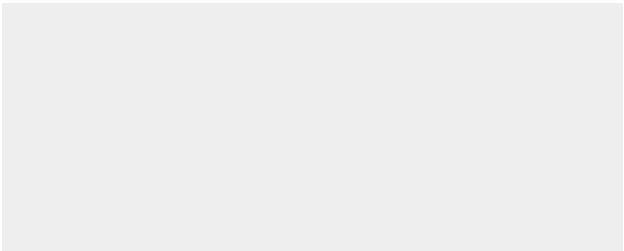
Следующий пример является упрощенной версией стратегии прорыва канала Ван Тарпа. Вместо того, чтобы отображать на графике ряды данных с уровнями прорыва, здесь используется метод PlotStops(). Этот метод позволяет нам визуальнo идентифицировать те бары, где ордера размещаются, перемещаются или отменяются.

Пример (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть) ?

```
1 /*Упрощенная стратегия прорыва уровней ВанТарпа*/
2 protected override void Execute()
3 {
4     int longPer = 55;
5     int shortPer = 21;
6
7     DataSeries H55 = Highest.Series(High, longPer);
8     DataSeries L55 = Lowest.Series(Low, longPer);
9     DataSeries H21 = Highest.Series(High, shortPer);
10    DataSeries L21 = Lowest.Series(Low, shortPer);
11
12    PlotStops();
13
14    for (int bar = longPer; bar < Bars.Count; bar++)
15    {
16        if (IsLastPositionActive)
17        {
18            Position Pos = LastPosition;
19
20            if (Pos.PositionType == PositionType.Long)
21
22                SellAtStop(bar + 1, LastPosition, L21[bar]);
23
24            else
25
26                CoverAtStop(bar + 1, LastPosition, H21[bar]);
27        }
28
29        else
30        {
31            RiskStopLevel = L21[bar];
32
33            if (BuyAtStop(bar + 1, H55[bar]) == null)
34            {
35                RiskStopLevel = H21[bar];
36                ShortAtStop(bar + 1, L55[bar]);
37            }
38        }
39    }
40 }
41 }
```

Как Вы видите, этот код является полноценной торговой системой, правда здесь пока не используется оптимизация и все предельно просто. В результате выполнения этой торговой системы мы увидим на графике такую картину:





Видео ВЕБИНАРОВ



Дмитрия Власова





МТС под микроскопом:

WLMA System

Подробный разбор торговой системы WLMA System

3900 руб.

 В корзину

Рубрики

Выберите рубрику

Обновления:

 Новые посты по Email

 Новые посты по RSS

 Комментарии по RSS

 Следим на Twitter

Метки

C# HowTo: Wealth-Lab

Видео Индикаторы МТС Торговые роботы Юмор алготрейдинг мероприятия обучение

Управление

- Регистрация
- Вход
- RSS Записей
- RSS Комментариив
- WordPress.org



Демонстрация прорывной системы Ван Тарпа

Обратите внимание на то, как работает метод PlotStops(). Там, где позиции не существует - стоп-приказы отображаются и сверху и снизу от текущей цены. Где есть позиция, там стоп приказ отображается только с одной стороны, выполняя функцию ограничения убытков.

Несмотря на то, что эта система разработана достаточно давно, для американского рынка акций и вообще не оптимизировалась, как ни странно, применяя ее к акциям Газпрома и проводя тест за последние 3 года на часовых графиках мы получим довольно красивую картинку эквити.

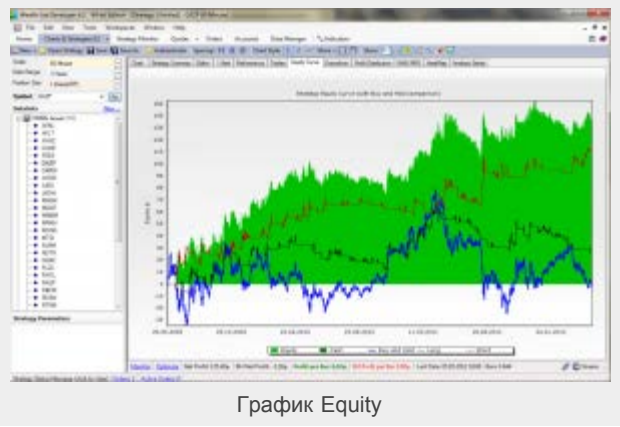


График Equity

Эта картинка свидетельствует о получении прибыли (даже с учетом комиссии). Так что берите на заметку - здесь есть над чем поработать. Можно на этой основе создать полноценную торговую стратегию.

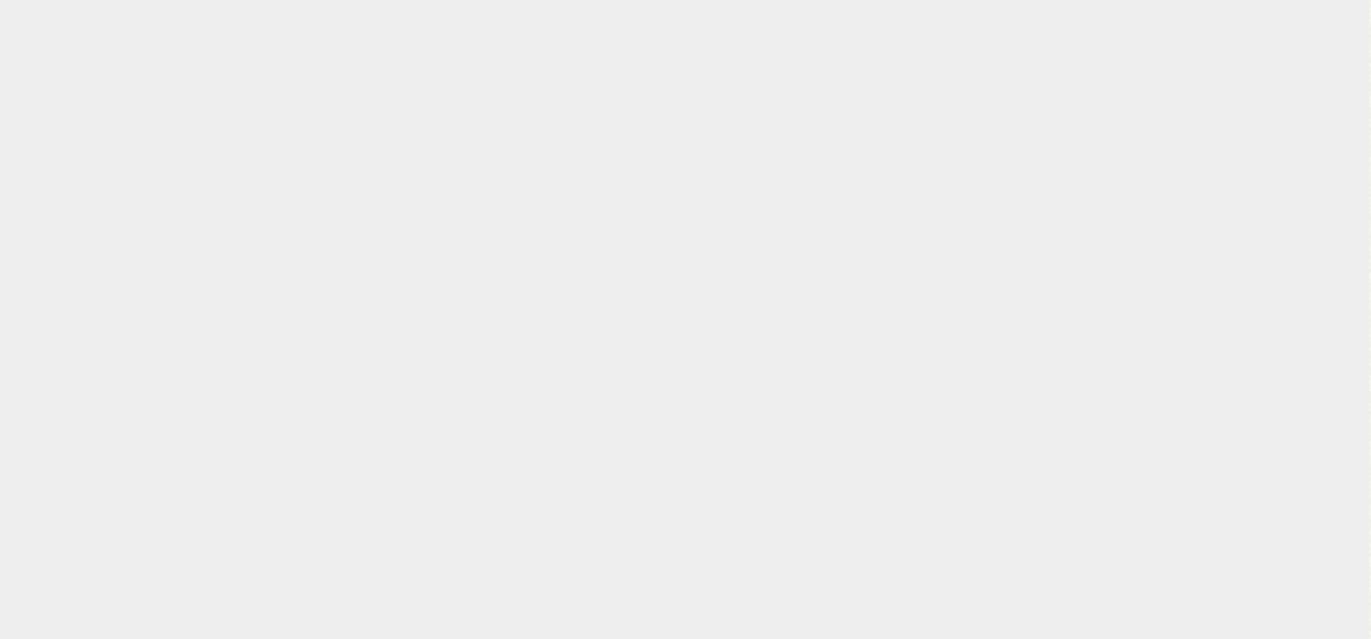
Как: отображать на графике стоп-приказы

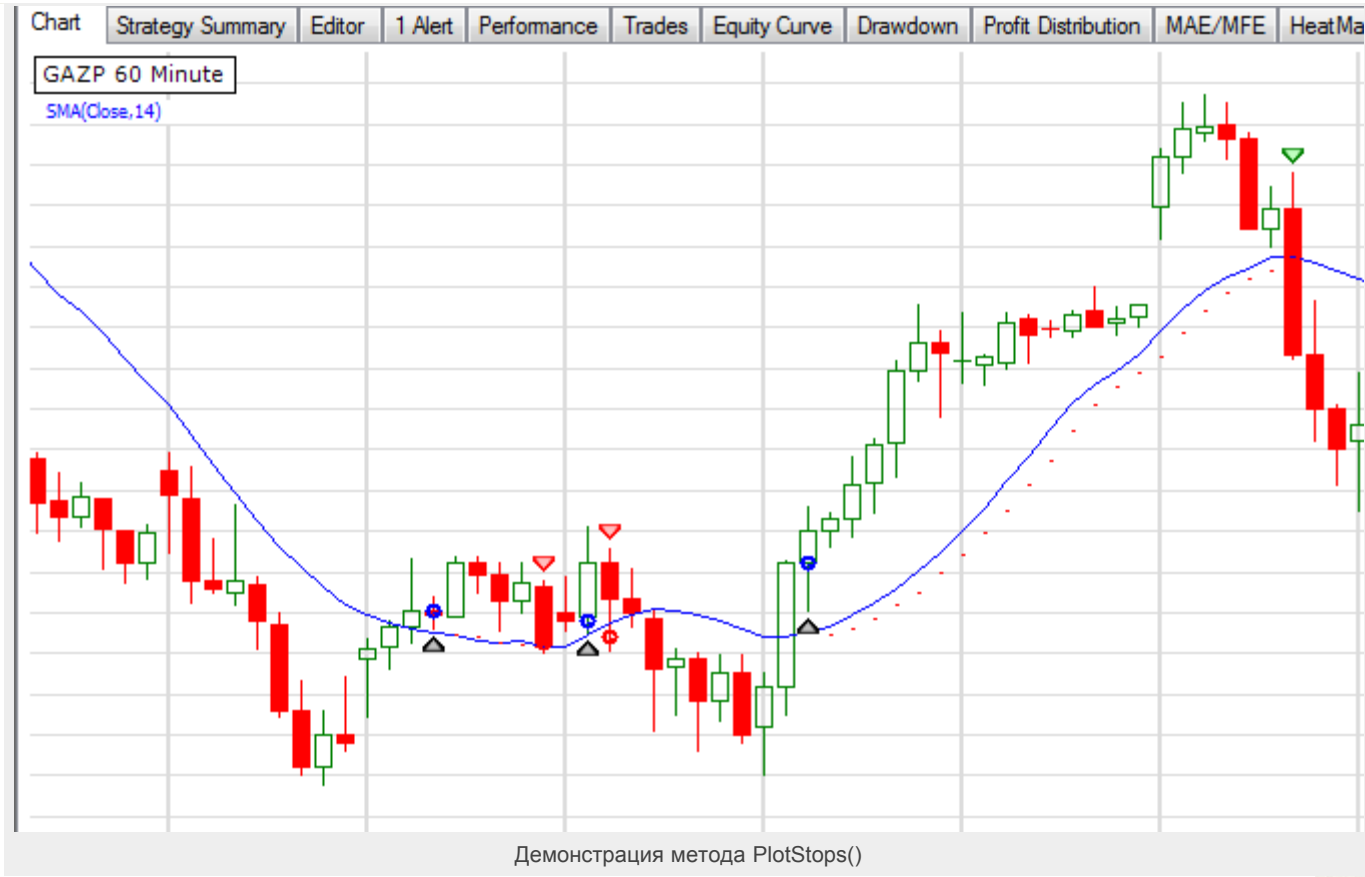
Несмотря на то, что предыдущий пример содержит прекрасную демонстрацию отображения на графике стоп ордеров с помощью метода PlotStops(), давайте взглянем на это еще раз с другой стороны. Рассмотрим как можно использовать SMA (простую скользящую среднюю) для того, чтобы устанавливать стоп-приказы.

Пример **(как выполнить пример приведенного кода)**...

```
Код примера на C# (кликните чтобы посмотреть)
1 | protected override void Execute()
2 | {
3 |     const int Period = 14;
4 |
5 |     PlotSeries(PricePane, SMA.Series(Close, Period), Color.Blue, LineStyle.Solid, 1);
6 |
7 |     PlotStops();
8 |
9 |     for(int bar = Period; bar < Bars.Count; bar++)
10 |    {
11 |        if (IsLastPositionActive)
12 |        {
13 |            SellAtStop(bar + 1, LastPosition, SMA.Series(Close, Period)[bar]);
14 |        }
15 |
16 |        else
17 |
18 |        {
19 |            if( CrossOver(bar, Close, SMA.Series(Close, Period)) )
20 |
21 |                BuyAtLimit(bar + 1, Close[bar]);
22 |        }
23 |    }
24 | }
```

В результате выполнения примера Вы увидите такой график:





Когда Вы будете анализировать результаты выполнения кода, наверняка обратите внимание на то, что стоп-приказы отображаются на графике не точно по линии скользящей средней. Это происходит потому, что метод PlotStops() показывает Вам бар и цену на которой стоп (или лимитный) ордер являются активными. Скользящая средняя рассчитывается и отображается на графике на текущем баре, но рассчитанное значение используется в качестве цены стоп-приказа только на следующем баре, т.е. на баре с номером (bar + 1).

Следующий пост расскажет Вам о том, [как планировать выходы, зависящие от времени удержания позиции](#). Не забывайте подписываться на новые посты по RSS.

Если Вам понравилась эта статья - подпишитесь на [новые СТАТЬИ по RSS](#) или получайте наши [новые СТАТЬИ на E-mail](#):

Рекомендую почитать также:

- [Программирование однопозиционных стратегий \(SP Strategies\) в программе Wealth-Lab](#)
- [Программирование выходов, основанных на времени удержания позиции в Wealth-Lab](#)
- [Создание многопозиционных стратегий в Wealth-Lab](#)
- [Что такое алерты и как их использовать в Wealth-Lab](#)
- [Анализ данных в Wealth-Lab с учетом разных таймфреймов](#)

 РУБРИКИ: [PROGRAMMING TRADING STRATEGIES](#), [WEALTHSCRIPT](#)

Комментариев: 2

Ответы на вопросы по программированию торговых стратегий в Wealth-Lab | Финансовая лаборатория пишет:

26 мая 2012 в 10:59

[...] 68. Как разместить лимитный или стоп приказ на покупку при... [...]

Ответить

Программирование однопозиционных стратегий (SP Strategies) в программе Wealth-Lab | Финансовая лаборатория пишет:

28 мая 2012 в 11:36

[...] Использование при программировании торговых стратеги... Главная Организация торгового цикла в [...]

Ответить

Оставить комментарий

Имя (Обязательно)

Почта (Обязательно)

Вэб-сайт

Получать новые комментарии по электронной почте. Вы можете [подписаться](#) без комментирования.

© 2012, Финансовая лаборатория. Все права защищены.
Любое использование материалов допускается только с указанием активной ссылки на источник.



489
243
188



В день мой блог
посещает **77**
человек,
сегодня **126**



mail.ru РЕЙТИНГ
757149 534
196



BY FEEDBURNER

Подписка

- [RSS](#)
- [Twitter](#)
- [ВКонтакте](#)
- [facebook](#)

Сайт

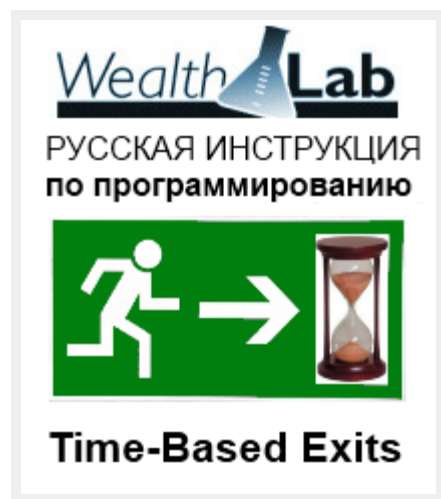
- [Создать блог](#)
- [Содержание](#)
- [О блоге](#)
- [Карта](#)

Программирование выходов, основанных на времени удержания позиции в Wealth-Lab

 **АВТОР** ДМИТРИЙ ВЛАСОВ  **МАЯ** - 28 - 2012  **КОММЕНТАРИЕВ:** 8

Очень часто при программировании торговых стратегий Вы будете сталкиваться с ситуациями, когда нужно выходить из позиции не на основе анализа движения цены, а просто по прошествии какого-то количества времени. Особенно популярны тесты торговых стратегий с применением выхода такого рода для определения дает-ли статистически значимое преимущество определенные входы.

Исходя из того, что такая схема выходов будет востребована - давайте сегодня более подробно изучим как можно конструировать выходы по времени.



Как: выполнить выход из позиции ориентируясь на время?

Если говорить в целом, то стратегии с выходами ориентирующимися на время делаются с помощью ордеров "по рынку", которые исполняются сразу же после удержания позиции в течение определенного количества баров. В однопозиционных стратегиях главным паттерном для выходов по времени является следующий:

Пример ([как выполнить пример приведенного кода](#))...

Код примера на C# (кликните чтобы посмотреть) 

```
1 int numberOfBars = 5; // определяем максимальное количество баров удержания позиции
2
3 Position p = LastPosition;
4
5 if( bar + 1 - p.EntryBar >= numberOfBars )
6
7     ExitAtMarket(bar + 1, p, "Выход по времени");
```

Здесь представлен код стратегии в которой применяется выход по времени. Эта стратегия покупает позицию по лимитному ордеру, который устанавливается на 3% ниже последней цены последнего закрытия и закрывает позицию после удержания в течение 5-ти баров.

Пример ([как выполнить пример приведенного кода](#))...

Код примера на C# (кликните чтобы посмотреть) 

```
1 protected override void Execute()
2 {
3     int numberOfBars = 5;
4
5     for(int bar = 1; bar < Bars.Count; bar++)
6     {
7         if (IsLastPositionActive)
8         {
9             Position p = LastPosition;
10            if( bar + 1 - p.EntryBar >= numberOfBars )
11                ExitAtMarket(bar + 1, p);
12        }
13    }
14    else
15
```

RSS товары



RSS новости



Twitter



Последние комментарии

макс: А где взять свечки с открытым интересом? на сайте финама нет



ring10: Добрый день Дмитрий Как сделать, чтобы открывался 1 велф лаб? Если открываешь под пользователем, т [...]



Дмитрий Власов: Буян, рад, что Вам пригодились материалы данной статьи!



Максим: Сам себе отвечу. protected override void Execute() { Bars rts = GetExternalSymbol("RTS", t [...]



Буян: Спасибо Дмитрий за статью! Мне удалось разработать хорошую рабочую торговую систему, используя возмо [...]



Последние посты



Бесплатный вебинар: 7 смертных грехов при оптимизации торговых систем

НОЯБРЯ - 5 - 2012



Отзывы участников курса до результата Wealth-Lab. С нуля до первой торговой системы за 5 дней.

НОЯБРЯ - 3 - 2012



Быстрее, Выше, Сильнее: Обновление Wealth-Lab до новой версии 6.4

ОКТАБРЯ - 23 - 2012



Курс до результата: Wealth-Lab – от нуля до первой торговой системы за 5 дней

ОКТАБРЯ - 23 - 2012



Обзорная экскурсия по достопримечательностям программы Wealth-Lab

ОКТАБРЯ - 20 - 2012

Берите

```
16         BuyAtLimit(bar + 1, Close[bar]* 0.97);
17     }
18 }
```

Вот так выглядит график стратегии:



Как: Не выполнять выходы, основанные на времени

Вам может показаться, что можно упростить код примера, приведенного чуть выше с помощью немедленного определения номера того бара, на котором должен происходить выход из позиции. Однако, если Вы используете конструкцию (Bar + n) в том случае, если n>1 для торговых сигналов, то Велс Лаб действительно выполняет эти сделки немедленно во время бектестинга. В самом деле, эти сделки будут выполнены только в будущем, но статус позиции поменяется на "активна" слишком рано. Очень часто это становится причиной того, что код, построенный по такой логике, начинает выполнять входы или выходы слишком рано (т.к. обычно такие действия базируются на статусе позиции) и именно из-за этого это является путем, который не совпадает с реальной торговлей.

Пример (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть)

```
1  protected override void Execute()
2  {
3      int numberOfBars = 5;
4      for(int bar = 1; bar < Bars.Count; bar++)
5      {
6
7          if (IsLastPositionActive)
8
9              // ЭТО НЕПРАВИЛЬНЫЙ СПОСОБ КОДИРОВАНИЯ ВЫХОДОВ ПО ВРЕМЕНИ (ИЛИ ЛЮБЫХ ДРУГИХ
10             ТОРГОВЫХ СИГНАЛОВ)
11
12             ExitAtMarket(bar + numberOfBars, LastPosition);
13
14         else
15
16             BuyAtLimit(bar + 1, Close[bar]* 0.97);
17     }
```

Прописывая в коде торговой стратегии правило, в соответствии с которым торговый сигнал должен выполняться на баре (bar + n), если n > 1 помните о том, что это всегда будет ошибочным решением как для бектестинга, так и для торговли потому что:

1. Wealth-Lab Developer выполняет торговые сигналы при бэктестинге немедленно. Выход из позиции на баре (bar + n), если n > 1 является причиной преждевременной смены статуса позиции из "active" на "not active". Это будет иметь разрушительный эффект для логики стратегии (смотри тему "Ловушки").
2. Невозможно торговать стратегии, которые генерируют Алерты на барах с номером (bar + n), где n > 1. В зависимости от стратегии результатом будет одна из двух ошибок: или алерт появится на n-1 бар раньше чем следовало, или же Вы будете получать алерты на выход из позиции для каждого из дней n.

На этом сегодняшний небольшой пост закончен. Важно навсегда запомнить, что при планировании выходов через определенное время необходимо использовать корректный код и не пользоваться теми конструкциями, которые вызывают торговые сигналы не на следующем баре, а на барах, которые находятся в будущем. В следующий раз мы коснемся темы, которая будет очень важна, чтобы не попасть впросак создавая свою торговую стратегию и не слить весь отведенный на торговлю депозит. Речь пойдет о **ловушках при построения торговых стратегий**. Поэтому, чтобы не попасть в такие ловушки не забудьте подписаться на новые статьи по RSS.

Если Вам понравилась эта статья - подпишитесь на **новые СТАТЬИ по RSS** или получайте наши **новые СТАТЬИ на E-mail**:
Рекомендую почитать также:

- Программирование однопозиционных стратегий (SP Strategies) в программе Wealth-Lab
- Создание многопозиционных стратегий в Wealth-Lab
- Анализ данных в Wealth-Lab с учетом разных таймфреймов
- Программирование торговых стратегий в программе Wealth-Lab
- Использование обоснованного заглядывания в будущее при программировании торговых стратегий в Велс Лаб

РУБРИКИ: PROGRAMMING TRADING STRATEGIES, WEALTHSCRIPT, ПРИМЕРЫ МТС



Видео ВЕБИНАРОВ



Дмитрия Власова

E-mail курсы



ХВАТИТ ЖАЛЕТЬ СВОЮ МТС





МТС под микроскопом:

WLMA System

Подробный разбор торговой системы WLMA System

3900 руб.

В корзину

Рубрики

Выберите рубрику

Обновления:

 Новые посты по Email

 Новые посты по RSS

 Комментарии по RSS

 Следим на Twitter

Метки

C# HowTo: **Wealth-Lab**
Видео Индикаторы МТС Торговые роботы Юмор алготрейдинг мероприятия обучение

Управление

Регистрация

Вход

RSS Записей

RSS Комментариев

WordPress.org

Использование при программировании торговых стратегий торговых приказов AtStop и AtLimit | Финансовая лаборатория пишет:

28 мая 2012 в 11:35

[...] пост расскажет Вам о том, как планировать выходы, зависящие от времени удержани.... Не забывайте подписываться на новые посты по RSS. Если [...]

Ответить

Игорь пишет:

29 мая 2012 в 11:58



“Особенно популярны тесты торговых стратегий с применением выхода такого рода для определения дает-ли статистически значимое преимущество определенные входы.” Дмитрий, а можно немного подробнее рассказать о подобного рода тестах. Тема весьма интересна, но, я не совсем понял, как можно определить хорошо вход или плох, на основании выходов по времени. Спасибо

Ответить

Дмитрий Власов пишет:

29 мая 2012 в 15:28



Игорь, приветствую. Такие тесты очень подробно описаны в моей любимой книге по трейдингу, которую написал Л. Вильямс “Долгосрочные секреты краткосрочной торговли”.

Смысл тестов в том, что он пытается найти хотя бы минимальное статистическое преимущество входов для этого делает вход, основанный на каком-либо предположении, а затем делает выход после определенного времени удержания позиции. После чего рассчитывает математическое ожидание. Если это МО больше 50%, то такой вход используется для дальнейшей работы с ним, т.е. в качестве одного из инструмента, помогающего делать торговую стратегию.

Ответить

Игорь пишет:

1 июня 2012 в 09:43



Спасибо, Дмитрий!

Ответить

Игорь пишет:

14 июня 2012 в 11:22



Приветствую Дмитрий, купил книгу Л. Вильямс “Долгосрочные секреты краткосрочной торговли”, решил, что ее стоит обязательно прочитать, раз уж умные люди считают ее своей любимой 😊 Не подскажешь в какой главе Вильямс разжевывает про тесты выходов по времени? Буду очень признателен. Спасибо

Ответить

Дмитрий Власов пишет:

14 июня 2012 в 12:25



Игорь, читай подряд всю книгу – это очень интересно. А тесты выходов по времени он в главе №6 разжевывает “Истина где-то рядом” в разделе “Рынок-это не игра в орлянку”. У меня в издании на странице 98.

Ответить

Игорь пишет:

14 июня 2012 в 16:57



Действительно, классная книга, буду подряд читать, просто думал гляну быстренько, чтобы врубится с чем это едят, ан нет, чтиво захватывает, так что мой тебе низкий поклон за подсказку 😊 И за видео уроки по квику, спасибо большущее, а то, я тоже 7 лет на смарттрейде просидел и квик сначала не понравился, а оказывается, я просто его не умел готовить 😊))

Ловушки при построении торговых систем, связанные с проблемой "подглядывания" в будущее |
Финансовая лаборатория пишет:

30 мая 2012 в 14:31

[...] Программирование выходов, основанных на времени удерж... [...]

Ответить

Оставить комментарий

Имя (Обязательно)

Почта (Обязательно)

Вэб-сайт

Получать новые комментарии по электронной почте. Вы можете [подписаться](#) без комментирования.

© 2012, Финансовая лаборатория. Все права защищены.
Любое использование материалов допускается только с указанием активной ссылки на источник.



489
243
188



В день мой блог
посещает **77**
человек,
сегодня — **126**



mail.ru РЕЙТИНГ
757151 536
196



BY FEEDBURNER

Подписка

- [RSS](#)
- [Twitter](#)
- [ВКонтакте](#)
- [facebook](#)

Сайт

- [Создать блог](#)
- [Содержание](#)
- [О блоге](#)
- [Карта](#)

Ловушки при построении торговых систем, связанные с проблемой подглядывания в будущее

 **АВТОР** ДМИТРИЙ ВЛАСОВ  **МАЯ** - 29 - 2012  **КОММЕНТАРИЕВ:** 3

Иногда случаются моменты, когда сердце системного трейдера начинает учащенно биться от предвкушения невероятных прибылей. Кажется, что вот он "святой грааль", найден и находится у тебя в руках. Эквити монотонно повышается в неведомые дали, а если использовать торговлю на весь капитал, то уходит вверх по экспоненте.

К несчастью, если Вы видите такую картину, то скорее всего Вы столкнулись с заглядыванием в будущее. Как ни печально, хотя бы однажды любой трейдер, тестирующий торговую стратегию попадает в такую ловушку. Чтобы заранее знать о подстерегающих Вас опасностях - обязательно внимательно прочитайте сегодняшний пост.



Являясь всего лишь средством программирования, Wealth-Lab не может искусственно ставить для Вас барьеры, указывая - как Вы должны строить программный код. При построении программного кода торговой стратегии возможны всевозможные подводные камни, такие, как заглядывание в будущее - специальное или нечаянное. В своей книге "Трейдинг - Ваш путь к финансовой свободе" Ван Тарп назвал такое заглядывание в будущее постдиктивными ошибками.

Хотя некоторые действительно используют выходы для "заглядывания в будущее", обычно такого заглядывания Вы хотите избежать. Непроизвольное подглядывание обычно не имеет отношения к логике построения кода. Внимательно изучив следующие подразделы, Вы научитесь избегать такие часто встречающиеся ловушки.

При тестировании не получайте доступа к данным из будущего

Убедитесь в том, что Ваша стратегия не извлекает преимущество из той информации, которую Вы никогда не сможете получить в реальном мире. Такого типа ошибки можно легко избежать если всегда получать доступ к значению ряда данных либо индикатора в главном цикле только на текущем (либо предыдущем) баре, не пытаясь получить доступ к данным на баре с номером (bar + 1) либо с любых других баров, которые находятся в будущем.

Чаще хитроумные ошибки совершаются следующим образом: перед главным циклом извлекается некоторая статистики или данные от индикатора, которые имеются уже в конце ряда данных. После чего эта информация используется при входе в главный торговый цикл. В следующем примере мы получаем стандартное отклонение проанализировав сразу весь ряд данных с ценами закрытия, но потом (естественно, это некорректно) используем это значение в торговой стратегии.

Пример ([как выполнить пример приведенного кода](#))...

Код примера на C# (кликните чтобы посмотреть)

```
1 //Иллюстрация того, ЧТО ДЕЛАТЬ НЕЛЬЗЯ!!!
2
```

RSS товары



RSS новости



Twitter



Последние комментарии

макс: А где взять свечки с открытым интересом? на сайте финама нет



ring10: Добрый день Дмитрий Как сделать, чтобы открывался 1 велф лаб? Если открываешь под пользователем, т [...]



Дмитрий Власов: Буян, рад, что Вам пригодились материалы данной статьи!



Максим: Сам себе отвечу. protected override void Execute() { Bars rts = GetExternalSymbol("RTS", t [...]



Буян: Спасибо Дмитрий за статью! Мне удалось разработать хорошую рабочую торговую систему, используя возмо [...]



Последние посты



Бесплатный вебинар: 7 смертных грехов при оптимизации торговых систем

НОЯБРЯ - 5 - 2012



Отзывы участников курса до результата Wealth-Lab. С нуля до первой торговой системы за 5

дней.

НОЯБРЯ - 3 - 2012



Быстрее, Выше, Сильнее: Обновление Wealth-Lab до новой версии 6.4

ОКТАБРЯ - 23 - 2012



Курс до результата: Wealth-Lab – от нуля до первой торговой системы за 5 дней

ОКТАБРЯ - 23 - 2012



Обзорная экскурсия по достопримечательностям программы Wealth-Lab

ОКТАБРЯ - 20 - 2012

Берите

```
3 | protected override void Execute()
4 | {
5 |     double sd = StdDev.Value(Bars.Count - 1, Close, Bars.Count,
6 | StdDevCalculation.Sample);
7 |
8 |     for(int bar = 20; bar < Bars.Count; bar++)
9 |     {
10 |
11 |         // Логика торговой стратегии, которая располагается здесь НЕ ДОЛЖНА
12 |         ИСПОЛЬЗОВАТЬ sd (стандартное отклонение)
13 |     }
14 | }
```

Выполняйте торговые сигналы только на баре с номером (bar + 1)

Если Вы не хотите получать доступ к данным бара, который находится в будущем, Ваша стратегия должна обрабатывать текущий бар и создавать торговый сигнал для следующего бара, т.е. для бара с номером (bar + 1). Все стратегии, которые поставляются вместе с программой Wealth-Lab Developer являются хорошими примерами для внимательного изучения. Обязательно посмотрите также **технику выходов, основанных на времени удержания позиции**.

Исключением из этого правила являются только торговые сигналы, которые используют цену закрытия (AtClose).

Хотя очень сложно (возможно, даже в большинстве случаев невозможно) получить доступ к ценам закрытия бара и выполнить сделку точно по ценам закрытия используя такой тип сигнала, как AtClose, это является позвоительной техникой проведения бектестинга для стратегий, применяющих цены закрытия дня. Смотрите как получить алерт для торговых сигналов типа AtClose. В большинстве случаев внутридневные стратегии не должны использовать сигналы AtClose за исключением только использования таких сигналов для выхода из позиции в конце торговой сессии.

Придерживайтесь порядка выполнения приказов, применяющихся для торговых сигналов

В стратегиях, которые используют многочисленные виды приказов одновременно (как для входов в позицию так и для выхода из позиции), приказы разных типов должны располагаться в таком порядке, который предполагает реалистичное срабатывание приказов. Вот такой порядок следования приказов будет правильным:

- 1. Приказы AtClose выполняются на текущем баре bar.
- Затем следуют приказы, выполняющиеся на баре bar+1 в следующем порядке:
- 2. Приказы AtMarket, затем
- 3. Приказы AtStop (ограничивающие убытки), и наконец, самыми последними выполняются
- 4. Приказы AtLimit (берущие целевую прибыль)

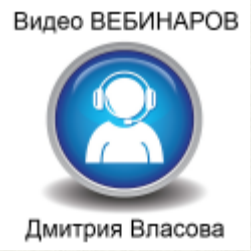
Стоп лоссы должны тестироваться так, чтобы они срабатывали перед тейк профитами (ордерами, берущими целевую прибыль) для того, чтобы моделировать самый пессимистичный из возможных результат (если оба сигнала могут сработать одновременно на одном и том же баре). Это правило вводится потому что заранее невозможно точно предсказать - какой из сигналов сработает раньше.

Пример (как выполнить пример приведенного кода)...

```
Код примера на C# (кликните чтобы посмотреть)
1 | protected override void Execute()
2 | {
3 |     for(int bar = 50; bar < Bars.Count; bar++) { if
4 | (IsLastPositionActive)
5 | { Position p =
6 | LastPosition; if( CumUp.Series(Close, 5)[bar] >= 1 )
7 |
8 |     ExitAtClose(bar, p);
9 |
10 |     else if( p.MAEAsOfBarPercent(bar) < -8)
11 |
12 |     ExitAtMarket(bar + 1, p);
13 |
14 |     else if( !ExitAtTrailingStop(bar + 1, p, SMA.Series(Close, 50)[bar]) )
15 |
16 |     ExitAtLimit(bar + 1, p, p.EntryPrice * 1.25);
17 | }
18 |
19 | else
20 | {
21 |
22 |     //здесь кодируются правила входа в позицию.
23 |
24 | }
```

Трейдинг, базирующийся на статусе активности позиции

Пожалуй наиболее хитроумными ошибками заглядывания в будущее, которые Вы можете совершить при программировании торговых стратегий, являются ошибки базирующиеся на следующей логике: Выполнение



МТС под микроскопом: WLMA System

Подробный разбор торговой системы WLMA System 3900 руб.

В корзину

Рубрики

Выберите рубрику

Обновления:

- Новые посты по Email
- Новые посты по RSS
- Комментарии по RSS
- Следим на Twitter

Метки

C# HowTo: Wealth-Lab Видео Индикаторы МТС Торговые роботы Юмор алготрейдинг мероприятия обучение

Управление

Регистрация

Вход

RSS Записей

RSS Комментариев

WordPress.org

торговых сигналов на текущем баре в зависимости от результатов срабатывания другого торгового сигнала на следующем баре (этот сигнал может закончиться сделкой, а может и не привести к сделке).

Для избегания таких ошибок запомните главное правило: никогда не проверяйте статус активности позиции после выполнения любого торгового сигнала. К примеру, в шаблоне кода однопозиционной стратегии, который изначально встроен в редактор стратегий вся логика торговли базируется на проверке является ли текущая позиция активной. Причем это происходит до выполнения торговых сигналов.

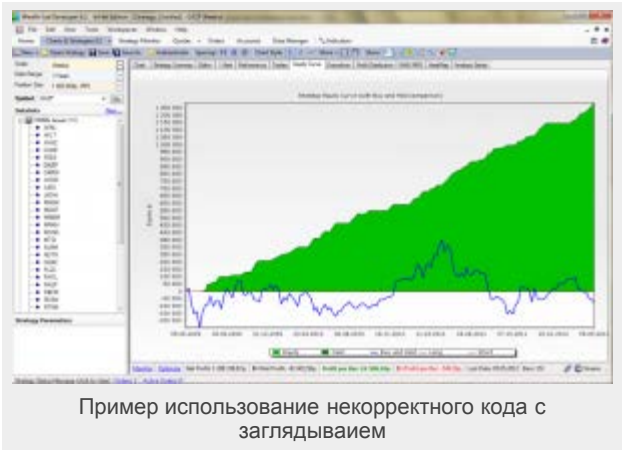
Внимательно рассмотрите следующую "проблемную стратегию", которая входит в позицию используя лимитные ордера на следующем баре, но которая тут же закрывает открытую позицию по цене открытия на том же самом баре. В результате получается невероятная доходность этой стратегии (ведь если нет открытия позиции, то нет и закрытия).

Пример (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть) ?

```
1 //Пример иллюстрирует то, ЧТО ДЕЛАТЬ НЕЛЬЗЯ!!!
2 protected override void Execute()
3 {
4     for(int bar = 5; bar < Bars.Count; bar++)
5     {
6         if( !IsLastPositionActive )
7
8             BuyAtLimit(bar + 1, Close[bar] * 0.98);
9
10        /* НИКОГДА НЕ ПРОВЕРЯЙТЕ СТАТУС АКТИВНОСТИ ПОЗИЦИИ ПОСЛЕ ТОРГОВОГО СИГНАЛА
11    */
12
13        if (IsLastPositionActive)
14
15            ExitAtMarket(bar+1, LastPosition);
16    }
```

Смотрите, какой прекрасный график роста капитала рисует нам проведенный бектестинг такой стратегии. Ну чем не "святой грааль"?



Пример использование некорректного кода с заглядываем

К сожалению, это всего лишь некорректный код стратегии, при входе в позицию стратегия подглядывает в будущее.

Искажения результатов из-за выбытия компаний

Вы должны иметь в виду, что возможны значительные искажения результатов тестирования даже в том случае, если код стратегии безупречен. Однако здесь вклинивается другая проблема - при бектестинге стратегий Вы обычно используете данные о финансовых инструментах, которые относятся только к тем компаниям, которые выжили в суровых условиях рыночных реалий. При этом финансовые инструменты, относящиеся к компаниям, которые больше не существуют или же больше не торгуются публично "выпадают" из Ваших тестов.

Отсутствие данных по таким финансовым инструментам при бектестировании может значительно исказить результаты тестирования стратегии (как в лучшую, так и в худшую сторону). Это происходит потому, что Вы заранее не знали бы какие бумаги выпадут и торговали бы всеми этими бумагами, а не только теми, которые оказались живучими в длительной перспективе.

Искажения результатов такого рода могут как оказать значительное воздействие на результаты тестирования, точно также этого влияния может вообще не произойти. Однако, как говорится, кто предупрежден - тот вооружен. Вам всегда нужно иметь в виду, что такая проблема существует.

Обратите внимание на то, что добыть данные о финансовых инструментах тех финансовых компаний, которые обанкротились достаточно сложно. Однако имеет смысл потестировать такие бумаги если Вы сможете найти их.

Может случиться и обратная ситуация - большое количество компаний выбывает в результате их поглощения другими компаниями. При этом зачастую платятся огромные премии к текущей цене финансовых инструментов. Благодаря процессам слияний и поглощений появляются ценовые всплески в сторону роста.

Сегодня мы рассмотрели опасные моменты, которые могут возникнуть при построении торговых стратегий. Не зря русская пословица говорит - "Знал бы где упасть - соломки бы подстелил". Те, кто прочитал сегодняшний пост могут заранее "подстелить соломки" и не делать ошибок при построении торговых систем, связанных с заглядыванием в будущее.

Однако, как ни странно, существуют торговые системы, которые заглядывают в будущее, но это заглядывание не является криминальным с точки зрения системного трейдера. В следующий раз поговорим как раз о таких разрешенных подглядываниях в будущее. Не забывайте подписываться на новые статьи по RSS.

Если Вам понравилась эта статья - подпишитесь на **новые СТАТЬИ по RSS** или получайте наши **новые СТАТЬИ на E-mail**:
Рекомендую почитать также:

- **Создание многопозиционных стратегий в Wealth-Lab**
- **Анализ данных в Wealth-Lab с учетом разных таймфреймов**
- **Программирование торговых стратегий в программе Wealth-Lab**
- **Программирование выходов, основанных на времени удержания позиции в Wealth-Lab**
- **Использование обоснованного заглядывания в будущее при программировании торговых стратегий в Велс Лаб**

 РУБРИКИ: **PROGRAMMING TRADING STRATEGIES, WEALTHSCRIPT**

Комментариев: 3

Программирование выходов, основанных на времени удержания позиции в Wealth-Lab | Финансовая лаборатория пишет:
30 мая 2012 в 10:03

[...] Ловушки при построении торговых систем, связанные с пр... Главная Торговая система HighLowLong – шаблон
[...]

Ответить

Юрий Шевелёв пишет:
28 сентября 2012 в 02:44



Дмитрий, здравствуйте!
Спасибо Вам за огромную работу, которую Вы проделали, сделав этот (так необходимый многим) перевод.
Хотелось бы знать, будет ли продолжен перевод оставшейся части инструкции. Мне кажется, что перевод порядка следования приказов точнее сделать примерно так:

1. Приказы AtClose выполняются на текущем баре bar.

Затем следуют приказы, выполняющиеся на баре bar+1 в следующем порядке:

2. Приказы AtMarket, затем
3. Приказы AtStop (ограничивающие убытки), и наконец, самыми последними выполняются
4. Приказы AtLimit (берущие целевую прибыль)

С уважением, Юрий Шевелёв

Ответить

Дмитрий Власов пишет:
30 сентября 2012 в 11:38



Спасибо, Юрий. Поправил.

Ответить

Оставить комментарий

Имя (Обязательно)

Почта (Обязательно)

Вэб-сайт

Получать новые комментарии по электронной почте. Вы можете [подписаться](#) без комментирования.

© 2012, Финансовая лаборатория. Все права защищены.
Любое использование материалов допускается только с указанием активной ссылки на источник.



489
243
188



В день мой блог
посещает **77**
человек,
сегодня — **126**



mail.ru РЕЙТИНГ
757152 537
196



Анонсы
BY FEEDBURNER

Подписка

- [RSS](#)
- [Twitter](#)
- [ВКонтакте](#)
- [facebook](#)

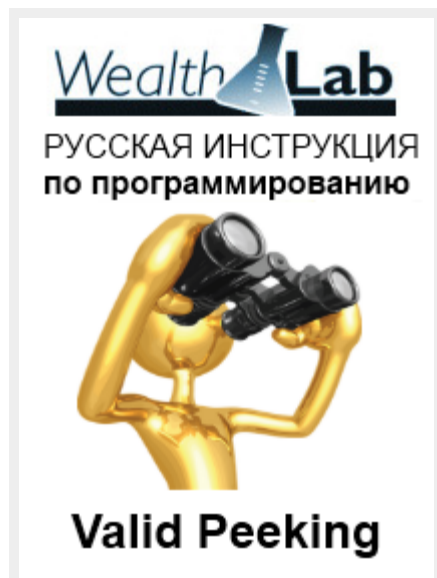
Сайт

- [Создать блог](#)
- [Содержание](#)
- [О блоге](#)
- [Карта](#)

Использование обоснованного заглядывания в будущее при программировании торговых стратегий в Велс Лаб

АВТОР [ДМИТРИЙ ВЛАСОВ](#) ⌚ МАЯ - 30 - 2012 💬 1 КОММЕНТАРИЙ

Возможно, это станет для Вас сюрпризом, но существует большое количество ситуаций в которых код программы по факту заглядывая в будущее тем не менее не является по своей сути ошибочным "заглядыванием". Далее в этой статье приводятся некоторые примеры таких ситуаций, однако в реальности таких примеров может быть намного больше.



Заглядывание в будущее с помощью кода программы при построении торговой стратегии в Велс Лаб будет оправданным в следующих случаях:

Проверка - не наступит ли на следующем баре заранее известное событие

Обоснованным заглядывание в будущее может быть для проверки - не наступит ли там некоторое заранее известное событие. Таким событием может быть:

- день экспирации опционов;
- последний торговый день месяца;
- день сплита акций;

Заглядывая в будущее, т.е. исследуя следующий за текущим бар мы проверяем - не там ли наступит известное событие.

Точно таким же образом, как Вы смотрите в календарь и определяете - не наступит ли завтра новый месяц, приводящийся в следующем примере метод `NextBarIsNewMonth()` проверяет - не является ли дата следующего бара датой другого месяца.

Однако обратите внимание, что скрипт может проверить дату на следующем баре только в том случае, если следующий бар в настоящее время уже существует на текущем графике. Именно по этой причине (и для упрощения) когда цикл достигает последний бар на графике, данный метод считает, что следующий бар является баром нового месяца. В результате этого, когда позиция не является активной, стратегия всегда будет генерировать алерт для открытия новой позиции.

Поэтому для того, чтобы выставить ордер на открытие позиции Вы должны будете лично заглянуть в календарь и убедиться, что следующий день действительно является днем нового месяца.

Пример ([как выполнить пример приведенного кода](#))...

Код примера на C# ([кликните чтобы посмотреть](#))

RSS товары



RSS новости



Twitter



Последние комментарии

макс: А где взять свечки с открытым интересом? на сайте финама нет



ring10: Добрый день Дмитрий Как сделать, чтобы открывался 1 велф лаб? Если открываешь под пользователем, т [...]



Дмитрий Власов: Буян, рад, что Вам пригодились материалы данной статьи!



Максим: Сам себе отвечу. protected override void Execute() { Bars rts = GetExternalSymbol("RTS", t [...]



Буян: Спасибо Дмитрий за статью! Мне удалось разработать хорошую рабочую торговую систему, используя возмо [...]



Последние посты



Бесплатный вебинар: 7 смертных грехов при оптимизации торговых систем

НОЯБРЯ - 5 - 2012



Отзывы участников курса до результата Wealth-Lab. С нуля до первой торговой системы за 5 дней.

НОЯБРЯ - 3 - 2012



Быстрее, Выше, Сильнее: Обновление Wealth-Lab до новой версии 6.4

ОКТАБРЯ - 23 - 2012



Курс до результата: Wealth-Lab – от нуля до первой торговой системы за 5 дней

ОКТАБРЯ - 23 - 2012



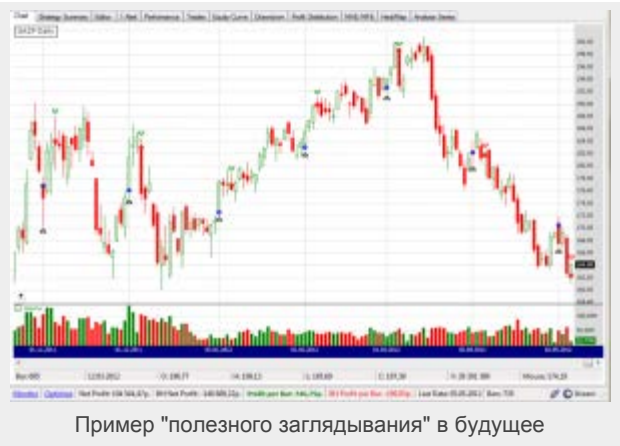
Обзорная экскурсия по достопримечательностям программы Wealth-Lab

ОКТАБРЯ - 20 - 2012

Берите

```
1 //создаем новый метод, проверяющий не является ли следующий бар днем нового месяца
2 public bool NextBarIsNewMonth(int bar)
3 {
4     if (bar < Bars.Count - 1)
5
6         return Date[bar + 1].Day < Date[bar].Day; //если номер следующего дня больше,
7 то это не новый месяц
8
9     else
10
11         /* Предполагается, что следующий бар является новым месяцем.
12            Взгляните в календарь и просто проигнорируйте алерт на вход в позицию если это
13 не так */
14
15         return true;
16
17 }
18
19 protected override void Execute()
20 {
21
22
23     for(int bar = 1; bar < Bars.Count; bar++)
24     {
25
26
27         if (IsLastPositionActive)
28         {
29
30
31             // выход после 3-х баров
32             Position p = LastPosition;
33
34             if (bar + 1 - p.EntryBar >= 3)
35
36                 SellAtMarket(bar + 1, p, "Time-based");
37
38         }
39
40         else if (NextBarIsNewMonth(bar)) // вход в первый торговый день нового месяца
41
42             BuyAtMarket(bar + 1);
43
44     }
45 }
```

В результате выполнения данного кода получается вот такой график:



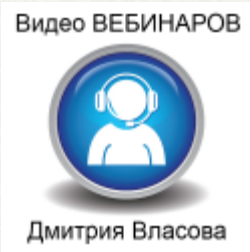
Здесь четко видно, что вход в позицию происходит в первый день месяца, а выход из позиции ровно через 3 бара.

Определение - исполнился ли приказ типа AtStop либо AtLimit по цене открытия следующего бара - для назначения такой позиции высшего приоритета

Если говорить в целом, то свойство Position.Priority сложно применять при проведении бектестинга стратегий, использующих закрытие позиций в конце дня и одновременно использующих стоп и лимитные ордера. Эти сложности возникают из-за того, что текущая последовательность сделок должна выстраиваться в зависимости от времени дня в которое текущая цена достигает цены торгового приказа. С другой стороны, достаточно легко определить - исполнился ли стоп или лимитный ордер в момент открытия торговой сессии. В этом случае, Вы можете установить приоритет больше чем единица для того, чтобы Велс Лаб выполнила этот ордер в первую очередь.

Пример (как выполнить пример приведенного кода)...

```
Код примера на C# (кликните чтобы посмотреть)
1 protected override void Execute()
2 {
3
4     for(int bar = 1; bar < Bars.Count; bar++)
5     {
6
7         if (IsLastPositionActive)
8         {
9             // правила для выхода из позиции
10        }
11
12        else
13
14        {
15
16            double limitPrice = Close[bar] * 0.98; // падение от цены закрытия
17            текущего бара в 2%
18
19            if ( BuyAtLimit(bar + 1, limitPrice, "") != null ) //если произошла сделка
20            на следующем баре
21
22                if ( bar < Bars.Count - 1 && Open[bar+1] <= limitPrice ) //если сделка
23                произошла в момент открытия торговой сессии
24
25            }
```



МТС под микроскопом: WLMA System

Подробный разбор торговой
системы WLMA System
3900 руб.

В корзину

Рубрики

Выберите рубрику

Обновления:

- Новые посты по Email
- Новые посты по RSS
- Комментарии по RSS
- Следим на Twitter

Метки

C# HowTo: **Wealth-Lab**
Видео Индикаторы МТС Торговые
роботы Юмор алготрейдинг
мероприятия обучение

Управление

Регистрация

Вход

RSS Записей

RSS Комментарии

WordPress.org

```
25 | LastActivePosition.Priority = 1.0;
    | }
    | }
    | }
```

Поиск подходящих условий для разделения позиции с целью частичного выхода из этой позиции

Когда Вы торгуете в реале, Вы не должны думать в терминах "разделения позиции" на части, ведь для закрытия половины позиции, состоящей из 1000 лотов Вы просто берете и вводите торговый приказ на закрытие пятисот лотов от текущей позиции. Если же Вы хотите смоделировать частичный выход из позиции при проведении бектестинга в программе Wealth-Lab, то для этого нужно будет сначала разделить первоначальную позицию на две самостоятельных позиции и только после этого выйти из одной из этих позиций - когда это будет необходимо.

Но что же делать, если Вы хотите разделить позицию для того, чтобы зафиксировать прибыль только по части этой позиции, но выйти из всей позиции только при срабатывании стоп-лосса?

Если Вы всегда будете разделять всю позицию на две части и продавать каждую из этих частей отдельно по исполнению стоп-лосса, Вы понесете дополнительные издержки в виде дополнительной комиссии. Вместо этого Вы можете дождаться, пока не наступят условия на выход из позиции (точно также, как и в реальном трейдинге) и определить именно в этот момент - когда необходимо разделение. В приведенном ниже примере происходит "заглядывание вперед" - там отслеживается максимальная цена следующего бара, чтобы определить - можно ли на следующем баре закрыть по тейк-профиту в размере 5% от цены входа половину начальной позиции. Если это так, то на текущем баре позиция разбивается на 2 части.

Пример (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть)

```
1 | protected override void Execute()
2 | {
3 |     bool hasSplit = false;
4 |
5 |     for(int bar = 1; bar < Bars.Count; bar++)
6 |     {
7 |         if (IsLastPositionActive)
8 |         {
9 |             Position p = LastPosition;
10 |
11 |             // сначала попробуйте выйти по Стоп-Лоссу
12 |
13 |             if (!SellAtStop(bar + 1, p, p.EntryPrice * 0.975, "Stop Loss"))
14 |
15 |                 if (!hasSplit) //если ожидается СПЛИТ
16 |                 {
17 |
18 |                     // выход половиной позиции по тейк-профиту размером 5%
19 |
20 |                     double target = p.EntryPrice * 1.05;
21 |
22 |                     if( bar < Bars.Count - 1 && target <= High[bar+1] )
23 |                     {
24 |
25 |                         Position s = SplitPosition( p, 49.99 );
26 |
27 |                         //
28 |                         s является теперь последней позицией (LastPosition), так что нужно
29 |                         продать позицию p для того, чтобы иметь возможность
30 |                         использовать логику LastPosition
31 |
32 |                         hasSplit = SellAtLimit(bar + 1, p, target, "5% Profit");
33 |
34 |                     }
35 |
36 |                 }
37 |
38 |             }
39 |
40 |             else
41 |
42 |                 // Sell the rest at 20% profit target
43 |                 SellAtLimit(bar + 1, p, p.EntryPrice * 1.2, "20% Profit");
44 |
45 |         }
46 |
47 |         else
48 |         {
49 |             BuyAtLimit(bar + 1, Close[bar] * 0.97);
50 |
51 |             hasSplit = false;
52 |         }
53 |     }
54 | }
```

Выполнив этот код Вы увидите следующую картину:



Здесь зеленой линией обозначена та часть позиции, которую удалось закрыть по тейкпрофиту (для этого на предыдущем от срабатывания тейк-профита баре изначальную позицию пришлось разбить на 2). Красной же позицией показана та часть изначальной позиции, которую пришлось закрыть по Стоп-Лоссу.

Чтобы Вы еще нагляднее смогли понять о чем здесь идет речь - посмотрите список трейдов. Красной рамкой выделена ситуация, отображенная на графике.

Chart	Strategy Summary	Editor	1 Alert	Performance	Trades	Equity Curve	Drawdown	Profit Distribution	MAE/MFE	HeatMap	Analysis Series
Position	Symbol	Quantity	Entry Date		Entry Price	Exit Date	Exit Price	Profit %	Profit \$	Bars Held	Exit Name
Long	GAZP	5 512	05.08.2011	10:00	180,11	08.08.2011 1...	175,31	-2,76	-27 437,15p.	10	Stop Loss
Long	GAZP	5 888	08.08.2011	18:00	169,83	09.08.2011 1...	162,88	-4,19	-41 901,07p.	2	Stop Loss
Long	GAZP	6 140	09.08.2011	11:00	162,84	09.08.2011 1...	158,76	-2,60	-26 038,51p.	3	Stop Loss
Long	GAZP	6 366	10.08.2011	18:00	157,06	11.08.2011 1...	153,13	-2,60	-26 005,71p.	8	Stop Loss
Long	GAZP	3 376	23.09.2011	10:00	148,08	26.09.2011 1...	155,49	4,90	24 502,46p.	12	5% Profit
Long	GAZP	3 377	23.09.2011	10:00	148,08	04.10.2011 1...	144,37	-2,60	-13 023,15p.	71	Stop Loss
Long	GAZP	3 554	05.10.2011	13:00	140,65	06.10.2011 1...	147,69	4,90	24 506,33p.	8	5% Profit
Long	GAZP	3 555	05.10.2011	13:00	140,65	24.10.2011 1...	168,78	19,89	99 458,04p.	122	20% Profit
Long	GAZP	2 822	03.11.2011	10:00	177,11	07.11.2011 1...	185,97	4,90	24 494,39p.	10	5% Profit
Long	GAZP	2 824	03.11.2011	10:00	177,11	21.11.2011 1...	172,68	-2,60	-13 002,22p.	102	Stop Loss
Long	GAZP	5 818	09.12.2011	10:00	171,88	12.12.2011 1...	167,58	-2,60	-26 004,89p.	17	Stop Loss

Список сделок для иллюстрации частичного закрытия

Все предыдущие позиции (до красной рамки) закрывались по Стоп-Лоссу, поэтому не приходилось прибегать к разделению позиции. Но так как в коде мы прописали, что если позиция может закрыться на следующем баре по тейк-профиту, то на текущем баре производится разделение (сплит) этой позиции. Что и произошло.

Как видите, заглядывание вперед позволило нам не принимать решение о вхождении или выходе из позиции, а помогает принять решение о расщеплении текущей позиции на две части, чтобы моделирование выглядело точно также, как и торговля в условиях реальной жизни.

Сегодня мы увидели, что при программировании торговых стратегий в программе Wealth-Lab не любое заглядывание в будущее является предосудительным. Иногда можно получать данные от баров, находящихся справа от текущего, но это можно делать только в тех случаях, когда эта информация была бы доступна и в условиях настоящих торгов.

- Если Вам понравилась эта статья - подпишитесь на **новые СТАТЬИ по RSS** или получайте наши **новые СТАТЬИ на E-mail**:
Рекомендую почитать также:
- Программирование торговых стратегий в программе Wealth-Lab
 - Виды торговых сигналов в Wealth-Lab
 - Программирование выходов, основанных на времени удержания позиции в Wealth-Lab
 - Ловушки при построении торговых систем, связанные с проблемой подглядывания в будущее
 - Анализ данных в Wealth-Lab с учетом разных таймфреймов

📁 РУБРИКИ: PROGRAMMING TRADING STRATEGIES, WEALTHSCRIPT

1 комментарий

Инструкция по программированию торговых стратегий в WealthLab | Финансовая лаборатория пишет:
20 июня 2012 в 12:20

[...] [...]

Ответить

Оставить комментарий

Имя (Обязательно)

Почта (Обязательно)

Вэб-сайт

Получать новые комментарии по электронной почте. Вы можете [подписаться](#) без комментирования.

© 2012, Финансовая лаборатория. Все права защищены.
Любое использование материалов допускается только с указанием активной ссылки на источник.



489
243
188



В день мой блог
посещает **77**
человек,
сегодня — **126**



рейтинг
757153538
196



BY FEEDBURNER

Подписка

- [RSS](#)
- [Twitter](#)
- [ВКонтакте](#)
- [facebook](#)

Сайт

- [Создать блог](#)
- [Содержание](#)
- [О блоге](#)
- [Карта](#)

Создание параметров торговой стратегии в Wealth-Lab

 **АВТОР ДМИТРИЙ ВЛАСОВ**  **МАЯ - 31 - 2012**  **КОММЕНТАРИЕВ: 2**

При создании торговых стратегий Вам нужно не только решать вопрос о том, какие индикаторы использовать, но и самое главное - какие параметры для индикаторов нужно применять для того, чтобы торговая стратегия приносила прибыль. Действительно, скользящая средняя с малым периодом усреднения будет генерировать торговые сигналы намного чаще чем та же самая скользящая средняя с периодом усреднения в несколько раз больше.

Для того, чтобы узнать - какие параметры индикаторов являются наиболее подходящими нужно изучить результаты тестирования стратегии с разными величинами. Вручную это был бы очень долгий, можно сказать - бесконечный процесс. Однако программа Wealth-Lab имеет целый арсенал средств для работы с параметрами торговых стратегий.



Параметры торговой стратегии (сокращенно называются - СТРАПы) дают возможность быстро, буквально на лету выполнить оптимизацию торговой стратегии используя удобные ползунки, которые располагаются в самом низу панели данных.

Что такое оптимизация торговых стратегий?

Процесс оптимизации торговой стратегии предполагает неоднократный запуск этой стратегии с использованием ряда различных значений параметров и сбор результатов каждого запуска.

К примеру, если Вы используете стратегию, входы и выходы которой происходят в результате закрытия цены выше или ниже скользящей средней, то можно оптимизировать эту стратегию для того, чтобы в результате определить какой период скользящей средней дает наилучшую прибыль, т. е. оптимальную отдачу.

В Wealth-Lab 6.3 Вы можете оптимизировать стратегию используя любое количество параметров (конечно же в рамках допустимого лимита времени и компьютерной памяти) для одного финансового инструмента либо для набора данных (DataSet) при бектестинге одновременно нескольких финансовых инструментов.

Wealth-Lab по умолчанию имеет два разных метода оптимизации: полный (Exhaustive) и метод Монте-Карло (об этих методах речь пойдет далее). Но хочется упомянуть, что можно использовать и другие методы оптимизации, которые создают сторонние разработчики.

Как: добавить в торговую стратегию, разработанную в программе Wealth-Lab оптимизируемые параметры

Добавить параметры оптимизации для торговой стратегии очень просто - просто выполняйте шаги, которые изложены далее.

Метод 1: автоматический метод создания параметров торговой стратегии

В окне, где располагается график стратегии захватите мышкой и бросьте на график один или более индикаторов. Для того, чтобы автоматически сгенерировать код графика и увидеть слайдеры (ползунки,

[RSS товары](#)



[RSS новости](#)



[Twitter](#)



Последние комментарии

макс: А где взять свечки с открытым интересом? на сайте финама нет



ring10: Добрый день Дмитрий Как сделать, чтобы открывался 1 велф лаб? Если открываешь под пользователем, т [...]



Дмитрий Власов: Буян, рад, что Вам пригодились материалы данной статьи!



Максим: Сам себе отвечу. protected override void Execute() { Bars rts = GetExternalSymbol("RTS", t [...]



Буян: Спасибо Дмитрий за статью! Мне удалось разработать хорошую рабочую торговую систему, используя возмо [...]



Последние посты



Бесплатный вебинар: 7 смертных грехов при оптимизации торговых систем

НОЯБРЯ - 5 - 2012



Отзывы участников курса до результата Wealth-Lab. С нуля до первой торговой системы за 5 дней.

НОЯБРЯ - 3 - 2012



Быстрее, Выше, Сильнее: Обновление Wealth-Lab до новой версии 6.4

ОКТАБРЯ - 23 - 2012



Курс до результата: Wealth-Lab – от нуля до первой торговой системы за 5 дней

ОКТАБРЯ - 23 - 2012



Обзорная экскурсия по достопримечательностям программы Wealth-Lab

ОКТАБРЯ - 20 - 2012

[Берите](#)

которые устанавливают значение оптимизируемых параметров), нужно сделать следующее:

После того, как Вы "бросили" на график индикаторы, Вы можете с помощью простого нажатия кнопки на панели инструментов функций получить код этих индикаторов и увидеть этот код в окне редактора стратегий. Имея доступ к коду отображения индикаторов, Вы автоматически получаете и синтаксис этих индикаторов, благодаря чему Вы можете создать из этих индикаторов ряды данных (DataSeries), которые требуются для Вашей стратегии.

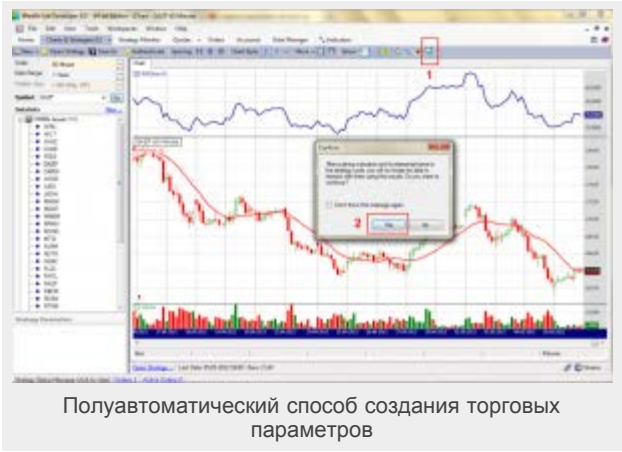
Помимо того, научившись преобразовывать индикаторы в их код, отображаемый в окне редактора стратегии, Вы приобретаете прекрасный способ автоматической генерации кода для слайдеров. Этот способ способен значительно экономить Ваше время в будущем.

Давайте более подробно рассмотрим - как это делается.

Предположим, что Вы собираетесь запрограммировать стратегию, которая использует в качестве индикаторов простую скользящую среднюю (SMA) и индикатор RSI. Конечно же, Вы можете использовать для этой цели построитель стратегий, однако в этот раз Вы хотите сделать это вручную.

Итак, какие же шаги нужно предпринять для этого:

1. Откройте новое окно графика (нажмите одновременно Ctrl+Shift+C)
2. "Бросьте" на график стандартные индикаторы SMA и RSI, изменив по Вашему желанию их параметры.
3. Нажмите на панели инструментов функций кнопку для получения кода. Нажатие этой кнопки приведет к автоматическому преобразованию графика в окно стратегии. Выполните эту появившуюся стратею и откройте окно с редактором стратегии.

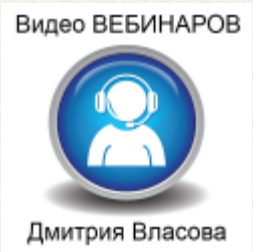


4. Эти действия создают код стратегии, который отображает на графике все те индикаторы, которые Вы изначально размещали на графике, а также здесь будут созданы все операторы и переменные необходимые для создания слайдеров. В зависимости от того, какой шаблон кода стратегии у Вас установлен по умолчанию, код такой стратегии будет выглядеть следующим образом:

Пример (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть)

```
1 using System;
2 using System.Collections.Generic;
3 using System.Text;
4 using System.Drawing;
5 using WealthLab;
6 using WealthLab.Indicators;
7
8 namespace WealthLab.Strategies
9 {
10     public class MyStrategy : WealthScript
11     {
12         //Pushed indicator StrategyParameter statements (объявление переменных типа
13         StrategyParameter для индикаторов)
14
15         private StrategyParameter slider1;
16
17         private StrategyParameter slider2;
18
19         public MyStrategy()
20         {
21
22             //Pushed indicator CreateParameter statements (инициализация переменных
23             типа StrategyParameter для индикаторов)
24             slider1 = CreateParameter("SMA_Period_1", 50, 2, 200, 20);
25             slider2 = CreateParameter("RSI_Period_2", 14, 2, 200, 20);
26         }
27
28         protected override void Execute()
29         {
30             for (int bar = 20; bar < Bars.Count; bar++)
31             {
32                 if (IsLastPositionActive)
33                 {
34                     //code your exit rules here (правила для выхода из позиции)
35                 }
36
37                 else
38                 {
39                     //code your entry rules here (правила для входа в позицию)
40                 }
41             }
42
43             //Pushed indicator ChartPane statements (создание области на графике для
```



МТС под микроскопом: WLMA System

Подробный разбор торговой
системы WLMA System
3900 руб.

В корзину

Рубрики

Выберите рубрику

Обновления:

- Новые посты по Email
- Новые посты по RSS
- Комментарии по RSS
- Следим на Twitter

Метки

C# HowTo: Wealth-Lab
Видео Индикаторы МТС Торговые
роботы Юмор алготрейдинг
мероприятия обучение

Управление

Регистрация

Вход

RSS Записей

RSS Комментарии

WordPress.org

```
44 | индикаторов)
45 |
46 |         ChartPane paneRSI1 = CreatePane(40, true, true);
47 |
48 |         //Pushed indicator PlotSeries statements (отображение индикаторов на
49 | графике)
50 |
51 |         PlotSeries(PricePane, SMA.Series(Low, slider1.ValueInt),
Color.FromArgb(255, 128, 0, 0), LineStyle.Solid, 1);
        PlotSeriesOscillator(paneRSI1,
            RSI.Series(Close, slider2.ValueInt), 70, 30, Color.FromArgb(63, 0, 0,
255), Color.FromArgb(63, 255, 0, 0), Color.FromArgb(255, 0, 0, 128),
LineStyle.Solid, 2);
        }
    }
}
```

5. В появившемся коде стратегии Вы увидите два метода PlotSeries*(), которые располагаются в самом низу кода. Скопируйте SMA.Series() и RSI.Series() внутри этих методов вместе со всеми их параметрами. Создав соответствующие ряды данных (DataSeries) и присвоив им скопированные значения, Вы сможете использовать эти ряды данных в Вашей стратегии. Готовый кусок кода будет выглядеть так:

Пример (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть) ?

```
1 | DataSeries sma = SMA.Series(Low,slider1.ValueInt);
2 | DataSeries rsi = RSI.Series(Close,slider2.ValueInt);
```

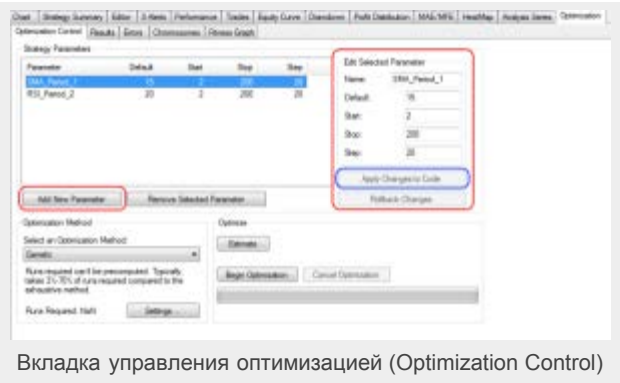
Обратите внимание, что этот кусок кода должен размещаться до торгового цикла.

Надеюсь, Вы убедились, что весь этот код может быть создан буквально за несколько секунд.

Метод 2: полуавтоматическое создание параметров

1. Из любого окна стратегии нажмите на ссылку "Optimize", которая располагается в левом нижнем углу статус бара.

В результате этого появится окошко "Optimization Control".



Вкладка управления оптимизацией (Optimization Control)

2. Используйте интерфейс окна оптимизации для:

- добавления новых параметров;
- регулирования величин параметров, которые будут использоваться по умолчанию;
- начальных и конечных значений параметров оптимизации;
- шага оптимизации

3. Нажмите на кнопку "Apply Changes to Code", что приведет к тому, что все Ваши пожелания будут внесены в код стратегии.

4. В готовом коде стратегии Вы можете получить доступ к значению параметров. Эти значения будут присвоены переменным strategyParameterN. Вы можете переименовать эти переменные по Вашему вкусу, придав им, к примеру, осмысленные названия.

Вы можете получить доступ к этим переменным, используя их свойства Value либо ValueInt для чисел с плавающей точкой либо для целых чисел соответственно.

Для получения дополнительной информации смотрите раздел "Optimize" в пользовательской инструкции.

Метод 3: ручное программирование параметров торговой стратегии

Давайте представим, что Вы работаете со стратегией, которая использует классическую схему с двумя простыми скользящими средними (SMA). Решение о сделках принимается при пересечении этих скользящих средних с разными периодами.

Переменными параметрами здесь будут периоды скользящих средних. Допустим, Вы хотите найти легкий способ регулирования этих периодов. В коде, который находится внизу начнем с рассмотрения этой простейшей стратегии, у которой еще нет страпов (нет параметров стратегии).

Пример (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть) ?

```
1 | namespace WealthLab.Strategies
2 | {
3 |     public class MyStrategy : WealthScript
4 |     {
5 |         protected override void Execute()
6 |         {
7 |             int fastPer = 20; //период усреднения для быстрой скользящей средней
```

```

8         int slowPer = 55; //период усреднения для медленной скользящей средней
9
10        SMA smaFast = SMA.Series(Close, fastPer); //быстрая скользящая средняя
11        (с коротким периодом усреднения)
12        SMA smaSlow = SMA.Series(Close, slowPer); //медленная скользящая средняя
13
14        PlotSeries(PricePane, smaFast, Color.Green, LineStyle.Solid, 2);
15        PlotSeries(PricePane, smaSlow, Color.Red, LineStyle.Solid, 2);
16
17        for (int bar = Math.Max(fastPer, slowPer); bar < Bars.Count; bar++)
18        {
19            if (IsLastPositionActive)
20            {
21                if (CrossUnder(bar, smaFast, smaSlow)) //при пересечении
22                короткой скользящей сверху вниз
23
24                    SellAtMarket(bar + 1, LastPosition);
25            }
26
27            else
28            {
29
30                if (CrossOver(bar, smaFast, smaSlow)) //при пересечении короткой
31                скользящей снизу вверх
32
33                    BuyAtMarket(bar + 1);
34            }
35        }
36    }
37 }

```

Шаг №1: объявляем переменные типа StrategyParameter

Сначала обратим внимание на класс с названием MyStrategy. Давайте изменим имя этого класса - дадим ему более осмысленное название, например SMACrossover.

Прямо здесь давайте объявим несколько частных переменных типа StrategyParameter для того, чтобы в дальнейшем работать со страпами (параметрами стратегии). Поскольку мы хотим регулировать периоды двух скользящих средних, создадим отдельную переменную для каждой из них. В итоге у нас появится переменная slowPeriod - для медленной скользящей средней (с длинным периодом усреднения) и переменная fastPeriod - для быстрой скользящей средней (с коротким периодом усреднения).

Пример (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть)

```

1 namespace WealthLab.Strategies
2 {
3     public class SMACrossver : WealthScript
4     {
5
6         /* Объявляем параметры торговой стратегии, создавая переменные типа
7         StrategyParameter*/
8
9         private StrategyParameter slowPeriod;
10        private StrategyParameter fastPeriod;
11
12        protected override void Execute()
13        {
14
15            int fastPer = 20; //параметр для периода быстрой скользящей средней
16            int slowPer = 55; //параметр для периода медленной скользящей средней
17
18        }
19    }
20 }

```

Шаг №2: создаем конструктор класса

На следующем шаге добавляем новый публичный конструктор класса. Обратите внимание на то, что название метода конструктора должно в точности соответствовать названию класса, для которого создается данный конструктор. В нашем случае это SMACrossover().

Внутри конструктора определяем метод CreateParameter() для каждой из переменных, которые мы объявили на шаге 1. Для более подробного изучения метода CreateParameter() обратитесь к справочнику QuikRef (F11).

Пример (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть)

```

1 namespace WealthLab.Strategies
2 {
3     public class SMACrossver : WealthScript
4     {
5         private StrategyParameter slow;
6         private StrategyParameter fast;
7
8         /* Добавляем публичный конструктор класса (обязательно должен иметь то же
9         самое имя, что и класс) */
10
11        public SMACrossver()
12        {
13            fast = CreateParameter("Fast Per", 20, 4, 100, 1); //начальное значение
14            параметра 20, пробегаемся от 4 до 100 с шагом 1
15            slow = CreateParameter("Slow Per", 55, 5, 300, 5); //начальное значение
16            параметра 55, пробегаемся от 5 до 300 с шагом 5
17        }
18
19        protected override void Execute()
20        {
21            int fastPer = 20;
22            int slowPer = 55;
23
24        }
25    }
26 }

```

Аргументы метода [CreateParameter\(string name, double value, double start, double stop, double step\)](#) имеют следующие значения:

name - эта переменная содержит текст, который располагается рядом с ползунком на панели данных и

используются для идентификации параметров торговой стратегии;
value - присваивает параметру стратегии значение по умолчанию;
start - нижняя граница возможных значений параметра стратегии;
stop - верхняя граница возможных значений параметра стратегии;
step - контролирует величину шага, с которым изменяется значение параметра торговой стратегии от нижней границе к верхней

Шаг 3: присваиваем переменным значения параметров торговой стратегии

И наконец, в методе Execute() мы конвертируем переменную типа StrategyParameter в соответствующий ей переменную, содержащее значение параметра. Для этого используется свойства переменной типа StrategyParameter или ValueInt - для присвоения целочисленного значения, либо Value - для присвоения значения типа float.

Пример (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть)

```
1 namespace WealthLabCompile
2 {
3     class MovingAverageCrossover : WealthScript
4     {
5         private StrategyParameter slow;
6         private StrategyParameter fast;
7
8         public MovingAverageCrossover()
9         {
10             fast = CreateParameter("Fast Per", 20, 4, 100, 1);
11             slow = CreateParameter("Slow Per", 55, 5, 300, 5);
12         }
13
14         protected override void Execute()
15         {
16             /* Получаем значения периодов от параметров */
17
18             int fastPer = fast.ValueInt;
19             int slowPer = slow.ValueInt;
```

Для того, чтобы увидеть конечный результат этого процесса - внимательно изучите стратегию "Moving Average Crossover", которая предустановленна в каждой программе Wealth-Lab Developer.

Код этой стратегии выглядит вот так:

Пример (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть)

```
1 using System;
2 using System.Collections.Generic;
3 using System.Text;
4 using WealthLab;
5 using WealthLab.Indicators;
6 using System.Drawing;
7
8 namespace WealthLabCompile
9 {
10     class MovingAverageCrossover : WealthScript
11     {
12         //Create parameters
13         private StrategyParameter slowPeriod;
14         private StrategyParameter fastPeriod;
15
16         public MovingAverageCrossover()
17         {
18             fastPeriod = CreateParameter("Fast Period", 20, 1, 100, 1);
19             slowPeriod = CreateParameter("Slow Period", 50, 20, 300, 5);
20         }
21
22         protected override void Execute()
23         {
24             //Obtain periods from parameters
25             int fastPer = fastPeriod.ValueInt;
26             int slowPer = slowPeriod.ValueInt;
27
28             SMA smaFast = SMA.Series(Close, fastPer);
29             SMA smaSlow = SMA.Series(Close, slowPer);
30
31             PlotSeries(PricePane, smaFast, Color.Green, LineStyle.Solid, 2);
32             PlotSeries(PricePane, smaSlow, Color.Red, LineStyle.Solid, 2);
33
34             for (int bar = Math.Max(fastPer, slowPer); bar < Bars.Count; bar++)
35             {
36                 if (IsLastPositionActive)
37                 {
38                     if (CrossUnder(bar, smaFast, smaSlow))
39                         SellAtMarket(bar + 1, LastPosition);
40                 }
41                 else
42                 {
43                     if (CrossOver(bar, smaFast, smaSlow))
44                         BuyAtMarket(bar + 1);
45                 }
46             }
47         }
48     }
49 }
```

Как видите, существует достаточно много способов создания параметров торговых стратегий. Вы можете выбрать любой метод создания страпов. Главное, чтобы Вы понимали как применять эти параметры в практической деятельности по оптимизации. В следующий раз рассмотрим тему многопозиционные стратегии, поэтому не забудьте подписаться на обновления по RSS.

Если Вам понравилась эта статья - подпишитесь на **новые СТАТЬИ по RSS** или получайте наши **новые**

СТАТЬИ на E-mail:

Рекомендую почитать также:

- Программирование однопозиционных стратегий (SP Strategies) в программе Wealth-Lab
- Использование при программировании торговых стратегий торговых приказов AtStop и AtLimit
- Создание многопозиционных стратегий в Wealth-Lab
- Что такое алерты и как их использовать в Wealth-Lab
- Анализ данных в Wealth-Lab с учетом разных таймфреймов



РУБРИКИ: PROGRAMMING TRADING STRATEGIES, WEALTHSCRIPT

Комментариев: 2

ring10 пишет:

2 сентября 2012 в 13:26



Здравствуйте Дмитрий!

Прослушал ваш вебинар по оптимизации торговой системы в ВЛД. Хочу изложить некоторые мысли по поводу тех процедур оптимизации, которые были изложены вами.

Мое глубокое убеждение и мои собственные расчеты (я эти занимаюсь уже несколько лет, поднимал свои записи и находил оптимизации которые я делал в 2004 году еще в метастоке) показывают, что оптимизация на одном участке рынка и инструмент должна занимать не более 5-10 минут. Например, на вебинаре мы проводили оптимизацию с мин 300 до макс 5000 при шаге в 5 и к этому еще был другой оптимизационный параметр. То есть получается $(5000-300)/5 = 940$ и еще, по моему, был другой оптимизационный параметр с макс 1500 мин 300 получается $(1500-300)/5 = 240$ и получается $940 \times 240 = 225600$ тестов. А тестировали мы 2 года это примерно 13 торговых часов в день, 250 рабочих дней и 2 года, перемножаем = 6500 часов.

$225600/6500=34$. Получатся 34 варианта оптимизируемых параметра на каждый торговый час. Это чистый подгон под кривую. Что и показал результат на неоптимизируемом периоде.

Так что я считаю все эти методы поиска по Монте-Карло и генетический оптимизатор не нужны, а более того вредны, для тех кто занимается поиском торговых алгоритмов. Так как вводят в заблуждение. Существует книга «Роберт Пардо Разработка, тестирование, оптимизация торговых систем для биржевого трейдера», где изложена неплохая система поиска и оптимизации торговых систем. Однако и она не дает гарантии прибыли. Я, например, по этой методике провел оптимизацию системы Черепашек для фьючерса индекса РТС на часовиках и для доллар-рубли на 30 мин. Все в тестировании показывало не плохо. К сожалению таблицу оптимизации я не могу сейчас найти. По этой системе я стал торговать на конкурсе ЛЧИ в 9 или 10 году. (кому интересно посмотрите ник ring10) У меня была чисто механическая система с интегрированной с Омегой и связка от zoran3.pard и маленький комп. типа современных планшетников. Я его таскал с собой на работу с работы, дома контролировал иногда. Ложился спать, он работал, в общем не большой голоняк на 3 месяца. Результат минус 10000 рублей. из 30000 капитала. В середине конкурса я отказался от системы с долларом, так как она генерила уж большие убытки. Хотя по индексу может и был бы и ноль, но я на одну неделю попал в больницу на операцию и не мог торговать. По закону подлости в это время было большое движение, а как вы знаете система Черепашек трендовая на пробой уровней макс./мин. 20 и 50 + управление капиталом.

Так что к оптимизации надо относится очень и очень аккуратно.

Вот почему я так настойчиво у вас спрашивал на вебинаре про оптимизацию на разных таймфреймах. Это очень хороший метод оптимизации. Зачастую хорошо прибыльная стратегия на часовиках прибыльна, а на 30 мин совсем, очень сильно убыточна. А спрашивается, что кардинально изменилось, что так переменялась эквити? Ведь идеи и принципы сохранились. Оптимизация по таймфрейму отсекает совсем не работающие системы. У меня еще много мыслей по поиску робастных систем, однако одному реализовать это не осилить.

Так вот у меня предложение создать на базе читателей вашего блога создать рабочую группу, которая совместными усилиями сможет во первых создать методику поиска робастных систем автоматизировать их поиск, а в дальнейшем делится опытом по применению, то есть реальной торговли.

Вот, например, сейчас я провожу работу по поиску индикаторов, которые имеют статистически значимые предсказательные результаты. В книге “Р.Колби,Т.Мейерс. Энциклопедия технических индикаторов рынка” рассказан метод ячеек с помощью которого, можно проводить эту работу. Могу потом когда появятся результаты расчетов поделиться.

Причем для сомневающихся, те кто думает, что чужие смогут воспользоваться его работой скажу следующее, что систем торговли так много и так много инструментов от акций до форека и что у каждого трейдера свои предпочтения (по отношению к убыткам к таймфрейму, технические ограничения), что конкуренции на рынке не возникнет, а польза от совместных усилий возрастет многократно.

Игорь 02.09.12

Ответить

Дмитрий Власов пишет:

2 сентября 2012 в 15:04



Игорь, спасибо за столь подробный комментарий. Я с Вами совершенно согласен в том, что система, которая дает хорошие результаты при оптимизации – совершенно не обязана приносить доход на том периоде, на котором мы ее не оптимизировали. При этом, к сожалению, это будет относиться к любым системам и к любым способам нахождения оптимальных параметров.

Но в то же время во время оптимизации мы делаем очень важное дело – отсекаем те параметры, которые никогда не работают и точно работать не будут. Таким образом, мы хотя бы немного повышаем свои шансы на то, что торговая система будет прибыльно работать в течение какого-то времени в будущем.

А сильно повышать свои шансы нам и не нужно. Например, владельцы казино живут и процветают получая преимущество всего в 2,7%....

Зная, что любая система со временем “сломается” нужно заранее разработать четкие правила дисквалификации боевых систем. И тогда, как только мы увидим что результат, который показывает система совсем не тот, который она показывала на истории – мы принимаем жесткое решение и убираем эту систему из торговли. Т.е. тут тоже подход как и в тренде....

Входим в тренд (запускаем торговую систему), ставим стоп-лосс – вернее сказать, трейлинг стоп(определяем правила дисквалификации системы) и отдаемся на волю рынка...

Те системы, которые “сломались” – выбиваются по стопу. Те системы, которые соответствуют текущему рынку находятся в тренде и мы стараемся в этом тренде продержаться как можно дольше (не трогаем систему и продолжаем в соответствии с модулем управления позицией увеличивать ставки на эту систему)...

Еще раз, как и всегда, я подчеркиваю, что это всего лишь мой взгляд на рынок и я его никому не навязываю. Я просто рассказываю то, во что верю в данный момент.

По поводу рабочей группы – у меня именно такие задумки – те участники курса, которые захотят работать дальше коллективно смогут это делать. Как первую пробу я попросил каждого из участников до 5 сентября придумать и запрограммировать по одной технике входа и одной технике выхода... Как результат каждый из нас получит по 10 разных техник входа и 10 разных техник выхода... В результате можно будет потестировать 100 различных торговых систем и наверняка из этого получится что-нибудь интересное.

Пока, к сожалению, не получил в ответ ни одного письма... Так что коллективная работа (несмотря на всю ее привлекательность) зачастую бывает очень затруднена. А так я очень надеюсь, что мы сможем работать совместно (тем более теперь после прохождения курса – мы говорим на одном языке).

Ответить

Оставить комментарий

Имя (Обязательно)

Почта (Обязательно)

Вэб-сайт

Получать новые комментарии по электронной почте. Вы можете [подписаться](#) без комментирования.

© 2012, Финансовая лаборатория. Все права защищены.
Любое использование материалов допускается только с указанием активной ссылки на источник.

489

243

188

В день мой блог посещает 77 человек, сегодня

126

mail.ru РЕЙТИНГ

757154

539196

А.ТВОИ

BY FEEDBURNER

Подписка

- [RSS](#)
- [Twitter](#)
- [ВКонтакте](#)
- [facebook](#)

Сайт

- [Создать блог](#)
- [Содержание](#)
- [О блоге](#)
- [Карта](#)



Создание многопозиционных стратегий в Wealth-Lab

 **АВТОР ДМИТРИЙ ВЛАСОВ**  **ИЮНЯ - 1 - 2012**  **НЕТ КОММЕНТАРИЕВ**

Когда трейдер строит сложную стратегию - ему не всегда хватает возможностей **шаблона однопозиционной стратегии**. Ведь действительно иногда хочется использовать доливки либо пирамидинг. При этом вход в следующую позицию может зависеть от того, есть ли уже хотя бы одна позиция. Всем этим можно управлять с помощью программирования в программе Wealth-Lab. Такие стратегии, которые используют более одной позиции одновременно называются мультипозиционными, либо многопозиционными.



До сих пор мы рассматривали только такие примеры, в которых схема построения кода стратегии гарантировала, что одновременно удерживается не более одной позиции в единицу времени. Многопозиционные стратегии (MP-Strategies) сконструированы таким образом, чтобы иметь возможность управлять одной или более позициями одновременно. К примеру, стратегия, которая пирамидит позиции (делает дополнительные покупки либо дополнительные продажи) на одном финансовом инструменте должна управлять каждой новой позицией отдельно от других.

Хоть Вы и можете использовать метод SplitPosition() для того, чтобы разделить существующую позицию на две части, в настоящее время невозможно "соединить" множество позиций в одну.

Советы по управлению мультипозицией

WealthScript содержит некоторое количество свойств, которые помогут Вам при построении торговых стратегий работать с информацией о позициях. Это такие свойства, как:

- Positions.Count - дает информацию об общем количестве позиций, которые были созданы;
- ActivePositios.Count - возвращает общее количество позиций, которые активны на текущий момент;

Каждая позиция, которая добавляется в ряд позиций имеет свой собственный набор свойств, которые содержат информацию о специфике позиции:

- .Active - показывает является ли позиция в текущий момент активной. Возможны два состояния: "false" - позиция неактивна или "true" - позиция активна.
- .BarsHeld - показывает на протяжении какого количества баров существует позиция.
- .EntryPrice - отражает цену входа в позицию;
- .EntryBar - содержит номер бара, на котором произошел вход в позицию;
- .ExitBar - содержит номер бара, на котором произошел выход из позиции.

Для получения дополнительной информации обратитесь к инструкции QuikRef (F11) - выбрав раздел Position Object.

Когда Вы работаете с мультипозициями, обычно Вы должны внутри Вашего главного торгового цикла организовать второй цикл, в котором обрабатывается каждая активная позиция для определения того, должна

RSS товары



RSS новости



Twitter



Последние комментарии

макс: А где взять свечки с открытым интересом? на сайте финама нет



ring10: Добрый день Дмитрий Как сделать, чтобы открывался 1 велф лаб? Если открываешь под пользователем, т [...]



Дмитрий Власов: Буян, рад, что Вам пригодились материалы данной статьи!



Максим: Сам себе отвечу. protected override void Execute() { Bars rts = GetExternalSymbol("RTS", t [...]



Буян: Спасибо Дмитрий за статью! Мне удалось разработать хорошую рабочую торговую систему, используя возмо [...]



Последние посты



Бесплатный вебинар: 7 смертных грехов при оптимизации торговых систем

НОЯБРЯ - 5 - 2012



Отзывы участников курса до результата Wealth-Lab. С нуля до первой торговой системы за 5 дней.

НОЯБРЯ - 3 - 2012



Быстрее, Выше, Сильнее: Обновление Wealth-Lab до новой версии 6.4

ОКТАБРЯ - 23 - 2012



Курс до результата: Wealth-Lab – от нуля до первой торговой системы за 5 дней

ОКТАБРЯ - 23 - 2012



Обзорная экскурсия по достопримечательностям программы Wealth-Lab

ОКТАБРЯ - 20 - 2012

Берите

ли она быть закрыта. Внимательно изучите как использовать ActivePositions.

Однако, Если логика стратегии предусматривает необходимость закрытия всех активных позиций одновременно, самым удобным способом осуществить это будет использования конструкции Position.AllPositions.

Заметка для программистов:

Располагайте код, отвечающий за логику выхода из позиции перед кодированием правил входа в позицию. Это делается для того, чтобы предотвратить выход из позиции на том же самом баре, где произошел вход в позицию, если на этом баре срабатывает и логика входа и логика выхода.

Использование логики LastPosition (особенно использование свойства IsLastPositionActive) обычно является ошибочным в многопозиционных стратегиях. Вы можете нечаянно закрыть позицию, открытую последней, оставив при этом без управления позиции, которые были открыты раньше.

Шаблон кода мультипозиционной стратегии в Wealth-Lab

В многопозиционной стратегии позиции обычно открываются на разных барах но для одного и того же финансового инструмента. В результате логика входов и логика выходов не может быть взаимоисключающей, как это принято в однопозиционных стратегиях. Именно из-за этого Вы должны очень внимательно следить за тем, что код Вашей стратегии при проверке статуса активности позиции не смотрит на следующую ей логику выхода из позиции. Другими словами, убедитесь в том, что логика входа в позиции не зависит от выходов, которые осуществляются на следующем баре.

Пример (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть)

```
1  protected override void Execute()
2  {
3      // главный торговый цикл
4
5      for(int bar = 20; bar < Bars.Count; bar++)
6      {
7          //логика выхода из позици - проверка каждой активной позиции
8
9          for(int pos = ActivePositions.Count - 1; pos >= 0; pos--)
10         {
11             // присваиваем ссылку на позицию используя индекс массива
12             Position p = ActivePositions[pos];
13
14             // Пример: обрабатываем текущую позицию "p"
15             if (bar - p.EntryBar > 10)
16
17                 ExitAtMarket(bar + 1, p, "по-времени");
18         }
19
20         /* здесь правила входа в позиции */
21     }
22 }
```

Никогда не используйте оператор foreach с коллекцией ActivePositions при программировании логики выхода из позиций. Такое удаление объектов из коллекций (которой по-сути являемся ActivePositions) является ошибкой при программировании в C#. А при закрытии позиций происходит именно удаление объекта Position из коллекции ActivePositions. Самым правильным подходом в этом случае будет запустить цикл по коллекции, начиная с последнего объекта по направлению к первому. Именно такой порядок работы отображен в предыдущем примере.

Как: использовать коллекцию ActivePositions

Wealth-Lab Developer версии 6 дает возможность прямого доступа к массиву, в котором содержатся только активные позиции. Этот массив и назван соответственно - ActivePositions. Посмотреть пример использования этого массива можно чуть выше, когда приводился пример к теме кода шаблона мультипозиционной стратегии.

```
// присваиваем ссылку на позицию используя индекс массива
Position p = ActivePositions[pos];
```

Логика работы здесь такая: выбирается конкретная позиция из массива и присваивается ссылка на эту позицию переменной, с которой происходит дальнейшая работа.

Как: использовать ссылку на Position.AllPositions

Если Ваша мультипозиционная стратегия использует одинаковые правила выхода для всех активных позиций, то удобно использовать в качестве ссылки на позицию в торговом сигнале на выход ссылку Position.AllPositions. Такое упрощение позволит Вам отказаться от создания цикла для проверки условий выхода из каждой активной позиции по отдельности.

Пример (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть)

```
1  /*
```



Видео ВЕБИНАРОВ



Дмитрия Власова



МТС под микроскопом: WLMA System

Подробный разбор торговой системы WLMA System
3900 руб.

В корзину

Рубрики

Выберите рубрику

Обновления:

- Новые посты по Email
- Новые посты по RSS
- Комментарии по RSS
- Следим на Twitter

Метки

C# HowTo: **Wealth-Lab**
Видео Индикаторы МТС Торговые
роботы Юмор алготрейдинг
мероприятия обучение

Управление

Регистрация

Вход

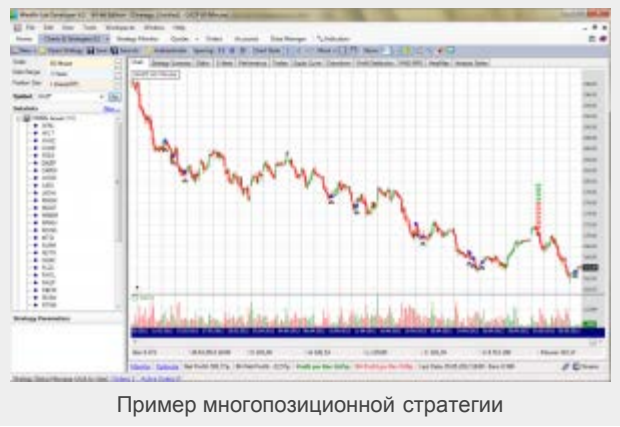
RSS Записей

RSS Комментарии

WordPress.org

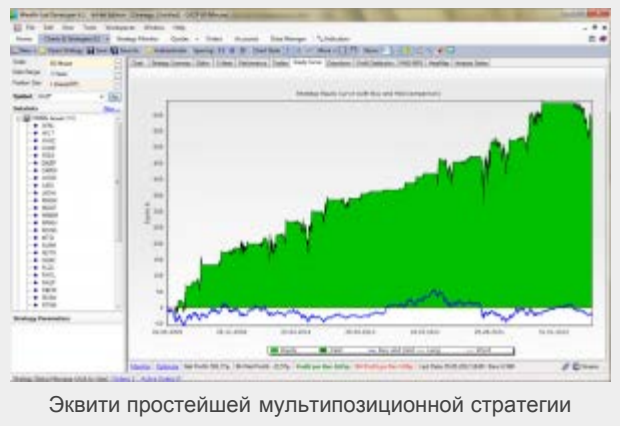
```
2      Эта торговая стратегия покупает когда RSI пересекает снизу вверх
3      уровень 30 и продает все позиции когда RSI пересекает уровень 70 сверху
4      вниз.*/
5
6      protected override void Execute()
7      {
8          RSI rsi = RSI.Series(Close, 14);
9
10         for(int bar = 42; bar < Bars.Count; bar++)
11         {
12             //первым делом кодируем правила выхода из позиции
13             if( ActivePositions.Count > 0 )
14                 if( CrossUnder(bar, rsi, 70) )
15                     SellAtMarket(bar + 1, Position.AllPositions); //обратите внимание на
16                     //ссылку на выход из всех позиций
17
18             if( CrossOver(bar, rsi, 30) )
19                 BuyAtMarket(bar + 1);
20         }
21     }
```

Вот так будет выглядеть окно стратегии:



Как видите, позиция набирается частями и затем при срабатывании условия на выход происходит выход сразу всех позиций одновременно.

Как ни странно, но даже такая простая стратегия за последние 3 года для акции Газпрома показывает вполне приемлемый результат.



Правда, результат кардинально изменится, если включить в период тестирования кризисный 2008 год.

Стратегии с несколькими финансовыми инструментами

Принимая во внимание особенность Wealth-Lab проводить бэктестинг нескольких финансовых инструментов, в общем-то не требуется явно указывать прямо в коде стратегии сделки с конкретным финансовым инструментом. Другими словами, скрипт использует для тестирования первичный финансовый инструмент - тот, который мы выбираем для тестирования.

Однако, когда стратегия содержит торговые правила, зависящие от поведения цены другого финансового инструмента, Вы должны полностью контролировать моделирование с помощью осуществления сделок со вторым финансовым инструментом, используя при этом метод SetContext(). Примером таких более сложных стратегий являются стратегии арбитража, парного трейдинга, а также стратегии, предполагающие ротацию финансовых инструментов в зависимости от некоторых условий.

Для примера, Вы можете изучить код предустановленных в Велс Лаб стратегий с названием "RSI Rotation" или "Dogs of the Dow".

Сегодня мы вкратце поговорили о возможности использовать не только однопозиционную стратегии, но и стратегий, которые могут управлять несколькими позициями одновременно. Также мы вкратце рассмотрели стратегии, которые работают не с несколькими позициями одного финансового инструмента, а с несколькими финансовыми инструментами одновременно, т.е. мультисимвольным стратегиям (MS-Strategies). Изучив возможности, которые появляются здесь Вы сможете программировать даже такие сложные торговые стратегии.

Следующая наша статья будет посвящена рассказу о том, как можно с помощью программы Велс Лаб своевременно получать торговые сигналы. Как Вы догадались, речь пойдет об алертах. Не забывайте подписываться по RSS на новые статьи нашего блога.

Если Вам понравилась эта статья - подпишитесь на **новые СТАТЬИ по RSS** или получайте наши **новые СТАТЬИ на E-mail**:

Рекомендую почитать также:

- **Анализ данных в Wealth-Lab с учетом разных таймфреймов**
- **Программирование однопозиционных стратегий (SP Strategies) в программе Wealth-Lab**
- **Программирование выходов, основанных на времени удержания позиции в Wealth-Lab**
- **Ловушки при построении торговых систем, связанные с проблемой подглядывания в будущее**
- **Программирование торговых стратегий в программе Wealth-Lab**



РУБРИКИ: PROGRAMMING TRADING STRATEGIES, WEALTHSCRIPT, ПРИМЕРЫ MTC

Оставить комментарий

Имя (Обязательно)

Почта (Обязательно)

Вэб-сайт

Получать новые комментарии по электронной почте. Вы можете **подписаться** без комментирования.

© 2012, Финансовая лаборатория. Все права защищены.
Любое использование материалов допускается только с указанием активной ссылки на источник.



489
243
188



В день мой блог
посещает **77**
человек,
сегодня — **126**



mail.ru РЕЙТИНГ
757155540
196



А вы?



BY FEEDBURNER

Подписка

- [RSS](#)
- [Twitter](#)
- [ВКонтакте](#)
- [facebook](#)

Сайт

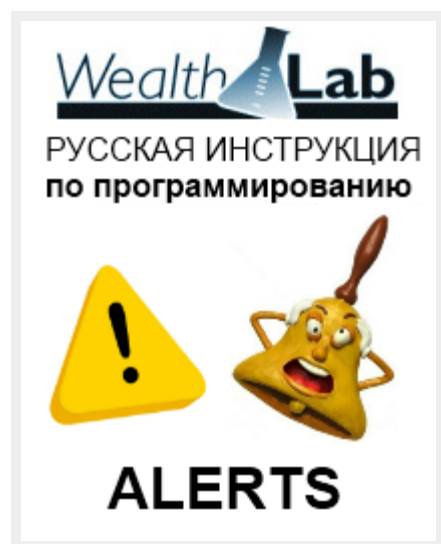
- [Создать блог](#)
- [Содержание](#)
- [О блоге](#)
- [Карта](#)

Что такое алерты и как их использовать в Wealth-Lab

 **АВТОР ДМИТРИЙ ВЛАСОВ**  **ИЮНЯ - 2 - 2012**  **КОММЕНТАРИЕВ: 2**

Торговая стратегия предназначена для того, чтобы генерировать торговые сигналы. Если сначала эти торговые сигналы мы используем для тестирования торговой стратегии и подбора оптимальных параметров, то когда стратегия уже "заточена" нужным образом - приходит пора использовать сигналы этой стратегии в реальной торговле.

Программа Wealth-Lab поможет нам и в этом. Она вовремя сообщит нам о том, что пришло время отдавать приказ брокеру. Такие информационные сообщения называются алертами. Именно об алертах сегодня и пойдет речь.



Торговые сигналы можно использовать двумя способами:

- Для создания теоретических трейдов для бектестинга
- Для генерации алертов чтобы отдавать торговые приказы. Эти приказы должны размещаться на баре идущем сразу же за последним баром графика.

Алерты это то, что делает возможным для Вас торговлю Вашей стратегии в реальной жизни. Алерты являются приказами, которые Вы размещаете на ближайшем баре после их появления (это споаведливо для любых таймфреймов - для интрадея, дневок, неделек и т.п.).

К примеру, когда торговля идет с использованием дневных баров, Вы можете запускать на своем компьютере торговую стратегию каждый вечер после окончания торгов и обновления данных о прошедшем торговом дне. Если торговая стратегия на основании обработки полученных данных сгенерирует алерт, то Вы должны будете отдать соответствующий торговый приказ на следующий день Вашему брокеру.

С другой стороны, если Вы торгуете интрадей, то Вы должны будете отдать приказ немедленно - сразу же после получения алерта.

При кодировании торговых систем с помощью WealthScript единственным требованием для получения алертов является сообщение торговому сигналу, что сделка должна произойти на баре с номером (bar + 1), т.е. сообщить, что сделка должна состояться за правой границей существующего графика.

Во время последней итерации в торговом цикле, когда номер текущего бара равен (Bars.Count - 1) такой торговый сигнал, как BuyAtMarket (bar + 1) обязательно создаст новый алерт, поскольку номер бара, который передается в данный торговый сигнал больше чем самый последний бар графика с номером (Bars.Count - 1).

Внимание!

RSS товары



RSS новости



Twitter



Последние комментарии

макс: А где взять свечки с открытым интересом? на сайте финама нет



ring10: Добрый день Дмитрий Как сделать, чтобы открывался 1 велф лаб? Если открываешь под пользователем, т [...]



Дмитрий Власов: Буян, рад, что Вам пригодились материалы данной статьи!



Максим: Сам себе отвечу. protected override void Execute() { Bars rts = GetExternalSymbol("RTS", t [...]



Буян: Спасибо Дмитрий за статью! Мне удалось разработать хорошую рабочую торговую систему, используя возмо [...]



Последние посты



Бесплатный вебинар: 7 смертных грехов при оптимизации торговых систем

НОЯБРЯ - 5 - 2012



Отзывы участников курса до результата Wealth-Lab. С нуля до первой торговой системы за 5 дней.

НОЯБРЯ - 3 - 2012



Быстрее, Выше, Сильнее: Обновление Wealth-Lab до новой версии 6.4

ОКТАБРЯ - 23 - 2012



Курс до результата: Wealth-Lab – от нуля до первой торговой системы за 5 дней

ОКТАБРЯ - 23 - 2012



Обзорная экскурсия по достопримечательностям программы Wealth-Lab

ОКТАБРЯ - 20 - 2012

Берите

Никогда не передавайте в торговый сигнал номер бара (bar + n) в том случае, если n > 1. НИКОГДА!
В торговые сигналы всегда нужно передавать только номер (bar + 1). Здесь может быть только одно небольшое исключение - когда сделка может быть совершена та текущем баре - при использовании торгового приказа AtClose. Для дополнительной информации смотрите "Алерты для приказов AtClose"

Мы уже **использовали алерты для создания скрининговой системы**. Для получения большего количества примеров - внимательно просмотрите некоторые из стратегий, которые предустановлены в программе Wealth-Lab.

Несмотря на то, что Вы должны использовать бар с номером (bar + 1) в торговых сигналах для создания торговой возможности для следующего бра, Вы не должны пытаться получить доступ к данным следующего после крайнего на графике бара. Это приведет к очевидной ошибке заглядывания в будущее, и как результат при обработке последнего бара графика приведет к ошибке кода.

Как: сохранить алерты стратегии в файл

Хоть большинство стратегий в основном генерируют только один алерт на один финансовый инструмент при использовании однопозиционную логику, полезно знать, что Wealth-Lab дает возможность работать с массивом алертов. Внутри этого массива алертов Вы можете создать цикл, пробежавшись по всем алертам, которые генерируются стратегией.

Приведенный в примере метод SaveAlerts() может быть повторно использован Вами для сохранения алертов в текстовый файл, находящийся в указанном месте на диске.

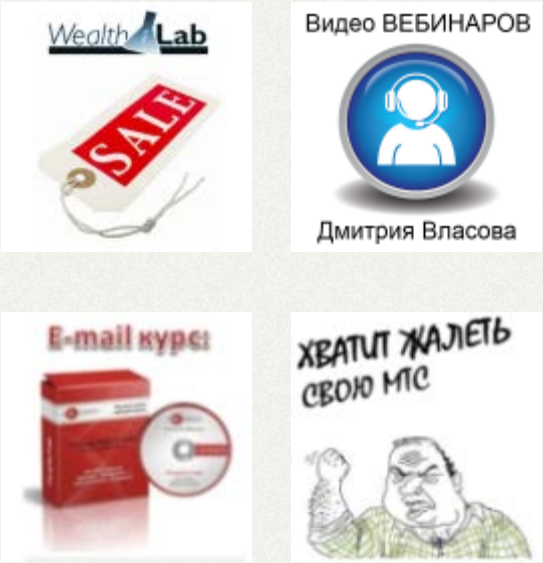
Пример (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть) ?

```
1  /* Должен использоваться в режиме "Raw Profit" для создания алерта для
2  соответствующих финансовых инструментов (размер) */
3
4  //создаем метод, позволяющий сохранять алерты в файл (как аргумент метода - путь
5  куда нужно записать данный файл).
6  public void SaveAlerts(string path)
7  {
8      string dateFormat = "yyyyMMdd";
9
10     if (Bars.IsIntraday) dateFormat = "yyyyMMdd HHmm";
11     if( Alerts.Count > 0 ) //если существует хотя бы один алерт
12     {
13         for( int i = 0; i < Alerts.Count; i++ ) //пробегаемся по всей коллекции
14 алертов
15
16         {
17             Alert a = Alerts[i]; //присваиваем ссылку на алерт переменной "a"
18
19             string s = a.AlertDate.ToString(dateFormat); //присваиваем текстовой
20 переменной "s" значение даты алерта в нужном формае
21
22             s = s + ";" + a.AlertType.ToString(); //добавляем в конец предыдущей записи
23 тип алерта
24             s = s + ";" + a.Shares.ToString(); // добавляем в конец количество лотов
25 (всегда равен единице в режиме "Portfolio Simulation")
26             s = s + ";" + a.Symbol; //добавляем в конец тикер
27             s = s + ";" + a.OrderType.ToString(); //добавляем в конец тип торгового
28 приказа
29
30             if (a.OrderType == OrderType.Limit || a.OrderType == OrderType.Stop) //если
31 это стоп-ордер или лимит-ордер - добавляем цену приказа
32             s = s + ";" + a.Price.ToString();
33             else //если не стоп и не лимит ордер, то добавляем базовую цену
34             s = s + ";" + a.BasisPrice.ToString();
35             s = s + ";" + a.SignalName + "\n"; //добавляем в конец имя сигнала.
36
37             System.IO.TextWriter tw = new System.IO.StreamWriter(path, true); //
38 показываем, куда записывать файл
39             tw.Write(s); //записываем файл
40             tw.Close(); //закрываем сущность, которая записывает файл
41         }
42     }
43 }
44
45 protected override void Execute()
46 {
47
48     // Генерируем 5 алертов, каждый на 1% ниже чем предыдущий.
49
50     int bar = Bars.Count - 1;
51     int pct = 99;
52
53     for(int numAlerts = 1; numAlerts <= 5; numAlerts++)
54
55         BuyAtLimit(bar + 1, Close[bar] * (pct - numAlerts)/100, numAlerts.ToString());
56
57     // Вызываем метод SaveAlerts() после главного торгового цикла
58
59     SaveAlerts(@"C:\Alerts.txt"); //показываем в качестве аргумента путь, куда текст
60 должен быть записан (имя файла)
```

Как: создать алерт для торгового сигнала AtClose

Стратегии, использующие приказы AtClose обычно намереваются открыть либо закрыть позицию по цене закрытия текущего бра, т.е. сразу после обработки. В результате, Вы обычно передаете торговому сигналу AtClose не бар с номером (bar + 1), как это делается во всех прочих случаях, а текущий бар с номером (bar). Поэтому Велс Лаб не в состоянии сообщить Вам о необходимости разместить торговый приказ, т.е. не в состоянии сгенерировать алерт.





**МТС под микроскопом:
WLMA System**

Подробный разбор торговой системы WLMA System

3900 руб.

 **В корзину**

Рубрики

Выберите рубрику

Обновления:

-  Новые посты по Email
-  Новые посты по RSS
-  Комментарии по RSS
-  Следим на Twitter

Метки

C# HowTo: **Wealth-Lab**
Видео Индикаторы МТС Торговые роботы Юмор алготрейдинг мероприятия обучение

Управление

Регистрация

Вход

RSS Записей

RSS Комментариев

WordPress.org

Теоретически Wealth-Lab без проблем может совершить сделку, однако на практике совершенно невозможно обработать данные текущего бара, который только находится в процессе формирования и сразу после полного формирования текущего бара выполнить сделку на том же самом баре.

Однако, если Вы можете получить доступ к данным еще не сформировавшегося бара (смотри инструкцию Wealth-Lab User Guide раздел Yahoo! Data Provider), то наверняка возможно обновить дневной бар прямо перед закрытием торговой сессии для получения предварительной оценки цены закрытия. Затем, используя полученную "прикидочную" цену закрытия, использовать ее для генерации алерта и совершения сделки.

Однако для того, чтобы выполнить все таким образом, необходимо прибегнуть к небольшой уловке. Вместо приказа AtClose нужно будет использовать приказ AtMarket в последнюю минуту торгов.

Именно такой подход демонстрируется в примере, приведенном ниже.

Пример (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть) ?

```
1  protected override void Execute()
2  {
3      SMA smaFast = new SMA(Close, 8, "Fast");
4      SMA smaSlow = new SMA(Close, 16, "Slow");
5      PlotSeries(PricingPane, smaFast, Color.Green, LineStyle.Solid, 1);
6      PlotSeries(PricingPane, smaSlow, Color.Red, LineStyle.Solid, 1);
7
8      for(int bar = 16; bar < Bars.Count; bar++)
9      {
10         if (IsLastPositionActive)
11         {
12             Position p = LastPosition;
13             if (CrossUnder(bar, smaFast, smaSlow))
14             {
15
16                 /* При выполнении условий на выход из позиции: */
17
18                 if (bar == Bars.Count - 1) //Если текущий бар является последним баром,
19                 то генерируется алерт с типом приказа AtMarket
20
21                     SellAtMarket(bar+1, p, "Alert");
22
23                 else //Если текущий бар не является последним, то генерируется обычный
24                 приказ AtClose на текущем баре
25
26                     SellAtClose(bar, p, "AtClose");
27             }
28         }
29         else
30         {
31             if (CrossOver(bar, smaFast, smaSlow))
32                 BuyAtMarket(bar + 1);
33         }
34     }
```

При торговле внутри дня разница между приказами AtClose и AtMarket составляет всего одну сделку. Поскольку торговые сигналы AtClose при торговле внутри дня не могут быть реализованы по практическим соображениям, они должны использоваться очень редко (а лучше вообще никогда не использоваться).

При торговле внутри дня нужно использовать сигналы AtMarket. Исключением может быть только использование сигналов AtClose для закрытия позиции в конце торговой сессии. Реализовать такие сигналы можно тем же самым способом, что и для дневного таймфрейма (пример мы разбирали чуть выше).

Итак, сегодня мы рассмотрели как в программе Wealth-Lab организована работа с алертами. Разобравшись с алертами мы сделали еще один решительный шаг к полной автоматизации биржевой торговли. Однако прежде чем начинать торговать нужно изучить еще несколько вопросов. В следующий раз расскажу Вам о том, как можно проводить анализ нескольких таймфреймов одновременно. Не забывайте подписываться на новые посты по RSS.

Если Вам понравилась эта статья - подпишитесь на **новые СТАТЬИ по RSS** или получайте наши **новые СТАТЬИ на E-mail**:
Рекомендую почитать также:

- **Использование при программировании торговых стратегий торговых приказов AtStop и AtLimit**
- **Программирование выходов, основанных на времени удержания позиции в Wealth-Lab**
- **Ловушки при построении торговых систем, связанные с проблемой подглядывания в будущее**
- **Создание многопозиционных стратегий в Wealth-Lab**
- **Анализ данных в Wealth-Lab с учетом разных таймфреймов**

 РУБРИКИ: **PROGRAMMING TRADING STRATEGIES, WEALTHSCRIPT**

Комментариев: 2

Автоматизация подачи торговых сигналов с помощью Wealth-Lab в связке с Quik | Финансовая лаборатория пишет:

6 июня 2012 в 16:37

[...] Алерт – это торговый сигнал, который сигнализирует о необходимости отправить торговый приказ указанного типа в систему. [...]

Ответить

Автоматическая подача поручений в QUIK на основе торговых сигналов Wealth-Lab | Финансовая лаборатория пишет:

21 июня 2012 в 11:04

[...] в монитор стратегий, ее сигналы даже не нужно превращать в алерты. Можно указывать там стратегию "как есть", т.е. вообще [...]

Ответить

Оставить комментарий

Имя (Обязательно)

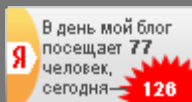
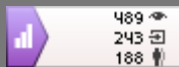
Почта (Обязательно)

Вэб-сайт

Получать новые комментарии по электронной почте. Вы можете [подписаться](#) без комментирования.

© 2012, Финансовая лаборатория. Все права защищены.

Любое использование материалов допускается только с указанием активной ссылки на источник.



А тайф



BY FEEDBURNER

Подписка

[RSS](#)

[Twitter](#)

[ВКонтакте](#)

[facebook](#)

Сайт

[Создать блог](#)

[Содержание](#)

[О блоге](#)

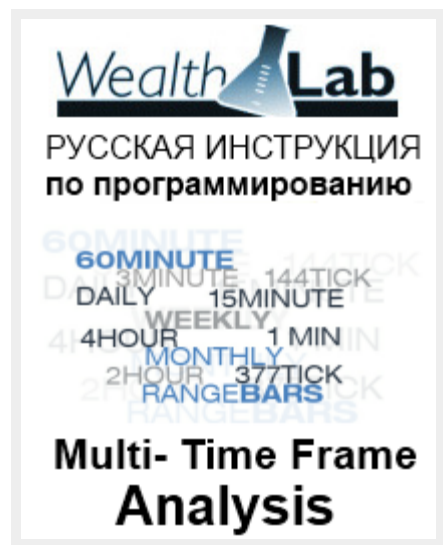
[Карта](#)

Анализ данных в Wealth-Lab с учетом разных таймфреймов

 АВТОР ДМИТРИЙ ВЛАСОВ  ИЮНЯ - 3 - 2012  КОММЕНТАРИЕВ: 2

Всем известна знаменитая торговая система Александра Элдера под названием "Три экрана". Смысл этой торговой стратегии заключается в том, что необходимо смотреть на рынок через призму трех разных временных измерений. Сначала на недельном графике определяется общая тенденция, на дневном графике определяются точки входа, а на внутридневном - моменты входа. Эта система является ярким примером работы с разными таймфреймами в одной стратегии.

Однако что делать системному трейдеру, который не хочет одновременно смотреть на три экрана, а хочет запрограммировать все правила торговой стратегии в одной торговой системе? К счастью, программа Wealth-Lab предоставляет возможность работать одновременно с необходимым количеством таймфреймов.



Большое количество популярных техник торговли использует доступ к данным с более высокого таймфрейма. WealthScript предоставляет целый набор методов, предназначенных специально для этих целей.

Например для дневных данных можно создать индикаторы на основе недельных или месячных графиков, а данные с внутридневных графиков могут быть сжаты как внутри дня (например с 5-ти минуток до часа) так и до дневных, недельных или даже месячных баров.

Анализ данных с учетом разных таймфреймов (мультитаймфреймовый анализ) позволяет Вам получить доступ к данным и индикаторам, которые базируются на таймфрейме более высокого порядка, но создавать трейда на графике базовой шкалы.

К примеру, предположим, что Вы хотели бы чтобы сделки в Вашей торговой системе осуществлялись в сторону лежащего в основе тренда, который измеряется с помощью скользящей средней на недельных барах. В программе Wealth-Lab 6 Вы можете определить этот недельный тренд и торговать в его направлении на дневном (или даже внутридневном) интервале.

Как: с помощью программного кода определить шкалу графика

Для того, чтобы определить таймфрейм графика (баров), Вы можете использовать структуру BarDataScale для того, чтобы получить значения простого типа, которые содержит свойства Bars.Scale (Tick - тики, Second - секунды, Minute - минуты, Daily - дни, Weekly - недели, Monthly - месяцы, Quarterly - кварталы, Yearly - годы) и свойство Bars.Interval (20-tick, 30-second, 60-minute). При этом нужно иметь в виду что BarInterval возвращает нулевое значение для не внутридневных шкал.

Пример ([как выполнить пример приведенного кода](#))...

Код примера на C# (кликните чтобы посмотреть)

```
1 | protected override void Execute()
```



RSS товары



RSS новости



Twitter



Последние комментарии

макс: А где взять свечки с открытым интересом? на сайте финама нет



ring10: Добрый день Дмитрий Как сделать, чтобы открывался 1 велф лаб? Если открываешь под пользователем, т [...]



Дмитрий Власов: Буян, рад, что Вам пригодились материалы данной статьи!



Максим: Сам себе отвечу. protected override void Execute() { Bars rts = GetExternalSymbol("RTS", t [...]



Буян: Спасибо Дмитрий за статью! Мне удалось разработать хорошую рабочую торговую систему, используя возмо [...]



Последние посты



Бесплатный вебинар: 7 смертных грехов при оптимизации торговых систем

НОЯБРЯ - 5 - 2012



Отзывы участников курса до результата Wealth-Lab. С нуля до первой торговой системы за 5

дней.

НОЯБРЯ - 3 - 2012



Быстрее, Выше, Сильнее: Обновление Wealth-Lab до новой версии 6.4

ОКТАБРЯ - 23 - 2012



Курс до результата: Wealth-Lab – от нуля до первой торговой системы за 5 дней

ОКТАБРЯ - 23 - 2012



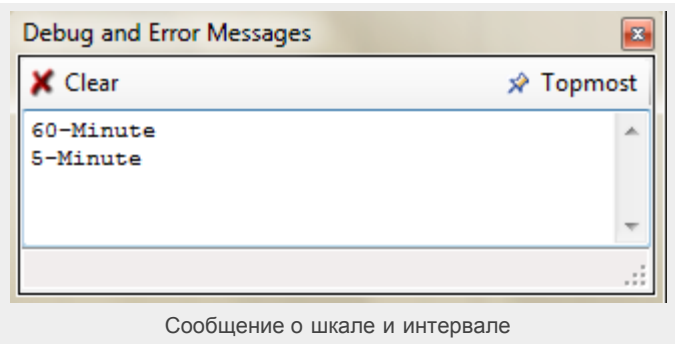
Обзорная экскурсия по достопримечательностям программы Wealth-Lab

ОКТАБРЯ - 20 - 2012

Берите

```
2 {
3     // Получаем информацию об интервале баров графика и типе шкалы графика.
4
5     BarDataScale ds = Bars.DataScale; //создаем переменную ds типа BarDataScale
6
7     if( ds.BarInterval == 0 ) //если не интрадейные данные
8
9         PrintDebug( ds.Scale ); //печатаем тип шкалы графика
10
11     else
12
13         PrintDebug( ds.BarInterval + "-" + ds.Scale ); //печатаем информацию об
14 интервале графика и типе шкалы
15 }
```

В результате появится вот такое сообщение с информацией о шкале и интервале:



Как: получить доступ к данным с верхнего таймфрейма на внутридневной шкале (Intraday - Intraday)

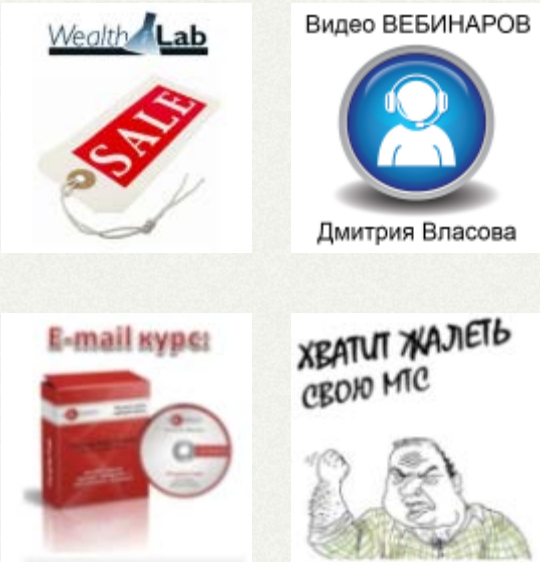
Для получения доступа к данным со старшего таймфрейма на внутридневном графике нужно проделать следующие шаги:

- Используйте метод SetCscaleCompressed() с требуемым периодом сжатия.
- После того, как Вы закончите работать с данными старшего таймфрейма - вызовите метод RestoreScale().
- Далее для получения нового синхронизированного с исходным таймфреймом ряда данных нужно вызвать метод Synchronize().

Этот пример демонстрирует как получить доступ к данным со старшей внутридневной шкалы (для целей иллюстрации рассмотрим 120-ти минутную шкалу) из шкалы с меньшим таймфреймом (в данном случае 15-ти минутной). Мы получим две разных скользящих средних с разными периодами на каждой из шкал и определим их пересечения (как снизу вверх, так и сверху вниз).

Пример (как выполнить пример приведенного кода)...

```
Код примера на C# (кликните чтобы посмотреть)
1 protected override void Execute()
2 {
3     // Выясняем - является ли исходная шкала внутридневной (выполнение _стратегии
4 требует внутридневных данных).
5     if( Bars.BarInterval == 0 )
6
7     {
8         DrawLabel(PricePane, "Данная стратегия работает только с внутридневными данными",
9 Color.Red);
10         return;
11     }
12
13     // Выясняем - достаточно ли данных для создания 60-ти барного 120-ти минутного
14 индикатора?
15
16     int barsRequired = 120 * (60 + 1) / Bars.BarInterval;
17
18     if (Bars.Count < barsRequired)
19     {
20         DrawLabel(PricePane, "Для выполнения стратегии необходимо минимум " +
21 barsRequired.ToString() + " баров", Color.Red);
22         return;
23     }
24
25     // Получаем две скользящие средние с периодом 20 и 60 баров на исходной шкале
26
27     DataSeries SMA20 = SMA.Series( Close, 20 );
28     DataSeries SMA60 = SMA.Series( Close, 60 );
29
30     //Меняем сжатие шкалы до 120-ти минут
31
32     SetScaleCompressed( 120 );
33
34     // Получаем две скользящие средние с периодом 20 и 60 баров на новой (120-ти
35 минутной) шкале
36
37     DataSeries SMA20_120 = SMA.Series( Close, 20 );
38     DataSeries SMA60_120 = SMA.Series( Close, 60 );
39
40     //Меняем шкалу назад к оригинальному таймфрейму
41     RestoreScale();
42
43     SMA20_120 = Synchronize( SMA20_120 ); //синхронизируем данные с исходной шкалой
44     SMA60_120 = Synchronize( SMA60_120 ); //синхронизируем данные с исходной шкалой
45
46     //Отображаем на графике скользящие средние
47
48     PlotSeries( PricePane, SMA20, Color.DarkRed, WealthLab.LineStyle.Solid, 2 );
49     PlotSeries( PricePane, SMA60, Color.DarkGreen, WealthLab.LineStyle.Solid, 2 );
50     PlotSeries( PricePane, SMA20_120, Color.Red, WealthLab.LineStyle.Solid, 2 );
51     PlotSeries( PricePane, SMA60_120, Color.Green, WealthLab.LineStyle.Solid, 2 );
52
53     // Выделяем на графике пересечения (сверху вниз и снизу вверх)
54
55     for(int bar = barsRequired; bar < Bars.Count; bar++)
56     {
```





МТС под микроскопом: WLMA System

Подробный разбор торговой системы WLMA System

3900 руб.

 **В корзину**

Рубрики

Выберите рубрику

Обновления:

-  Новые посты по Email
-  Новые посты по RSS
-  Комментарии по RSS
-  Следим на Twitter

Метки

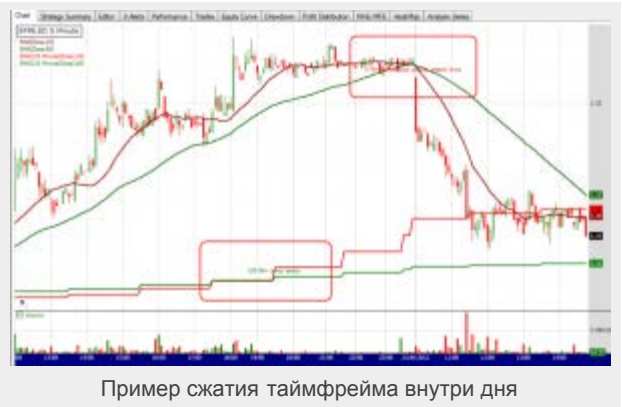
C# HowTo: **Wealth-Lab**
Видео Индикаторы МТС Торговые
роботы Юмор алготрейдинг
мероприятия обучение

Управление

- Регистрация
- Вход
- RSS Записей
- RSS Комментарии
- WordPress.org

```
57
58         if( CrossOver( bar, SMA20, SMA60 ) )
59             AnnotateChart( PricePane, Bars.DataScale + "Исходная шкала: снизу вверх",
60 bar, SMA20[bar], Color.Red );
61
62         if( CrossOver( bar, SMA20_120, SMA60_120 ) )
63             AnnotateChart( PricePane, "120 Min: снизу вверх", bar, SMA20_120[bar],
64 Color.Green);
65
66         if( CrossUnder( bar, SMA20, SMA60 ) )
67             AnnotateChart( PricePane, Bars.DataScale + "Исходная шкала: сверху вниз",
68 bar, SMA20[bar], Color.Red );
69
70         if( CrossUnder( bar, SMA20_120, SMA60_120 ) )
71             AnnotateChart( PricePane, "120 Min: сверху вниз", bar, SMA20_120[bar],
72 Color.Green);
73     }
74 }
```

Вот такой график получится в итоге:



Как: получить доступ к дневным данным с внутрисуточного графика (Intraday - Daily)

Пример демонстрирует как вычислить так называемые уровни Пивот ("Floor Trader Pivots"), известные также как дневные линии поддержки и сопротивления. Для расчета этих уровней используется сжатие внутрисуточных данных до дневной шкалы. Как и всегда при работе с несколькими таймфреймами отображение индикаторов и выполнение сделок должно осуществляться на исходной шкале с меньшим таймфреймом.

синхронизация данных, сдвинутых по шкале времени вправо

Обратите внимание на то, что в этом примере дневные данные сдвинуты вправо по шкале времени. Таким образом, на внутрисуточной шкале текущий бар всегда синхронизируется с данными предыдущего дня. С другой стороны, новые пивоты для текущего дня будут отображаться на последнем баре графика, обновляющегося в реальном времени.

Пример (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть) ?

```
1  protected override void Execute()
2  {
3      const int NumTicks = 2; // Количество тиков после пивот-уровня
4
5      SetScaleDaily();
6
7      DataSeries hPivot = AveragePriceC.Series(Bars);
8      DataSeries hDayHigh = High;
9      DataSeries hDayClose = Close;
10     DataSeries hDayLow = Low;
11     RestoreScale();
12
13     hPivot = Synchronize( hPivot ) >> 1;
14     hDayHigh = Synchronize( hDayHigh ) >> 1;
15     hDayClose = Synchronize( hDayClose ) >> 1;
16     hDayLow = Synchronize( hDayLow ) >> 1;
17
18     hPivot.Description = "Pivot";
19
20     DataSeries hR1 = ( 2 * hPivot - hDayLow );
21     hR1.Description = "R1";
22
23     DataSeries hS1 = ( 2 * hPivot - hDayHigh );
24     hS1.Description = "S1";
25
26     DataSeries hR2 = ( hPivot - ( hS1 - hR1 ) );
27     hR2.Description = "R2";
28
29     DataSeries hS2 = ( hPivot - ( hR1 - hS1 ) );
30     hS2.Description = "S2";
31
32     HideVolume();
33
34     PlotSeries( PricePane, hR2, Color.Maroon, LineStyle.Dotted, 1);
35     PlotSeries( PricePane, hR1, Color.Olive, LineStyle.Solid, 1 );
36     PlotSeries( PricePane, hPivot, Color.Blue, LineStyle.Solid, 1 );
37     PlotSeries( PricePane, hS1, Color.Fuchsia, LineStyle.Dotted, 1 );
38     PlotSeries( PricePane, hS2, Color.Red, LineStyle.Solid, 1 );
39
40     // Значение прорыва уровня (будет добавлено и вычтено из уровней поддержки/
41     сопротивление)
42
43     double boVal = Bars.SymbolInfo.Tick * NumTicks; //в нашем случае - 2 минимальных
44     шага цены (тика)
45
46     // Find the StartBar of the 2nd day
47     int StartBar = 0;
48     int cnt = 0;
49 }
```

```

50     for(int bar = 0; bar < Bars.Count; bar++)
51     {
52         if( Bars.IntradayBarNumber(bar) == 0 ) cnt++;
53         if( cnt == 2 )
54         {
55             StartBar = bar;
56             break;
57         }
58     }
59
60     for(int bar = StartBar; bar < Bars.Count; bar++)
61     {
62
63         // Не торговать на первых двух барах в начале дня (на барах с номерами "0" и
64         "1"
65
66         if( Bars.IntradayBarNumber(bar) < 2 ) continue; //переходим к следующей
67         итерации цикла
68
69         if (IsLastPositionActive)
70         {
71             Position p = LastPosition;
72
73             if( p.PositionType == PositionType.Long )
74             {
75                 if( !SellAtStop(bar + 1, p, hS1[bar], "StopLoss" ) )
76
77                     SellAtLimit(bar + 1, p, hR2[bar], "PftTgt" );
78             }
79
80             else if( !CoverAtStop(bar + 1, p, hR1[bar], "StopLoss" ) )
81
82                 CoverAtLimit(bar + 1, p, hS2[bar], "PftTgt" );
83         }
84
85         else if( Bars.IsLastBarOfDay( bar ) == false ) //если не является последним
86         баром дня
87
88         {
89             bool NewPositionEntered = false; //произведен ли вход в новую позицию
90
91             if( Close[bar] < hR1[bar] )
92
93                 NewPositionEntered = BuyAtStop(bar + 1, hR1[bar] + boVal) != null;
94
95             if( !NewPositionEntered )
96
97                 if( Close[bar] > hS1[bar] )
98
99                     ShortAtStop(bar + 1, hS1[bar] - boVal);
100         }
101     }
102 }

```

В результате на графике внутри дня мы можем просматривать пивот-уровни, которые рассчитываются по дневным барам.



Как: наложить дневные свечи на внутридневной график

Следующий пример показывает Вам как можно отобразить на внутридневном графике искусственные дневные свечи.

Пример (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть) 

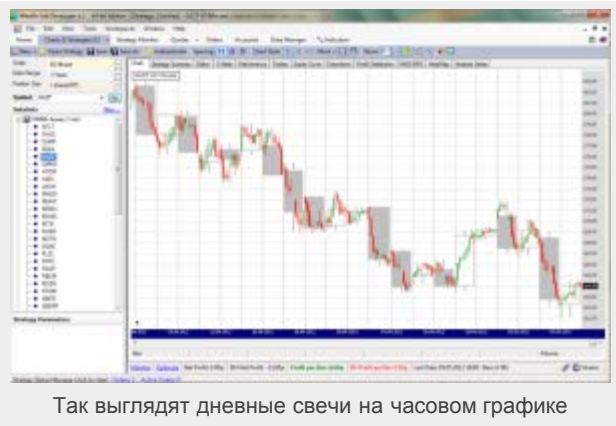
```

1     protected override void Execute()
2     {
3         HideVolume();
4         // Убедимся в том, что работаем с внутридневными данными:
5         if ( !Bars.IsIntraday ) return;
6
7         // Получаем дневные бары:
8
9         SetScaleDaily();
10        Bars iDay = Bars; //присваиваем переменной iDay типа Bars бары дневного графика
11        RestoreScale();
12
13        iDay = Synchronize( iDay ); // Стандартная (быстрая) синхронизация
14
15        for(int bar = 1; bar < Bars.Count; bar++)
16        {
17
18            if ( Bars.IsLastBarOfDay(bar) || (bar == Bars.Count-1) ) //если текущий бар
19            является последним баром дня или последним баром на графике
20
21            {
22
23                int o = bar - Bars.IntradayBarNumber( bar ); // рассчитываем номер первого
24                бара текущего дня
25
26                double p1 = Open[o]; //Цена открытия первого бара текущего дня
27                double p2 = Close[bar]; //Цена закрытия текущего дня
28                double p3 = iDay.High[bar]; //максимальная цена текущего дня

```

```
29         double p4 = iDay.Low[bar]; //минимальная цена текущего дня
30
31         double[] rectangle = { o, p1, o, p2, bar, p2, bar, p1 }; // создаем массив
32 с координатами 4-х точек, рисующих прямоугольник тела свечи (против часовой стрелки)
33
34         int avgBar = bar - Bars.IntradayBarNumber(bar)/2; //находим середину дня:
35
36         if( p2 > p1 ) //если цена закрытия больше цены открытия (если свеча белая)
37         {
38             DrawPolygon( PricePane, Color.Silver, Color.Empty,
39 WealthLab.LineStyle.Invisible, 2, true, rectangle ); //рисуем прямоугольник,
40 используя заготовленный массив (тело свечи)
41             DrawLine( PricePane, avgBar, p4, avgBar, p1, Color.Silver,
42 WealthLab.LineStyle.Solid, 3 ); //рисуем линию от минимума до октрытия (нижнюю тень
43 свечи)
44             DrawLine( PricePane, avgBar, p3, avgBar, p2, Color.Silver,
45 WealthLab.LineStyle.Solid, 3 ); //рисуем линию от максимума до закрытия (верхнюю тень
46 свечи)
47         }
48
49         else //если цена закрытия меньше цены открытия (если свеча черная)
50         {
51             DrawPolygon( PricePane, Color.Silver, Color.Silver,
52 WealthLab.LineStyle.Invisible, 2, true, rectangle ); //рисуем тело свечи (закрашивая
53 его)
54             DrawLine( PricePane, avgBar, p3, avgBar, p4, Color.Silver,
55 WealthLab.LineStyle.Solid, 3 ); //рисуем линию от низа до верха (сразу две тени,
56 т.к. тело свечи непрозрачно)
57         }
58     }
59 }
60 }
```

Вот так будет выглядеть этот график.



Так выглядят дневные свечи на часовом графике

Тут сразу видно и внутридневной график и свечи дневного графика. Это демонстрирует мощность работы программы Wealth-Lab.

Intraday - Weekly, Monthly

К сжатию данных нужно подходить разумно. Конечно, внутридневную шкалу графика можно без проблем сжать до дневной либо даже до недельной или месячной шкалы. Однако нужно учитывать размер данных на внутридневном графике с исходным таймфреймом. К примеру, Вам понадобится более 8000 одноминутных баров для создания всего лишь одной месячной свечи.

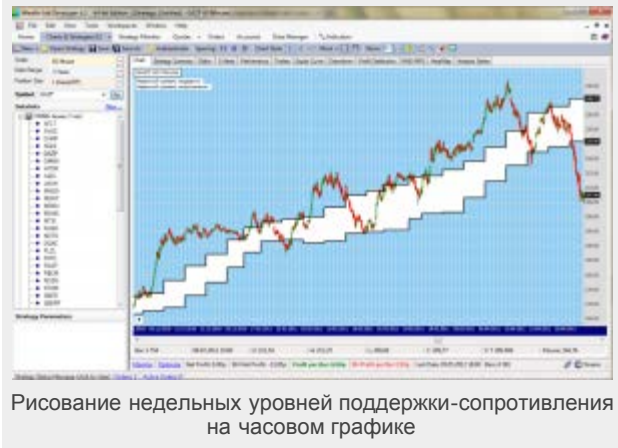
Как: отобразить на внутридневном графике недельные уровни поддержки и сопротивления

В стратегии возможно использовать не только старшие таймфреймы на внутридневном графике или дневную шкалу. Еще одним шагом вперед станет использование недельных или даже месячных данных. В следующем примере будет продемонстрировано как с графика, построенного на внутридневном таймфрейме взглянуть на уровни поддержки и сопротивления, построенные по недельным барам. Эти уровни поддержки и сопротивления формируются как скользящие средние максимальных и минимальных цен недельных баров и отражаются на внутридневных графиках.

Пример (как выполнить пример приведенного кода)...

```
Код примера на C# (кликните чтобы посмотреть)
1 protected override void Execute()
2 {
3     // Ищем недельные уровни поддержки - сопротивления на внутридневных графиках
4     // Уровни поддержки / сопротивления это скользящие средние максимальных и
5 минимальных цен недельных баров
6
7     if( Bars.Scale != BarScale.Minute ) return; //если шкала не внутридневная -
8 ничего не делаем
9
10    SetScaleWeekly(); //сжимаем шкалу до недельной
11
12    DataSeries WeeklySup = SMA.Series( Bars.Low, 5 );
13    DataSeries WeeklyRes = SMA.Series( Bars.High, 5 );
14
15    RestoreScale(); //восстанавливаем шкалу в первоначальное состояние
16
17    WeeklySup = Synchronize( WeeklySup );
18    WeeklySup.Description = "Недельный уровень поддержки";
19
20    WeeklyRes = Synchronize( WeeklyRes );
21    WeeklyRes.Description = "Недельный уровень сопротивления";
22
23    HideVolume();
24
25    for(int bar = 0; bar < Bars.Count; bar++)
26
27        SetBackgroundColor( bar, Color.LightSkyBlue );
28
29    PlotSeriesFillBand(PricePane, WeeklySup, WeeklyRes, Color.Black, Color.White,
30 LineStyle.Solid, 2); //рисуем на внутридневном графике уровни
31 }
```

График этой стратегии будет выглядеть так:



Как: получить доступ к ряду данных индикатора, построенного по недельным барам из графика дневного таймфрейма (Daily - Weekly)

Для того, чтобы проиллюстрировать идею, следующая стратегия покупает когда дневная MACD пересекает нулевой уровень снизу вверх и продает когда недельный MACD пересекает нулевой уровень сверху вниз.

Пример (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть)

```
1 | protected override void Execute()
2 | {
3 |     /*
4 |     Открываем длинную позицию при пересечении дневным MACD нулевого уровня снизу вверх
5 |     Позиция закрывается когда недельный MACD пересекает нулевой уровень сверху вниз
6 |     */
7 |
8 |     if( Bars.Scale != BarScale.Daily ) return; //если шкала не дневная, то ничего не
9 |     делаем.
10 |
11 |     SetScaleWeekly(); //сжимаем шкалу до недельного уровня
12 |
13 |     DataSeries WeeklyMACD = MACD.Series( Bars.Close );
14 |
15 |     RestoreScale(); //восстанавливаем шкалу в исходное (дневное) состояние
16 |
17 |     WeeklyMACD = Synchronize( WeeklyMACD );
18 |     WeeklyMACD.Description = "Weekly MACD";
19 |
20 |     for(int bar = 60; bar < Bars.Count; bar++)
21 |     {
22 |         if (IsLastPositionActive)
23 |         {
24 |             if( CrossUnder( bar, WeeklyMACD, 0 ) )
25 |
26 |                 SellAtMarket( bar+1, LastPosition, "Недельный MACD спустился ниже 0" );
27 |
28 |         }
29 |         else
30 |         {
31 |
32 |             if( CrossOver( bar, MACD.Series( Bars.Close ), 0 ) )
33 |
34 |                 BuyAtMarket( bar+1, "Дневной MACD поднялся выше 0" );
35 |
36 |         }
37 |     }
38 |
39 |     HideVolume();
40 |
41 |     ChartPane WeeklyMACDPane = CreatePane( 30, false, true );
42 |     DrawHorzLine( WeeklyMACDPane, 0, Color.Red, WealthLab.LineStyle.Solid, 2 );
43 |     PlotSeries( WeeklyMACDPane, WeeklyMACD, Color.Blue, WealthLab.LineStyle.Histogram,
44 |         2 );
45 | }
```

Как: получить доступ к ряду данных индикатора, построенного по месячным барам из графика дневного таймфрейма (Daily - Monthly)

Для получения доступа к недельным и месячным данным из стратегии, которая работает с дневными данными, нужно проделать следующие шаги:

- вызовите методы SetScaleWeekly() и SetScaleMonthly() соответственно;
- после того, как Вы закончите оперировать данными в старшем таймфрейме, вызовите метод RestoreScale();
- следующим шагом станет вызов нового синхронизированного ряда данных (DataSeries) либо нового объекта баров (Bars), которые активируются при вызове метода Synchronize().

Здесь приводится код простой стратегии, которая использует в приказе сразу три таймфрейма (дневной, недельный и месячный) для того, чтобы принять решение.

Если и недельный и месячный индикатор CCI больше чем 100, а дневной CCI пересекает уровень 100 снизу вверх, то создается приказ на вход на следующем баре по рынку. Эта длинная позиция существует до тех пор, пока дневной CCI не пересечет уровень -200 сверху вниз.

Пример (как выполнить пример приведенного кода)...

Код примера на C# (кликните чтобы посмотреть)

```
1 | using System;
2 | using System.Collections.Generic;
3 | using System.Text;
4 | using System.Drawing;
```

```

5  using WealthLab;
6  using WealthLab.Indicators;
7
8  namespace WealthLab.Strategies
9  {
10     public class CCItriplex : WealthScript
11     {
12         private StrategyParameter cciPeriod; // parameter for period
13
14         public CCItriplex()
15         {
16             cciPeriod = CreateParameter("CCI Period", 10, 10, 60, 5);
17         }
18
19         protected override void Execute()
20         {
21             int Period = cciPeriod.ValueInt;
22
23             if (Bars.Scale != BarScale.Daily) return; //если шкала не является
дневной, то ничего не делаем
25
26             DataSeries cciD = CCI.Series(Bars, Period); //дневной CCI
27
28             SetScaleWeekly(); //сжимаем шкалу до недельной
29
30             DataSeries cciW = CCI.Series(Bars, Period); //недельный CCI
31
32             RestoreScale(); //приводим шкалу в первоначальный вид
33
34             SetScaleMonthly(); //сжимаем шкалу до месячной
35
36             DataSeries cciM = CCI.Series(Bars, Period); //месячный CCI
37
38             RestoreScale(); //приводим шкалу в первоначальный вид
39
40             cciW = Synchronize(cciW);
41             cciW.Description = "Недельный CCI";
42
43             cciM = Synchronize(cciM);
44             cciM.Description = "Месячный CCI";
45
46             ChartPane cciPane = CreatePane(50, true, true);
47
48             PlotSeries(cciPane, cciD, Color.Green, WealthLab.LineStyle.Solid, 1);
49             PlotSeries(cciPane, cciW, Color.Blue, WealthLab.LineStyle.Solid, 1);
50             PlotSeries(cciPane, cciM, Color.Red, WealthLab.LineStyle.Solid, 1);
51
52             for (int bar = Period; bar < Bars.Count; bar++)
53             {
54                 if (IsLastPositionActive)
55                 {
56                     if (CrossUnder(bar, cciD, -200))
57
58                         SellAtMarket(bar + 1, LastPosition);
59
60                 }
61
62                 else if ((cciW[bar] >= 100) && (cciM[bar] >= 100) && CrossOver(bar,
63 cciD, 100))
64
65                     BuyAtMarket(bar + 1);
66
67             }
68         }
69     }
70 }

```

Вот так будет выглядеть график:



Сегодня мы рассмотрели очень большую и интересную тему - анализ данных в разрезе разных таймфреймов в рамках одной торговой стратегии. Овладев этими несложными методами работы с разными таймфреймами и научившись сжимать шкалу графика до нужного Вам состояния Вы сможете создавать стратегии позволяющие Вам видеть одновременно несколько измерений рынка и ситуации на нем. Следующий раз мы рассмотрим довольно специфический класс WealthLab.Rules. Чтобы не пропустить новые статьи - подписывайтесь на обновления по RSS.

Если Вам понравилась эта статья - подпишитесь на **новые СТАТЬИ по RSS** или получайте наши **новые СТАТЬИ на E-mail**:

Рекомендую почитать также:

- [Создание многопозиционных стратегий в Wealth-Lab](#)
- [Программирование однопозиционных стратегий \(SP Strategies\) в программе Wealth-Lab](#)
- [Программирование выходов, основанных на времени удержания позиции в Wealth-Lab](#)
- [Ловушки при построении торговых систем, связанные с проблемой подглядывания в будущее](#)
- [Использование обоснованного заглядывания в будущее при программировании торговых стратегий в Велс Лаб](#)

Инструкция по программированию торговых стратегий в WealthLab | Финансовая лаборатория пишет:

20 июня 2012 в 12:23

[...] 12. Анализ данных в Wealth-Lab с учетом разных таймфреймов [...]

Ответить

Максим пишет:

17 сентября 2012 в 10:15



Дмитрий, поясни, пожалуйста, зачем нужен при расчете уровней пивот этот цикл?

// Find the StartBar of the 2nd day

```
int StartBar = 0;
```

```
int cnt = 0;
```

```
for(int bar = 0; bar < Bars.Count; bar++)
{
    if( Bars.IntradayBarNumber(bar) == 0 ) cnt++;
    if( cnt == 2 )
    {
        StartBar = bar;
        break;
    }
}
```

Ведь мы этого этого и так рассчитали все уровни.

Ответить

Оставить комментарий

Имя (Обязательно)

Почта (Обязательно)

Вэб-сайт

Получать новые комментарии по электронной почте. Вы можете [подписаться](#) без комментирования.

© 2012, Финансовая лаборатория. Все права защищены.

Любое использование материалов допускается только с указанием активной ссылки на источник.

491

245

190

В день мой блог посещает 77 человек, сегодня 126

А вы?

mail.ru РЕЙТИНГ

757157542196

BY FEEDBURNER

Подписка

[RSS](#)
[Twitter](#)
[ВКонтакте](#)
[facebook](#)

Сайт

[Создать блог](#)
[Содержание](#)
[О блоге](#)
[Карта](#)

Market Manager – управляем торговыми площадками в Wealth-Lab

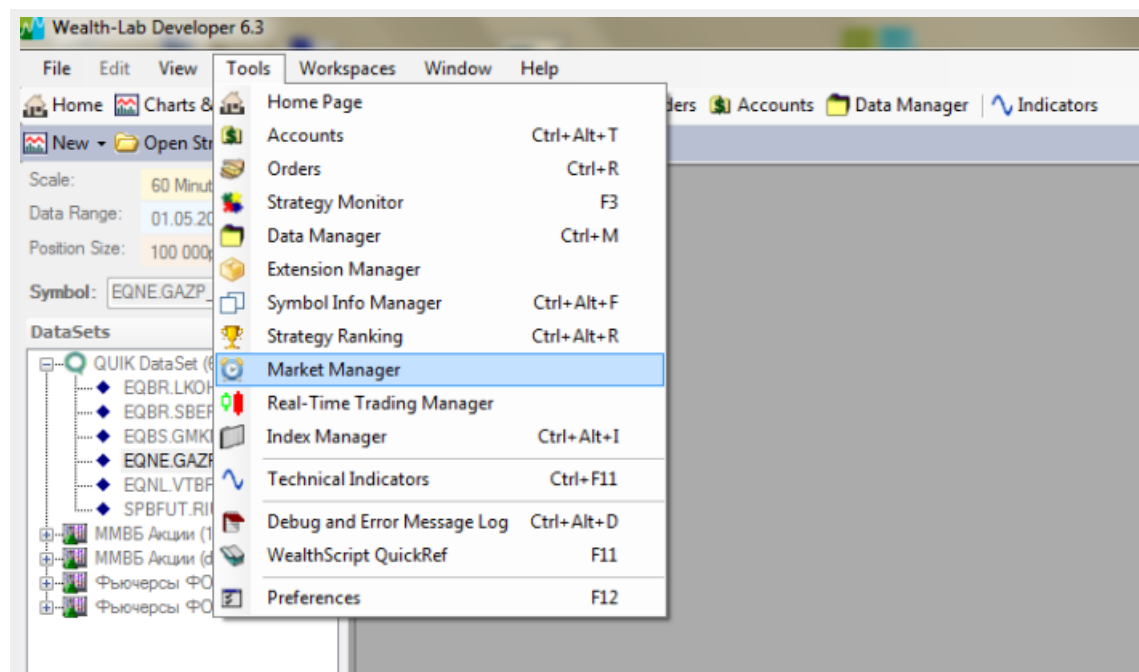
 АВТОР ДМИТРИЙ ВЛАСОВ  ИЮНЯ - 18 - 2012  НЕТ КОММЕНТАРИЕВ

Когда я начал настраивать Wealth-Lab для создания связи с программой QUIK в качестве отдельной задачи значилось настройка утилиты Market Manager. Я начал искать информацию об этой утилите и кроме отрывочных сведений ничего конкретного не нашел. К счастью, у Wealth-lab есть специальная база знаний, в которой можно найти буквально любую информацию, правда на английском языке. Поэтому я решил перевести [статью, посвященную Market Manager](#). Если кому-нибудь пригодится - буду рад.



Что такое Market Manager?

Market Manager - это специальная утилита к программе Wealth-Lab, запустить которую можно через главное меню:



Вызов Market Manager через главное меню Wealth-Lab

Эта утилита дает возможность пользователям WLD 6 учитывать особенности торговых площадок, на которых торгуются финансовые инструменты.

Для того, чтобы лучше понять - что такое Market Manager - представьте, что это посредник между провайдером данных и программой Wealth-Lab.

Информация о ценах и объемах, которую Вы получаете от поставщика данных, хранится в провайдере данных. И когда Вы используете Market Manager, эти данные проходят предварительную обработку "на лету" с помощью этой утилиты, и лишь затем попадают в программу Wealth-Lab.

Для чего нужен Market Manager?

[RSS товары](#)



[RSS новости](#)



[Twitter](#)



Последние комментарии

макс: А где взять свечки с открытым интересом? на сайте финама нет



ring10: Добрый день Дмитрий Как сделать, чтобы открывался 1 велф лаб? Если открываешь под пользователем, т [...]



Дмитрий Власов: Буян, рад, что Вам пригодились материалы данной статьи!



Максим: Сам себе отвечу. protected override void Execute() { Bars rts = GetExternalSymbol("RTS", t [...]



Буян: Спасибо Дмитрий за статью! Мне удалось разработать хорошую рабочую торговую систему, используя возмо [...]



Последние посты



Бесплатный вебинар: 7 смертных грехов при оптимизации торговых систем

НОЯБРЯ - 5 - 2012



Отзывы участников курса до результата Wealth-Lab. С нуля до первой торговой системы за 5 дней.

НОЯБРЯ - 3 - 2012



Быстрее, Выше, Сильнее: Обновление Wealth-Lab до новой версии 6.4

ОКТАБРЯ - 23 - 2012



Курс до результата: Wealth-Lab – от нуля до первой торговой системы за 5 дней

ОКТАБРЯ - 23 - 2012



Обзорная экскурсия по достопримечательностям программы Wealth-Lab

ОКТАБРЯ - 20 - 2012

Берите

Зачем вообще нужно предварительно обрабатывать данные о ценах и объемах? Вот перечисление нескольких случаев, при которых нужна предварительная обработка данных:

- Вы хотите скрыть данные из предторгового и послеторгового периода (точно также Вы можете скрыть первые минуты торгов, когда цена очень волатильна и по цене которой практически невозможно войти в рынок).
- Данные содержат бары, которые здесь вообще не должны содержаться.
- Нужно просто определить разные торговые площадки и настроить их свойства (такие, как время открытия, время закрытия, нерабочие дни и т.п.).

Настройки утилиты Market Manager

Окно утилиты Market Manager разбито на две области:

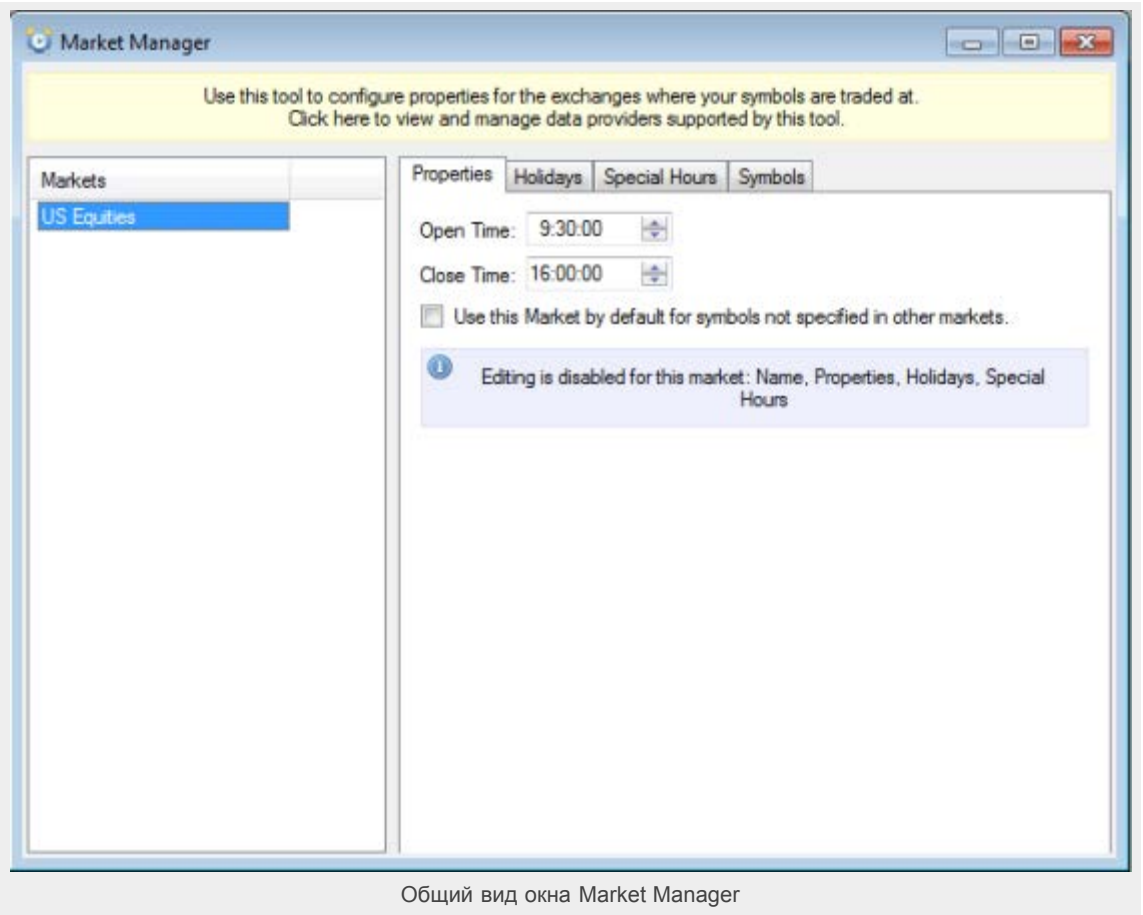
- в левой области отображается список торговых площадок
- в правой области находятся элементы управления, с помощью которых Вы сможете настроить свойства торговых площадок.

Для того, чтобы помочь Вам разобраться с утилитой Market Manager - по умолчанию уже существует одна площадка с названием "US Equities".

Обратите внимание на то, что Вы сможете изменить только список финансовых инструментов, которые торгуются на этой площадке, все остальные опции останутся недоступными для редактирования.

Если Вы кликните правой кнопкой в левой области окна Market Manager, то появится контекстное меню, с помощью которого Вы сможете:

- создать новую торговую площадку;
- переименовать уже существующую торговую площадку;
- удалить существующую торговую площадку (это можно сделать также выделив существующую площадку и нажав на кнопку Delete).



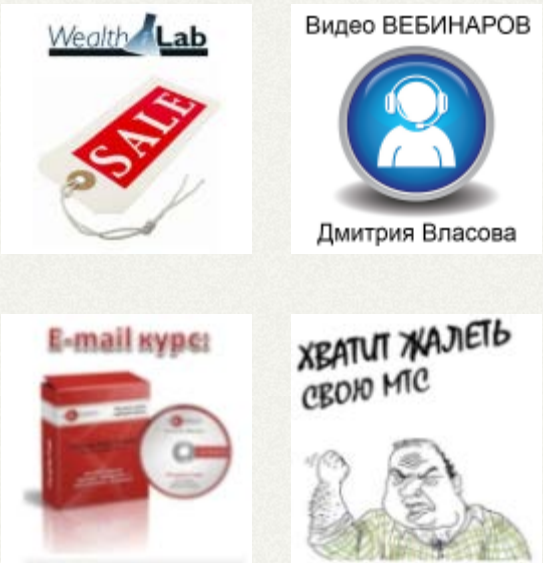
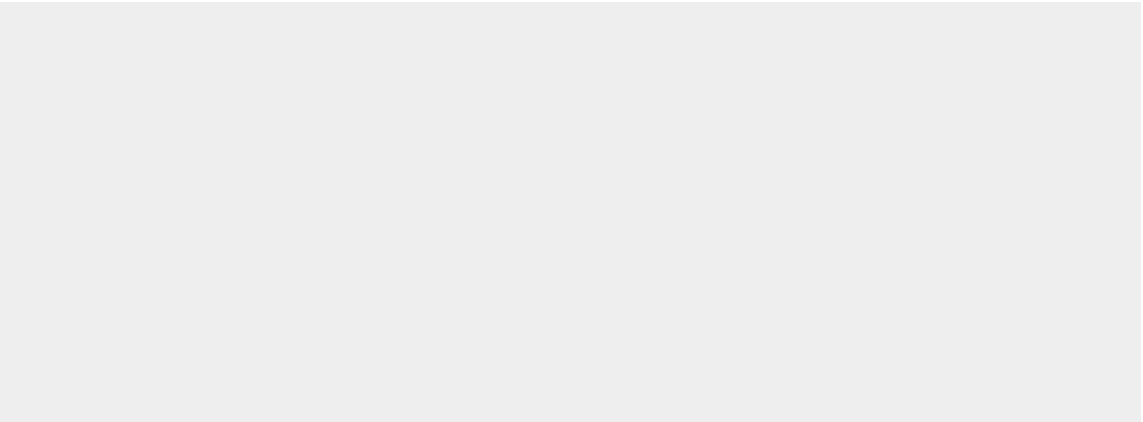
Общий вид окна Market Manager

Желтая полоска, которая находится сверху (сразу под названием формы и чуть выше левой и правой области утилиты Market Manager) является кликабельной.

После щелчка мышкой по этой желтой полоске, появится диалог, в котором отобразится список установленных провайдеров данных (data providers), которые поддерживают работу с утилитой Market Manager.

На текущий момент не все провайдеры данных совместимы с этой утилитой, однако постепенно такая совместимость будет внедряться. Точно поддерживают эту утилиту провайдеры данных FINAM и QUIKStaticProvider (от компании Real- Time Trading).

Поставив либо сняв галочку напротив соответствующего провайдера данных Вы сможете глобально включить либо отключить для данного провайдера данных эффект от использования утилиты Market Manager.





МТС под микроскопом:
WLMA System
Подробный разбор торговой системы WLMA System
3900 руб.
[В корзину](#)

Рубрики

Выберите рубрику

Обновления:

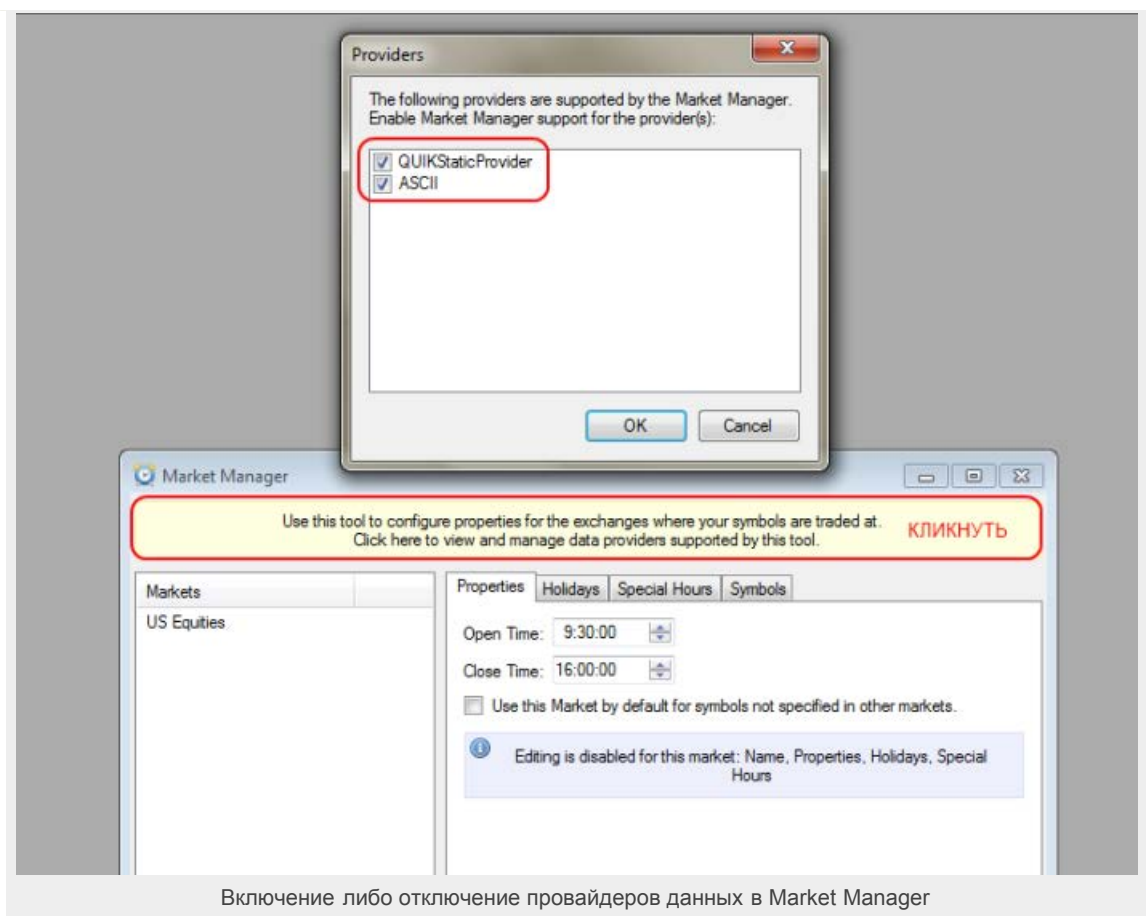
-  Новые посты по Email
-  Новые посты по RSS
-  Комментарии по RSS
-  Следим на Twitter

Метки

С# HowTo: **Wealth-Lab**
Видео **Индикаторы** **МТС** **Торговые**
роботы **Юмор** **алготрейдинг**
мероприятия **обучение**

Управление

- Регистрация
- Вход
- RSS Записей
- RSS Комментариев
- WordPress.org



Включение либо отключение провайдеров данных в Market Manager

Свойства утилиты Market Manager

Пользователи более ранних версий программы Wealth-Lab конечно же вспомнят, что в зависимости от источника данных, к которому была подключена торговля в реальном времени, контроль над загружаемыми данными можно было менять по обстоятельствам.

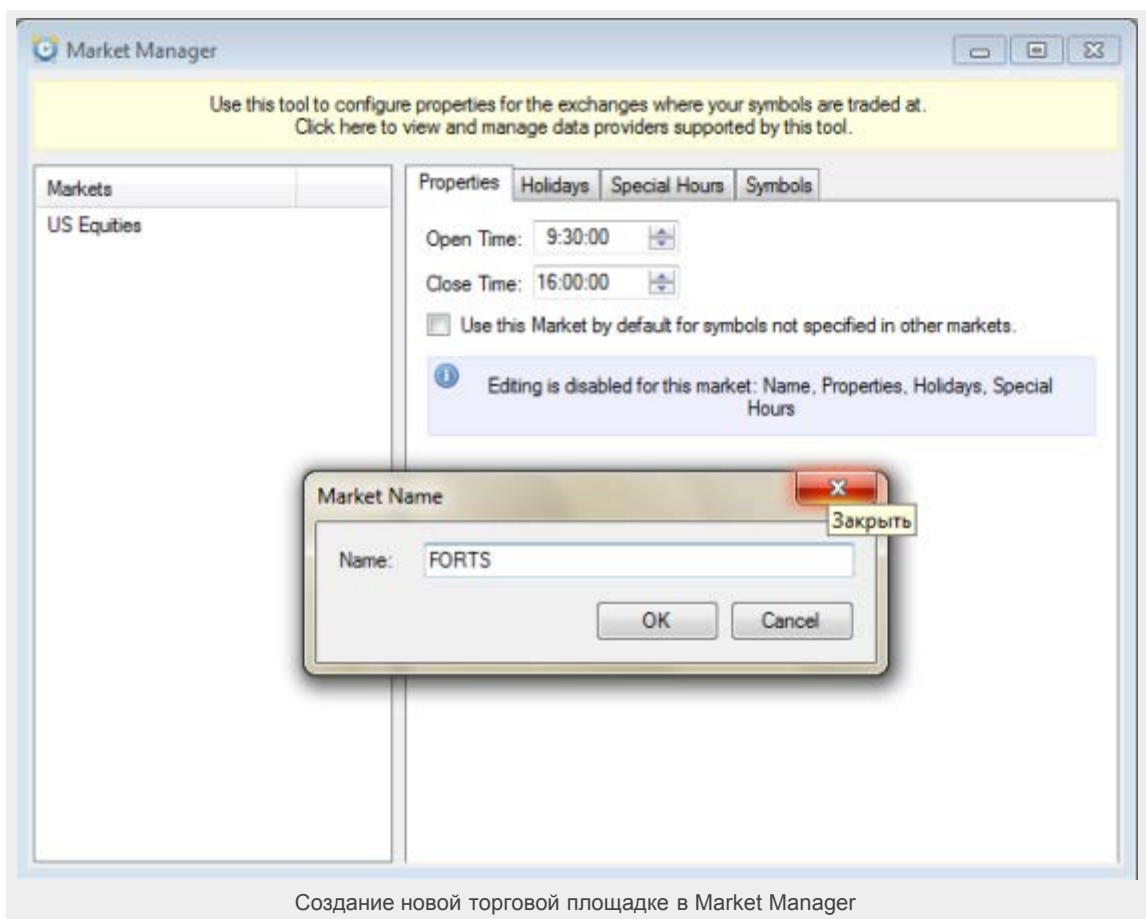
Среди этих возможностей была возможность выбирать часы торговли и ставить фильтры по времени, чтобы не торговать в реальном времени в те часы, когда Вы этого не хотите.

В дополнение к этим, уже привычным, функциям, фильтрация данных с помощью утилиты MarketManager применима не только к потоковым данным для реальной торговли, но и к историческим данным.

Благодаря этой утилите многочисленные запросы пользователей о создании фильтра для исторических данных, которыми пользуется программа Wealth-Lab были полностью удовлетворены.

Вы можете настроить время торговли для конкретной торговой площадки и отфильтровать данные, которые не входят в этот временной промежуток. Это будет полезными, когда Вы, к примеру, не хотите брать на себя риск низкой ликвидности, широких спредов между спросом и предложением и высокой волатильности.

Особенно это касается первых минут торговли на российском срочном рынке FORTS. Имеет смысл игнорировать предторговую и послеторговую активность, а также первые минуты торгов на таких площадках.



Создание новой торговой площадке в Market Manager

Возможность определить "Время открытия" и "Время закрытия" торговой площадки используется именно для этих целей и является отличительной особенностью утилиты Market Manager.

Определяя "Время открытия" и / или "Время закрытия" отличные от "0:00" для конкретной торговой площадки, Вы автоматически фильтруете данные для всех финансовых инструментов, которые торгуются на данной площадке.

Wealth-Lab скроет все данные вне временных рамок, которые были заданы для этой торговой площадки.

Возможно, Вам потребуется обновить график с историческими данными после того, как Вы внесете изменения в настройку утилиты Market Manager.

Если Вы хотите отключить фильтрацию по времени, просто установите "Время открытия" и "Время закрытия" на конкретной торговой площадке равные "0:00".

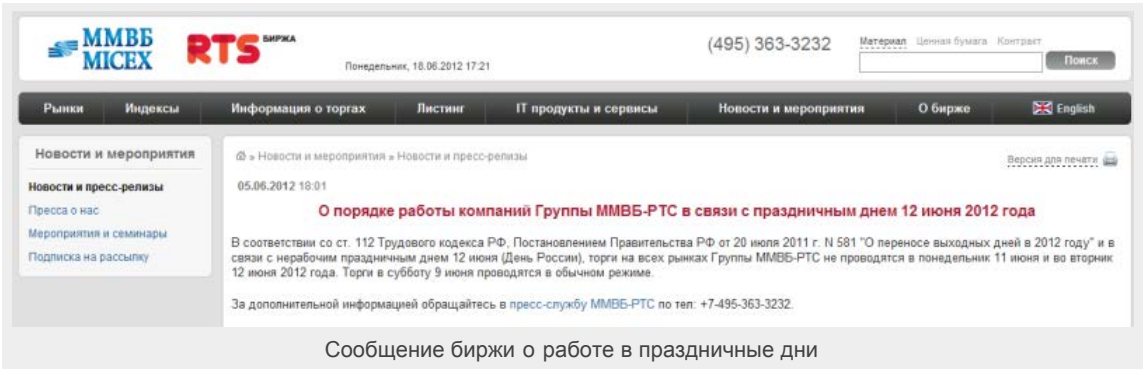
Если Вы поставите галочку напротив опции "Use this Market by default for symbols not present in other markets", то Вы указываете программе использовать настройки данной торговой площадки по умолчанию для всех финансовых инструментов, которые не представлены на других торговых площадках.

Таким образом, Wealth-Lab будет обходиться со всеми новыми, пока еще не отнесенными к какой-либо конкретной торговой площадке финансовыми инструментами точно также, как и к инструментам, представленным на этой торговой площадке.

Праздничные дни (Holidays)

Цель этой таблицы понятна из названия. Таблица "Holidays" предназначена для исключения с графика тех дней недели, в которые торговля не ведется.

Посмотрите, к примеру, сообщение биржи о работе в праздничные дни.

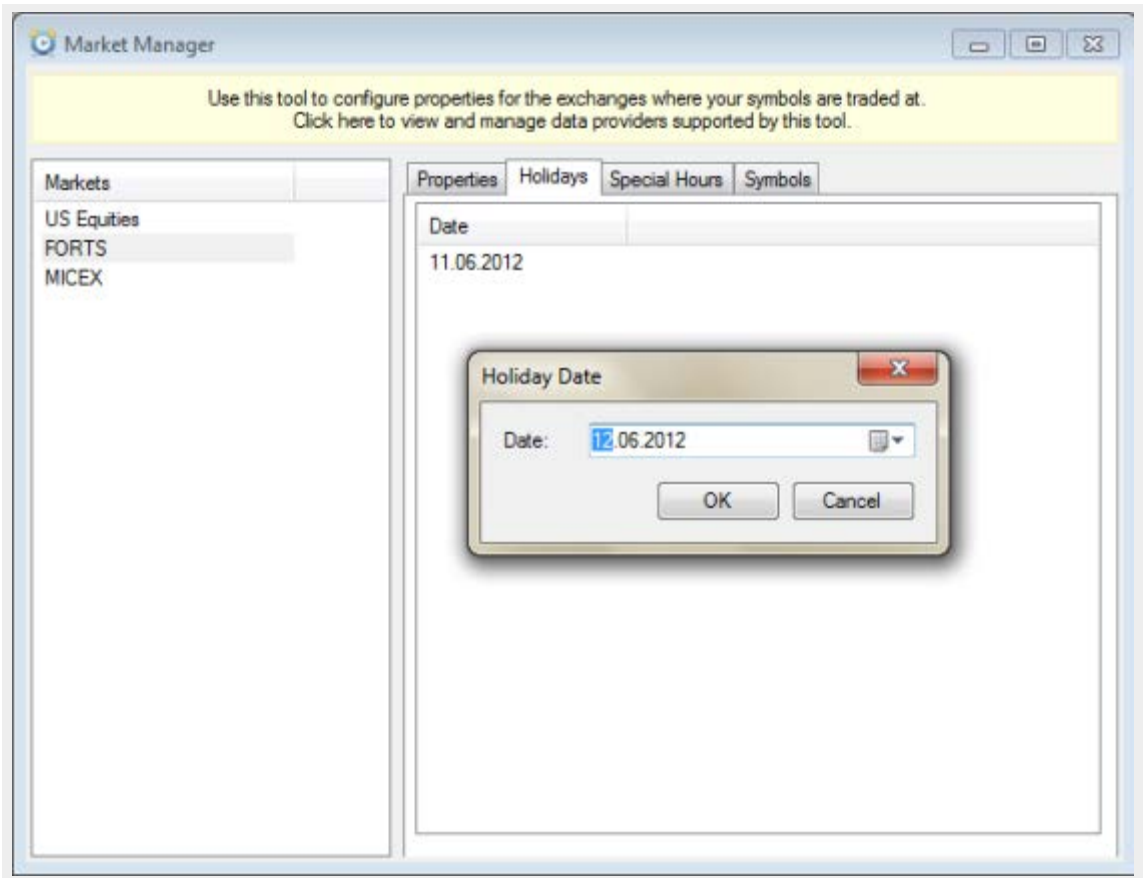


Сообщение биржи о работе в праздничные дни

Если вдруг Ваши данные содержат те дни, которых не должно быть на графике - их легко исключить.

- Просто выбираете вкладку "Holidays" утилиты Market Manager,
- нажимаете правой кнопкой мыши в правой области окна
- из контекстного меню выбираете "New" для того, чтобы определить нежелательный бар.

Благодаря этим действиям целый торговый день будет устранен с Ваших графиков и не будет обрабатываться торговой стратегией.



Настройка вкладки "Holidays" - для указания нерабочих дней

Специальные часы (Special Hours)

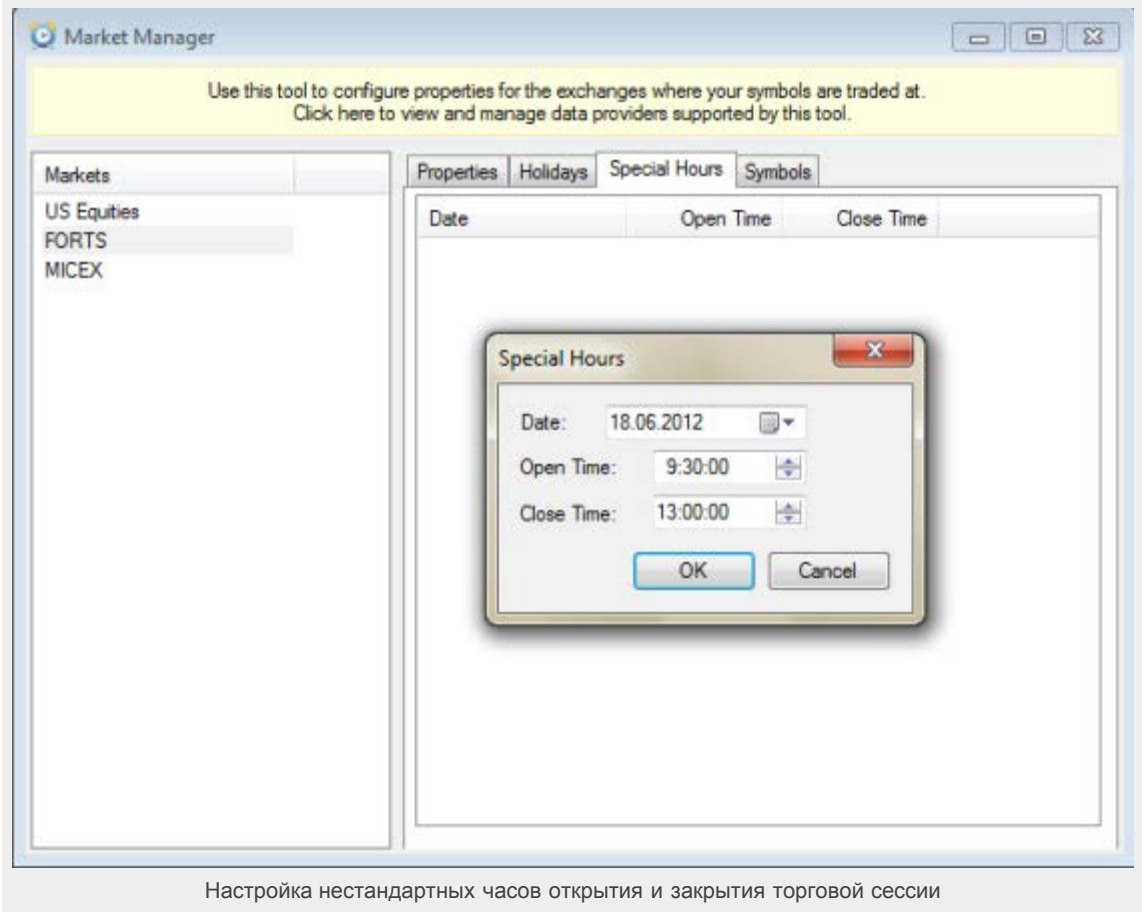
Эти фильтры позволят Вам управлять временем в дни, когда действуют укороченные торговые сессии (это характерно для праздничных и предпраздничных дней). В такие укороченные сессии время открытия и закрытия торгов может сдвигаться по сравнению с обычными днями.

Если Вы хотите начинать торговать в такие "короткие сессии" в определенное время - просто

- перейдите на вкладку "Special Hours";
- нажмите правой кнопкой по правой области таблицы;
- из контекстного меню выбираете пункт "New"
- заполняете дату, время открытия и время закрытия
- нажимаете на кнопку "OK"

Эти действия позволят Вам установить специфическое время торговой сессии.

Как и обычно, для редактирования данных вызовите правой мышью контекстное меню и выберете "Edit". Для удаления выберете из контекстного меню "Delete".



Финансовые инструменты (Symbols)

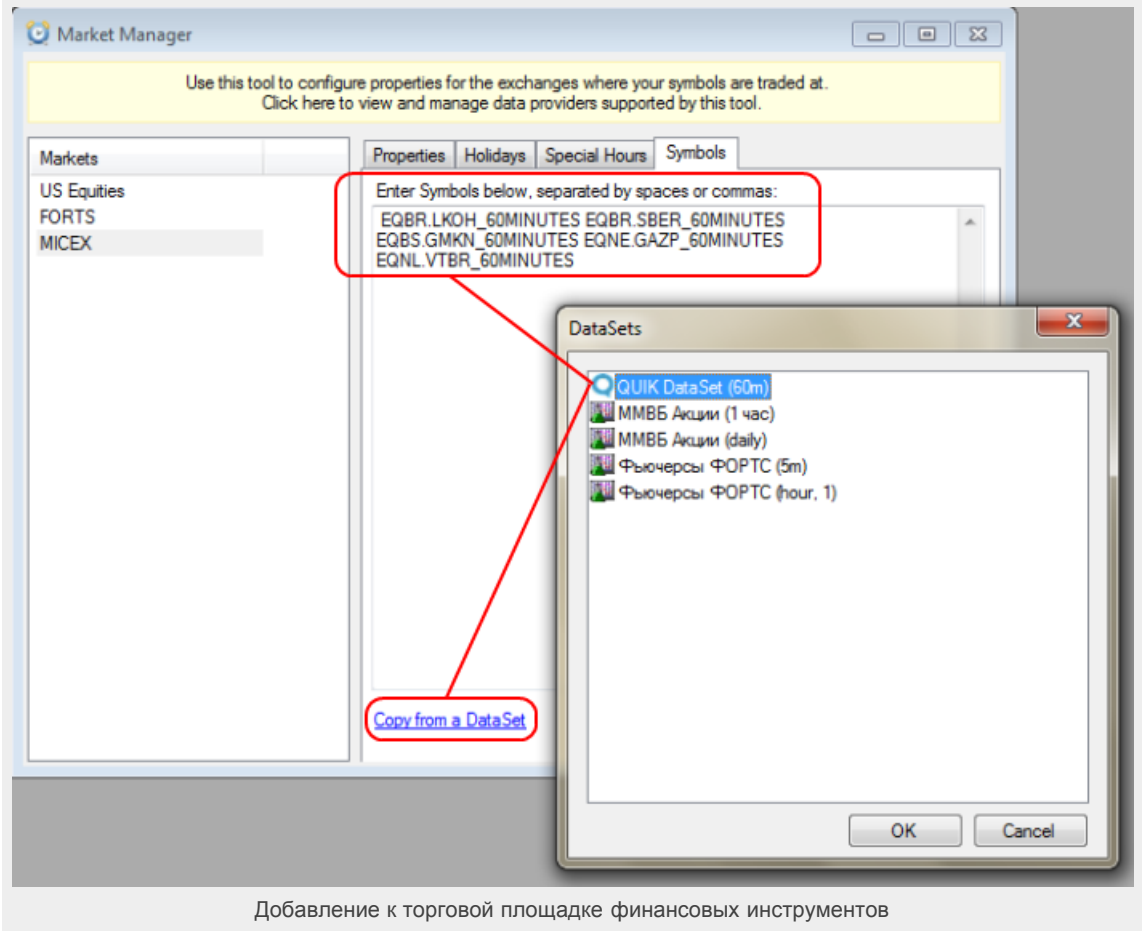
Последняя вкладка позволяет определить - какие финансовые инструменты относятся к данной торговой площадке. Эти финансовые инструменты будут подчиняться тем же правилам, которые Вы установите для данной торговой площадки.

Для того, чтобы отнести конкретные финансовые инструменты к данной торговой площадке, нужно просто внести их название на вкладку SYMLOS, разделив их пробелами.

Способ работы с финансовыми инструментами на вкладке Symbols утилиты Market Manager очень похож на тот способ, который Вам уже привычен при работе с наборами данных (DataSet) программы Wealth-Lab.

Вы можете напечатать название тикеров вручную, или же просто кликнув по ссылке "Copy from a DataSet", сохранив себе время.

Если вдруг один и тот же финансовый инструмент окажется на нескольких торговых площадках, Market Manager сообщит Вам об этом и поможет таким образом разрешить конфликт.



Как отключить утилиту Market Manager

Несмотря на то, что обработка данных с помощью утилиты Market Manager занимает очень незначительное время даже для очень больших выборок, Вы можете ее отключить.

Если Вы сделаете действия, описанные ниже, данные не будут подвергаться обработке с помощью утилиты Market Manager.

Частично: для того, чтобы отключить действие утилиты Market Manager для определенных торговых площадок, просто удалите финансовые инструменты, которые торгуются на этой площадке с вкладки

"Symbols", либо просто удалите эту торговую площадку из утилиты Market Manager.
Полное: Для того, чтобы не подвергать данные конкретного провайдера данных воздействию утилиты Market Manager - кликните желтую полосу, о которой уже говорилось в данной статье и снимите галочку напротив конкретного провайдера данных.

Замечания, проблемы и ограничения в использовании Market Manager

Если использование утилиты Market Manager не вызывает никаких проблем, Вы можете просто пропустить данный раздел.
Ошибка: Когда включены потоковые данные (Streaming) для конкретного финансового инструмента, перенос этого финансового инструмента на другую торговую площадку в утилите Market Manager может привести к смешиванию потоковых данных с потоковыми данными другого финансового инструмента.
Ограничение: Не показываются данные для внутридневной торговли финансовыми инструментами, которые торгуются круглые сутки в том случае, если время открытия наступает позже, чем время закрытия (например, открытие в 6:00pm, а закрытие в 5:15pm).
Фильтрация происходит вне зависимости от того, определен ли финансовый инструмент в "Symbol Info Manager" или нет.
При конвертации и фильтрации данных Market Manager не затрагивает данные, таймфрейм которых базируется на неделях, месяцах, кварталах и годах. Обрабатываются только данные, базовой шкалой которых являются внутридннные и дневные таймфреймы. Причем для дневных данных применимыми являются только фильтры нерабочих дней.

На сегодня все. Вы получили исчерпывающую информацию о том, как в программе Wealth-Lab работает утилита для настройки особенностей различных торговых площадок Market Manager. Чтобы не пропустить новые статьи по особенностям использования программы Wealth-Lab подписывайтесь на RSS обновления блога "Финансовая лаборатория".

Если Вам понравилась эта статья - подпишитесь на **новые СТАТЬИ по RSS** или получайте наши **новые СТАТЬИ на E-mail**:
Рекомендую почитать также:
• [Подготовка к построению МТС \(настройка Symbol Manager в Wealth Lab\)](#)
• [Финансовые инструменты: как найти российские ликвидные акции и добавить их в Wealth Lab?](#)
• [Как установить программу QUIKLiveTrading для создания связи между торговой системой QUIK и программой технического анализа Wealth-Lab](#)
• [Как использовать QUIK в качестве источника исторических данных для Wealth-Lab](#)
• [Окончательная настройка программ для автоматизации торговли и превращения алертов Wealth-Lab в торговые поручения QUIK](#)

 РУБРИКИ: [WEALTH-LAB](#), [ПРОГРАММЫ ТРЕЙДЕРУ](#)

Оставить комментарий

Имя (Обязательно)

Почта (Обязательно)

Вэб-сайт

Получать новые комментарии по электронной почте. Вы можете [подписаться](#) без комментирования.

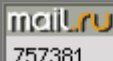
© 2012, Финансовая лаборатория. Все права защищены.
Любое использование материалов допускается только с указанием активной ссылки на источник.



213 👁
75 📄
70 ⬆



В день мой блог
посещает **84**
человека, сегодня
уже **43**



РЕЙТИНГ
757381
245
75

А, т, е, б, ь?

BY FEEDBURNER

Подписка

[RSS](#)
[Twitter](#)
[ВКонтакте](#)
[facebook](#)

Сайт

[Создать блог](#)
[Содержание](#)
[О блоге](#)
[Карта](#)



ТОРГОВЫЕ РОБОТЫ (открытая база знаний)

Главная

Обсуждения

Люди

Главная / Wealth-Lab



Дмитрий задал вопрос 18 августа 2012 в 09:28

Пользователь

С ОТВЕТОМ

Вопрос по фьючерсу в WLD

Здравствуйте, Дмитрий!

Возник вопрос по поводу фьючерсов, которые мы забиваем в Symbol Info Manager. Мы делаем так, как будто фьючерс на индекс PTC с его ценой (150000) является рублевым. То есть как будто он стоит 150000 рублей, правильно? А потом в расчете размера позиции указываем необходимое плечо Margin Factor как 8 к 1.

Вопросы:

1. Мы уходим от проблемы изменения ГО на истории?
2. При этом Point Value = 0,64, что означает неизменный курс доллара на всей истории тестирования. Приходим к проблеме неизменного курса доллара, который на самом деле менялся и достаточно сильно. Например, в 2008 году. Зачем?
3. А что если поставить ГО 333\$ на всем периоде и PointValue = 0,02. Тогда получим результат в долларах за весь период. Не будет ли это более корректным?
4. Как я понимаю, если мы торгуем не на весь депо в ГО, а более-менее консервативно, тогда проблемы изменения ГО на истории вообще не возникает? Например: счет 10 000\$. ставлю ГО 333\$. Размер позиции в очередной сделке допустим 5 контрактов. В ГО уходит 1665\$. Если ГО на истории менялось, ну пусть стало 666\$. Вход на 5 контрактов по счету в 10000\$ будет осуществлен без проблем (в ГО уйдет 3330\$).
5. При переходе в рублевый эквивалент мы реальный размер счета начинаем регулировать через Margin Factor. Но меняющееся на истории ГО по сути будет менять этот Margin Factor. На счету 10000\$. Го 1000. Margin Factor 8 в рублевом эквиваленте прокатывает. го стало 2000. Margin Factor 8 уже не прокатывает. Получается уходим от проблемы ГО на истории, но попадаем на Margin Factor, на который ГО влияет?
6. Зачем концептуально переходить в рублевый эквивалент.

Спасибо

0



У меня этот вопрос. Голосовать!

Создать обсуждение

- Следить за вопросом
- Подписаться на RSS
- Пожаловаться

Поделиться:

Сохранить



Мне нравится

1

Поделиться



Wealth-Lab

6 комментариев

Добавить комментарий



Власов Дмитрий написал более 4 месяцев назад
Создатель сообщества

По поводу оптимизации фьючерса следующий подход можно применять:

1) Действительно в SymbolInfoManager лучше указывать режим не Futures, а Equity. Тогда количество контрактов рассчитывается не исходя из величины Margin (гарантийное обеспечение на один контракт) а исходя из текущей цены финансового инструмента.

На мой взгляд это лучше, т.к. ГО зависит от цены фин. инструмента (для фьючерса на индекс PTC = 12%), а в Wealth-Lab Margin всегда одно и то же число на всем периоде тестирования.

Плюс биржа иногда меняет величину ГО, т.е. есть зависимость еще от волатильности и от праздничных дней.

Поэтому лучше выставить режим для фьючерса на Индекс PTC = Equity. Margin Factor указываем = 8. (чтобы установить предел для максимального плеча по инструменту). А на какой процент от эквити торговать - указываем в модуле управления позицией (PosSizer) в зависимости от характеристик торговой стратегии - ориентируясь на максимально допустимую величину просадки.

2) Установка величины PointValue - никак не будет сказываться на оптимизации. Просто это дает возможность увидеть финансовый результат тестирования не в пунктах, а в рублях.

3. Так нельзя делать, ведь цена финансового инструмента дается в пунктах. Соответственно и величину ГО тоже нужно давать в пунктах.

4. Если Вы используете режим Futures в Symbol InfoManager, все равно Вам с помощью модуля управления позиции нужно управлять размером позиции и Вашими рисками. Величина

ГО влияет тут только на "максимальное плечо", которое Вы можете использовать. Размер же позиции нужно определять не уровнем максимального плеча, а уровнем риска, который Вы готовы терпеть.

5. 6. Еще раз выскажу свое мнение, лучше всего устанавливать для работы с фьючерсами режим "Equiti" в SymbolInfoManager, Величину MarginFactor (максимальное плечо) находить расчетным путем: $100 / (\text{ГО в рублях} / \text{рублевая стоимость контракта} * 100)$. Фактическое плечо устанавливать в % от Equiti в зависимости от характеристик торговой стратегии и максимальному риску, который Вы готовы терпеть.

Если Equity устанавливаете в размере 100% - торгуете без плеча, если 200% - на один рубль Ваших средств используете 1 рубль заемных, если 300% - то на одну Вашу часть используете 2 заемных части и т.п.

★ Это — ответ на вопрос! Ответить Пожаловаться



Игорь Афанасьев написал более 4 месяцев назад
Пользователь

Дмитрий, не могли бы пояснить пункт 5.6? Если находить максимально плечо расчетным путем, по указанной формуле, то у меня получается какой то бред. На примере Si. Исходные данные - ГО 1130, цена, например 32249, тогда получается $100 / (1130 / 32249 * 100) = 350,39$ Если же просто разделить цену на ГО, то максимальное плечео получится 28,53, что больше похоже на правду. Может, я что то не так понял?

★ Это — ответ на вопрос! Ответить Пожаловаться



Власов Дмитрий написал более 4 месяцев назад
Создатель сообщества

Давайте по шагам:

1) Находим сколько % составляет величина ГО от стоимости контракта:

$1130 / 32249 * 100 = 3,5\%$ (т.е. для того, чтобы купить 1000 долларов нужно зарезервировать 3,5% от текущей стоимости).

2) Находим плечо "рычаг":

$100 / 3,5 = 28,5$ (т.е. Мы можем при торговле валютой использовать плечо 28,5 - привет форексникам...)

Так что формула корректна...
Точно также можно для любого контракта посчитать.

★ Это — ответ на вопрос! Ответить Пожаловаться



Игорь Афанасьев написал более 4 месяцев назад
Пользователь

Теперь, понятно, меня смутило именно нахождение процента :) Я сразу не врубился, что в Excel, надо было результат привести к процентному формату

★ Это — ответ на вопрос! Ответить Пожаловаться



Дмитрий написал более 4 месяцев назад
Пользователь

А почему нельзя просто разделить стоимость инструмента на ГО?
 $32249 / 1130 = 28,53$
зачем использовать в формуле числа 100, которые сокращаются математически?

★ Это — ответ на вопрос! Ответить Пожаловаться



Власов Дмитрий написал более 4 месяцев назад
Создатель сообщества

Можно, конечно. Просто для ясности приводил такие расчеты, чтобы было видно, как величина ГО в процентах связана с величиной рычага (плечом)

★ Это — ответ на вопрос! Ответить Пожаловаться



В I U

Добавить комментарий

Как установить связь между Wealth-Lab и Visual Studio для отладки торговой стратегии

 **АВТОР** ДМИТРИЙ ВЛАСОВ  **АВГ - 5 - 2012**  **КОММЕНТАРИЕВ:** 18

Давайте сегодня посмотрим как можно из Wealth-Lab прямо напрямую подключиться к Visual Studio.

Для чего это нужно: Если мы научимся этому, то сможем отлаживать любые торговые стратегии, любые индикаторы, любые модули управления размером позиции (PosSizers) прямо в Visual Studio.

Для того, чтобы это сделать необходимо проделать следующее.

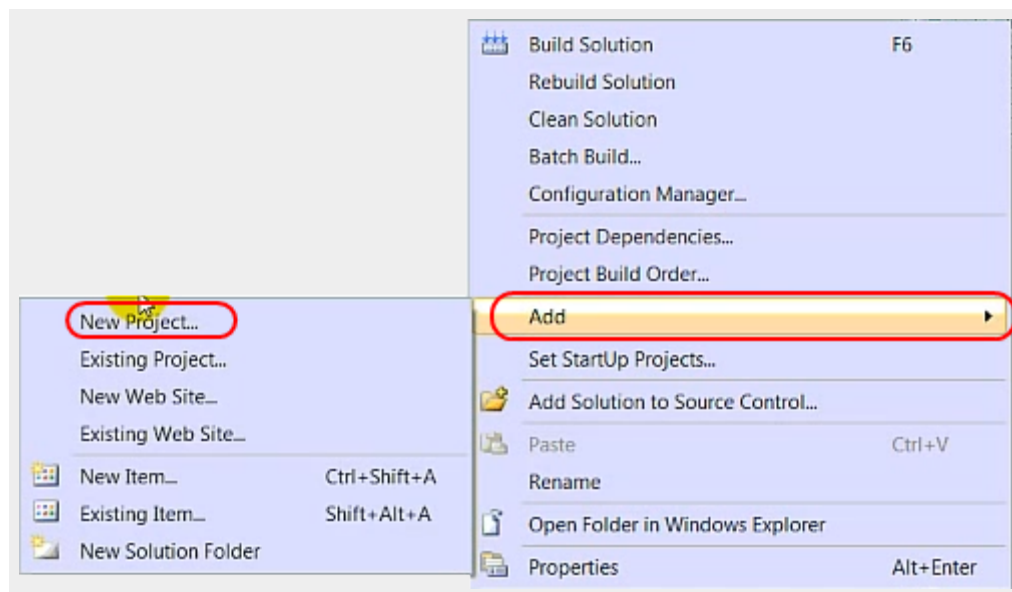
Во-первых в Visual Studio необходимо создать новый проект. Давайте назовем этот проект "Otladka"

Для этого делаем следующие шаги:

1) Находим Solution Explorer и нажимаем правой кнопкой мыши по названию решения (Solution).

2) В появившемся контекстном меню выбираем пункт:

Add >> New Project... (Добавить >> Новый проект)



Сразу после нажатия появится окно добавления нового проекта

3) В окне добавления нового проекта проделываем следующие шаги:

в самом верху выбираем .NET Framework 3.5

выбираем тип проекта "Class Library" (библиотека классов).

В поле Name: даем название проекту (к примеру, "05_Otladka")

В поле Location: указываем где должно располагаться решение (Solution), содержащее данный создаваемый проект.

Нажимаем кнопку "OK", подтверждая создание нового проекта

RSS товары



RSS новости



Twitter



Последние комментарии

макс: А где взять свечки с открытым интересом? на сайте финама нет



ring10: Добрый день Дмитрий Как сделать, чтобы открывался 1 велф лаб? Если открываешь под пользователем, т [...]



Дмитрий Власов: Буян, рад, что Вам пригодились материалы данной статьи!



Максим: Сам себе отвечу. protected override void Execute() { Bars rts = GetExternalSymbol("RTS", t [...]



Буян: Спасибо Дмитрий за статью! Мне удалось разработать хорошую рабочую торговую систему, используя возмо [...]



Последние посты



Бесплатный вебинар: 7 смертных грехов при оптимизации торговых систем

НОЯБРЯ - 5 - 2012



Отзывы участников курса до результата Wealth-Lab. С нуля до первой торговой системы за 5 дней.

НОЯБРЯ - 3 - 2012



Быстрее, Выше, Сильнее: Обновление Wealth-Lab до новой версии 6.4

ОКТАБРЯ - 23 - 2012



Курс до результата: Wealth-Lab – от нуля до первой торговой системы за 5 дней

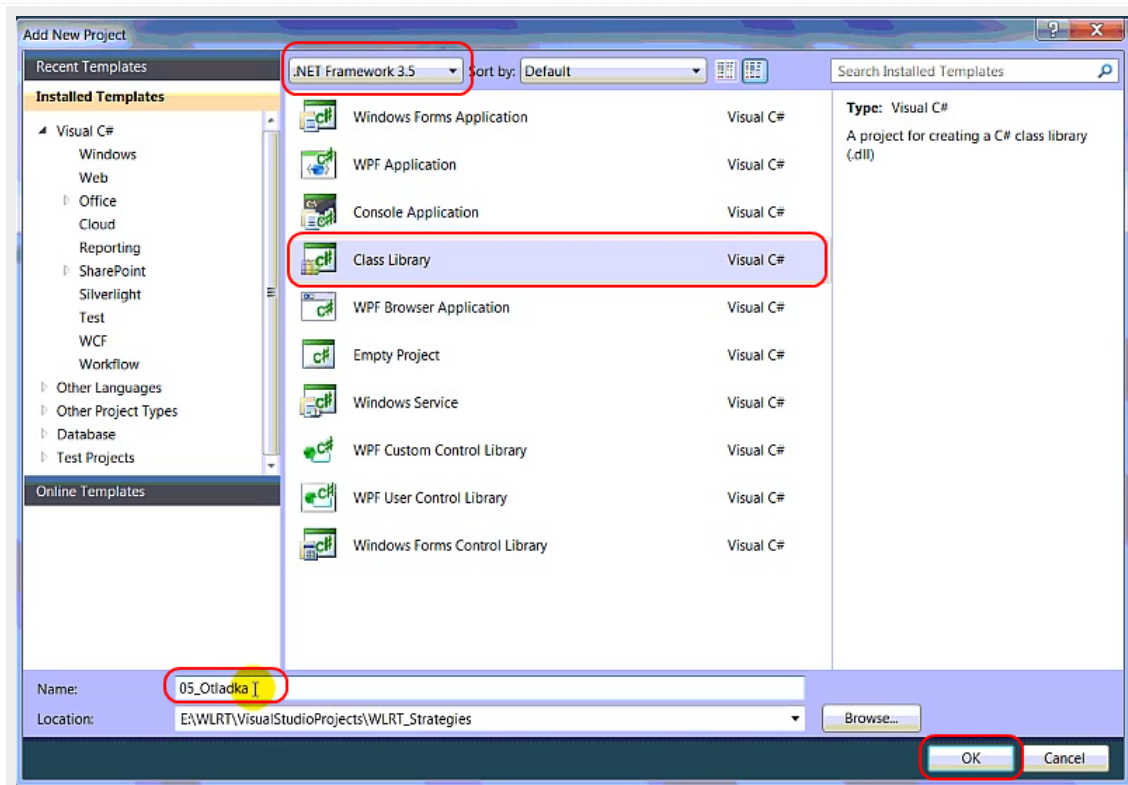
ОКТАБРЯ - 23 - 2012



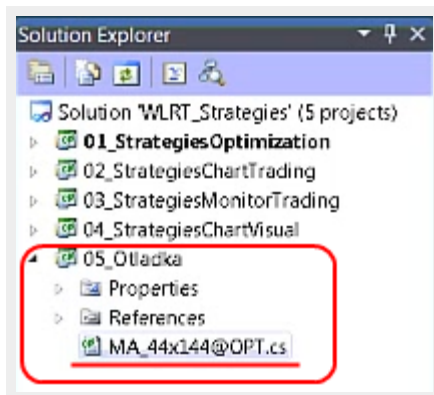
Обзорная экскурсия по достопримечательностям программы Wealth-Lab

ОКТАБРЯ - 20 - 2012

Берите



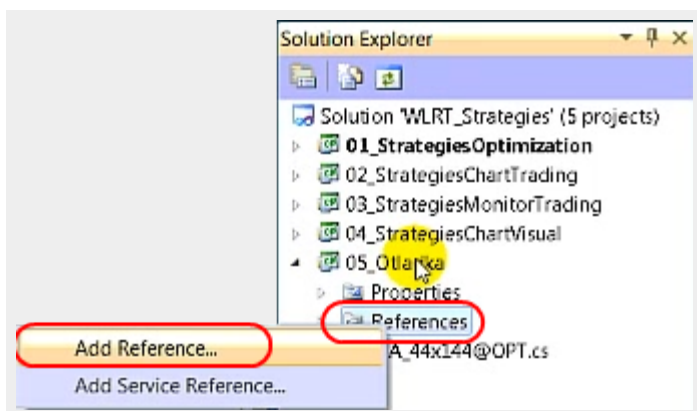
4) Теперь удаляем созданный в новом проекте класс по умолчанию (обычно называется Class1.cs) и добавляем в проект класс, который содержит код торговой стратегии. В нашем случае для демонстрации процесса подключения Wealth-Lab к Visual Studio воспользуемся простейшей торговой стратегией пересечения двух скользящих средних (название класса стратегии: MA_44144@OPT)



5) Теперь для того, чтобы была возможность в коде стратегии пользоваться методами и свойствами библиотеки WealthScript, нужно добавить в наш проект ссылки на библиотеки WealthLab.dll и WealthLab.Indicators.dll

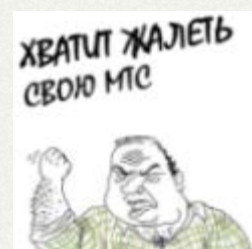
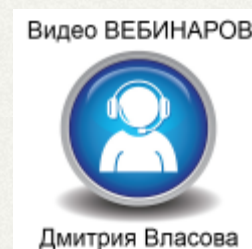
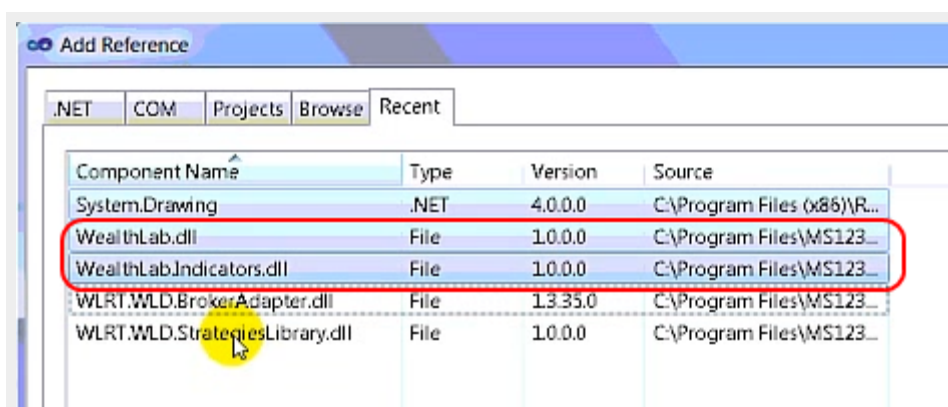
Делается это так:

В Solution Explorer нажимаем правой кнопкой на строку "References" ("Ссылки") и в появившемся контекстном меню кликаем по "Add Reference..."



Затем переходим на закладку "Browse", находим там папку, в которой находится файл, запускающий Wealth-Lab и в этой папке находим необходимые нам файлы (WealthLab.dll и WealthLab.Indicators.dll). Выделяем их и нажимаем "OK", добавляя таким образом ссылки на выбранные библиотеки.

Если Вы уже ранее в данном решении уже добавляли ссылки, легче всего будет перейти на закладку "Recent" (Недавно использовались) и найти данные библиотеки там.



МТС под микроскопом: WLMA System

Подробный разбор торговой
системы WLMA System

3900 руб.

[В корзину](#)

Рубрики

Выберите рубрику

Обновления:

- Новые посты по Email
- Новые посты по RSS
- Комментарии по RSS
- Следим на Twitter

Метки

C# HowTo: **Wealth-Lab**
Видео Индикаторы МТС Торговые
роботы Юмор алготрейдинг
мероприятия обучение

Управление

Регистрация

Вход

RSS Записей

RSS Комментариев

6) Теперь добавляем в класс стратегии дополнительный специфический класс Helper.

Здесь приведен пример кода простейшей стратегии совместно с классом Helper...

Код примера на C# (кликните чтобы посмотреть)



```

1  /*
2  * Это стратегия, для демонстрации возможности подключения Wealth-Lab к Visual Studio
3  * Подробная статья об этой стратегии располагается по адресу:
4  * http://finlabportal.ru/2012/08/kak-ustanovit-svyaz-mezhdu-wealth-lab-i-visual-studio/
5  * Видео процесса подключения можно найти тут: http://www.youtube.com/watch?
6  feature=player_embedded&v=y8mInMDvZhU
7  */
8
9  /*
10 using System;
11 using System.Collections.Generic;
12 using System.Text;
13 using WealthLab;
14 using WealthLab.Indicators;
15 using System.Drawing;
16
17 namespace Example.Strategies.Optimization
18 {
19     //это основной код стратегии
20     class MA_44x144_otl : WealthLab.WealthScript
21     {
22         #region Объявляем параметры торговой стратегии
23
24         private StrategyParameter _slowPeriod;
25         private StrategyParameter _fastPeriod;
26
27         #endregion
28
29         #region Конструктор класса
30
31         public MA_44x144_otl()
32         {
33             _fastPeriod = CreateParameter("Fast Period", 44, 1, 100, 1); //Устанавливаем
34             значение параметров по умолчанию от, до и шаг
35             _slowPeriod = CreateParameter("Slow Period", 144, 20, 200, 1);
36         }
37         #endregion
38
39         protected override void Execute()
40         {
41             int firstValidValue = 1; // Первое значение свечи при которой существуют
42             индикаторы
43
44             #region Индикаторы
45
46             // Быстрая скользящая средняя
47             int fastPeriod = _fastPeriod.ValueInt; //Определяем период скользящей
48             средней
49             DataSeries smaFast = SMA.Series(Close, fastPeriod); //Создаем ряд(массив
50             данных в каждой ячейке которого храниться значение индикатора
51             PlotSeries(PricePane, smaFast, Color.Green, LineStyle.Solid, 2); //
52             Отображаем наш ряд данных на графике цен в WLD
53             firstValidValue = Math.Max(firstValidValue, smaFast.FirstValidValue);
54
55             // Медленная скользящая средняя
56             int slowPeriod = _slowPeriod.ValueInt; //Определяем период скользящей
57             средней
58             DataSeries smaSlow = SMA.Series(Close, slowPeriod); //Создаем ряд(массив
59             данных в каждой ячейке которого храниться значение индикатора
60             PlotSeries(PricePane, smaSlow, Color.Red, LineStyle.Solid, 2); //
61             Отображаем наш ряд данных на графике цен в WLD
62             firstValidValue = Math.Max(firstValidValue, smaSlow.FirstValidValue);
63             Присваиваю переменной firstValidValue максимальное значение
64             самого длинного периода из существующих индикаторов
65
66             #endregion
67
68             for (int bar = firstValidValue; bar < Bars.Count-1; bar++)
69             {
70                 if (IsLastPositionActive) //Если есть открытая позиция
71                 {
72                     if (CrossUnder(bar, smaFast, smaSlow)) //Быстрая пересекает медленную
73                     сверху вниз
74                     {
75                         SellAtMarket(bar + 1, LastPosition); //Продаем по рынку на
76                         следующем баре
77                     }
78                     else //Активных позиций нет
79                     {
80                         if (CrossOver(bar, smaFast, smaSlow)) //Быстрая пересекает медленную
81                         снизу вверх
82                         {
83                             BuyAtMarket(bar + 1); //Покупаем на следующем баре по цене
84                             открытия
85                         }
86                     }
87                 }
88             }
89         }
90     }
91
92     //для отладки стратегии и связки Wealth-Lab с Visual Studio нужно добавить класс Helper
93     //это код стратегии HELPER
94     public class MA_44x144_otlHelper : StrategyHelper
95     {
96         public override string Author
97         {
98             get { return "Власов Дмитрий Викторович"; } //в кавычках пишем автора стратегии
99         }
100
101         public override DateTime CreationDate
102         {
103             get { return new DateTime(2012, 08, 02); } //в скобках указываем дату создания
104             стратегии (год, месяц, дата)
105         }
106
107         public override string Description //здесь необходимо сделать краткое описание

```

```

108 стратегии (оставив открывающий тег) - будет виднов велсе
109 {
110     get { return "Покупка на пересечении скользящих средних (Description в кл
111 Helper)"; }
112 }
113
114 public override Guid ID
115 {
    get { return new Guid("7817cb9f-b00d-499f-9739-33fd59653efc"); } //правой
    кнопкой по проекту, затем "Свойства" и в "Assembly Information" находим ID
}

public override DateTime LastModifiedDate
{
    get { return new DateTime(2012, 08, 02); } //указываем дат модификации
стратегии
}

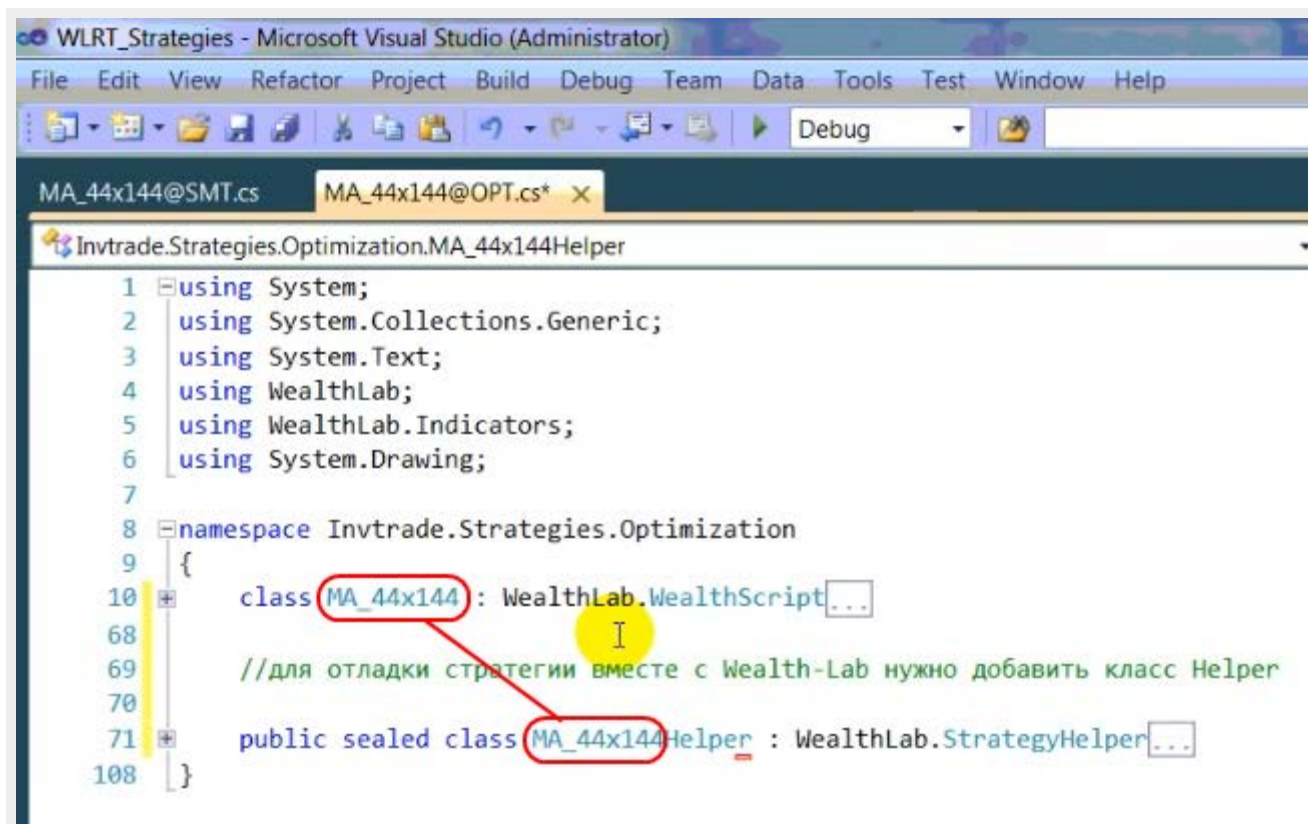
public override string Name
{
    get { return "MA_44x144_otl"; } //здесь в кавычках указываем название стр
}

public override System.Type WealthScriptType
{
    get { return typeof(MA_44x144_otl); } //в скобках указываем точное имя
основного класса стратегии.
}
}
}

```

Этот класс необходим для того, чтобы осуществить процесс стыковки Wealth-Lab с Visual Studio.

Название данного класса до слова Helper должно в точности соответствовать названию основного класса стратегии.



7) Теперь этот класс необходимо отредактировать.

Делается это очень просто - давайте пробежимся по каждому из пунктов:

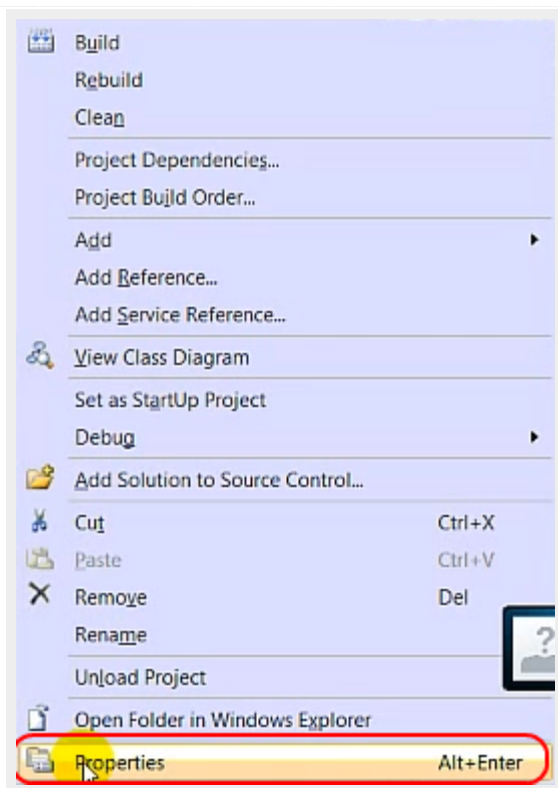
Name:

Внутри кавычек здесь пишется название стратегии... Точное совпадение с названием класса стратегии здесь не обязательно, но чтобы в последующем не путаться - лучше написать именно оригинальное название стратегии

System.Guid ID

Для того, чтобы найти значение ID нужно проделать следующее:

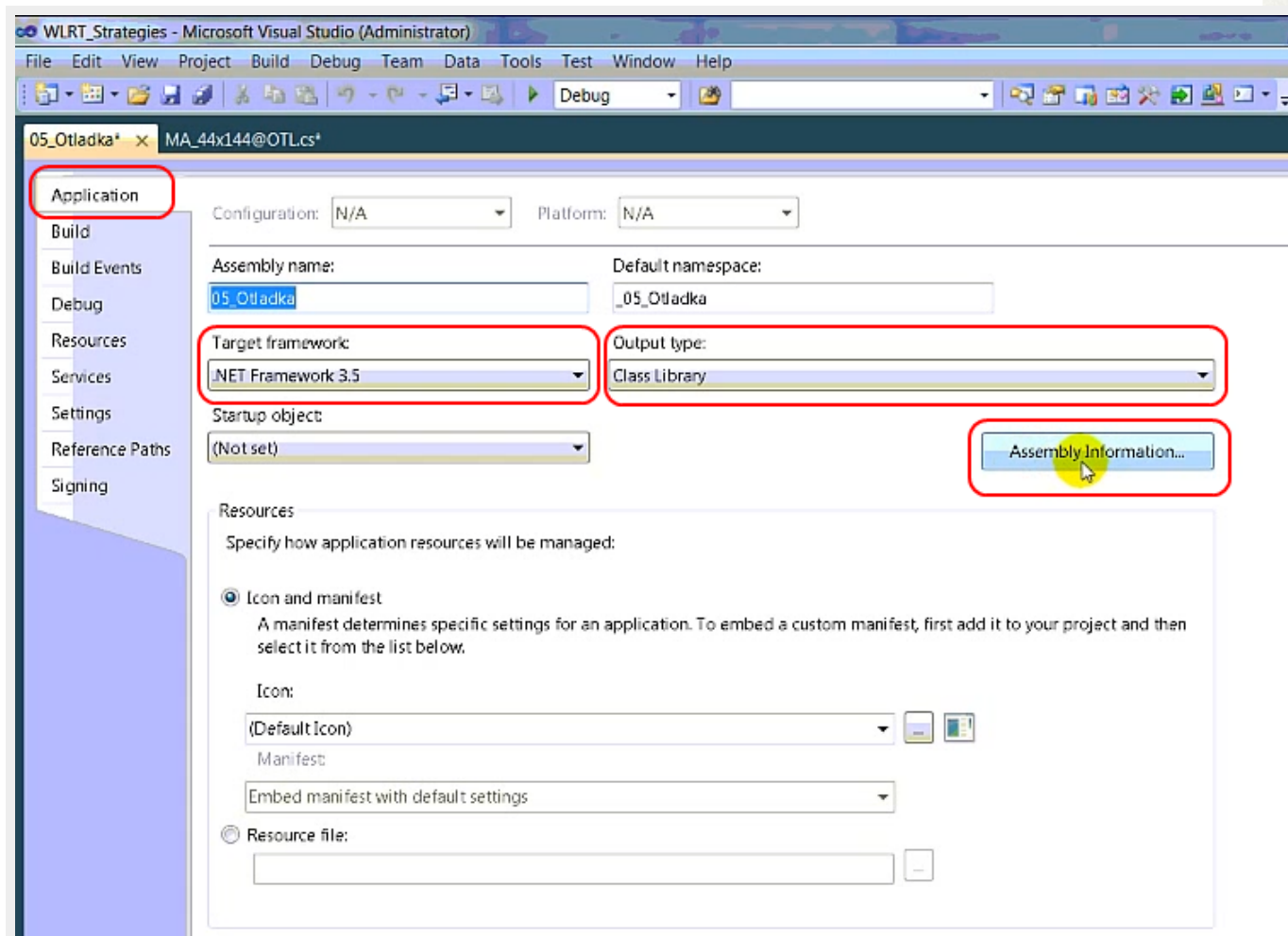
нажимаем правой кнопкой по названию проекта:



В открывшемся окне переходим на закладку "Application" и проверяем:

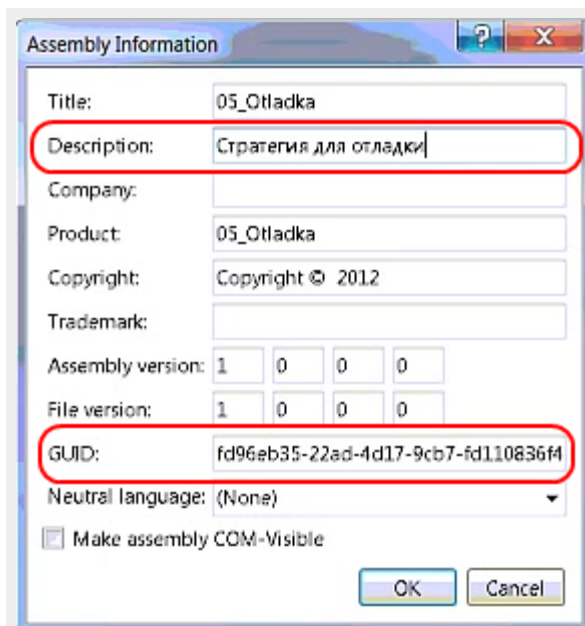
Target Framework: .NET Framework 3.5

Output type: Class Library



Нажимаем на кнопку "Assembly Information..." (Информация о сборке)

В результате появляется форма "Assembly Information"



Здесь нужно заполнить два поля:

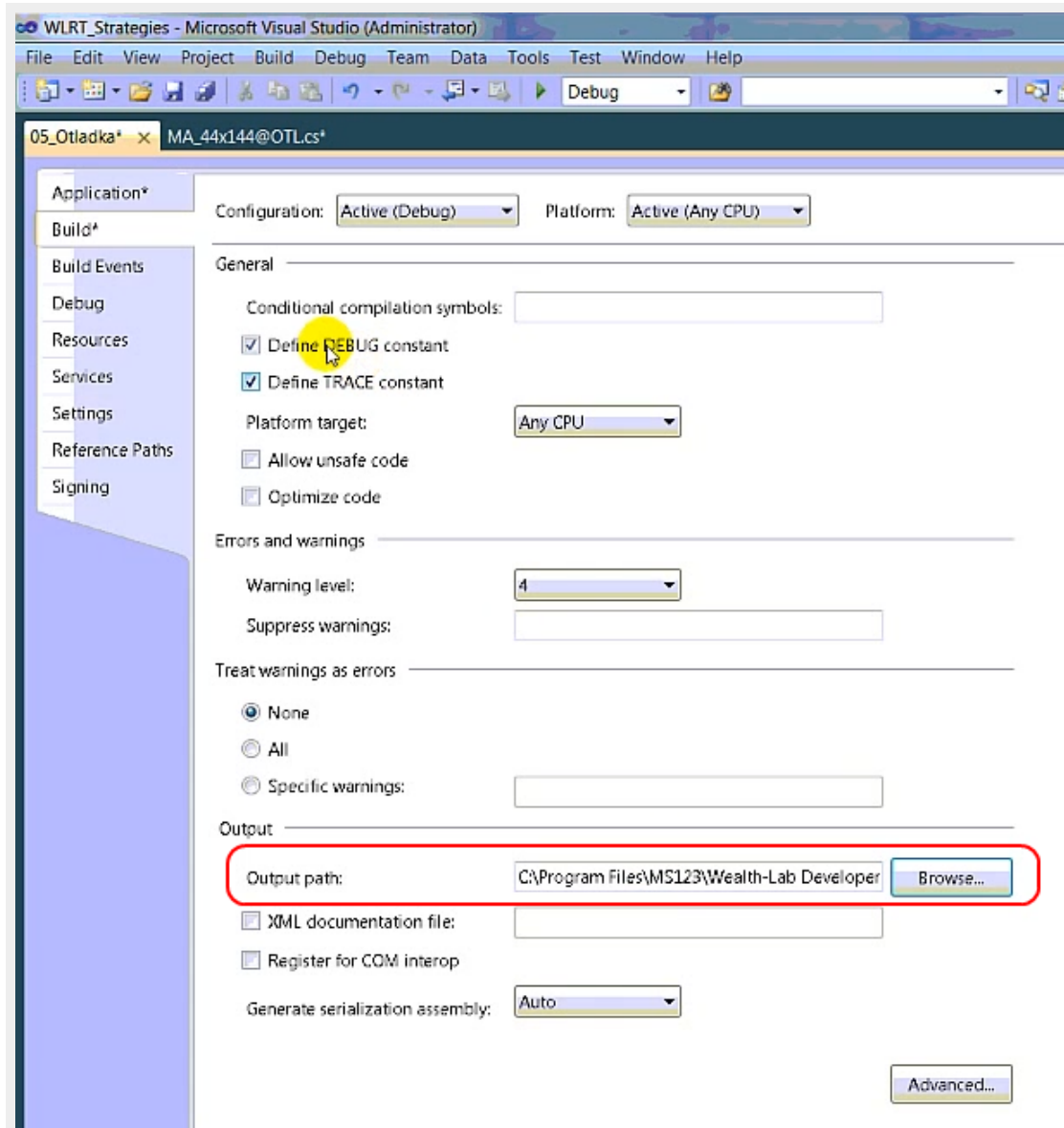
Discription (Описание) - здесь мы должны написать осмысленную фразу. Эта фраза будет видна из Wealth-Lab

когда мы будем выбирать папку, в которой находятся те стратегии, которые мы поместили в проект для отладки.

GUID: - здесь ничего менять не нужно, просто выделяем этот код и копируем его в буфер обмена.

Делаем все остальные настройки, необходимые для настройки соединения:

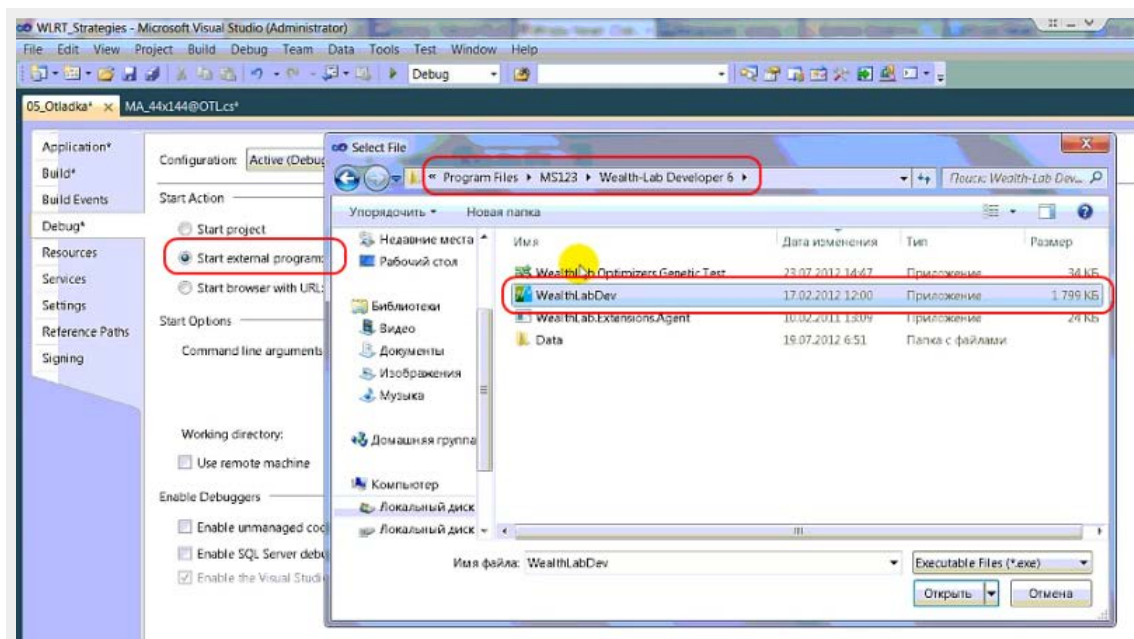
Переходим в закладку "Build" ("Построение")



Здесь мы должны указать папку, куда будет выкладываться dll (библиотека классов), содержащая все стратегии, которые содержатся в данном проекте.

Указать мы должны ту папку, в которой находится файл, запускающий Wealth-Lab.

Теперь переходим в закладку "Debug" (Отладка) и выбираем опцию "Start external program" и указываем файл, запускающий Wealth-Lab.



Заполняем System.Guid ID

В скобках открываем кавычки и внутрь кавычек вставляем тот код, который мы скопировали ранее, после чего закрываем кавычки.

Author: внутри кавычек пишется имя автора стратегии

System.Type WealthScriptType

Здесь в скобках должно быть указано точное имя класса нашей стратегии.

Discription (Описание):

Здесь мы пишем краткое описание стратегии. Это описание в дальнейшем будет видно в программе Wealth-Lab в Strategy Explorer.

Чтобы описание в Wealth-Lab отображалось правильно (чтобы не было псевдосимволов) здесь нужно внести вот такую надпись:

```
get { return "<meta http-equiv=\"Content-Type\" content=\"text/html; charset=utf-8\" />Покупка на пересечении скользящих средних"; }
```

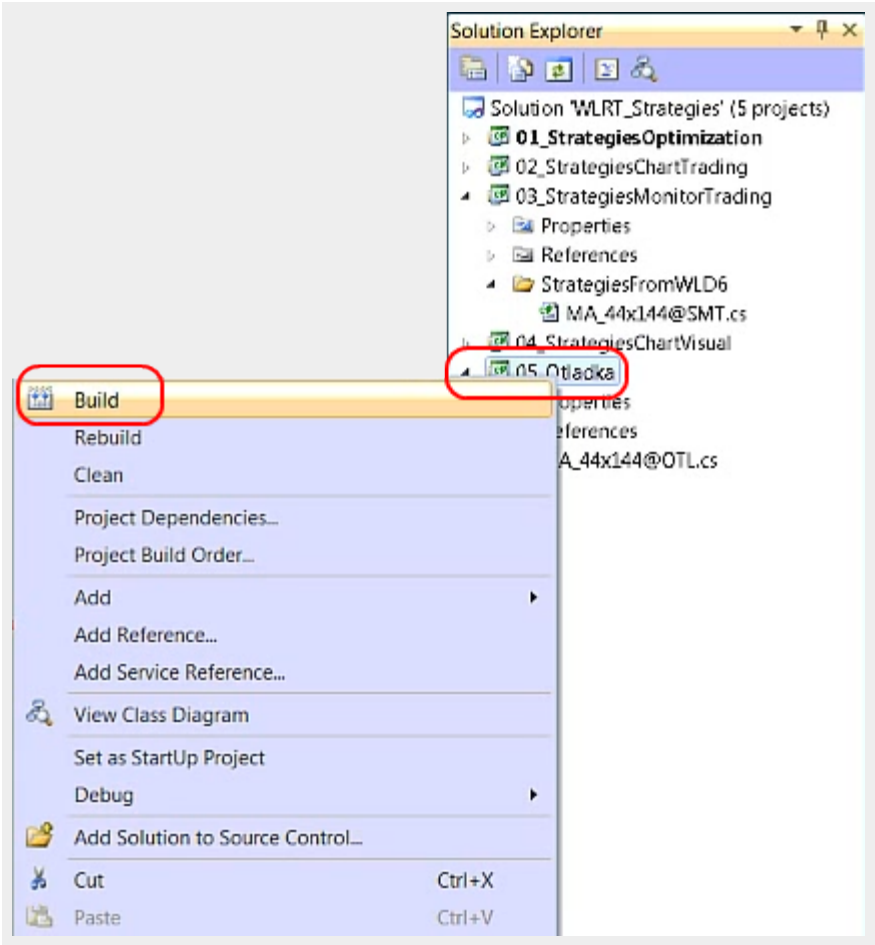
Даная конструкция показывает, какого типа должен быть текст и если она будет отсутствовать, то когда мы перейдем к соответствующей стратегии в Wealth-Lab, то увидим не осмысленный текст из русских букв, а набор символов, который невозможно прочитать.

Далее вносим дату создания и дату модификации стратегий. Как это сделать - можно увидеть в примере.

Теперь сохраняем результаты.

8) Строим проект:

Нажимаем правой кнопкой по названию проекта и из контекстного меню выбираем пункт "Build" ("Построить").



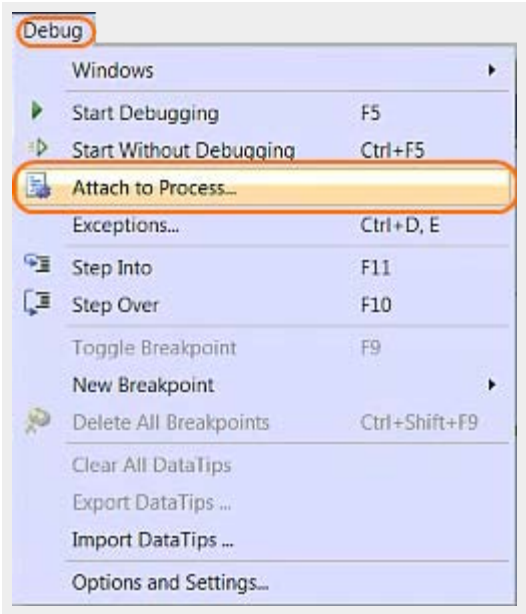
Это делается для того, чтобы во-первых, проверить нет ли ошибок в коде. Если такие ошибки есть, то у нас перед глазами появится сообщение об ошибке.

Во-вторых, сразу после процесса построения в папке, где находится файл, запускающий Wealth-Lab появится dll, которая содержит все те стратегии, которые мы "упаковали" в данный проект.

Благодаря наличию данной dll мы сможем подключиться к нашей стратегии из Wealth-Lab.

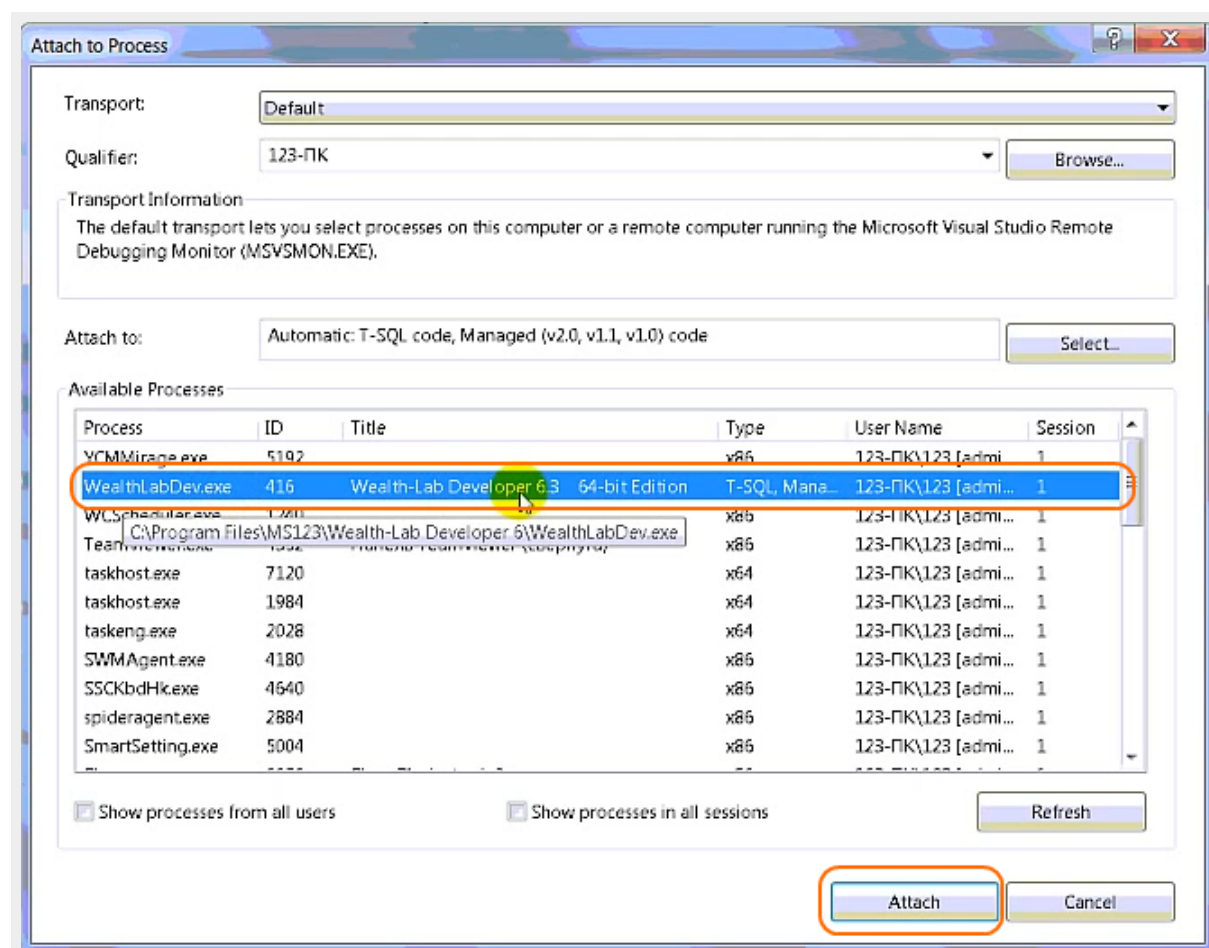
9) Теперь открываем программу Wealth-Lab. После того, как программа Wealth-Lab запустилась, мы снова переходим в Visual Studio и в главном меню находим пункт, который позволит нам подключиться к процессу.

Debug >> Attach to Process...



Сразу после этого выскочит формочка подключения к процессу. Здесь нужно в списке доступных процессов найти Wealth-Lab Dev. Для этого лучше отсортировать все процессы, нажав на столбец Process и в самом

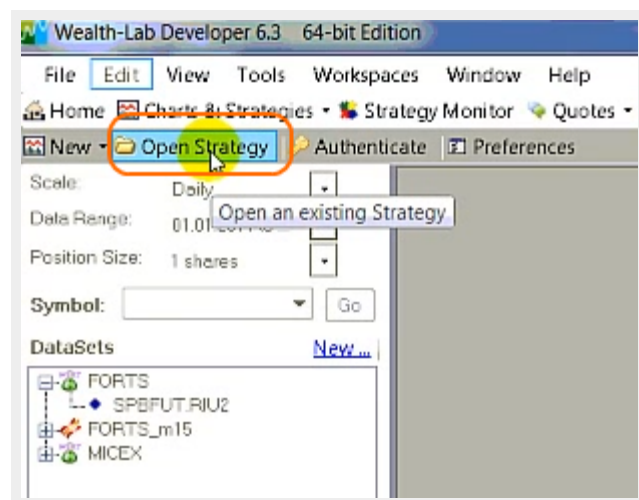
верху обнаружим нужный нам процесс.



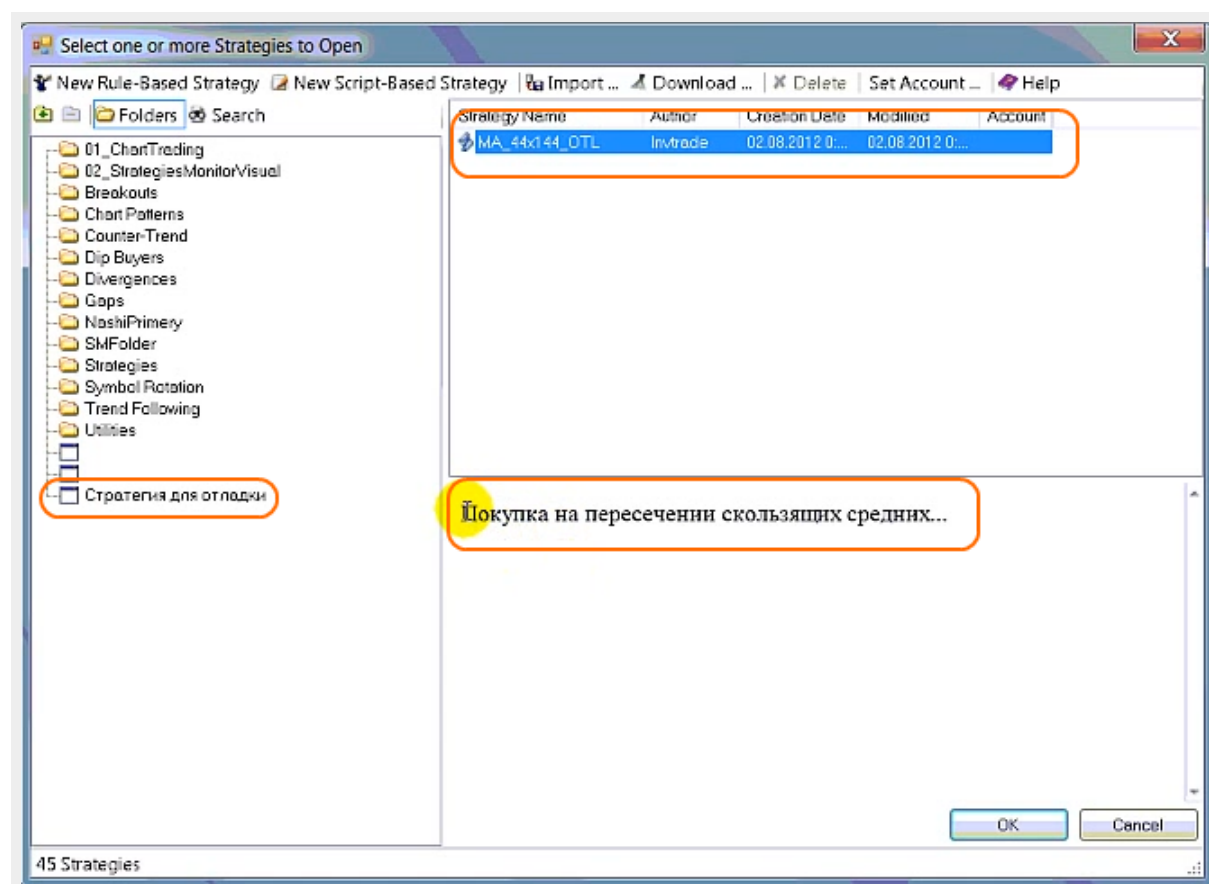
Это процесс называется WealthLabDev.exe. Выбираем его левой кнопкой мыши и нажимаем кнопку внизу формы "Attach".

10) Теперь снова переходим в программу Wealth-Lab.

В Wealth-Lab нам нужно открыть Strategy Explorer - для чего нажимаем на кнопку "Open Strategy"



В папках открытого окна с существующими стратегиями мы увидим ссылку с тем названием, которое вводили в свойствах проекта при вводе информации о сборке.



Выбрав эту ссылку справа мы увидим все стратегии, которые мы включили в наш проект при его построении в Visual Studio. Причем у каждой из этих стратегий можно будет прочесть четко по-русски описание

(напоминаю, мы делали это описание в классе Helper).

Открываем стратегию, подготовленную нами в Visual Studio, щелкнув по ее названию и кликнув на кнопку "OK". Выбираем финансовый инструмент и видим, что стратегия благополучно открылась .

Особенностью данного окна стратегий является то, что мы вообще не видим закладки "Editor".

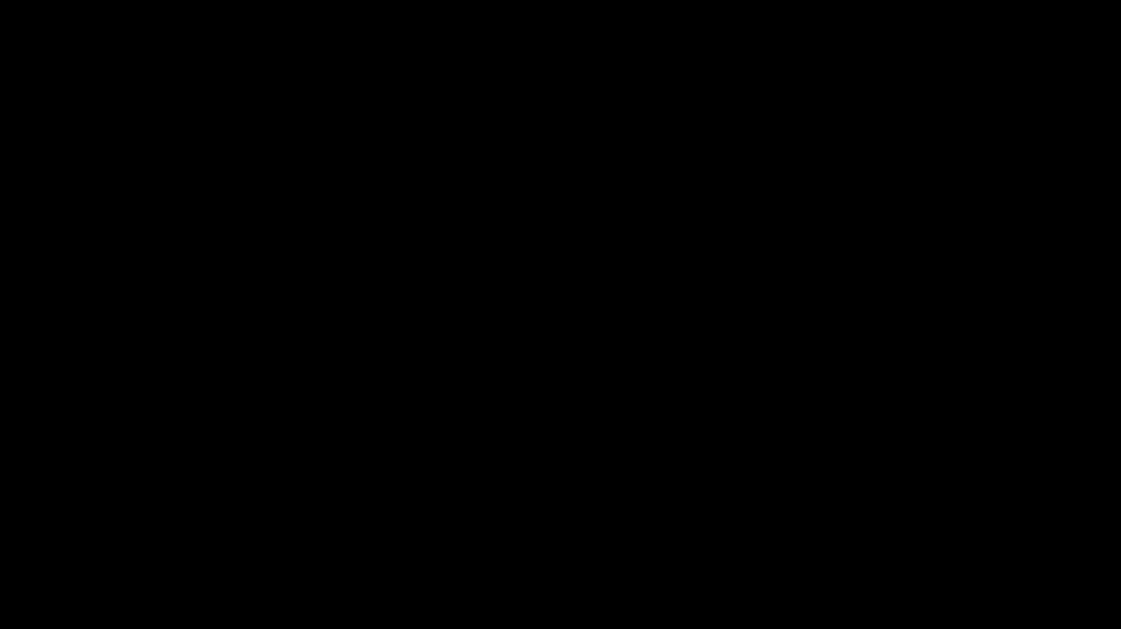


В качестве редактора будет выступать Visual Studio. Именно здесь мы можем править код стратегии, отлаживать стратегию и делать другие, очень интересные вещи. Это все равно, что писать текст не в обыкновенном блокноте (которым, по сути, является родной редактор программы Wealth-Lab), а в навороченном текстовом редакторе.

Подключив подобным образом Wealth-Lab к Visual Studio мы можем воспользоваться всем богатством данной программы для отладки кода - делать точки остановки, следить в режиме реального времени за значением любых переменных, маленькими шагами проходить весь код стратегии.

Все это очень удобно и сильно поможет Вам при написании сложных стратегий, которые содержат в себе многие страницы кода. Используя кнопки F11, F5 и точки остановок - можно найти любую ошибку либо проблемный кусок кода.

Более подробно за всем процессом подключения Wealth-Lab к Visual Studio и способам отладки стратегии в Visual Studio Вы можете посмотреть в данном видео.



Это видео, отображающее процесс подключения Wealth-Lab к Visual Studio и отладки стратегий, было записано мною в процессе **индивидуального обучения**, которое я провожу для всех желающих лучше узнать Wealth-Lab и язык программирования C# в процессе индивидуальных занятий (с разрешения того, для кого данное занятие проводилось). Каждое такое занятие я снимаю на видео и отдаю этот "минифильм" в течение нескольких дней после проведения занятия.

В конце сегодняшнего поста напоминаю Вам о том, что в понедельник 06 августа 2012 года в 15 часов начнется **бесплатный вебинар**, который я буду проводить для всех желающих получше узнать о программе Wealth-Lab и языке программирования C#. К моему удивлению, на текущий момент записалось уже более 80 человек из 90 возможных. Если хотя бы половина из записавшихся посетит данный вебинар - будет очень круто. Кто записался - не забывайте о времени начала.

Если Вам не хватило мест в этот раз Вы можете записаться на точно такой же вебинар, который состоится несколькими днями позже и который организован для тех, кому удобнее посещать вебинары в вечернее время. Этот вебинар состоится 9-го августа в 19 часов по Москве. Там места еще точно есть, так что кто записался, но забыл прийти или не успел записаться - **записывайтесь на этот вебинар**.

Если Вам понравилась эта статья - подпишитесь на **новые СТАТЬИ по RSS** или получайте наши **новые СТАТЬИ на E-mail**:
Рекомендую почитать также:

- **Видео вебинара по программе Wealth-Lab в рамках курса "Wealth-Lab: C# на службе трейдера"**
- **Ответы на вопросы по программированию торговых стратегий в Wealth-Lab.**
- **Курс для тех, кто хочет создавать собственные торговые стратегии: Wealth-Lab: C# на службе трейдера**
- **Проектирование и построение индикаторов в Wealth-Lab**

- [Использование при программировании торговых стратегий торговых приказов AtStop и AtLimit](#)



РУБРИКИ: [WEALTH-LAB](#), [WEALTHSCRIPT](#)

Комментариев: 18

Роман пишет:

6 августа 2012 в 17:27



интересуюсь созданием торгового робота в Wealth-Lab для торговли на американском рынке. можете помочь?

Ответить

Дмитрий Власов пишет:

14 августа 2012 в 10:41



Роман, кажется, я Вам по почте уже отвечал. Для американского рынка написать торговую стратегию в велсе можно, конечно, но вот чтобы автоторговлю использовать с применением Wealth-Lab – здесь возникают трудности. Если получится у Fidelity счет открыть, то там версия Wealth-Lab PRO с уже готовым подключением к этому брокеру. А как – то связать велс с другими американскими брокерами... не знаю, как это сделать.

Ответить

Stas пишет:

8 августа 2012 в 12:17



А какую “студию” использовать? Я установил 10, но в ней нет некоторых меню, в частности: при создании проекта, нет возможности выбрать версию фреймворка; в “свойствах проекта” в вкладке “дебаг” нет возможности указать экзешный файл велза; так же нельзя загрузить ссылку на using System.Drawing.

Подскажите пожалуйста как быть?

Ответить

Дмитрий Власов пишет:

14 августа 2012 в 10:42



У меня Visual Studio 10 версия PRO. Именно в ней и показывал, как все это сделать.

Ответить

Alexey пишет:

11 августа 2012 в 18:56



Хочу Вам сказать огромное спасибо за детальный план связи.

Тот, что описан у вэлса в вики, сложно применить.

Там нет описания Helpaga.

Ответить

Дмитрий Власов пишет:

14 августа 2012 в 10:42



Алексей, рад, что Вам пригодилась информация, которую я опубликовал здесь.

Ответить

Vitaliy пишет:

13 августа 2012 в 19:17



Stas.

Как мне кажется, в данном примере использовалось Visual Studio не экспресс-выпуска, а более профессиональный (т.е «платный»).

Что касается System.Drawing, его надо подключать дополнительно – как мы делали с WealthLab.dll и WealthLab.Indicators.dll, только вкладка .Net.

Ответить

Дмитрий Власов пишет:

14 августа 2012 в 10:44



По поводу платности – есть возможность в течение 30 дней использовать бесплатно Visual Studio pro. Затем продлить это бесплатное использование еще на 30 дней. Кроме того у Microsoft есть специальные предложения для учителей и преподавателей, студентов, маленьких компаний... Так что воспользоваться программами Microsoft можно не платя больших денег.

Ответить

Игорь Афанасьев пишет:

16 августа 2012 в 15:45



Дмитрий, что то, я затупил. Не могу понять, а где заполняется System.Guid ID, что не могу понять где его искать?

Ответить

Игорь Афанасьев пишет:

16 августа 2012 в 15:56



Ой, извини 😊 предыдущий вопрос по System.Guid ID снимается, действительно тупанул 😊 Разобрался

Ответить

Vitaliy пишет:

22 августа 2012 в 12:56



Настроил все как показано в данной статье.

Создаю ДЛЛ в папке с WLD.

Пытаюсь подключиться к процессу, вроде все проходит, но «стопы» не работаю, да и при перестроении пишет ошибку.

Подскажите, в чем может быть причина?

Текст ошибки в VS 2010:

Ошибка 1 не удалось скопировать файл “obj\Debug5_Otladka.dll” в “..\..\..\..\Program Files\MS123\Wealth-Lab Developer 65_Otladka.dll”. Процесс не может получить доступ к файлу “..\..\..\..\Program Files\MS123\Wealth-Lab Developer 65_Otladka.dll”, так как этот файл используется другим процессом. 05_Otladka

Ответить

Дмитрий Власов пишет:

22 августа 2012 в 19:49



Копировать файл *.dll тогда, когда Wealth-Lab не запущен. Т.е. закрываете программу, копируете в нужное место dll-ку и потом открываете программу Wealth-Lab.

Ответить

Vitaliy пишет:

23 августа 2012 в 12:41



В видео показано, как Вы перехватываете выполнение ДЛЛ и переходите в VS (время на видео 19:15). Но как бы я не пытался, перехватывать выполнение не получается.

Подскажите, в чем может быть причина?

Делал все в точности как показано на видео.

ПО использую:

VS PRO – trial

WLD – trial

Ответить

Vitaliy пишет:

23 августа 2012 в 20:21



При отключении WLD, перестроить DLL удастся, как только включаю WLD, DLL становится не доступной.

Ответить

Вууап пишет:

25 октября 2012 в 17:28



Дмитрий, огромное спасибо за такую подробную инструкцию! Ваше руководство намного лучше, чем руководство самих велслабовцев!

Ответить

Дмитрий Власов пишет:

26 октября 2012 в 01:27



Спасибо, я старался. Правда все, что делаю на этом блоге – делаю для себя, чтобы если вдруг прекращу заниматься этим делом, а потом решу возобновить – смог разобраться и вспомнил что раньше делал... 😊

Ответить

Роман пишет:

1 ноября 2012 в 16:41



Подскажите – при запуске Wealthlab, после создания dll в каталоге велза возникает ошибка “System.TypeInitializationException” и он не запускается. С чем это может быть связано?

Ответить

Роман пишет:

2 ноября 2012 в 07:19



Сделал весь процесс в windows7, запускается без проблем.
Видимо ошибка вылетает из-за восьмой винды, так как после создания otladka.dll, wealthlab просто не запускается, а в studio вылетает вышеобозначенная ошибка

Ответить

Оставить комментарий

Имя (Обязательно)

Почта (Обязательно)

Вэб-сайт

Получать новые комментарии по электронной почте. Вы можете [подписаться](#) без комментирования.





ТОРГОВЫЕ РОБОТЫ (открытая база знаний)

Главная

Обсуждения

Люди

[Главная](#) / [WLRT \(Wealth-Lab по русски\)](#)

Дмитрий Власов задал вопрос 09 августа 2012 в 17:42

С ОТВЕТОМ

Пользователь

Настройка Data Range при реальной торговле из чарта - что нужно указывать?

Когда я запускал реальную торговлю из чарта в настройках DataRange (в левом верхнем углу) программы Wealth-Lab был указан временной промежуток с 01.03.2012 по 05.08.2012.

Торговля проводится 08.08.2012 года. Вопрос заключается в следующем - будет ли заход в позицию в соответствии с правилами торговой стратегии и будет ли выход из позиции в соответствии с правилами торговой стратегии?

0



У меня этот вопрос. Голосовать!

Создать обсуждение

Следить за вопросом

Подписаться на RSS

Пожаловаться

Поделиться:

Сохранить



Мне нравится

0

Поделиться

Официальный ответ компании



gorda написал 5 месяцев назад

Представитель компании

Диапазон данных влияет на запрос исторических данных, потоковые данные не будут фильтроваться.

Торговля будет идти как обычно. Для торговли рекомендуется использовать интервал типа последние несколько дней(например 10), а не фиксированный интервал, т.к. при этом ваши индикаторы могут иметь некорректные значения, из-за пропущенных данных за пропущенный период(6е и 7е августа).



WLRT (Wealth-Lab по русски)

1 комментарий

1 участник

1 подписчик

нет комментариев

[Добавить комментарий](#)

B *I* U

Добавить комментарий

Создание библиотеки индикаторов в программе Wealth-Lab

 **АВТОР ДМИТРИЙ ВЛАСОВ**  **АВГ - 22 - 2012**  **КОММЕНТАРИЕВ: 6**

Эта статья объясняет как можно создать библиотеку технических индикаторов для программы Wealth-Lab. Такая библиотека содержит скомпилированные индикаторы, которые интегрированы в программу Wealth-Lab и ведут себя точно также, как и индикаторы, которые поставляются вместе с программой. Созданные индикаторы появляются в собственной папке в списке индикаторов прямо в программе Wealth-Lab. Вы сможете схватить мышкой и перетащить любой из индикаторов, которые находятся в данной папке, прямо на график. Также Вы без проблем сможете использовать эти индикаторы в мастере стратегий.



Для того, чтобы создать библиотеку индикаторов, Вам потребуется специальная программа для написания кода на языке C#. Для этих целей можно использовать, к примеру, Visual Studio или бесплатную программу SharpDeveloper. Те кто знаком с этими программами, сможет без труда создать библиотеку индикаторов. Если Вы не обладаете нужными знаниями - найдите, что Вы можете создать и отобразить на графике собственные индикаторы прямо в коде стратегии в редакторе программы Wealth-Lab.

Что такое библиотека индикаторов

Раз уж Вы дочитали статью до этого места, давайте более подробно разберемся - что такое библиотека индикаторов.

В программе Wealth-Lab каждый отдельный индикатор реализован как класс, который наследуется от базового класса DataSeries.

Класс DataSeries представляет собой массив, в каждой ячейке которого содержится число типа double (числа с цифрами после запятой), каждая ячейка данного массива связана с массивом, содержащим дату и время.

Конструктор класса

Ряд данных DataSeries может быть создан с помощью одного из двух конструкторов. В конструкторе доступны данные, используя которые и создаются ряды данных типа DataSeries. Это может быть или или другой ряд данных типа DataSeries, либо совокупность свечей (объект типа Bars). С помощью объекта Bars можно получить доступ ко всей истории данных о ценах открытия, закрытия, максимальных и минимальных ценах а также объемах соответствующего финансового инструмента.

Когда Вы создаете новый класс индикатора, создавайте именно такой конструктор, который более всего подходит для Ваших задач. Если Вы строите индикатор, который будет использовать только единичный ряд данных (к примеру, для расчета скользящей средней), выбирайте конструктор, который использует в качестве параметра переменную типа DataSeries.

Однако, если для расчета Вашего индикатора потребуется информация о ценах OHLC либо об объемах (например Вы собираетесь строить индикатор типа Стохастика), либо любую другую комбинацию, используйте конструктор, в качестве параметра в котором используется переменная типа Bars.

Примеры конструктора, который использует переменную типа DataSeries:

```
public SMA(DataSeries ds, int period, string description)
```

[RSS товары](#)



[RSS новости](#)



[Twitter](#)



Последние комментарии

макс: А где взять свечки с открытым интересом? на сайте финама нет



ring10: Добрый день Дмитрий Как сделать, чтобы открывался 1 велф лаб? Если открываешь под пользователем, т [...]



Дмитрий Власов: Буян, рад, что Вам пригодились материалы данной статьи!



Максим: Сам себе отвечу. protected override void Execute() { Bars rts = GetExternalSymbol("RTS", t [...]



Буян: Спасибо Дмитрий за статью! Мне удалось разработать хорошую рабочую торговую систему, используя возмо [...]



Последние посты



Бесплатный вебинар: 7 смертных грехов при оптимизации торговых систем

НОЯБРЯ - 5 - 2012



Отзывы участников курса до результата Wealth-Lab. С нуля до первой торговой системы за 5

дней.

НОЯБРЯ - 3 - 2012



Быстрее, Выше, Сильнее: Обновление Wealth-Lab до новой версии 6.4

ОКТАБРЯ - 23 - 2012



Курс до результата: Wealth-Lab – от нуля до первой торговой системы за 5 дней

ОКТАБРЯ - 23 - 2012



Обзорная экскурсия по достопримечательностям программы Wealth-Lab

ОКТАБРЯ - 20 - 2012

Берите

: base(ds, description)

Примеры конструктора, который использует переменную типа Bars:

```
public StochD(Bars bars, int period, int smooth, string description)
```

: base(bars, description)

Обратите внимание на то, что базовый конструктор будет вызываться каждый раз при создании индикатора. Пока этот конструктор заполняет ряд данных нулями, поэтому уже сейчас можно начинать работу с полным рядом значений. Теперь это Ваша забота - прямо в конструкторе нужно посчитать каждое значение индикатора бар за баром. Как это сделать - написано ниже.

Обратите внимание также на то, что одним из параметров конструктора является текстовая переменная, называемая "description" (описание). Эта текстовая строка представляет собой уникальное описание индикатора, включая его параметры. Этот параметр создается с помощью статического метода Series(), который также описан ниже.

Ваш индикатор может также иметь некоторое количество параметров, подобных параметру "period", который применяется при расчете скользящей средней. Эти дополнительные параметры могут иметь только следующие типы данных:

[DataSeries](#)
 any enumerated type
 Bars (дают доступ к последовательности свечей и их ценам OHLC и объемам)
 Int32
 Double
 Boolean
 String
 DateTime

Рассчитываем значения индикатора

Значения индикатору нужно присвоить прямо в конструкторе. Первоначально каждая ячейка в ряде данных, содержащем значения индикатора заполняется нулями. Используйте свойство Count для того чтобы определить - сколько значений индикатора нужно рассчитать и сколько значений нужно присвоить.

Используйте индексный метода доступа (this[]) для того, чтобы присвоить значения каждой ячейке индикатора как только эти значения будут рассчитаны. Для доступа к конкретному значению ряда данных, который является источником, используйте номер конкретного бара, заключив его в квадратные скобки ([]). Для доступа к полям класса Bars, используйте его свойства Open, High, Low, Close и Value. Эти поля сами являются объектами типа DataSeries (ряды данных).

Приведенный код является примером того, как рассчитывается индикатор простой скользящей средней (SMA).

Код примера на C# (кликните чтобы посмотреть)

```
1 //Рассчитываем значения индикатора SMA:
2
3 for (int bar = period - 1; bar < ds.Count; bar++) {           double sum = 0;           for
4 (int innerBar = bar; innerBar > bar - period; innerBar--)
5     sum += ds[innerBar];
6     this[bar] = sum / period;
7 }
```

Контроль первого бара, отображаемого на графике

Вы можете спрятать первоначальные значения Вашего индикатора. Для того, чтобы достичь этого, присвойте свойству FirstValidValue значение номера первого бара, который должен быть отображен на графике.

Для нашего примера, в котором рассчитывается простая скользящая средняя, FirstValidValue должно принять значение равное периоду скользящей средней минус единица. К примеру, если у нас есть 30-ти периодная скользящая средняя, значения первых 29 баров (которые расположены в ячейках с номерами от 0 до 28) не являются валидными. Первым валидным баром в данном случае будет 30-й бар (его номер 29, т.к. первая ячейка имеет номер 0).

Статический метод Series

Статичный метод Series это именно тот метод, код которого будет выполняться программой Wealth-Lab для создания конкретного экземпляра класса индикатора. Этот код будет выполняться и тогда, когда Вы бросаете индикатор на график цены с помощью мышки. Метод Series имеет точно такие же параметры, что и конструктор класса индикатора, за исключением параметра "description".

Внутри метода Series создается экземпляр Вашего индикатора, при этом используется служебное слово "new", сразу после этого Вы сразу получаете доступ к данному экземпляру.

Выполните следующие шаги, создавая статичный метод Series:

Создайте текстовое описание индикатора, используя при этом текстовый параметр конструктора "description".
 Посмотрите - не существует ли уже индикатор с точно таким же описанием в памяти компьютера (в кэше) - об этом будет упомянуто чуть позже.
 Если в кэше уже есть индикатор с соответствующим описанием, то нужно вернуть экземпляр закешированного индикатора.
 Если же такой индикатор еще не рассчитывался и отсутствует в кэше, то
 нужно создать экземпляр индикатора, использовав при этом оператор "new" а также все параметры



Видео ВЕБИНАРОВ



Дмитрия Власова



МТС под микроскопом: WLMA System

Подробный разбор торговой
системы WLMA System

3900 руб.

В корзину

Рубрики

Выберите рубрику

Обновления:



Новые посты по Email



Новые посты по RSS



Комментарии по RSS



Следим на Twitter

Метки

C# HowTo: **Wealth-Lab**
Видео Индикаторы МТС Торговые
роботы Юмор алготрейдинг
мероприятия обучение

Управление

Регистрация

Вход

RSS Записей

RSS Комментариев

WordPress.org

(включая описание), которые упоминались ранее в пункте 1.
сохраняем экземпляр индикатора в кэш (как это сделать - смотрите ниже).
возвращаем ссылку на экземпляр индикатора.

Ниже приведен код примера для метода Series, который рассчитывает индикатор простой скользящей средней (SMA).

Код примера на C# (кликните чтобы посмотреть) ?

```
1 //Статический метод Series(), который возвращает экземпляр класса indicator
2 public static SMA Series(DataSeries ds, int period)
3 {
4 //Создаем описание индикатора (далее будет индикатором конкретно этого экземпляра
5 индикатора):
6 string description = "SMA(" + ds.Description + "," + period + ")";
7
8 //Проверяем - не имеется ли уже в кэше аналогичный индикатор (совпадающий по
9 описанию)
10 if (ds.Cache.ContainsKey(description))
11 return (SMA)ds.Cache[description];
12
13 //Создаем экземпляр индикатора SMA, помещаем этот экземпляр в кэш и возвращаем
14 экземпляр индикатора
15
16 SMA sma = new SMA(ds, period, description); //объявляем
    переменную sma типа SMA и с помощью конструктора класса SMA создаем
    экземпляр и присваиваем ссылку на этот экземпляр нашей переменной.
    ds.Cache[description] = sma; //поещаем в кэш
    return sma; //возвращаем полученный экземпляр индикатора
}
```

Создание уникального описания индикатора

Первым шагом при реализации статического метода для создания индикатора является создание текстового описания ("Description"), которое уникальным образом идентифицирует этот индикатор.

Это описание формируется из названия индикатора ("name"), сразу за которым следуют параметры индикатора (конечно, если такие существуют), заключенные в круглые скобки. Между параметрами индикатора должны стоять запятые.

Обратите внимание на то, что при создании описания индикатора также используется свойство "Description" того ряда данных (DataSeries), который является параметром для нашего индикатора. Именно это обеспечивает уникальность текстового описания нашего индикатора.

К примеру, если индикатор использует для своего построения цену закрытия и период 20, то свойство "description" нашего индикатора будет выглядеть так:
SMA(Close,20)

Извлечение существующего экземпляра индикатора из кэша

У объектов класса DataSeries (ряды данных) и Bars (совокупность свечей) есть свойство, которое называется "Cache". Это свойство является "справочником". Несмотря на то, что работа с кэшем не является обязательной, Вы обязательно должны понять и применять в программировании механизм кеширования. Это нужно сделать для того, чтобы исключить создание множества совершенно одинаковых копий Вашего индикатора с аналогичными параметрами.

Метод Series() четко определяет уже созданный и находящийся в кэше индикатор. Это достигается благодаря использованию метода ContainsKey() объекта типа Dictionary. Текстовое описание индикатора всегда является ключом, т.е. уникальной записью точно идентифицирующей данный индикатор с данными параметрами.

Если в кэше уже есть соответствующий индикатор, то метод Series() вынимает этот индикатор из кеша и затем забрасывает его назад к индикаторам соответствующего типа. Кэш хранит все индикаторы как объекты типа DataSeries, поэтому Вам нужно перед возвращением объекта в кэш привести его к нужному типу данных.

Создание нового экземпляра индикатора

Если индикатор не найден в кэше, то метод Series() создает экземпляр данного индикатора используя при это конструктор. Для вызова конструктора используется оператор new.
Ваш индикатор должен иметь публичный конструктор, который использует точно такие же параметры, что и метод Series(), но дополнительно в конструкторе должен быть еще один параметр типа string который представляет собой описание, которое обязательно должно присутствовать в индикаторе. Использование описания индикатора в конструкторе в качестве дополнительного параметра преследует две цели:

1. Обеспечивается использование аналогичных текстовых строк как в кэше, так и в свойстве "Description" создаваемого индикатора.
2. Наличие такого параметра делает возможным использование оператора new для создание экземпляра класса создаваемого индикатора и предоставляют созданным таким образом экземплярам собственное описание. Здесь следует заметить, что индикаторы, созданные с помощью оператора new не участвуют в механизме кеширования.

Как только метод Series() создает экземпляр индикатора, он сразу же помещает этот экземпляр в кэш и отдает экземпляр данного индикатора в точку вызова метода.

Создание класса IndicatorHelper

Для того, чтобы иметь возможность использовать созданный Вами индикатор непосредственно в программе

Wealth-Lab, Вам необходимо создать специальный класс Helper, который описывает созданный Вами индикатор.

Класс Helper является объектом, который наследуется от базового класса IndicatorHelper, описание которого Вы найдете чуть ниже. Вы можете создать класс Helper в том же самом пространстве имен, где Вы создаете и основной класс индикатора.

Свойства базового класса IndicatorHelper:

public abstract Type IndicatorType

Возвращает ссылку на объект типа именно того индикатора, который поддерживается данным классом Helper.

public abstract IList <string>ParameterDescriptions

Возвращает строковый массив, каждая ячейка которого содержит параметр, который используется в конструкторе Вашего индикатора. Дает возможность дать краткое описание (одно или два слова) каждого параметра. Эти строковые короткие описания используются как метки в форме построения индикатора в программе Wealth-Lab для описания параметров индикатора.

Внутри кода Вы можете использовать любой класс, который поддерживает интерфейс IList. Однако обычно используется именно текстовый массив, который заполняется в момент объявления (смотри пример создания индикатора SMA).

public abstract IList <object> ParameterDefaultValues

Возвращает массив объектов, один объект соответствует в этом массиве одному параметру Вашего индикатора, которые указаны в конструкторе. Заполните данный массив теми значениями, которые параметры Вашего индикатора должны принимать когда в программе Wealth-Lab Вы будете хватать данный индикатор мышкой и бросать на график цен.

Вы можете использовать любой класс, который поддерживает интерфейс IList для того, чтобы реализовать его в Вашем классе Helper. Обычно используется статичный массив объектов, которые заполняются в момент их объявления (смотри пример создания индикатора SMA). Для определенных типов параметров используйте следующие значения в массиве:

- Параметры типа Bars: Вы должны установить тип данному объекту используя перечисление BarsDataType.Bars. Это явно укажет программе Wealth-Lab на то, что индикатор в качестве входного параметра использует объект типа Bars.
- Параметры типа DataSeries: Вы должны присвоить объекту одно из значений из перечисления CoreDataSeries. К примеру, это могут быть такие объекты, как open, high, low, close, или volume.
- Параметры типа enum (перечисления): Вы можете использовать любые созданные Вами либо уже существующие перечисления в качестве параметров. Форма, которая устанавливает параметры индикаторов в программе Wealth-Lab будет отображать раскрывающийся список (со стрелочкой вниз), который будет заполнен строковыми значениями, которые пользователь сможет выбрать из предлагающегося списка.
- Параметры типа int: Вы можете устанавливать значения типа int которые соответствуют значениям по умолчанию, которые должны принимать параметры. Или же Вы можете для установки параметра типа int использовать класс RangeBoundInt32, который устанавливает значение по умолчанию, минимальные, максимальные значения в своем конструкторе. Благодаря такому подходу пространство значений параметров, которые пользователь может использовать при использовании индикатора ограничивается.
- Параметры типа double: Точно также, как и параметры типа int, Вы можете присваивать значения типа double напрямую, либо используя класс RangeBoundDouble для ограничения области значений данного параметра.
- Параметры типа string: Вы можете устанавливать переменные типа string в качестве значений по умолчанию, а также такие параметры позволят пользователям вводить текстовые значения прямо в форме настройки параметров индикатора.
- Параметры типа DateTime: Если Вы хотите, чтобы пользователи получали возможность выбрать дату и время прямо в форме настройки параметров индикатора установите значение переменной типа DateTime по умолчанию, заполнив нулями поля, предназначенные для года, месяца и дневных настроек времени.

public abstract Color DefaultColor

Возвращает цвет, который должен использоваться по умолчанию для прорисовки индикаторов на графике.

public virtual LineStyle DefaultStyle

Возвращает Стиль линии (Solid, Dotted, Dashed or Histogram) который должен использоваться при прорисовке индикатора. Стиль "Solid" (сплошная) является параметром по умолчанию, поэтому если Вы хотите, чтобы линия индикатора была сплошной, Вам вообще не требуется переопределять значение этого свойства.

public virtual int DefaultWidth

Возвращает толщину линии, которая используется по умолчанию при прорисовке индикатора. Если Вы не переопределяете это свойство, то толщина линии устанавливается равной 1.

public abstract string Description

Возвращает описание индикатора. При этом рекомендуется использовать лишь несколько предложений. Это описание появится внизу списка индикаторов сразу после того, как Вы выберете мышкой конкретный индикатор в окне выбора индикаторов программы Wealth-Lab.

`public virtual string` URL

Возвращает ссылку на вебресурс (URL), на который пользователь должен перейти для того, чтобы получить дополнительную информацию о конструируемом Вами индикаторе. После нажатия на ссылку "More Info" в окне описания индикатора - программа Wealth-Lab запустит браузер и перейдет по этой ссылке на указанную Вами страницу. Если Вы не хотите отсылать пользователей ни на какую страницу - просто присвойте пустое строговое значение.

`public virtual string` TargetPane

Это свойство определяет - на какой области графика должен отображаться создаваемый Вами индикатор. Если Вы хотите, чтобы индикатор всегда отображался только в области цен (PricePane) - возвращайте здесь значение "P", если хотите, чтобы индикатор отрисовывался в области объемов - возвращайте значение "V". Возвращайте пустое значение строгового типа в том случае, если индикатор нужно отображать в той области, куда его "кинут" с помощью мышки.

Во всех прочих случаях, передайте другую строчку (обычно область называют по имени класса индикатора) для того, чтобы создать новую область на которой должен отображаться индикатор и вызвать эту область. Когда индикатор хватают мышкой и отпускают, программа Wealth-Lab пробует найти область цен, которая уже существует и в точности совпадает по названию с той строкой, которую Вы указали. Если такая область цен находится - то индикатор будет отображаться именно на этой области цен, а новая область цен при этом не будет создаваться. Эта особенность может использоваться для того, чтобы вызывать "группы" связанных индикаторов для отображения их на одной и той же области цен. Это касается таких индикаторов, как: ADX, ADXR, DI+, DI-.

`public virtual Type` PartnerBandIndicatorType

Если Вы строите индикатор, который всегда имеет пару в виде другого индикатора, например как верхняя линия Болинджера (BBandUpper) или нижняя линия Болинджера (BBandLower), Вы можете переопределив это свойство вернуть тип индикатора - партнера. Если Вы сделаете это, программа Wealth-Lab позволит пользователям прорисовать сразу две линии, когда одна из этих линий захватывается мышкой и бросается на график.

`public virtual Color` DefaultBandColor

Если Ваш индикатор является частью пары, Вы должны определить здесь цвет по умолчанию, который должен будет предлагаться когда пользователи собираются закрасить пространство между индикаторами.

`public virtual bool` IsOscillator

Переопределите и верните true в том случае, если Ваш индикатор должен отображаться как осциллятор и области перекупленности и перепроданности должны заполняться другим цветом. Если Вы сделаете именно так, то Вы сможете переопределить также методы осциллятора для модификации поведения индикатора по умолчанию.

`public virtual double` OscillatorOverboughtValue

Определите уровень перекупленности осциллятора. Значения, которые превышают этот уровень будут закрашены цветом, который будет определяться ниже. Уровнем по умолчанию является 0.

`public virtual double` OscillatorOversoldValue

Определите уровень перепроданности осциллятора. Значения, которые будут отображаться ниже чем этот уровень будут закрашиваться цветом, который определяется дальше. Уровнем по умолчанию является 0.

`public virtual Color` OscillatorOverboughtColor

Определите цвет, который будет использоваться для закрашивания области на осцилляторе, находящейся в зоне перекупленности. Цветом по умолчанию для этой области будет голубой.

`public virtual Color` OscillatorOversoldColor

Определите этот цвет - он будет использоваться для закрашивания области осциллятора, который располагается под уровнем перепроданности. Цвет по умолчанию - красный.

`public virtual Bitmap` Glyph

Переопределите свойство Glyph (логотип) в том случае, если хотите получить самодельный значок для отображения его в списке индикаторов программы Wealth-Lab. Картинка, которая используется в качестве логотипа должна быть размером 16x16, и в качестве фона использовать цвет Fuchsia.

Разработка статического метода Value()

Разработка такого метода является не обязательной. Вы можете это сделать по желанию. Многие индикаторы позволяют работать с методом Value(). Этот статический метод Value() позволяет классу Вашего индикатора просчитать и вернуть в точку вызова метода Value() значение "на лету" для определенного номера бара. При этом не происходит создания нового экземпляра индикатора.

Этот способ расчета соответствует режиму простой калькуляции, который применялся в предыдущей версии Wealth-Lab.

Если Вы примете решение поддержать это соглашение, то создавайте этот метод таким образом, чтобы он использовал точно такие же параметры, что и конструктор класса Вашего индикатора плюс еще один дополнительный параметр, который указывает номер соответствующего бара, для которого нужно просчитать значение индикатора.

Ниже приведен пример создания статического метода для индикатора, рассчитывающего простую скользящую среднюю.

Код примера на C# (кликните чтобы посмотреть)

```
1 //This static method allows ad-hoc calculation of SMAs (single calc mode)
2 public static double Value(int bar, DataSeries ds, int period)
3 {
4     if (ds.Count < period) return 0; else { double sum = 0; for(int i = bar; i > bar -
5     period; i--)
6     sum += ds[i];
7     return sum / period;
8 }
```

Использование индикаторов в коде WealthScript

Торговая система использует индикаторы в основном с использованием статического метода Series().

Пример:

Код примера на C# (кликните чтобы посмотреть)

```
1 //Create moving averages
2 DataSeries SMA20 = SMA.Series(Close, 20);
3 DataSeries SMA30 = SMA.Series(Close, 30);
```

Опытные пользователи могут использовать оператор new для создания экземпляра класса с помощью конструктора, попутно создавая и описание индикатора указывая его как параметр конструктора.

Пример:

Код примера на C# (кликните чтобы посмотреть)

```
1 DataSeries SMA20 = new SMA(Close, 20, "20 period SMA");
2 DataSeries SMA30 = new SMA(High, 10, "10 day SMA of Highs");
```

Создание собственного логотипа для библиотеки индикаторов

В программе Wealth-Lab созданная Вами библиотека индикаторов отображается как новая на панели со списком индикаторов. Индикаторы в Вашей библиотеке отображаются внизу папки, содержащей индикаторы после щелчка мышью по этой папке. По умолчанию, программа Wealth-Lab отображает папку, содержащую индикаторы с помощью стандартной иконки открытия/закрытия папки.

Вы можете поменять этот рисунок, использующийся для отображения папки на свой собственный рисунок. Для осуществления такой задумки - добавьте изображение в сборку Вашей библиотеки индикаторов. Убедитесь, что в Visual Studio свойств данного рисунка "Действие при построении" имеет значение "Внедренный ресурс". Обратите внимание также на то, что:

- название рисунка должно быть "glyph.bmp"
- размер рисунка должен быть 16x16 pixels
- в качестве заднего фона нужно использовать цвет "Fuchsia".

Пример создания индикатора простой скользящей средней

Представленный в качестве примера код содержит полную реализацию класса индикатора простой скользящей средней (SMA) и ассоциированным с ним класса Helper.

Код примера на C# (кликните чтобы посмотреть)

```
1 /*
2     SMA
3     Создание собственной библиотеки индикаторов для Wealh-Lab
4     на примере создания индикатора простой скользящей средней
5     Власов Дмитрий
6
7     http://finlabportal.ru
8
9 */
10
11 using System;
12 using System.Collections.Generic;
13 using System.Text;
14 using System.Drawing;
15 using WealthLab;
16
17 namespace IndicatorsExample
18 {
19     //создаем 2 класса - класс индикатора и класс IndicatorHelper
20
21     public class SMA : DataSeries //Класс, который создает SMA Indicator
22     (наследуется от DataSeries)
23     {
24         #region Приватные поля класса:
25
26         DataSeries _sourceSeries;
```

```

27         int _period;
28
29     #endregion
30
31     #region Публичный интерфейс
32
33     #region Статический метод Series() - возвращает экземпляр индикатора
34
35     public static SMA Series(DataSeries ds, int period) //в качестве параметра
36     здесь используется ряд данных и период
37     {
38
39         string description = "SMA(" + ds.Description + "," + period + ")"; //
40     Создаем описание:
41
42
43         if (ds.Cache.ContainsKey(description)) //Используя описание как ключ
44     смотрим - нет ли такого экземпляра индикатора в кеше
45         return (SMA)ds.Cache[description]; //в круглых скобках нужно
46     постаить название класса индикатора
47
48         //Если в кеше нужного экземпляра индикатора не нашли, то создаем,
49     помещаем в кеш, и возвращаем
50
51         SMA sma = new SMA(ds, period, description); //создаем
52         ds.Cache[description] = sma; //помещаем в кэш
53         return sma; //возвращаем в точку вызова метода
54     }
55
56     #endregion
57
58     #region Статический метод Value() - позволяет добавить вычисление
59     индикатора SMA на каждом конкретном баре
60
61     public static double Value(int bar, DataSeries ds, int period) //
62     дополнительный параметр метода - номер бара
63     {
64         if (ds.Count < period) //если данных не хватает для расчета индикатора
65         return 0;
66         else
67         {
68             double sum = 0;
69             for (int i = bar; i > bar - period; i--) //пробегаемся по всем
70     барам от текущей свечи влево для вычисления
71
72                 sum += ds[i];
73
74             return sum / period;
75         }
76     }
77
78     #endregion
79
80     #region Конструктор класса индикатора
81
82     public SMA(DataSeries ds, int period, string description)
83     : base(ds, description) //в качестве параметров здесь используется
84     DataSeries (а может быть bars)
85     {
86         //Запоминаем значение параметров в переменных
87         _sourceSeries = ds;
88         _period = period;
89
90         //Определяем номер первого бара, на котором наш индикато будет
91     корректно считаться.
92         FirstValidValue = period - 1 + ds.FirstValidValue;
93
94         //Расчитываем значение скользящей средней на каждом из баров и
95     присваиваем значение каждой из ячеек.
96         for (int bar = period - 1; bar < ds.Count; bar++)
97         {
98             double sum = 0;
99             for (int innerBar = bar; innerBar > bar - period; innerBar--)
100                 sum += ds[innerBar];
101             this[bar] = sum / period;
102         }
103     }
104
105     #endregion
106
107     #region Метод для расчета значения индикатора для "живого", еще
108     неполностью сформировавшегося бара.
109
110     public override void CalculatePartialValue()
111     {
112         if (_sourceSeries.Count < _period - 1 || _sourceSeries.PartialValue ==
113     Double.NaN)
114             PartialValue = Double.NaN;
115         else
116         {
117             double sum = 0;
118             for (int bar = _sourceSeries.Count - _period + 1; bar <
119     _sourceSeries.Count; bar++)
120                 sum += _sourceSeries[bar];
121             sum += _sourceSeries.PartialValue;
122             PartialValue = sum / _period;
123         }
124     }
125     #endregion
126
127
128     #endregion
129 }
130
131
132 public class SMAHelper : IndicatorHelper //Helper class - этот класс позволяет
133 работать индикатору в программе WealthLab 6
134 {
135
136     #region Приватные поля класса SmaHelper
137
138     private static object[] _paramDefaults = { CoreDataSeries.Close, new
139     RangeBoundInt32(20, 2, Int32.MaxValue) };
140     private static string[] _paramNames = { "Финансовый инструмент", "Период
141     усреднения" };

```

```
142
143         #endregion
144
145         public override IList<object> ParameterDefaultValues //Возвращает значения
146 параметров по умолчанию
147     {
148         {
149             {
150                 return _paramDefaults;
151             }
152         }
153
154         public override IList<string> ParameterDescriptions //Возвращает описание
155 параметров
156     {
157         {
158             {
159                 return _paramNames;
160             }
161         }
162
163         public override Color DefaultColor //Устанавливает цвет индикатора по
164 умолчанию
165     {
166         {
167             {
168                 return Color.Red;
169             }
170         }
171
172         public override Type IndicatorType //Возвращает ссылку на поддерживаемый тип
173 индикатора
174     {
175         {
176             {
177                 return typeof(SMA);
178             }
179         }
180
181         public override string Description //возвращает текстовое описание
182 индикатора (видно в Wealth-Lab)
183     {
184         {
185             {
186                 return @"Индикатор рассчитывает скользящую среднюю и используется
187 как пример на курсе Власова Дмитрия.";
188             }
189         }
190
191         public override string URL //возвращает ссылку, перейдя по которой можно
192 получить дополнительную информацию
193     {
194         {
195             {
196                 return @"http://finlabportal.ru/2012/05/proektirovanie-i-postroenie-
197 indikatorov-v-wealth-lab/";
198             }
199         }
200     }
201 }
202 }
```

Если Вам понравилась эта статья - подпишитесь на **новые СТАТЬИ по RSS** или получайте наши **новые СТАТЬИ на E-mail**:
Рекомендую почитать также:

- Торговая система HighLowLong – шаблон кода и оптимизация.
- Анализ данных в Wealth-Lab с учетом разных таймфреймов
- Схема генератора механических торговых систем и создание новой торговой техники
- Как установить связь между Wealth-Lab и Visual Studio для отладки торговой стратегии
- Видео вебинара по программе Wealth-Lab в рамках курса “Wealth-Lab: C# на службе трейдера”

 РУБРИКИ: **WEALTH-LAB**

Комментариев: 6

Vladimir пишет:

23 августа 2012 в 14:29



Здравствуйте. Подскажите пожалуйста почему в параметрах оптимизации в WLD 6.3 у меня методы – Перебором и Монте – Карло, а генетической оптимизации нет? Может какой плагин установить нужно.

Ответить

Vladimir пишет:

23 августа 2012 в 14:53



Все разобрался, скачал Addins и все заработало

Ответить

Дмитрий Власов пишет:

23 августа 2012 в 18:15



Владимир, я даже ответить не успеваю, на Ваши вопросы... 😊

Ответить

Mr_oz пишет:

31 августа 2012 в 17:35



Доброго дня. Подскажите пожалуйста данная возможность сборки собственных индикаторов в библиотеке присутствует в версии 5.4 ?

Ответить

Дмитрий Власов пишет:

31 августа 2012 в 21:17



Да, конечно.

Ответить

sa пишет:

3 октября 2012 в 15:50



Дмитрий, а можно где-нибудь увидеть пример создания коллекций индикаторов?
Что-то не могу допетрить как это делается. Задача состоит в том, чтобы создать несколько разнопериодных ета и сохранить их параметры в массив, который потом можно будет перебирать в циклах?
Спасибо.

Ответить

Оставить комментарий

Имя (Обязательно)

Почта (Обязательно)

Вэб-сайт

Получать новые комментарии по электронной почте. Вы можете [подписаться](#) без комментирования.

© 2012, Финансовая лаборатория. Все права защищены.
Любое использование материалов допускается только с указанием активной ссылки на источник.

213

75

70

👁

📄

👤

Я

В день мой блог посещает 84 человека, сегодня уже 43

mail.ru

РЕЙТИНГ

757382

246

75

Анонсы

BY FEEDBURNER

Подписка

- [RSS](#)
- [Twitter](#)
- [ВКонтакте](#)
- [facebook](#)

Сайт

- [Создать блог](#)
- [Содержание](#)
- [О блоге](#)
- [Карта](#)



ТОРГОВЫЕ РОБОТЫ (открытая база знаний)

Главная

Обсуждения

Люди

[Главная](#) / [WLRT \(Wealth-Lab по русски\)](#)

Дмитрий Власов задал вопрос 02 августа 2012 в 22:32

С ОТВЕТОМ

Пользователь

Что указывать в методе SetShareSize() когда торгуем акциями - лоты или штуки акций?

Подскажите, плз., в методе SetShareSize() когда мы торгуем акцией количество нужно показывать акций или лотов?

Т.е. например чтобы один лот по газпрому и сбербанку купить - что нужно указать внутри скобок метода SetShareSize()?

0



У меня этот вопрос. Голосовать!

Официальный ответ компании



Власов Дмитрий написал 5 месяцев назад

Создатель сообщества

Подробный ответ был дан в комментариях.

2 комментария

[Добавить комментарий](#)

Дмитрий Власов написал более 5 месяцев назад

Пользователь

Внутри скобок метода SetShareSize() нужно указывать не лоты, а акции. Однако, количество акций должно быть кратно лоту.

На самом деле логика довольно проста. Цена идет за акцию, поэтому при покупке 1 лота денег тратится денег за столько акций, сколько содержится в лоте. Поэтому объем в штуках акций.

Чтобы добиться кратности можно воспользоваться при программировании следующей логики:

```
int lot = 10; //показываем, сколько акций содержится в лоте (для каждого вида акций свое количество акций в лоте)
int vol = 118; //каким либо методом рассчитываем, сколько нужно акций купить.
```

```
SetShareSize(vol - vol % lot); //определяем количество акций, кратное лоту.
```

Выражение: (vol % lot) дает в итоге остаток от деления 118 на 10 (т.е. 8)

В итоге получается: 118-8 = 110

110 кратно десяти, так что в итоге купится 110 акций (или 11 лотов).

★ Это — ответ на вопрос! [Ответить](#) [Пожаловаться](#)



Дмитрий Власов написал более 5 месяцев назад

Пользователь

Вот пример определения количества акций:

```
double Equity = 200000; //Указываем здесь размер счета
int SharesInLot = 10; //показываем, сколько акций содержится в лоте (для каждого вида акций свое количество акций в лоте)
double Shares = 0; //расчетное количество акций (еще не кратное лотам) изначально равняется нулю рассчитывается при входе в позицию.
double InitialStop = 0; //объявляем переменную, где будет храниться величина стоп-лосса (после расчета)
```

//Чтобы добиться кратности можно воспользоваться при программировании следующей логики:

Создать обсуждение

[Следить за вопросом](#)[Подписаться на RSS](#)[Пожаловаться](#)

Поделиться:

Сохранить



Мне нравится

1

[Поделиться](#)

WLRT (Wealth-Lab по русски)

3 комментария

1 участник

1 подписчик

//int SharesInLot = 10; //показываем, сколько акций содержится в лоте (для каждого вида акций свое количество акций в лоте)
//int Shares = 118; //каким либо методом рассчитываем, сколько нужно акций купить.
//SetShareSize(Shares - Shares % SharesInLot); //определяем количество акций, кратное лоту.
//Выражение: (Shares % SharesInLot) дает в итоге остаток от деления 118 на 10 (т.е. 8)
//В итоге получается: 118-8 = 110
//110 кратно десяти, так что в итоге купится 110 акций (или 11 лотов).

Shares = (int) ((Equity * 0.03) / (Close[bar] - InitialStop)); //определяем количество акций для покупки (не кратно лоту) в зависимости от риска на одну сделку и отдаленности Стоп-Лос
SetShareSize(System.Math.Floor(Shares - Shares % SharesInLot)); //внутри метода определяем количество акций, которое кратно лоту.

★ Это — ответ на вопрос! [Ответить](#) [Пожаловаться](#)



B *I* U

Добавить комментарий