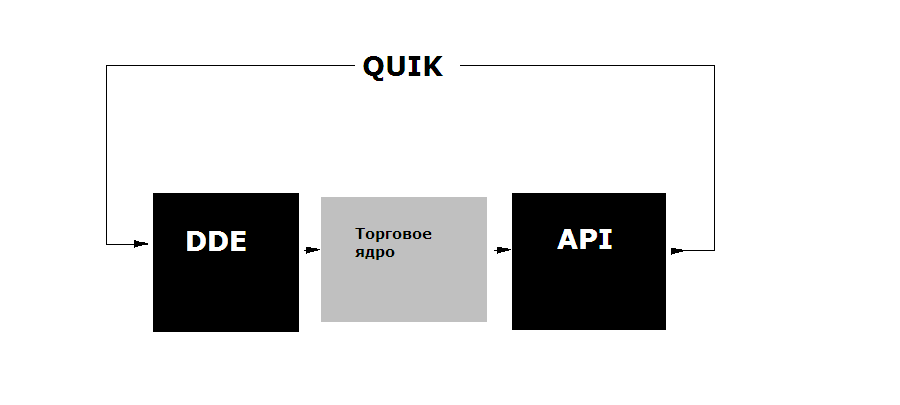
Бизнес-требования

Алгоритм считывает котировки из текстового файла и на выходе дает торговые сигналы.

Задача заключается в создании платформы обладающей функциями экспорта котировок из QUIK (через DDE) и обратной отправки торговых сигналов (через API).

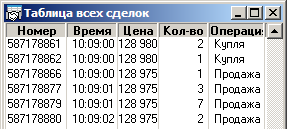


На данной схеме показана общая идея, как это должно выглядеть. Вы создаете DDE и API блоки, которые имеют точки присоединения моего торгового ядра. Ядро в дальнейшем может модифицироваться (вплоть до коренных изменений технологии), но универсальность платформы должна позволять встраивать в нее абсолютно любые торговые алгоритмы.

К данному письму прикладывается проект VB 8.0 с наброском, как это может выглядеть на форме (Windows окне).

БЛОК Экспорт котировок через DDE

1. Экспорт котировок должен происходить из таблицы всех сделок.



1. Колонки таблицы «**Всех сделок**» настраиваются согласно изображению: Номер, Время, Кол-во, Операция.
2. Название в QUIKе самой таблицы всех сделок должно соответствовать названию заданной на форме. Например, это может быть RIU2 или SiU2. Название таблицы задается в поле, напротив «**Name T.Sd:».** Поле имеет маску ввода, только на латинском и не более 4 символов.
3. QUIK выбирается согласно заданному пути. Путь задается вручную, либо через расположенную рядом кнопку «Обзор».
4. Задается время, в которое платформа знает что сейчас должны поступать котировки. Время прописывается в промежутках от «Начала осн.» до «Нач. клиринга» и т.д. Полям задается маска. ЧЧ.ММ.СС.
5. Создается контрольное поле «Текущее время» в котором отображается время по которому живет робот. Чтобы можно было отследить насколько оно соответствует времени QUIK.
6. Задается таймфрейм, в который DDE конвертирует таблицу всех сделок на выходе. В результате получаем свечу с заданным таймфреймов стандартном формате: **инструмент,** **дата, время,** **цена открытие, максимум, минимум, закрытие, объем**. Построение свечи должно происходить в реальном времени. Это значит, что если например таймфрейм 5 минут, то за это время могут происходить изменения (изменится минимум или максимум). Все это должно учитываться, чтобы торговое ядро могло оперативно реагировать в момент срабатывания заявки или стоп сигнала.
7. Таймфрейм должен изменятся в широких пределах. От 10 сек до стандартных диапазонов. Фактически в поле таймрейма задается интервал в секундах. 10 сек, 30 сек, 60 сек = 1 мин, 300 сек = 5 мин.
8. Так же нужно задать искусственный ограничитель по времени обновления (перерисовки свечи в нутрии таймфрейма). Это скажем полезно при работе с часовыми барами. Каждую секунду перестраивать свечи не нужно. А полезно например раз в минуту. В поле «Частота обращений» задается в секундах ограничитель.
9. Экспорт котировок из QUIK начинается и заканчивается при нажатии кнопки «Начать/Закончить DDE».
10. Во время DDE передачи все котировки из таблицы всех сделок должны одновременно сохранятся в текстовой файл (система архивирования данных). Папка для хранения файла с котировками задается. Поле «Архив T.Sd». Кнопкой обзор можно выбрать папку. Название файла присваивается в формате: «ЧЧ.ММ.ГГ». По окончанию торговой сессии, время которой соответствует заданным интервалам файл перестает записываться. И на следующий торговый день ведется новый файл с котировками.
11. Альтернативно во время DDE передачи полученные свечи так же должны архивироваться в файлы. Место для хранения которых задается в поле «Архив тайм».

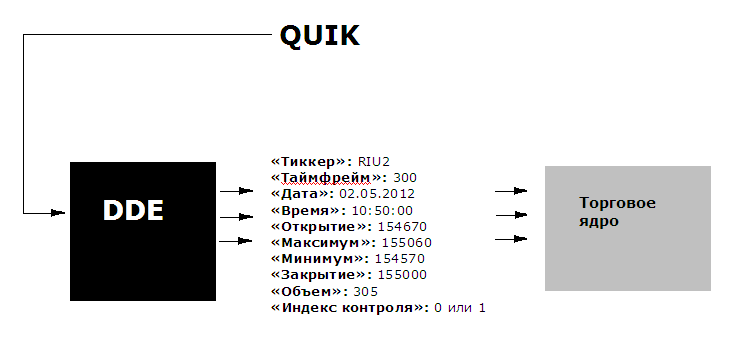
Исключение в том, что файл с данными котировками должен вестись непрерывно. В одном файле несколько торговых сессий. Менять файлы будут руками сам.

Форма записи данных в файл:

RIU2,5,20100111,120500,155565.00,155640.00,155405.00,155440.00

1. Все введенные настройки должны так же сохранятся в единый файл настроек робота. Путь к расположению которого расположен в поле «Настройки». При включении робота, он считывает настройки из файла и загружает их. По окончанию работы с роботом, настройки перед закрытием программы сохраняются.
2. В поле «Лог действий» должны отображаться все действия которые производит программа, или те ошибки которые она выдает. Например в случае успешного подключения пишет, что все хорошо. Или пишет что не так. Что нужно подправить.
3. Текущая свеча которая идет в ходе построения должна отображаться в последних полях.

На выходе модуля DDE должны получить следующий массив данных:



На выходе имею массив готовых свечей. При появлении новой свечи, ее параметры записываются в новую строку массива. Если таймфрейм не закончен тогда в столбец «Индекс контроля» записываем «0». Если свечка закончена формироваться, тогда Индекс = 1. Внимание, что свечка меняет свои параметры в он-лайне, с заданным временным ограничителем обновления данных.

Система отказоустойчивости

Блок DDE должен проверять, активен ли QUIK. Когда нажимаем кнопку «Начать/закончить DDE» подсвечивается поле QUIK активен если все хорошо. Если DDE работает нормально, то подсвечивается поле DDE активно.

В ходе работы проверяется поступление котировок из таблицы всех сделок. Состояние отображается в поле «Передача активна». Если по каким либо причинам котировки не поступают в таблицу всех сделок в течение 30 минут и более (срок контроля задается полем «Срок контроля» в секундах) происходит автоматическое переключение на другой сервер расположенный следующим в таблице QUIK «Доступные соединения».

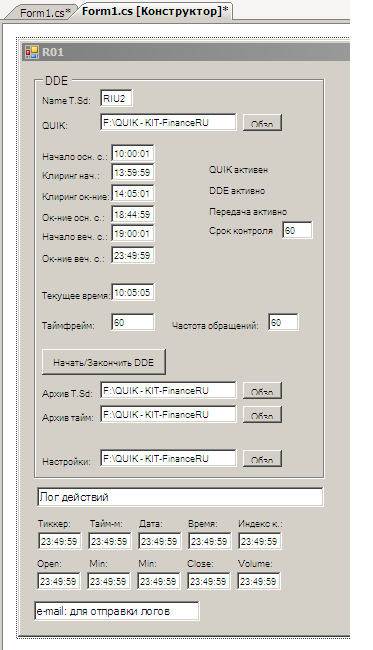
Когда происходит переключение на другой сервер, таблица всех сделок в QUIK перезаписывается заново. Желательно чтобы пропущенные данные дозаписывались в массив и файл а не перезаписывались полностью.

Так же нужно предусмотреть, что в случае если мы отключаем DDE передачу через кнопку «Начать/Закончить DDE», то при обратном включении DDE файлы дозаписывались пропущенными котировками за текущую торговую сессию. Даже если мы запустили робота позже начала торговой сессии. Это происходит через контроль времени начала/окончания основной, вечерней сессий.

Требуется предусмотреть минимизацию потерь данных в случае внештатных ситуаций и как это отразится на торговом ядре. Внештатные ситуации могут быть разные (при DDE передачи) обрыв канала связи, сбой на сервере у брокера или проблемы у биржи. Во время ремонтных работ будет провал в поступающих котировках. И в момент восстановления работы мы получим существенное изменение в цене.

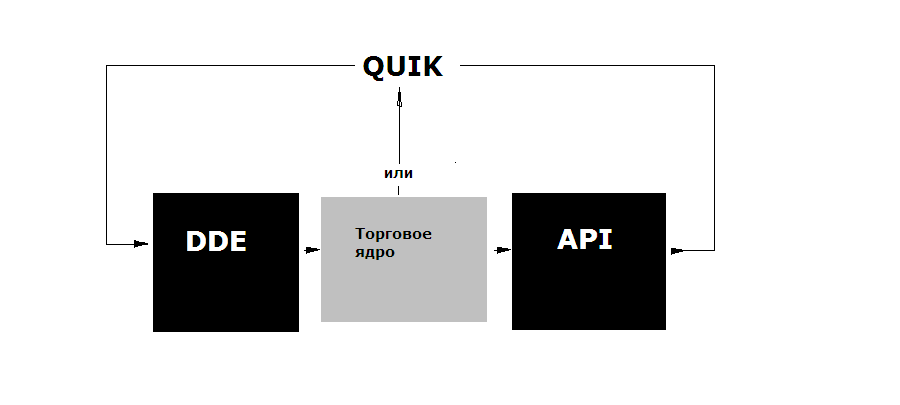
Хотелось бы, чтобы все логи, которые поступают в поле «Лог действий» дублировались на заданную электронную почту (e-mail: для отправки логов). В случае получения письма, что котировки не идут, - на ПК с роботом останавливаем робота через кнопку остановить DDE передачу, самостоятельно переключаюсь на другой сервер и через кнопку запускаю заново DDE передачу.

Предусмотреть реализацию возможности, чтобы котировки по DDE могли поступать с одной квика, а торговые поручения могли идти на любой другой QUIK. Это по крайней мере позволит держать несколько резервных терминалов, т.к. у брокеров вылеты в работе сервера наиболее часто встречаются.



На данном изображении показано расположение на форме зон по заданию настроек.

Система взаимодействия модулей



Как видно из изображения - полный цикл работы робота разбит до трех этапов.

В начале мы получаем котировки из DDE. Получили котировку, конвертировали их в свечу, сделали обращение к массиву хранения свечей (МХС), записали или перезаписали данные.

Далее Торговое ядро обращается к МХС считывает от туда последние данные. Смотрит, если новая цена привела к генерации торговых сигналов, то вызывает блок API, если нет, то возвращает контроль Блоку DDE.

БЛОК подачи торговых поручений API

1. Данный блок должен быть универсальным. В нем необходимо предусмотреть прописание всех возможных начальных полей идентификаторов клиента, необходимых для подачи торговых поручений на все возможные площадки:

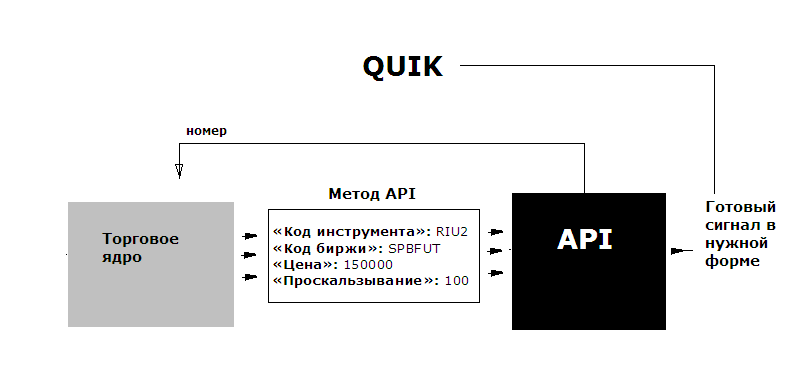
Торговый счет и счета депо и прочее. Это необходимо для того, чтобы можно было торговать абсолютно на любой российской бирже.

1. Задаем поле расположения QUIK. Данная функция все же предусматривает возможность получения котировок через один квик и отправки торговых получений на другой терминал.
2. Если API передача активна, QUIK активен, Интеренет передача активна сигнализируем об этом в соответствующих полях альтернативно DDE передачи.
3. Кнопка «Начать/Закончить API» альтернативно DDE подключает или отключает торговый блок. Если торговый блок выключен, тогда при посылки в него торговых поручений от торгового ядра, они не попадают в QUIK и на оборот. Это позволит работать в демо режиме.
4. В файле «Архив сделок» отображаются все сделки поступившие на биржу, а так же контроль их исполнения.
5. Так же как и в блоке по DDE все заполненные поля запоминаются в файл настроек.
6. Поле «частота проверки исполнений» задается в секундах и служит для проверки отданных поручений.
7. Все логи действий блока API отображаются в «Логе действий» расположенном в DDE передаче и альтернативно по возможности дублируются на почту.
8. В период перерывов (клиринга) или отсутствия торгов (время задается в DDE блоке начало и оконочание, клиринги), торговые сигналы не поступают в QUIK. Если же в рабочее время по каким то причинам торговые сигналы в Квик не поступают (например сбой у брокера, нужно уведомить сигнальными индикаторами, в логи и на почту).
9. Кнопки снять заявки и стоп заявки, закрыть все позиции, докупить/допрадать заданное количество выполняют простые оперативные функции согласно их названиям и предназначены для внештатных ситуаций. Они выполняют поручения по тем инструментам которые прописаны в поле Код инструмента и Код биржи.
10. Контроль исполнения всех заявок Блоком API происходит самостоятельно.

Точкой для подключения Торгового Ядра к API служат команды вызова торговых поручений. Сам блок должен быть выполнен в виду набора методов. Каждый метод выполняет отдельную торговую операцию. Задача API вызвать нужный метод и передать ему все необходимые параметры.

Метод из формы берет нужный ему торговый счет и отправляет заявку в заданный QUIK. Самостоятельно проверяет исполнение торгового поручения и по результату отправляет в лог действий, в файл и на почту отчет.

Взаимодействие блока API с Торговым ядром



Торговое ядро должно вызвать нужный ему метод блока API и передать ему список необходимых параметров который ограничивается максимум теми, что перечислено на картинке. Размер забыл довить

Например, Ядро вызывает метод API «Купить по рынку», передает ему бумагу, код биржи, цену и цена + просакальзывание = цена исполнения (если это фортс).

x.BayMarket(“RIU2”,”SPBFUT”,1500000,150100,10);

Торговые счета робот сам вставляет исходя из данных на форме

В роботе необходимо использование всех возможных торговых поручений:

Купить/продать по рынку

Лимитированные заявки на покупку/продажу

Стоп заявки (стоп лосс, тейк профит)

Снять все заявки,

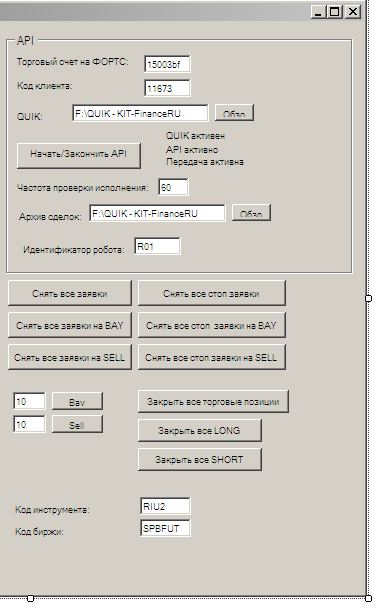
Снять заявку на покупку/продажу,

Снять стоп заявки/только на покупку/только на продажу

Заменить заявку/стоп заявку.

Снять определенную заявку или стоп заявку.

После операции, блок API должен вернуть номер сделки или выставленной стоп заявки, чтобы в случае необходимости можно было снять или заменить конкретную заявку.



Так выглядит панель управления API передачей

Хотелось бы, чтобы и API и DDE блоки располагались на отдельных вкладках в роботе. Т.е. всего нужно три вкладки (две для блоков и одна для торгового ядра).

Окно могла подстраиваться в размерах под размер монитора, и были полосы прокрутки, чтобы можно было увидеть скрытые кнопки.

В целом, пока все.

Мне нужно понимать, из какого массива считывать котировки (и формат данных в нем) и описание методов (для тупых) по вызову торговых функций в блоке API.

Так же стоит указать в какое место кода нужно вставлять торговое ядро.

Хотелось бы чтобы каждая технология (БЛОК) писался в виде отдельных прикрепленных к проекту классам.

Жду от Вас замечаний, вопрос и предложений по дополнению. А так же хотелось бы понять, что на выходе после оказания услуги я получаю, как это выглядит.