

TraderAssistant

Руководство пользователя

Версия 1.0

Содержание

Основные функции TraderAssistant	3
Описание структуры программного комплекса	4
Описание папок и файлов	5
Ограничения демонстрационной версии	5
1. Подготовка к работе	6
1.1 Установка, настройка и запуск программы	6
1.2 Получение лицензии на использование	9
2. Работа с программой TAOOrderSrvr	10
2.1 Режимы функционирования и общий алгоритм работы с TAOOrderSrvr	11
2.2 Настройка программы	12
2.3 Сохранение и загрузка конфигурации окон	13
2.4 Журнал работы программы	13
2.5 Установка соединения	13
2.6 Управление торговыми счетами	14
2.6.1 Таблица "Счета"	14
2.6.2 Добавление/редактирование счета	14
2.7 Управление стратегиями	15
2.7.1 Таблица "Стратегии"	15
2.7.2 Добавление/редактирование стратегии	16
2.8 Лимиты – автоматизированная торговля	16
2.8.1 Таблица "Лимиты – автоматизированная торговля"	16
2.8.2 Добавление/редактирование лимита	18
2.9 Лимиты – торговля в ручном режиме	19
2.9.1 Таблица "Лимиты – ручная торговля"	19
2.10 Сигналы	19
2.10.1 Таблица "Сигналы"	20
2.11 Заявки	20
2.12 Сделки	20
2.12.1 Добавление/редактирование сделки	21
2.13 Отложенные заявки	21
2.13.1 Таблица "Отложенные заявки"	22
2.14 Аналитические сделки	22
2.15 Котировки	23
2.15.1 Таблица "Котировки"	23
3. Работа с программой TAlerter	23
3.1. Настройка программы	24
3.2. Проверка настроек программы	25
4. Работа с библиотекой AMI.dll	25
4.1. AMI_IsConnected	25
4.2. AMI_GetCurrentFORTSPosition	25
4.3. AMI_GetCurrentMICEXPosition	25
4.4. AMI_GetCurrentMICEXMoney	26
4.5. AMI_GetCurrentFORTSMoney	26

4.6. AMI_GetCurrentPosition	26
4.7. AMI_GetCurrentPositionNoOrder	26
4.8. AMI_GetCurrentMoneyLimit	27
4.9. AMI_PushOrder	27
4.10. AMI_PushOCOOrder	28
4.11. AMI_KillOrders	28
4.12. AMI_PushStopLoss	29
4.13. AMI_PushTakeProfit	29
4.14. AMI_PushStopTake	30
4.15. AMI_GetLastOrderStatus	31
4.16. AMI_GetLastOrderRemainder	31
4.17. AMI_GetBuyDeals	32
4.18. AMI_GetSellDeals	32
5. Таблицы базы данных	33

Введение

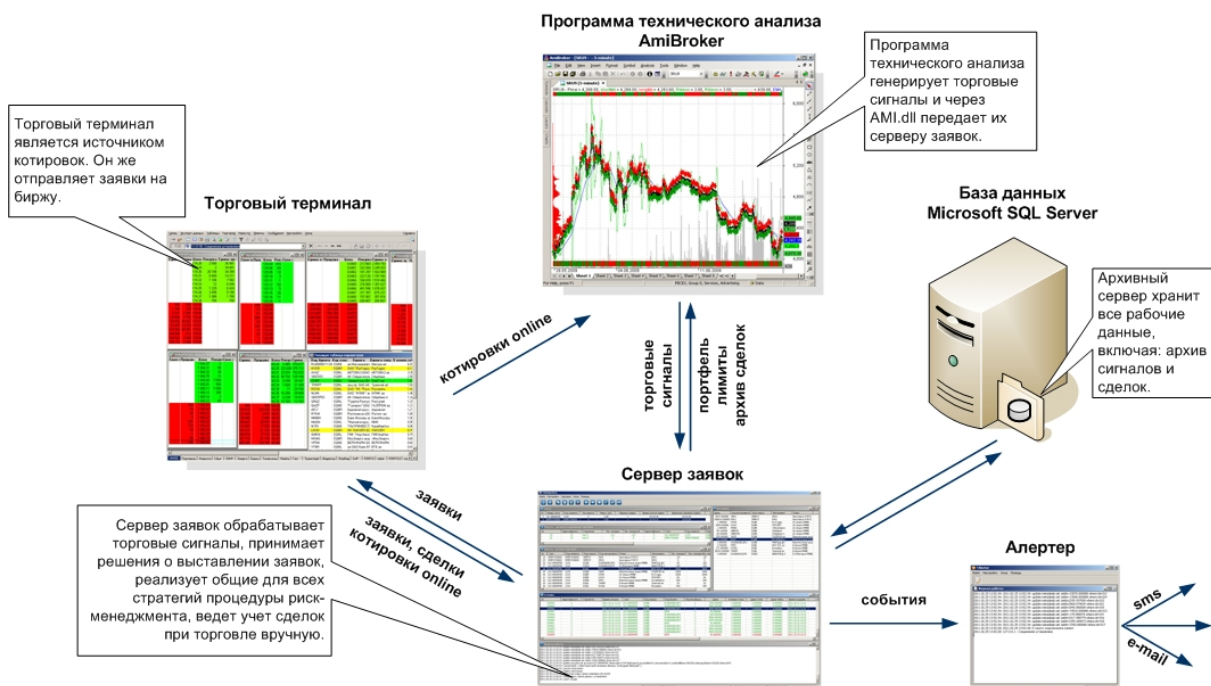
В этом Руководстве по эксплуатации описаны правила работы с программным комплексом **TraderAssistant**, его основные функции, порядок установки и настройки.

Основные функции TraderAssistant

1. Для торговли в ручном режиме:
 - Работа с несколькими торговыми счетами.
 - Перенос заявок между торговыми сессиями, принудительное снятие заявок в задно пользователем время.
 - Автоматическая загрузка сделок из QUIK, возможность ввода информации о сделках вручную
 - Сохранение сделок в базе данных для последующего анализа.
 - Расчет в реальном времени доходности завершенных сделок и открытых позиций.
 - Просмотр сделок в Amibroker на графике цены.

2. Для автоматизации торговых стратегий:
 - Работа с несколькими торговыми счетами.
 - Автоматизированное выставление заявок из неограниченного числа торговых стратегий под управлением Amibroker, ведение журнала торговых сигналов, заявок и сделок.
 - Просмотр сделок в Amibroker на графике цены.
 - Расчет в реальном времени доходности завершенных сделок и открытых позиций.
 - Работа с различными типами заявок: простые заявки, связанные заявки (OCO), стоп-лоссы, тейк-профиты.
 - Сохранение сделок в базе данных для последующего анализа.
 - Поддержка реализации различных стратегий управлений капиталом (за счет собственных средств и средств программы технического анализа).
 - Перенос заявок между торговыми сессиями, принудительное снятие заявок в задно пользователем время.
 - Информирование пользователя о поступлении торговых сигналов, проведении сделок, возникновении ошибок: разрыве соединения с сервером брокера или потере работоспособности вследствие программных сбоев (e-mail, sms, всплывающие окна, звуковые сигналы).
 - Работа в тестовом режиме: заявки от автоматизированных торговых стратегий не отправляются в торговый терминал, а исполняются непосредственно сервером заявок. Данный режим позволяет тестировать стратегии в условиях, максимально приближенных к реальным.

Описание структуры программного комплекса



В состав TraderAssistant входит:

- Плагин AMI.dll, встраиваемый в программу технического анализа Amibroker и позволяющий организовать обмен данными между торговой стратегией и сервером обработки заявок TAOOrderSrvr.
- Сервер обработки заявок TAOOrderSrvr, реализующий основные функции управления капиталом и обеспечивающий двустороннюю связь между AmiBroker и QUIK. Сервер заявок должен быть установлен на том же компьютере, где работает QUIK и AmiBroker. Сервер заявок может одновременно обрабатывать сигналы множества торговых стратегий.
- Алертер TAlerter, информирующий трейдера о важных моментах работы системы, таких как выставление заявок, совершение сделок, потеря соединения с сервером QUIK или отказ одного из модулей системы. Алертер может устанавливаться как на компьютере, на котором запущен сервер, так и на любом другом компьютере (связь сервером осуществляется через TCP/IP). Способ информирования о событиях выбирается трейдером из поддерживаемых вариантов: смс, e-mail, звуковое оповещение, всплывающие окна.

СОВЕТ! В целях повышения надежности функционирования системы TAlerter рекомендуется устанавливать на отдельный компьютер. Таким образом при любом аппаратном или программном сбое, в результате которого выключится/перезагрузится компьютера с TAOOrderSrvr алертер сможет незамедлительно уведомлять об этом трейдера.

- База данных для хранения истории работы программного комплекса, включая архив сделок. База данных работает по управлению Microsoft SQL Server и может быть установлена на любом компьютере, к которому есть сетевой доступ.

ВНИМАНИЕ! Программа осуществляет доступ к базе данных через ODBC источник TraderAssistant. В руководстве по установке описан вариант, при котором база данных и сервер заявок располагаются на одном компьютере. Отредактируйте ODBC источник

данных вручную для получения более широких возможностей по управлению местонахождением вашей базы данных.

Описание папок и файлов

Папка	Файл	Описание
Корневая папка ¹	TAOrderSrvr.exe ²	Сервер заявок, модуль управления
Корневая папка	AMI.dll	Библиотека, встраиваемая в Amibroker
Корневая папка	TRANS2QUIK.dll	Библиотека-интерфейс QUIK для выставления заявок
Корневая папка	TAlerter.exe	Алертер
Корневая папка	OrderSrvrSettings.xml	Файл конфигурации для TAOrderSrvr.exe
Корневая папка	license.rtf	Лицензионное соглашение
Корневая папка\Strategies		В этой папке будут храниться стратегии по мере создания их пользователем
Корневая папка\Logs		В эту папку сервер заявок сохраняет журналы работы
Корневая папка\Database	TA.mdf	База данных для Microsoft SQL Server 2005

Ограничения демонстрационной версии

Сервер заявок TAOrderSrvr

- Не получает из QUIK сделки, однако может получать все остальные данные.
- Не передает никакие сигналы в QUIK.
- Работает только в режиме тестирования.
- От автоматизированных стратегий обрабатываются только сигналы PushOrder.

Библиотека для Amibroker

- Работают все функции получения значений (текущий портфель, лимиты и т.п).
- Из функции отправки заявок доступна только PushOrder, номер счета и код клиента для работы: DemoAccount и DemoClientCode соответственно.
- Функции получения сделок (GetBuyDeals и GetSellDeals) возвращают не более трех сделок.

Алертер

Интерфейс пользователя доступен полностью, программа принимает все сообщения, но не сигнализирует звуковым сигналом и всплывающим окном, не отправляет сигналы по смс и электронной почте. Только мигает иконкой на панели инструментов.

¹ Корневая папка – папка, в которую установлена программа

² В демонстрационной версии к названиям файлов в конце добавляется 'Demo'

1. Подготовка к работе

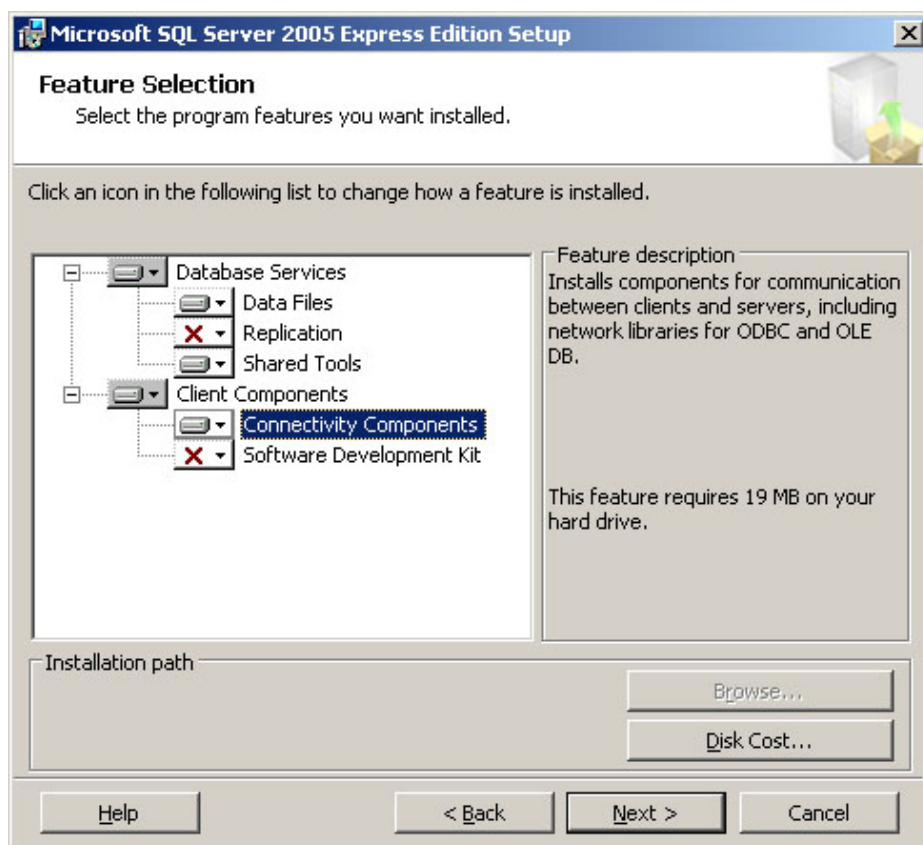
1.1 Установка, настройка и запуск программы

ВАЖНО! Выполняйте каждый следующий шаг только при успешном завершении предыдущего. Подобный подход позволит избежать множества возможных проблем, связанных с невнимательностью.

Шаг 0. Установка Microsoft SQL Server 2005

1. Скачайте и установите бесплатную СУБД Microsoft SQL Server Express: <http://www.microsoft.com/downloads/ru-ru/details.aspx?familyid=220549B5-0B07-4448-8848-DCC397514B41&displaylang=ru>

При установке в списке выборы обязательно отметьте установку компонентов возможности соединения (см. рис)



В остальном для наших целей достаточно следовать опциям, предлагаемым установщиком по умолчанию. Для более детальной информации об установке Microsoft SQL Server обращайтесь к сопроводительной документации.

2. Установите бесплатное приложение Microsoft SQL Server Management Studio для удобной работы с базами данных:

<http://www.microsoft.com/downloads/ru-ru/details.aspx?FamilyID=c243a5ae-4bd1-4e3d-94b8-5a0f62bf7796>

Следуйте опциям, предлагаемым установщиком по умолчанию.

Шаг 1. Установка программы

Запустите TraderAssistant.msi и следуйте инструкциям установщика. Программа не установится, если на машине отсутствует Microsoft SQL Server – в этом случае программа выдаст сообщение о необходимости установки Microsoft SQL Server.

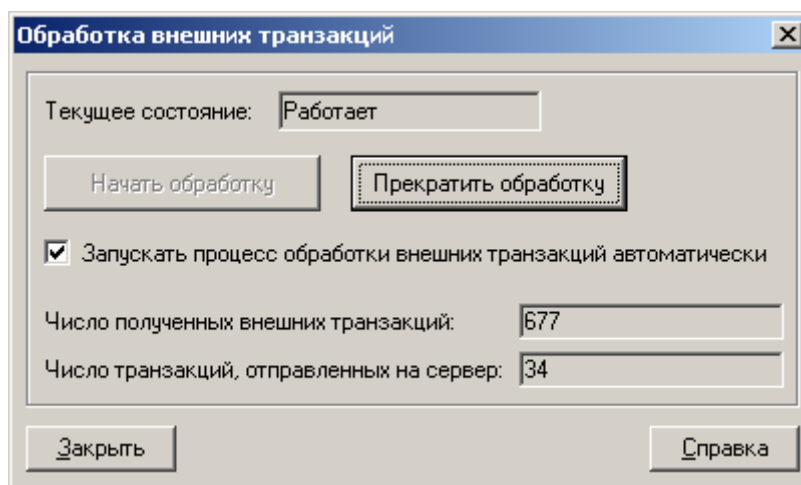
ВАЖНО! Проследите, чтобы в пути установки отсутствовали русские символы и пробелы. Это необходимо для корректной работы драйвера базы данных ODBC, т.к. замечены случаи сбоев, связанных с некорректными (с точки зрения драйвера) путями.

Шаг 2. Настройки соединения

Запустите TAOrderSrv.exe, выберите пункт меню "Настройки->Общие настройки". В открывшемся диалоговом окне укажите путь к QUIK.

Шаг 3. Настройка импорта транзакций QUIK

Для того, чтобы посылаемые программой заявки обрабатывались в QUIK необходимо включить импорт транзакций. Сделать это нужно в меню Quik **Торговля/Внешние транзакции**



Установите флажок "Запускать процесс...", чтобы не возиться с этим каждый раз при запуске QUIK и нажмите кнопку "Начать обработку". После этого окно можно закрывать, теперь при каждом старте QUIK будет включать обработку внешних транзакций автоматически.

Заявки и сделки экспортируются из QUIK через интерфейс TRANS2QUIK.dll.

Шаг 4. Настройка экспорта данных о инструментах и котировках

- Создайте таблицу текущих параметров вызвав пункт меню **Таблицы/Текущая Таблица**.
- Отредактируйте таблицу, оставив в ней поля: Код класса, Код бумаги, Цена последней сделки, Точность, Класс, Бумага сокр., Размер лота.
- Настройте экспорт таблицы через ODBC. В качестве имени источника данных выберите TraderAssistant. Таблица для экспорта – Quotes. Сопоставление полей: Код класса = ClassCode, Код бумаги = SecCode, Цена последней сделки = LastPrice, Точность = Precision, Класс = ClassName, Бумага сокр. = SecName, Размер лота = LotSize.
- Установите флажки "Чистить таблицу перед выводом" и "Вывод после создания".

ВАЖНО! Программа будет работать только с инструментами, указанными в этой таблице.

ВАЖНО! После завершения добавления инструментов необходимо перезапустить TAOrderSrvr, чтобы программа подписалась на обновление заявок и сделок. В противном случае новые заявки и сделки в TAOrderSrvr приходиться не будут.

Шаг 5. Настройка экспорта данных о денежных средствах на ФОРТС

- Создайте таблицу ограничений по клиентским счетам вызвав пункт меню **Торговля/Фьючерсы/Ограничения по клиентским счетам**.
- Отредактируйте таблицу, оставив в ней поля: Фирма, Торговый счет, Лимит по откр. поз., Тип лимита.
- Настройте экспорт таблицы через ODBC. В качестве имени источника данных выберите TraderAssistant. Таблица для экспорта – MoneyFORTS. Сопоставление полей: Фирма = Firm, Торговый счет = Account, Лимит по откр. поз. = Position, Тип лимита = Type.
- В настройках таблицы оставьте только флажок "Показывать лимиты по ден. средствам", остальные лимиты не нужны.
- Установите флажки "Чистить таблицу перед выводом" и "Вывод после создания".

Шаг 6. Настройка экспорта данных о позициях на ФОРТС

- Создайте таблицу ограничений по клиентским счетам вызвав пункт меню **Торговля/Фьючерсы/Позиции по клиентским счетам**.
- Отредактируйте таблицу, оставив в ней поля: Фирма, Торговый счет, Код инструмента, Текущая чистая позиция.
- Настройте экспорт таблицы через ODBC. В качестве имени источника данных выберите TraderAssistant. Таблица для экспорта – PositionsFORTS. Сопоставление полей: Фирма = Firm, Торговый счет = Account, Код инструмента = Seccode, Текущая чистая позиция = Position.
- Установите флажки "Чистить таблицу перед выводом" и "Вывод после создания".

Шаг 7. Настройка экспорта данных о денежных средствах на ММВБ

- Создайте таблицу ограничений по клиентским счетам вызвав пункт меню **Лимиты/Клиентский портфель**
- Отредактируйте таблицу, оставив в ней поля: Фирма, Код клиента, Сумма денежных остатков.
- Настройте экспорт таблицы через ODBC. В качестве имени источника данных выберите TraderAssistant. Таблица для экспорта – MoneyMICEX. Сопоставление полей: Фирма = Firm, Код клиента = ClientCode, Сумма денежных остатков = Position.
- Установите флажки "Чистить таблицу перед выводом" и "Вывод после создания".

Шаг 8. Настройка экспорта данных о позициях на ММВБ

- Создайте таблицу ограничений по клиентским счетам вызвав пункт меню **Лимиты/Лимиты по бумагам**
- Отредактируйте таблицу, оставив в ней поля: Фирма, Код бумаги, Код клиента, Счет депо, Текущий остаток.

- Настройте экспорт таблицы через ODBC. В качестве имени источника данных выберите TraderAssistant. Таблица для экспорта – PositionsMICEX. Сопоставление полей: Фирма = Firm, Код бумаги = SecCode, Код клиента = ClientCode, Текущий остаток = Position.
- В настройках таблицы установите флажок "Показывать нулевые лимиты".
- Установите флажки "Чистить таблицу перед выводом" и "Вывод после создания".

Шаг 9. Настройка взаимодействия с AmiBroker

Для использования возможностей автоматизированной торговли скопируйте AMI.dll в папку Amibroker\Plugins и перезапустите Amibroker.

Шаг 10. Запуск программы

1. Запустите QUIK. Установите соединение с сервером брокера.
2. Запустите TAOrderSrvr.exe, установите соединение с базой данных.
Если все настройки заданы верно и QUIK соединен с брокером, в файле журнала появится сообщение: " Соединение с QUIK установлено. Состояние QUIK: TRANS2QUIK_QUIK_CONNECTED ".

ВАЖНО! После перезапуска QUIK необходимо вручную переподключить к нему TAOrderSrvr.

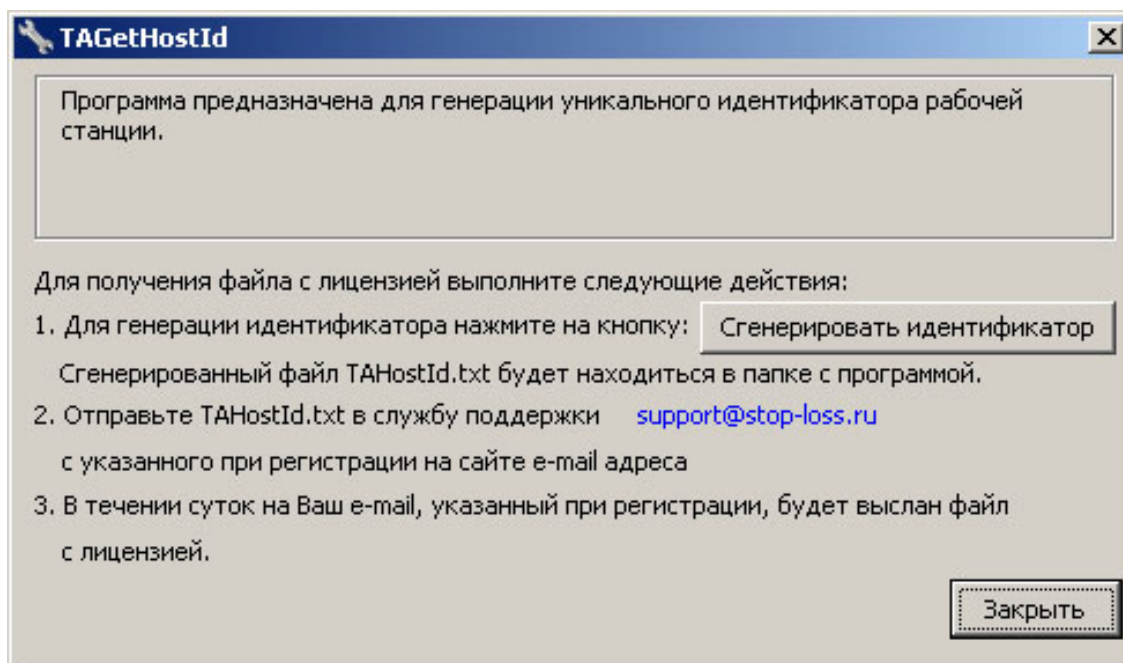
ВАЖНО! Настройте QUIK для автоматического переподключения с сервером брокера при потере соединения. Переподключения TAOrderSrvr для ситуации потери QUIK соединения с сервером брокера не требуется.

1.2 Получение лицензии на использование

Для получения лицензии на использование **TAWorkBook** необходимо сгенерировать идентификатор. Лицензия привязывается к рабочей станции. Таким образом программа может быть использована только на одной рабочей станции – той, на которой был сгенерирован идентификатор. Лицензия допускает двукратную замену рабочей станции, при такой замене бесплатно генерируется новая лицензия, а старая аннулируется. Лицензия не имеет ограничения по времени действия (предоставляется навсегда) и дает возможность на получение бесплатных обновлений программы.

Для получения лицензии следуйте перечисленным ниже шагам.

Шаг1: Генерация идентификатора рабочей станции. Для генерации идентификатора рабочей станции необходимо воспользоваться программой **TAGetHostId.exe**. Внешний вид программы представлен на рисунке.



Нажмите на кнопку "Сгенерировать идентификатор". Программа создаст файл **TAHostId.txt**

Шаг2: Отправка идентификатора в службу поддержки. Отправьте полученный файл **TAHostId.txt** на адрес службы поддержки: support@stop-loss.ru (**ВНИМАНИЕ! Файл необходимо отправлять с адреса электронной почты, указанного при регистрации на сайте www.stop-loss.ru**). В течении суток служба поддержки вышлет Вам лицензию на использование TAWorkbook.

Шаг3: Получение и установка лицензии. Полученную лицензию положите в директорию, в которой находится **TAOrderSrvr.exe**. При запуске программа автоматически обнаружит лицензию и можно будет начать работу.

- * Если Вы потеряете файл с лицензией – его всегда бесплатно восстановят в службе поддержки.
- * Если после изменения аппаратной конфигурации Вашего компьютера программа перестанет работать, ссылаясь на неверный файл лицензии – служба поддержки бесплатно и оперативно сгенерирует для вас новую лицензию.

2. Работа с программой TAOrderSrvr

Программа TAOrderSrvr является центральным звеном программного комплекса и связывает между собой все остальные компоненты:

- На основе котировок из QUIK рассчитывает в реальном времени портфели и лимиты.
- Предоставляет Ami.dll информацию о портфеле и лимитах.
- Получает от Ami.dll сигналы, сохраняет их в базе данных, отправляет заявки в QUIK и отслеживает их исполнение.
- Отправляет информацию о событиях в TAlerter.

- * Щелчок левой кнопкой мыши по иконке программы в панели задач позволяет спрятать окно программы, повторный щелчок снова покажет окно.
- * Все основные пункты меню для удобства продублированы кнопками на панели задач. Кнопки сгруппированы по функциональному назначению. Наведите мышью на кнопку и программа отобразит всплывающее окно с ее описанием, также в статусной строке внизу главного окна программы появится расширенное описание кнопки.

2.1 Режимы функционирования и общий алгоритм работы с TAOrderSrvr

Предусмотрено два режима работы с программой:

- Автоматизированная торговля – решения о сделках принимают автоматизированные торговые стратегии и через Ami.dll передают их в TAOrderSrvr, который достаточность лимитов и регистрирует заявку в QUIK. После исполнения заявки TAOrderSrvr автоматически регистрирует сделку и связывает ее с торговой стратегией. В автоматическом режиме также отслеживаются доходности сделок.

ВНИМАНИЕ! Для автоматизированной торговли предусмотрен тестовый режим, при котором заявки не уходят в QUIK а исполняются виртуально в программе по текущим котировкам.

- Торговля в ручном режиме – решения о сделках принимает трейдер и совершает сделки вручную в терминале QUIK. TAOrderSrvr в этом режиме работает как регистратор сделок, автоматически связывая сделки трейдера со стратегией "Ручной режим" и рассчитывая доходность.

Общий алгоритм работы с программой:

- Перед началом работы необходимо настроить параметры (меню **Настройки/Общие настройки**).
- Установить соединение с QUIK и базой данных (меню **Файл/Соединение**).
- Зарегистрировать торговые счета (меню **Настройки/Счета**)
- Проверить правильность экспорта данных из QUIK, проверить, что обновляется таблица котировок и в ней присутствуют все необходимые инструменты (меню **Торговля/Котировки**). Необходимо организовать экспорт из QUIK недостающих инструментов.

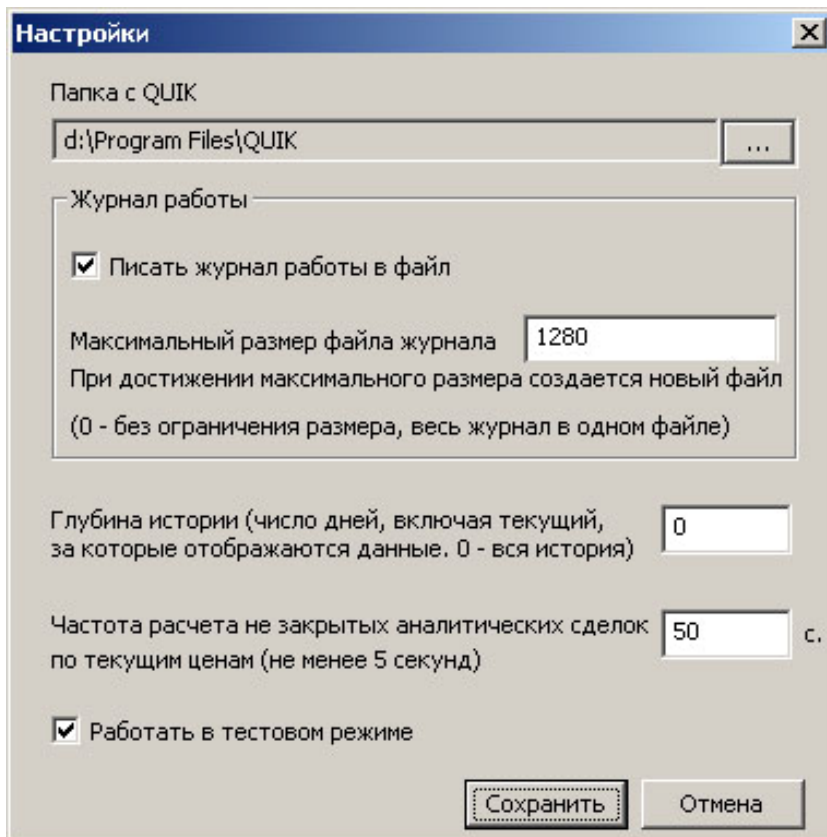
ВАЖНО! После добавления в экспорт новых документов необходимо перезапустить TAOrderSrvr.

ВНИМАНИЕ! После выполнения этого шага уже можно начинать работать в ручном режиме.

- Для автоматизированной торговли зарегистрировать торговые стратегии (меню **Настройки/Стратегии**)
- Для автоматизированной торговли зарегистрировать и запустить лимиты для торгуемых инструментов (меню **Настройки/Лимиты – автоматизированная торговля**).

2.2 Настройка программы

Перед началом работы необходимо осуществить настройку программы. Доступ к окну настроек осуществляется при помощи меню **Настройки/Общие настройки**.



Описание элементов окна

Поле ввода **Папка с QUIK** содержит путь к папке с QUIK. Для выбора пути необходимо нажать на кнопку "..." справа от поля ввода и в открывшемся диалоговом окне выбрать папку, в которую установлен QUIK.

Блок **Журнал работы** управляет параметрами ведения журнала. При установленном флажке **Писать журнал работы в файл** журнал работы ведется, при снятом – нет. Журнал работы сохраняется в текстовом файле в папке **Корневая папка\Logs**. Поле ввода **Максимальный размер журнала** содержит максимальное число строк в файле журнала. При достижении максимального количества строк программа начинает новый файл журнала, старый не удаляется.

* Для упрощения анализа рекомендуется вести журнал работы программы, размер журнала не стоит делать слишком большим (например, свыше 10000 строк) во избежании проблем с задержками при операциях чтение/запись файла на диске.

Поле ввода **Глубина истории** содержит число дней за которые загружаются данные в таблицы программы (0 – вся история).

* Ограничение глубины истории актуально при большом числе совершаемых сделок и предназначено для повышения скорости работы за счет оперирования меньшим набором данных.

Поле ввода **Частота расчета не закрытых аналитических сделок** содержит период пересчета аналитических сделок в секундах. Минимальное значение 5.

Флажок **Работать в тестовом режиме** – если установлен, то приходящие от автоматизированных торговых стратегий сигналы не передаются в QUIK а “исполняются” по переданной в заявке цене и регистрируются в виде сделок внутри программы. Режим предназначен для тестирования торговых идей.

2.3 Сохранение и загрузка конфигурации окон

TAOrderSrvr позволяет запоминать расположение дочерних окон и сохранять его для автоматической загрузки при следующих запусках программы.

Для сохранения конфигурации окон выберите пункт меню **Окна/Сохранить конфигурацию окон в файл**. Программа предложит выбрать имя файла для сохранения конфигурации и запишет в него список открытых окон, их расположение, размеры, параметры отображения таблиц и фильтры данных.

При последующих запусках программа после соединения с базой данных будет автоматически открывать последнюю сохраненную конфигурацию.

Для загрузки ранее сохраненной конфигурации окон выберите пункт меню **Окна/Загрузить конфигурацию окон из файл**. Программа предложит выбрать имя файла для загрузки конфигурации и загрузит выбранный файл.

2.4 Журнал работы программы

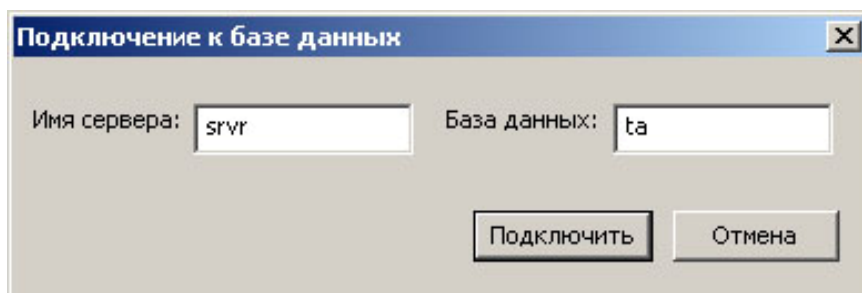
В журнале работы программы отображается вся значимая информация. Журнал работы записывается, если в настройках программы установлен флажок **Писать журнал работы в файл**.

Для показа/скрытия окна с журналом работы на панели инструментов предусмотрена кнопка



2.5 Установка соединения


Установка соединения с базой данных и QUIK осуществляется через пункт меню **Файл/Соединение**. В открывшемся диалоговом окне необходимо выбрать имя сервера и название базы данных.



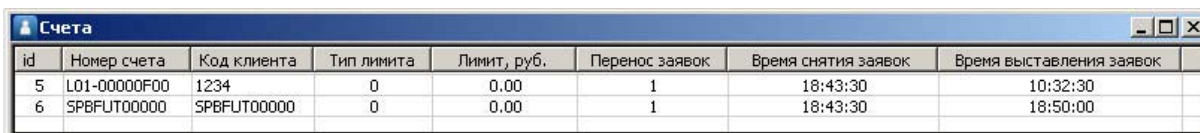
В поле ввода **Имя сервера** необходимо ввести имя Microsoft SQL server, установленного на вашей машине. При первом запуске программа сама определяет имя первого из установленных на машине серверов и подставляет его в качестве имени по умолчанию.

В поле ввода **База данных** необходимо ввести имя базы данных на выбранном сервере. По умолчанию программа устанавливается с базой данных ТА. При первом запуске программа регистрирует источник данных TraderAssistant, присоединяет к Microsoft SQL Server базу данных ТА и связывает их между собой. В дальнейшем базу данных можно заменить или даже сделать их несколько под разные цели.

2.6 Управление торговыми счетами

Программа поддерживает неограниченное множество торговых счетов. Для управления торговыми счетами перейдите в пункт меню **Настройки/Счета** или воспользуйтесь кнопкой  на панели инструментов. Программа откроет таблицу со списком счетов.

2.6.1 Таблица "Счета"



id	Номер счета	Код клиента	Тип лимита	Лимит, руб.	Перенос заявок	Время снятия заявок	Время выставления заявок
5	L01-00000F00	1234	0	0.00	1	18:43:30	10:32:30
6	SPBFUT00000	SPBFUT00000	0	0.00	1	18:43:30	18:50:00

В поле **Id** содержится идентификатор счета в TraderAssistant.

Поле **Тип лимита** может содержать значения 0 (для счетов без ограничения по размеру) или 1 (для искусственно ограниченных счетов). В поле **Лимит, руб** для счетов без ограничения по размеру содержится 0, для ограниченных счетов – искусственно установленный размер ограничения.

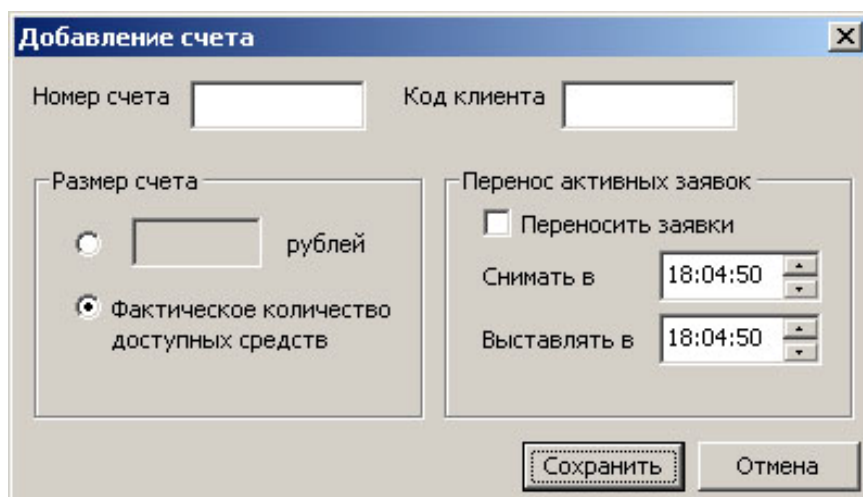
В поле **Перенос заявок** может находиться значение 1 (для заявок с переносом) или 0 (для заявок без переноса).

Для добавления нового счета щелкните правой кнопкой мыши по таблице и в появившемся выпадающем меню выберите пункт **Добавить**. Программа отобразит диалоговое окно добавления счета.

Для редактирования счета щелкните правой кнопкой мыши по строке счета в таблице и в появившемся выпадающем меню выберите пункт **Редактировать**. Программа отобразит диалоговое окно редактирования счета.

Для удаления счета щелкните правой кнопкой мыши по строке счета в таблице и в появившемся выпадающем меню выберите пункт **Удалить**. Программа запросит подтверждение на удаление счета и, в случае согласия трейдера, удалит его.

2.6.2 Добавление/редактирование счета



Описание элементов окна

Счет характеризуется непосредственно номером, который необходимо ввести в поле **Номер счета** и кодом клиента (поле ввода **Код клиента**). Для ММВБ данные о коде клиента необходимо взять из QUIK, для ФОРТС код клиента может быть любым.

Блок **Размер счета** устанавливает ограничение на размер денежных средств на счету, доступных автоматизированным стратегиям. Можно либо задать конкретный денежный лимит, либо разрешить автоматизированным стратегиям работать со всеми средствами на счету.

Блок **Перенос активных заявок** управляет функцией снятия/выставления заявок в заданное трейдером время. При установленном флажке **Переносить заявки** все активные заявки, имеющиеся в QUIK на момент времени, указанном в поле **Снимать в..** будут сняты и размещены в таблице **Отложенных** заявок программы. В момент времени, указанном в поле **Выставлять в..** отложенные заявки будут выставлены в QUIK.

2.7 Управление стратегиями


Для каждого автоматизированного торгового алгоритма в программе необходимо задать **Стратегию**.

Стратегия – набор правил, результатом обработки которых является сгенерированный и переданный в торговую систему сигнал. Правила описываются на языке программы **Amibroker**. Каждая стратегия представляет из себя законченный набор правил и размещается в отдельном файле .afl.

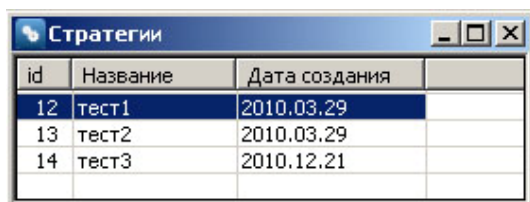
Программа позволяет вести список стратегий, автоматически присваивая каждой новой стратегии уникальный номер.

Программа хранит файлы стратегий в архиве (Корневая папка\Strategies), добавление файлов в архив осуществляется трейдером из интерфейса управления стратегиями. Для каждой стратегии внутри папки \Strategies программа автоматически создает собственную папку, название которой равно идентификатору стратегии.

ВНИМАНИЕ! Стратегиям автоматизированной торговли автоматически присваиваются положительные целочисленные идентификаторы. Для сделок, совершенных пользователем вручную идентификатор стратегии всегда равен "-1".

Для управления торговыми счетами перейдите в пункт меню **Настройки/Стратегии** или воспользуйтесь кнопкой  на панели инструментов. Программа откроет таблицу со списком стратегий.

2.7.1 Таблица "Стратегии"



id	Название	Дата создания
12	тест1	2010.03.29
13	тест2	2010.03.29
14	тест3	2010.12.21

В поле **Id** содержится идентификатор стратегии в TraderAssistant.

Поля **Название** и **Дата создания** содержат название стратегии и дату создания соответственно.

Для добавления новой стратегии щелкните правой кнопкой мыши по таблице и в появившемся выпадающем меню выберите пункт **Добавить**. Программа отобразит диалоговое окно добавления стратегии.

Для редактирования счета щелкните правой кнопкой мыши по строке в таблице и в появившемся выпадающем меню выберите пункт **Редактировать**. Программа отобразит диалоговое окно редактирования стратегии.

Для удаления стратегии щелкните правой кнопкой мыши по строке в таблице и в появившемся выпадающем меню выберите пункт **Удалить**. Программа запросит подтверждение на удаление и, в случае согласия трейдера, удалит стратегию.

2.7.2 Добавление/редактирование стратегии


Описание элементов окна

В поля ввода **Дата создания** и **Название** необходимо ввести дату создания и название стратегии соответственно.

Нажатие на кнопку **Открыть папку** приведет к открытию папки Корневой каталог\Strategies\[id], где id – идентификатор стратегии, присвоенный ей программой после создания.

2.8 Лимиты – автоматизированная торговля

Лимит представляет из себя ограничение, накладываемое на автоматизированную стратегию в целях реализации задач управления капиталом. Если для стратегии не задан или задан не активирован лимит – заявки выставляться не будут. Перед выставлением каждой заявки проверяется соответствие ее размера лимитам. Только в том случае, когда лимит позволяет пропустить заявку она попадает на биржу.

Для управления лимитами перейдите в пункт меню **Настройки/Лимиты – автоматизированная торговля** или воспользуйтесь кнопкой  на панели инструментов. Программа откроет таблицу со списком лимитов.

2.8.1 Таблица “Лимиты – автоматизированная торговля”

Id	Стратегия	Тек. позиция	Тек. позиция без учета заявок	Счет	Код клиента	Код класса	Код инструмента	Тип лимита по деньгам	Лимит по деньгам	Тип лимита по бумагам	Лимит по бумагам	Статус
19	тест1	1110	1120	L01-00000F00	1234	EQBR	SBER03	1	500000.00	2	0.00	1
22	тест3	0	0	SPBFUT00000	SPBFUT00000	SPBFUT	SRH1	3	0.00	2	0.00	1

Активные лимиты отображаются в таблице зеленым цветом, останавливаемые – синим, остановленные – красным.

Описание некоторых столбцов таблицы

В поле **Id** содержится идентификатор лимита в TraderAssistant.

Тип лимита по деньгам – Возможные значения: 1 – лимит задан в рублях, 2 – в % от доступных средств, 3 – лимит не задан.

Лимит по деньгам - Содержит максимально допустимый размер позиции в денежной оценке для стратегии, если лимит по деньгам задан. Если лимит по деньгам не задан – 0.-

Тип лимита по бумагам – Возможные значения: 1 – лимит задан, 2 – лимит не задан.

Лимит по бумагам – Содержит максимально допустимое число лотов для стратегии в портфеле, если лимит по бумагам задан. Если лимит по бумагам не задан – 0.

Статус – Состояние лимита. Возможные значения: 1 - лимит активен, 2 – отслеживаются текущие заявки и лимит деактивируется, 3 – лимит не активен.

Для просмотра информации об аналитических сделках, совершенных в рамках лимита, необходимо щелкнуть правой кнопкой мыши по строке с лимитом и в появившемся меню выбрать пункт **Аналитические сделки**. Программа откроет таблицу аналитических сделок по выбранному лимиту.

Для немедленного перерасчета аналитических сделок и текущей позиции по лимиту необходимо щелкнуть правой кнопкой мыши по строке с лимитом в таблице и в появившемся меню выбрать пункт **Перерасчитать аналитические сделки**.

Для немедленного запуска не работающего лимита необходимо щелкнуть правой кнопкой мыши по строке с лимитом в таблице и в появившемся меню выбрать пункт **Запустить**.

Для немедленной остановки работающего лимита необходимо щелкнуть правой кнопкой мыши по строке с лимитом в таблице и в появившемся меню выбрать пункт **Остановить**. Программа снимет все активные заявки и остановит лимит.

Для редактирования лимита щелкните правой кнопкой мыши по строке в таблице и в появившемся выпадающем меню выберите пункт **Редактировать**. Программа отобразит диалоговое окно редактирования лимита.

Для удаления лимита необходимо щелкнуть правой кнопкой мыши по строке с лимитом в таблице и в появившемся меню выбрать пункт **Удалить**.

2.8.2 Добавление/редактирование лимита

Настройка лимитов по стратегии

Остановлено

Включить Завершить сделки и отключить Отключить немедленно

Идентификатор лимита Идентификатор стратегии

Счет

Идентификатор счета Код площадки (класса)

Номер счета Код инструмента

Код клиента

Лимиты по деньгам

 рублей

 % от доступных средств

 Все доступные средства

Лимиты по бумагам

 штук

 Без ограничений

Использовать свободные денежные средства из других стратегий

id	Стратегия	Код класса	Код инструм...	+

Сохранить Отмена

Описание элементов окна

В верхней части окна располагается цветное поле состояния лимита, под ним – кнопки изменяющие состояние: **Включить**, **Завершить сделки и отключить** и **Отключить немедленно**. Изменение состояния лимита возможно для уже созданных лимитов. При отключении лимита блокируется возможность выставления по нему заявок.

Поле ввода **Идентификатор лимита** заполняется программой автоматически и содержит уникальный идентификатор лимита.

Поле ввода **Идентификатор стратегии** содержит идентификатор стратегии, с которой связан лимит. Для изменения значения необходимо нажать на кнопку "...", расположенную справа от поля и в открывшемся списке выбрать стратегию.

Блок **Счет** содержит информацию о торговом счете, с которым связан лимит. Для изменения значения необходимо нажать на кнопку "...", расположенную справа от поля ввода **Идентификатор счета** и в открывшемся списке выбрать счет.


Блок **Инструмент** содержит информацию о инструменте, с которым связан лимит. Для изменения значения необходимо нажать на кнопку "...", расположенную справа от поля ввода **Код площадки** и в открывшемся списке выбрать из списка инструмент. Список формируется на основе таблицы котировок.

Блоки **Лимит по бумагам** и **Лимит по деньгам** содержат ограничение на максимальный размер позиции, который может иметь стратегия в рамках лимита. Для проверки лимита по деньгам рассчитывается текущая оценка портфеля в деньгах и сравнивается с величиной,

указанной в лимите. Во время проверки лимиты по бумагам и лимиты по деньгам объединяются через логическое И.

2.9 Лимиты – торговля в ручном режиме

Лимит для торговли в ручном режиме – условное понятие, не подразумевающее решение задач аналогичных задачам, решаемым лимитами для автоматизированной торговли. При торговле в ручном режиме таблица лимитов отображает текущие позиции по инструментам и, фактически, используется как иллюстрация портфеля.

Для управления лимитами перейдите в пункт меню **Настройки/Лимиты – ручная торговля** или воспользуйтесь кнопкой  на панели инструментов. Программа откроет таблицу со списком лимитов.

2.9.1 Таблица “Лимиты – ручная торговля”

id	Счет	Код клиента	Код класса	Код инструмента	Класс	Инструмент	Тек. позиция	Тек. позиция без учета заявок
25	SPBFUT00000	SPBFUT00000	SPBFUT	SRH1	Фьючерсы FORTS	SRH1	0	0
24	SPBFUT00000	SPBFUT00000	SPBFUT	RIH1	Фьючерсы FORTS	RIH1	-20	-20
27	L01-00000F00	1234	EQNO	RU000A0JQ5V9	Б-Облигации ММББ	АВАНГАРД-3	150	150
28	L01-00000F00	1234	EQNL	FEES	Б-Акции ММББ	ФСК ЕЭС ао	8000	8000
34	L01-00000F00	1234	EQNL	TRNFP	Б-Акции ММББ	Трансф ап	20	20
35	L01-00000F00	1234	EQNL	ROSN	Б-Акции ММББ	Роснефть	490	490
26	L01-00000F00	1234	EQNE	RU000A0EQ3R3	Внесписочные акции ММББ	ПИФТрд-ДН	40	100
32	L01-00000F00	1234	EQNE	NTRI	Внесписочные акции ММББ	НутриФивХол	250	250
29	L01-00000F00	1234	EQNE	GAZP	Внесписочные акции ММББ	ГАЗПРОМ ао	2000	2000
30	L01-00000F00	1234	EQBR	HYDR	A1-Акции ММББ	РусГидро	2000	2000
31	L01-00000F00	1234	EQBR	LKOH	A1-Акции ММББ	ЛУКОЙЛ	50	50

Таблица содержит информацию о текущих позициях (с учетом и без учета активных заявок) по инструментам, зарегистрированным в таблице **Котировок**, сделки по которым совершались трейдером вручную в QUIK.

Таблица перерасчитывается с периодичностью, определяемой параметром **Частота расчета не закрытых аналитических сделок** в настройках программы.

Для просмотра информации об аналитических сделках, совершенных в ручном режиме по инструменту, необходимо щелкнуть правой кнопкой мыши по строке с инструментом и в появившемся меню выбрать пункт **Аналитические сделки**. Программа откроет таблицу аналитических сделок, совершенных в ручном режиме, по выбранному лимиту.


Для немедленного перерасчета аналитических сделок и текущей позиции по лимиту необходимо щелкнуть правой кнопкой мыши по строке с инструментом в таблице и в появившемся меню выбрать пункт **Перерасчитать аналитические сделки**.

Для немедленного перерасчета аналитических сделок и текущих позиций по ВСЕМ лимитам в таблице необходимо щелкнуть правой кнопкой мыши по таблице и в появившемся меню выбрать пункт **Перерасчитать ВСЕ “ручные” аналитические сделки**.

2.10 Сигналы

Ami.dll генерирует торговые сигналы и передает их в TAOrderSrvr на обработку. Таблица сигналов содержит полную информацию обо всех поступивших сигналах и пути их выполнения. При успешном выполнении транзакции строка сигнала в таблице окрашивается в зеленый цвет, при возникновении ошибок – в красный.

* Причину возникновения ошибки при отправке транзакции можно посмотреть в столбце **Статус**.

Для просмотра таблицы сигналов перейдите в пункт меню **Торговля/Сигналы** или воспользуйтесь кнопкой  на панели инструментов. Программа откроет таблицу со списком сигналов.

2.10.1 Таблица “Сигналы”

Описание некоторых столбцов таблицы

Id - содержит идентификатор сигнала в TraderAssistant.

Покупка/продажа – Содержит 1 для сигналов на покупку или -1 для сигналов на продажу.

Статус – Числовое поле содержит статус отправки транзакции в QUIK. В текстовом поле – расшифровка статуса отправки транзакции.

№ Заявки – В случае успешного завершения транзакции по выставлению заявки содержит номер заявки на бирже.

Тип – содержит тип сигнала. 1 – заявка, СОС заявка. 2- стоп-лосс. 3 – тейк-профит. 4 – стоп-тейк. 5 – снятие заявок по условию.

2.11 Заявки


Для просмотра таблицы заявок перейдите в пункт меню **Торговля/Заявки** или воспользуйтесь кнопкой  на панели инструментов. Программа откроет таблицу со списком заявок.

Таблица заявок заполняется автоматически из QUIK и имеет стандартный набор отображаемых параметров. Заявки на покупку отображаются зеленым цветом, на продажу – красным.

Для снятия заявки вручную щелкните правой кнопкой мыши по строке таблицы с заявкой и в появившемся меню выберите пункт **Снять**. Программа отправит запрос на снятие заявки в QUIK.

Для снятия **всех** заявок вручную щелкните правой кнопкой мыши по таблице и в появившемся меню выберите пункт **Снять все**. Программа отправит запрос на снятие всех заявок в QUIK.

2.12 Сделки


Для управления таблицей сделок перейдите в пункт меню **Торговля/Сделки** или воспользуйтесь кнопкой  на панели инструментов. Программа откроет таблицу со списком сделок.

Таблица сделок заполняется автоматически из QUIK и имеет стандартный набор отображаемых параметров. Сделки, совершенные для заявок по сигналам из автоматизированных стратегий автоматически связываются с соответствующими стратегиями. Сделки, совершенные трейдерами вручную автоматически привязываются к стратегии “Ручной режим” (идентификатор стратегии = -1). Также предусмотрена возможность ручной коррекции информации о сделках – любую сделку можно отредактировать, удалить или зарегистрировать в системе абсолютно новую сделку вручную.

Для добавления новой сделки щелкните правой кнопкой мыши по таблице и в появившемся выпадающем меню выберите пункт **Добавить**. Программа отобразит диалоговое окно добавления сделки.

Для редактирования сделки щелкните правой кнопкой мыши по строке счета в таблице и в появившемся выпадающем меню выберите пункт **Редактировать**. Программа отобразит диалоговое окно редактирования сделки.

Для удаления сделки щелкните правой кнопкой мыши по строке счета в таблице и в появившемся выпадающем меню выберите пункт **Удалить**.

2.12.1 Добавление/редактирование сделки

Информация о сделке

Дата и время сделки
27.02.2011 16:13:05

Ручной режим
 Автоматизированная торговля

Идентификатор стратегии

Счет

Идентификатор
Номер счета
Код клиента

Инструмент

Код площадки (класса)
Код инструмента
Цена
Количество лотов

Комиссия биржи
Комиссия брокера

Покупка Продажа

Сохранить Отмена

Описание элементов окна

Сделка может быть связана как с торговлей в ручном режиме, так и с одной из автоматизированных стратегий. Выбор стратегии происходит при помощи переключателя **Ручной режим/Автоматизированная торговля**. Для автоматизированной торговли необходимо задать стратегию, с которой связана сделка. Для этого необходимо нажать на кнопку "...", расположенную справа от поля ввода **Идентификатор стратегии** и в открывшемся списке выбрать стратегию.

Поле ввода **Идентификатор стратегии** содержит идентификатор стратегии, с которой связан лимит. Для изменения значения необходимо нажать на кнопку "...", расположенную справа от поля и в открывшемся списке выбрать стратегию.

Блок **Счет** содержит информацию о торговом счете, для которого регистрируется сделка. Для изменения значения необходимо нажать на кнопку "...", расположенную справа от поля ввода **Идентификатор** и в открывшемся списке выбрать счет.

Блок **Инструмент** содержит информацию о инструменте, с которым совершается сделка. Для изменения значения необходимо нажать на кнопку "...", расположенную справа от поля ввода **Код площадки** и в открывшемся списке выбрать из списка инструмент. Список формируется на основе таблицы котировок.


Поля ввода **Цена** и **Количество лотов** определяют цену сделки и число лотов инструмента в ней.

В поля ввода **Комиссия биржи** и **Комиссия брокера** вводятся соответствующие комиссии в рублях. Это необязательные поля.

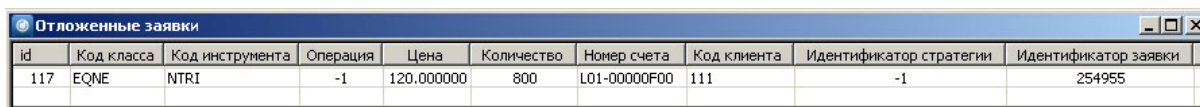
2.13 Отложенные заявки

Механизм отложенных заявок предназначен для переноса заявок между торговыми сессиями. Если в настройках счета установлен флажок **Переносить заявки**, то в заданное в настройках

время снятия активные заявки будут сниматься и попадать в отложенные. А в заданное время выставления – снова размещаться на бирже.

Для просмотра таблицы отложенных заявок перейдите в пункт меню **Торговля/Отложенные заявки** или воспользуйтесь кнопкой  на панели инструментов. Программа откроет таблицу со списком отложенных заявок.

2.13.1 Таблица “Отложенные заявки”



id	Код класса	Код инструмента	Операция	Цена	Количество	Номер счета	Код клиента	Идентификатор стратегии	Идентификатор заявки
117	EQNE	NTRI	-1	120.000000	800	L01-00000F00	111	-1	254955

Описание некоторых столбцов таблицы

Id - содержит идентификатор отложенной заявки в TraderAssistant.

Операция - содержит направление заявки. Возможны значения 1 (покупка) или -1 (продажа).

Идентификатор стратегии – стратегия, к которой привязана заявка. Для примера на рисунке – заявка выставлена в ручном режиме.

Идентификатор заявки – содержит номер заявки на бирже или 0, если заявка отсутствует. После успешной регистрации отложенной заявки на бирже это поле заполняется номером выставленной заявки.

Для немедленного выставления заявки щелкните правой кнопкой мыши по строке в таблице и в появившемся выпадающем меню выберите пункт **Выставить**.

Для немедленного выставления всех заявок щелкните правой кнопкой мыши по строке в таблице и в появившемся выпадающем меню выберите пункт **Выставить все**.


Для немедленного удаления отложенной заявки щелкните правой кнопкой мыши по строке в таблице и в появившемся выпадающем меню выберите пункт **Удалить**.

Для немедленного удаления всех отложенных заявок щелкните правой кнопкой мыши по строке в таблице и в появившемся выпадающем меню выберите пункт **Удалить все**.

2.14 Аналитические сделки

Аналитическая сделка представляет из себя совокупность простых сделок между нулевыми состояниями счета.

* Таким образом, например, операции покупки 10 контрактов с последующей их продажей составляют одну аналитическую сделку, в то время как простых биржевых сделок для выполнения этой операции может быть несколько, минимум 2.

Для просмотра таблицы аналитических сделок перейдите в пункт меню **Торговля/Аналитические сделки** или воспользуйтесь кнопкой  на панели инструментов. Программа откроет таблицу со списком аналитических сделок.

Для каждой аналитической сделки программа автоматически рассчитывает величину прибыли/убытка на основании цен открытия и закрытия аналитической сделки. Для еще не закрытых аналитических сделок прибыль/убыток рассчитываются на основе текущих биржевых котировок.

Таблица аналитических сделок рассчитывается только в автоматическом режиме. Перерасчет таблицы вызывается из всплывающего меню при работе с таблицей лимитов (см. соответствующий раздел документа).

Для просмотра сделок, входящих в состав конкретной аналитической сделки необходимо щелкнуть правой кнопкой мыши по аналитической сделке в таблице и в появившемся меню выбрать пункт **Сделки**.


ВНИМАНИЕ! Текущая версия программы в таблице биржевых сделок, входящих в состав аналитической сделки всегда показывает полный объем сделки независимо от того какая ее часть вошла в данную аналитическую сделку.

2.15 Котировки

Таблица котировок носит исключительно информационный характер и предназначена для визуального контроля над работоспособностью экспорта котировок из QUIK. Таблица котировок обновляется на экране с периодичностью, заданной в параметре **Частота расчета незакрытых аналитических сделок** в настройках программы.

* В базе данных котировки обновляются в реальном времени вне зависимости от заданных настроек обновления интерфейса, таким образом параметр **Частота расчета незакрытых аналитических сделок** влияет **ТОЛЬКО** на частоту обновления таблицы на экране программы и не влияет на получение котировок. Котировки всегда поступают в программу в режиме реального времени.

ВАЖНО! Программа может работать только с инструментами, указанными в таблице котировок. После добавления новых инструментов в таблицу необходимо пересоединиться с базой данных и QUIK. Необходимость обусловлена тем, что подписка на заявки и сделки осуществляется программой в момент осуществления соединения.

Для просмотра таблицы котировок перейдите в пункт меню **Торговля/Котировки** или воспользуйтесь кнопкой  на панели инструментов.

2.15.1 Таблица "Котировки"

Цена	Код инструмента	Код класса	Инструмент	Класс
9837.000000	SRH1	SPBFUT	SRH1	Фьючерсы FORTS
188815.000000	RIH1	SPBFUT	RIH1	Фьючерсы FORTS
1.458000	HYDR	EQBR	РусГидро	A1-Акции ММВБ
1984.030000	LKOH	EQBR	ЛУКОЙЛ	A1-Акции ММВБ
2.964000	MSNG	EQBR	+МосЭнерго	A1-Акции ММВБ
98.120000	SBER03	EQBR	Сбербанк	A1-Акции ММВБ
68.200000	SBERP03	EQBR	Сбербанк-п	A1-Акции ММВБ
205.690000	GAZP	EQNE	ГАЗПРОМ ао	Внесписочные акции ММВБ
65.910000	NTRI	EQNE	НутриньХол	Внесписочные акции ММВБ
0.000000	RU000A0EQ3R3	EQNE	ПИФТрд-ДН	Внесписочные акции ММВБ
0.392000	FEES	EQNL	ФСК ЕЭС ао	Б-Акции ММВБ
271.050000	ROSN	EQNL	Роснефть	Б-Акции ММВБ
38915.000000	TRNFP	EQNL	Трансф ап	Б-Акции ММВБ
0.000000	RU000A0JQSV9	EQNO	АВАНГАРД-3	Б-Облигации ММВБ

3. Работа с программой TAlerter

Программа **TAlerter** предназначена для информирования трейдера о существенных событиях функционирования программного комплекса. После запуска TAlerter соединяется с TAOOrderSrvr по протоколу tcp/ip (порт 2049) и получает информационные сообщения, данные о сделках, сигналах, ошибках. В зависимости от настроек для каждого типа сообщений TAlerter TraderAssistant. Руководство пользователя. стр. 23/39

предпринимает те или иные действия. TAlerter может размещаться на том же компьютере, что и TAOOrderSrvr или на любом другом.

СОВЕТ! В целях повышения надежности функционирования системы TAlerter рекомендуется устанавливать на отдельный компьютер. Таким образом при любом аппаратном или программном сбое, в результате которого выключится/перезагрузится компьютера с TAOOrderSrvr алертер сможет незамедлительно уведомлять об этом трейдера.

При получении сообщения трижды мигает иконка TAlerter на панели задач, программа издает звуковой сигнал, отображает всплывающее окошко с основными параметрами сообщения и, в зависимости от настроек оповещения, высылает/не высылает e-mail и/или sms с сообщением.

* Щелчок левой кнопкой мыши по иконке программы в панели задач позволяет спрятать окно программы, повторный щелчок снова покажет окно.

3.1. Настройка программы

Перед началом использования необходимо настроить параметры уведомления о поступающих сигналах. Для этого выберите пункт меню **Настройки/Настройки оповещения...** Программа отобразит диалоговое окно настроек.

The screenshot shows a dialog box titled "Настройки оповещения" (Notification Settings). It is organized into several panels:

- Настройки E-mail:** Includes input fields for "E-mail получателя:" (recipient), "E-mail отправителя:" (sender), "SMTP сервер:" (server), "Логин отправителя:" (login), and "Пароль отправителя:" (password). Below these is a warning: "ВАЖНО! Программа устанавливает соединение с почтовым сервером при каждой отправке письма. Для предотвращения зависания программы из-за задания неверных настроек всегда отправляйте тестовые e-mail при любых изменениях в этом блоке."
- Настройки SMS:** Includes input fields for "Телефон получателя:" (phone number, with "7xxxxxxxxx" as an example) and "COM-порт модема:" (COM port, with "COM1" as an example).
- Информация:** Contains checkboxes for "SMS" and "E-mail".
- Ошибки:** Contains checkboxes for "SMS" and "E-mail", both of which are checked.
- Сигналы:** Contains checkboxes for "SMS" and "E-mail", with "E-mail" checked.
- Сделки:** Contains checkboxes for "SMS" and "E-mail", with "SMS" checked.

At the bottom right, there are two buttons: "Сохранить" (Save) and "Отмена" (Cancel).

Окно разбито на несколько логических блоков. Левый верхний блок "Настройки E-mail" отвечает за настройку параметров отправки электронной почты, левый нижний – за настройку параметров отправки смс.

* Формат номера телефона для отправки смс: 7xxxxxxxxx, где x – цифра номера вашего телефона. Com-порт, на котором находится gsm-модем необходимо задавать в формате "COMx" (латиницей), где x – номер порта.

В правой части диалогового окна располагается ряд блоков с настройками поведения TAlerter при получении того или иного типа сообщений. Например, на картинке указаны настройки, при которых при приходе информационного сообщения сигналы не отправляются вообще, при приходе ошибки отправляются и смс и e-mail, при поступлении сигнала от торговой системы отправляется только e-mail, а при заключении сделки только смс.

3.2. Проверка настроек программы

Перед началом работы рекомендуется проверить правильность задания настроек оповещения. Для этого в меню выберите пункт **Настройки/Послать тестовую SMS** для отправки тестовой sms или **Настройки/Послать тестовый E-mail** для отправки тестового сообщения по электронной почте. Программа отобразит диалоговое окно, в котором нужно будет ввести текст тестового сообщения.

4. Работа с библиотекой AMI.dll

Ami.dll работает через программу TAOrderSrvr. Для работы с библиотекой необходимо разместить **ami.dll** в каталоге **Plugins** программы Amibroker и перезапустить Amibroker. После этого функции библиотеки автоматически появятся в списке доступных функций.

Ниже приведен перечень функций ami.dll:

4.1. AMI_IsConnected

```
AMI_IsConnected()
```

Возвращает текущий статус соединения.

4.2. AMI_GetCurrentFORTSPosition

```
AMI_GetCurrentFORTSPosition(sAccount, sSeccode);
```

Возвращает текущий размер портфеля на FORTS для заданных счета и кода инструмента.

Параметры

sAccount - счет.

sSeccode - код инструмента.

Пример использования

```
CurrentPosition = AMI_GetCurrentFORTSPosition("SPBFUT00111", "RIM0");
```

4.3. AMI_GetCurrentMICEXPosition

```
AMI_GetCurrentMICEXPosition(sAccount, sClientCode, sSeccode);
```

Возвращает текущий размер портфеля на ММВБ для заданных счета, кода клиента и кода инструмента.

Параметры

sAccount - счет.

sClientCode - код клиента.

sSeccode - код инструмента.

Пример использования

```
CurrentPosition = AMI_GetCurrentMICEXPosition("L01-00000F00", "111", "SBER03");
```

4.4. AMI_GetCurrentMICEXMoney

```
AMI_GetCurrentMICEXMoney(sAccount, sClientCode);
```

Возвращает текущий свободный денежный остаток на ММВБ для заданных счета и кода клиента.

Параметры

sAccount - счет.

sClientCode - код клиента.

Пример использования

```
CurrentPosition = AMI_GetCurrentMICEXMoney("L01-00000F00", "111");
```

4.5. AMI_GetCurrentFORTSMoney

```
AMI_GetCurrentFORTSMoney(sAccount);
```

Возвращает текущий свободный денежный остаток на ММВБ для заданного счета.

Параметры

sAccount - счет.

Пример использования

```
CurrentPosition = AMI_GetCurrentFORTSMoney("SPBFUT00111");
```

4.6. AMI_GetCurrentPosition

```
AMI_GetCurrentPosition(sAccount, sClientCode, sClasscode, sSeccode, iStrategyId);
```

Возвращает текущий размер позиции по лимиту (лимит определяется по инструменту и идентификатору стратегии). Размер позиции определяется как сумма текущей позиции и объема выставленных активных заявок по бумаге.

Параметры

sAccount - счет.

sClientCode - код клиента.

sClasscode - код класса.

sSeccode - код инструмента.

iStrategyId - идентификатор стратегии.

Пример использования

```
CurrentPosition = AMI_GetCurrentPosition(("SPBFUT00111", "SPBFUT00111", "SPBFUT", "SRM0", 12);
```

4.7. AMI_GetCurrentPositionNoOrder

```
AMI_GetCurrentPositionNoOrder(sAccount, sClientCode, sClasscode, sSeccode, iStrategyId);
```

Возвращает текущий размер позиции по лимиту (лимит определяется по инструменту и идентификатору стратегии). В отличие от функции AMI_GetCurrentPosition размер позиции НЕ ВКЛЮЧАЕТ выставленные заявки.

Параметры

sAccount - счет.
sClientCode - код клиента.
sClasscode - код класса.
sSeccode - код инструмента.
iStrategyId - идентификатор стратегии.

Пример использования

```
CurrentPosition = AMI_GetCurrentPositionNoOrder("SPBFUT00111", "SPBFUT00111",  
"SPBFUT", "SRM0", 12);
```

4.8. AMI_GetCurrentMoneyLimit

```
AMI_GetCurrentMoneyLimit(sAccount, sClientCode, sClasscode, sSeccode,  
iStrategyId);
```

Возвращает текущий лимит для стратегии в рублях.

Параметры

sAccount - счет.
sClientCode - код клиента.
sClasscode - код класса.
sSeccode - код инструмента.
iStrategyId - идентификатор стратегии.

Пример использования

```
Limit = AMI_GetCurrentMoneyLimit("SPBFUT00111", "SPBFUT00111", "SPBFUT",  
"SRM0", 12);
```

4.9. AMI_PushOrder

```
AMI_PushOrder(sAccount, sClientCode, sClasscode, sSeccode, iStrategyId,  
iPrecision, iDirection, fPrice, iSize, Date, Time);
```

Выставляет заявку.

Параметры

sAccount - счет.
sClientCode - код клиента.
sClasscode - код класса.
sSeccode - код инструмента.
iStrategyId - идентификатор стратегии.
iPrecision - точность цены инструмента.
iDirection - направление заявки (Для покупки iDirection>0, для продажи iDirection<0).

fPrice - цена заявки.

iSize - размер заявки.

Date - дата в числовом выражении (10000 * (year - 1900) + 100 * month + day).

Time - время в числовом выражении (10000 * hour + 100 * minute + second).

Пример использования

```
AMI_PushOrder("SPBFUT00111", "111", "SPBFUT", "RIM0", 12, 0, 1, 140000, 10, LastValue(DateNum()), LastValue(TimeNum()));
```

4.10. AMI_PushOCOOrder

```
AMI_PushOCOOrder(sAccount, sClientCode, sClasscode, sSeccode, iStrategyId, iPrecision, iDirection, fPrice, iSize, Date, Time, fStopLoss, fStopPrice, fTakeProfit);
```

Выставляет связанную заявку (One Cancel Other).

Параметры

sAccount - счет.

sClientCode - код клиента.

sClasscode - код класса.

sSeccode - код инструмента.

iStrategyId - идентификатор стратегии.

iPrecision - точность цены инструмента.

iDirection - направление заявки (Для покупки iDirection>0, для продажи iDirection<0).

fPrice - цена заявки.

iSize - размер заявки.

Date - дата в числовом выражении (10000 * (year - 1900) + 100 * month + day).

Time - время в числовом выражении (10000 * hour + 100 * minute + second).

fStopLoss - цена стоп-лосса.

fStopPrice - уровень срабатывания стоп-лосса.

fTakeProfit - цена тейк-профита.

Пример использования

```
AMI_PushOCOOrder("SPBFUT00111", "111", "SPBFUT", "SRM0", 12, 0, 1, 8500, 1, LastValue(DateNum()), LastValue(TimeNum()), 8400, 8450, 9000);
```

4.11. AMI_KillOrders

```
AMI_KillOrders(sAccount, sClientCode, sSeccode, iStrategyId, iDirection, Date, Time);
```

Снимает все активные заявки по инструменту на покупку или продажу (в зависимости от указанного направления).

Параметры

sAccount - счет.
sClientCode - код клиента.
sSeccode - код инструмента.
iStrategyId - идентификатор стратегии.
iDirection - направление (1 - покупка, -1 - продажа).
Date - дата в числовом выражении (10000 * (year - 1900) + 100 * month + day).
Time - время в числовом выражении (10000 * hour + 100 * minute + second).

Пример использования

```
AMI_KillOrders("SPBFUT00111", "SPBFUT00111", "SRM0", 12, 1, LastValue(DateNum()), LastValue(TimeNum()));
```

4.12. AMI_PushStopLoss

```
AMI_PushStopLoss(sAccount, sClientCode, sClasscode, sSeccode, sExpiryDate, iStrategyId, iPrecision, iDirection, fPrice, iSize, fStopPrice, Date, Time);
```

Выставляет стоп-лосс.

Параметры

sAccount - счет.
sClientCode - код клиента.
sClasscode - код класса.
sSeccode - код инструмента.
sExpiryDate - срок действия в формате ггггммдд или слово GTC.
iStrategyId - идентификатор стратегии.
iPrecision - точность цены инструмента.
iDirection - направление заявки (Для покупки iDirection>0, для продажи iDirection<0).
fPrice - цена.
iSize - размер.
fStopPrice - стопцена.
Date - дата в числовом выражении (10000 * (year - 1900) + 100 * month + day).
Time - время в числовом выражении (10000 * hour + 100 * minute + second).

Пример использования

```
AMI_PushStopLoss("SPBFUT00111", "111", "SPBFUT", "SRM0", "20100413", 12, 0, 1, 9000, 1, 8900, LastValue(DateNum()), LastValue(TimeNum()));
```

4.13. AMI_PushTakeProfit

```
AMI_PushTakeProfit(sAccount, sClientCode, sClasscode, sSeccode, sExpiryDate, iStrategyId, iPrecision, iDirection, fStopPrice, iSize, fOffset, iOffsetUnits, fSpread, iSpreadUnits, Date, Time);
```

Выставляет тейк-профит.

Параметры

sAccount - счет.
sClientCode - код клиента.
sClasscode - код класса.
sSeccode - код инструмента.
sExpiryDate - срок действия в формате ггггммдд или слово GTC.
iStrategyId - идентификатор стратегии.
iPrecision - точность цены инструмента.
iDirection - направление заявки (Для покупки iDirection>0, для продажи iDirection<0).
fStopPrice - стопцена.
iSize - размер.
fOffset - смещение.
iOffsetUnits - единицы, в которых задан параметр fOffset(1, если в процентах и 2, если в пунктах цены).
fSpread - защитный спрэд.
iSpreadUnits - единицы, в которых задан параметр fSpread(1, если в процентах и 2, если в пунктах цены).
Date - дата в числовом выражении (10000 * (year - 1900) + 100 * month + day).
Time - время в числовом выражении (10000 * hour + 100 * minute + second).

Пример использования

```
AMI_PushTakeProfit("SPBFUT00111","111","SPBFUT","SRM0","GTC",12,0,-1, 9000, 1, 5, 2, 0.1, 1,LastValue(DateNum()),LastValue(TimeNum()));
```

4.14. AMI_PushStopTake

```
AMI_PushTakeProfit(sAccount, sClientCode, sClasscode, sSeccode, sExpiryDate, iStrategyId, iPrecision, iDirection, fStopPrice, iSize, fOffset, iOffsetUnits, fSpread, iSpreadUnits, fPrice, fStopPrice2 Date, Time);
```

Выставляет тейк-профит со связанной стоп заявкой.

Параметры

sAccount - счет.
sClientCode - код клиента.
sClasscode - код класса.
sSeccode - код инструмента.
sExpiryDate - срок действия в формате ггггммдд или слово GTC.
iStrategyId - идентификатор стратегии.
iPrecision - точность цены инструмента.
iDirection - направление заявки (Для покупки iDirection>0, для продажи iDirection<0).
fStopPrice - стопцена.

iSize - размер.

fOffset - смещение.

iOffsetUnits - единицы, в которых задан параметр fOffset(1, если в процентах и 2, если в пунктах цены).

fSpread - защитный спрэд.

iSpreadUnits - единицы, в которых задан параметр fSpread(1, если в процентах и 2, если в пунктах цены).

Date - дата в числовом выражении (10000 * (year - 1900) + 100 * month + day).

Time - время в числовом выражении (10000 * hour + 100 * minute + second).

fPrice - цена для противоположной заявки.

fStopPrice2 - стопцена для противоположной заявки.

Пример использования

```
AMI_PushStopTake("SPBFUT00111", "111", "SPBFUT", "SRM0", "GTC", 12, 0, -1, 9000, 1, 5, 2, 0.1, 1, 7800, 7740, LastValue(DateNum()), LastValue(TimeNum()));
```

4.15. AMI_GetLastOrderStatus

```
AMI_GetLastOrderStatus(sAccount, sClientCode, sSeccode, iStrategyId, iDirection);
```

Возвращает статус последней выставленной из стратегии заявки.

Отрицательное значение - признак ошибки.

Значение -11 соответствует отсутствию заявки по текущей активной аналитической сделке.

Параметры

sAccount - счет.

sClientCode - код клиента.

sSeccode - код инструмента.

iStrategyId - идентификатор стратегии.

iDirection - направление (1 - покупка, -1 - продажа).

Пример использования

```
AMI_GetLastOrderStatus("SPBFUT00111", "SPBFUT00111", "SRM0", 12, 1);
```

4.16. AMI_GetLastOrderRemainder

```
AMI_GetLastOrderRemainder(sAccount, sClientCode, sSeccode, iStrategyId, iDirection);
```

Возвращает неисполненный остаток последней выставленной из стратегии заявки.

Отрицательное значение - признак ошибки.

Значение -1 соответствует отсутствию заявки.

Параметры

sAccount - счет.

sClientCode - код клиента.

sSeccode - код инструмента.
iStrategyId - идентификатор стратегии.
iDirection - направление (1 - покупка, -1 - продажа).

Пример использования

```
AMI_GetLastOrderRemainder("SPBFUT00111","SPBFUT00111","SRM0",12,1);
```

4.17. AMI_GetBuyDeals

```
AMI_GetBuyDeals(sAccount, sClientCode, sClasscode, sSeccode, iStrategyId);
```

Возвращает цены сделок на покупку для заданного инструмента и стратегии.

Параметры

sAccount - счет.
sClientCode - код клиента.
sClasscode - код площадки.
sSeccode - код инструмента.
iStrategyId - идентификатор стратегии.

Пример использования

```
BuyDeals = AMI_GetBuyDeals("acc","cl","class","secc",-1);  
SellDeals = AMI_GetSellDeals("acc","cl","class","secc",-1);
```

```
for(i = 0; i < BarCount; i++)  
{  
    if(BuyDeals[i] > 0)  
    {  
        Buy[i] = 1;  
        BuyPrice[i] = BuyDeals[i];  
    }  
    else  
    {  
        Buy[i] = 0;  
        BuyPrice[i] = 0;  
    }  
  
    if(SellDeals[i] > 0)  
    {  
        Sell[i] = 1;  
        SellPrice[i] = SellDeals[i];  
    }  
    else  
    {  
        Sell[i] = 0;  
        SellPrice[0] = 0;  
    }  
}  
PlotShapes(IIf(Buy, shapeUpArrow, 0), colorGreen, 0, BuyPrice, 0);  
PlotShapes(IIf(Sell, shapeDownArrow, 0), colorRed, 0, SellPrice, 0);
```

4.18. AMI_GetSellDeals

```
AMI_GetSellDeals(sAccount, sClientCode, sClasscode, sSeccode, iStrategyId);
```

Возвращает цены сделок на продажу для заданного инструмента и стратегии.

Параметры

sAccount - счет.

sClientCode - код клиента.
 sClasscode - код площадки.
 sSeccode - код инструмента.
 iStrategyId - идентификатор стратегии.

Пример использования

```
BuyDeals = AMI_GetBuyDeals("acc","cl","class","secc",-1);
SellDeals = AMI_GetSellDeals("acc","cl","class","secc",-1);

for(i = 0; i < BarCount; i++)
{
    if(BuyDeals[i] > 0)
    {
        Buy[i] = 1;
        BuyPrice[i] = BuyDeals[i];
    }
    else
    {
        Buy[i] = 0;
        BuyPrice[i] = 0;
    }

    if(SellDeals[i] > 0)
    {
        Sell[i] = 1;
        SellPrice[i] = SellDeals[i];
    }
    else
    {
        Sell[i] = 0;
        SellPrice[0] = 0;
    }
}
PlotShapes (IIf(Buy, shapeUpArrow, 0), colorGreen, 0, BuyPrice, 0);
PlotShapes (IIf(Sell, shapeDownArrow, 0), colorRed, 0, SellPrice, 0);
```

5. Таблицы базы данных

Таблица Signals

Сигналы торговых систем. Заполняется программой автоматически

Id_signal	Уникальный идентификатор сигнала, автоинкремент
Id_strategy	Уникальный идентификатор стратегии. Каждая стратегия знает свой идентификатор.
SignalTime	Дата/время появления сигнала
Account	Счет
Classcode	Код класса
Seccode	Код инструмента
BuySell	Направление – покупка или продажа. 1 – buy -1 - sell
Price	Цена
StopPrice	Стопцена

StopLoss	Стоплосс
TakeProfit	Тейкпрофит
Quantity	Количествов лотах
CreateTime	Время создания записи в базе данных
Status	0 – получен сигнал от торговой системы 1 – заявка отправлена в торговый терминал 2 – заявка исполнена 3 – заявка зарегистрирована
Precision	Число значащих знаков после запятой для инструмента
OrderNum	Номер заявки из QUIK
StatusText	Расшифровка поля Status
SignalType	1 – заявка, OCO заявка 2- стоп-лосс 3 – тейк-профит 4 – стоп-тейк 5 – снятие заявок по условию
ExpiryDate	Для стоп-лоссов и тейк-профитов содержит срок жизни в формате ггггммдд или слово GTC для его отсутствия
Offset	Смещение. Используется для тейк-профитов
OffsetUnits	Размерность смещения 1 – проценты 2 – единицы цены
Spread	Защитный спрэд. Используется для тейк-профитов
SpreadUnits	То же, что и OffsetUnits
StopPrice2	Используется как стопцена в сигналах типа стоп-тейк

Таблица Deals

Сделки. Заполняется программой автоматически.

DealNum	Номер сделки из QUIK
OrderNum	Номер заявки из QUIK
Classcode	Код класса
Seccode	Код инструмента
DealPrice	Цена
Quantity	Количествов лотах
Operation	Направление – покупка или продажа. 1 – buy -1 – sell

Account	Счет
ClientCode	Код клиента
DealDate	Дата сделки
DealTime	Время сделки
ExchangeComm	Комиссия биржи
TradeComm	Комиссия брокера
Id_strategy	Уникальный идентификатор стратегии. Каждая стратегия знает свой идентификатор.

Таблица MetaDeals

Аналитические сделки. Содержит информацию об аналитических сделках (аналитическая сделка представляет из себя совокупность простых сделок между нулевыми состояниями счета. Т.о. операции покупки 10 контрактов с последующей их продажей составляют одну аналитическую сделку, в то время как простых биржевых сделок для выполнения этой операции может быть несколько, минимум 2).

Id	Идентификатор метасделки
Id_limit	Идентификатор лимита
StartDate	Дата начала сделки
StartTime	Время начала сделки
FinishDate	Дата окончания сделки
FinishTime	Время окончания сделки
Operation	Направление – покупка или продажа. 1 – buy -1 – sell
Yield	Доходность, без учета комиссий

Таблица MetaDealDeals

Вспомогательная таблица, содержит связи между простыми сделками и аналитическими.

Id_Metadeal	Идентификатор метасделки
DealNum	Идентификатор простой сделки
Quantity	Число контрактов из простой сделки, вошедшее в данную метасделку

Таблица Orders

Заявки. Заполняется программой автоматически.

OrderNum	Номер заявки из QUIK
Classcode	Код класса
Seccode	Код инструмента
OrderPrice	Цена

Operation	Направление – покупка или продажа. 1 – buy -1 – sell
InitialAmount	Начальное количество
CurrentAmount	Текущее количество
Account	Счет
ClientCode	Код клиента
OrderDate	Дата заявки
OrderTime	Время заявки
Status	Статус из QUIK 1 – активна 2 – снята и т.д.
Id_strategy	Уникальный идентификатор стратегии. Каждая стратегия знает свой идентификатор.

Таблица DelayedOrders

Отложенные заявки. Заполняется программой автоматически.

Id	Внутренний идентификатор
Classcode	Код класса
Seccode	Код инструмента
OrderPrice	Цена
Operation	Направление – покупка или продажа. 1 – buy -1 – sell
Amount	Количество
Account	Счет
ClientCode	Код клиента
Id_strategy	Уникальный идентификатор стратегии. Каждая стратегия знает свой идентификатор.

Таблица PositionsFORTS

Позиции FORTS. Необходимо настроить экспорт данных из QUIK (Торговля->Фьючерсы->Позиции по клиентским счетам).

Account	Счет
Seccode	Код инструмента
Position	Текущая позиция

Таблица PositionsMICEX

Позиции ММВБ. Необходимо настроить экспорт данных из QUIK (Лимиты->Лимиты по бумагам).

Position	Текущая позиция
ClientCode	Код клиента (для ММВБ)

Таблица MoneyFORTS

Свободные деньги на FORTS. Необходимо настроить экспорт данных из QUIK (Торговля->Фьючерсы->Ограничения по клиентским счетам).

Account	Счет
Type	Программой не используется. Обязательное поле для экспорта в самом QUIK
Position	Текущая позиция

Таблица MoneyMICEX

Свободные деньги на ММВБ. Необходимо настроить экспорт данных из QUIK (Лимиты->Клиентский портфель).

Account	Счет
Seccode	Код инструмента
Position	Текущая позиция
ClientCode	Код клиента (для ММВБ)

Таблица Accounts

Счета

Id	Идентификатор счета
Account	Счет
ClientCode	Код клиента (для ММВБ)
LimitType	Тип лимита счета 1- установлена сумма в поле AccountLimit 0 – все доступные средства
AccountLimit	Величина лимита (для LimitType=1)

Таблица Strategy

Стратегии

Id	Идентификатор стратегии
Title	Название
Creation_date	Дата создания
IsMoveOrders	Признак необходимости переноса заявок 1 – перенос

	0 – без переноса
OrdersKillTime	Время снятия заявок при необходимости переноса (формат ЧЧММСС)
OrdersPushTime	Время выставления заявок при необходимости переноса (формат ЧЧММСС)

Таблица Limits

Лимиты по бумагам

Id	Идентификатор лимита
Id_strategy	Идентификатор стратегии
Id_account	Идентификатор счета
Classcode	Код класса
Seccode	Код инструмента
LimitType	Вид лимита 0 – лимит для автоматизированной торговли (создаются пользователями) 1 – лимит для ручной торговли (создаются автоматически программой)
MoneyLimitType	Тип лимита по деньгам 1 - лимит в рублях 2 – в процентах от доступных средств 3 – нет лимита
MoneyLimitValue	Величина лимита по деньгам
SecurityLimitType	Тип лимита по бумагам 1- лимит задан 2- нет лимита
SecurityLimitValue	Величина лимита по бумагам
Status	1 - используется 2 – отслеживает последнюю сделку и завершает работу 3 – не используется
CurrentPosition	Текущая позиция по инструменту с учетом активных и частично исполненных заявок
CurrentPositionNoOrder	Текущая позиция по инструменту без учета активных и частично исполненных заявок

Таблица UseOtherLimits

Содержит информацию по связанным лимитам

Id_limit	Идентификатор лимита
Id_limit_ref	Идентификатор связанного лимита

Таблица Quotes

Содержит текущие цены по инструментам. В эту таблицу необходимо экспортировать текущую таблицу параметров

Security	Программой не используется. Обязательное поле при экспорте в самом QUIK
Classcode	Код класса
Seccode	Код инструмента
LastPrice	Цена последней сделки
Precision	Точность цены (число знаков после запятой)