



Открытый интерфейс “SmartCOM”

Версия 4.0

Оглавление

1.	Изменения в версии 4.0.....	4
2.	Установка и удаление.....	5
3.	Взаимодействие с открытым интерфейсом.....	6
3.1.	Подключение к своему приложению.....	6
3.2.	Работа в торговой системе Matrix.....	7
3.3.	Дополнительная настройка.....	7
4.	Справочник функций.....	9
5.	События.....	11
5.1.	Событие AddBar.....	11
5.2.	Событие AddPortfolio.....	12
5.3.	Событие AddSymbol.....	12
5.4.	Событие AddTick.....	13
5.5.	Событие AddTickHistory.....	13
5.6.	Событие AddTrade.....	14
5.7.	Событие Connected.....	14
5.8.	Событие Disconnected.....	14
5.9.	Событие OrderCancelFailed.....	14
5.10.	Событие OrderCancelSucceeded.....	15
5.11.	Событие OrderMoveFailed.....	15
5.12.	Событие OrderMoveSucceeded.....	15
5.13.	Событие OrderFailed.....	16
5.14.	Событие OrderSucceeded.....	16
5.15.	Событие SetMyClosePos.....	16
5.16.	Событие SetMyOrder.....	17
5.17.	Событие SetMyTrade.....	18
5.18.	Событие SetPortfolio.....	18
5.19.	Событие UpdateBidAsk.....	19
5.20.	Событие UpdatePosition.....	19
5.21.	Событие UpdateOrder.....	20
5.22.	Событие UpdateQuote.....	21
5.23.	Событие OrderErrorDescription.....	22

Открытый интерфейс “SmartCOM”

6.	Методы.....	23
6.1.	Метод CancelBidAsk	23
6.2.	Метод CancelOrder	23
6.3.	Метод CancelPortfolio	23
6.4.	Метод CancelQuotes	24
6.5.	Метод CancelTicks	24
6.6.	Метод Connect	24
6.7.	Метод Disconnect.....	24
6.8.	Метод GetBars.....	25
6.9.	Метод GetMyPortfolioData	25
6.10.	Метод GetTrades	26
6.11.	Метод GetPrortfolioList.....	26
6.12.	Метод GetSymbols	26
6.13.	Метод IsConnected.....	26
6.14.	Метод ListenBidAsks.....	26
6.15.	Метод ListenPortfolio	27
6.16.	Метод ListenQuotes	27
6.17.	Метод ListenTicks	27
6.18.	Метод MoveOrder	27
6.19.	Метод PlaceOrder.....	28
7.	Пример использования.....	29
7.1.	Программа TestConnect	29

Открытый интерфейс “SmartCOM”

1. Изменения в версии 4.0

1.1. Новая версия клиентского интерфейса SmartCOM 4.0 работает во многопоточном окружении (Multithreaded Apartments), в связи с этим:

- Все клиентские события (AddPortfolio, AddTrade, и т.д.) вызываются из разных потоков.
- Для наилучшего быстродействия в SmartCOM 4.0 отсутствует дополнительная буферизация данных, благодаря этому все события вызываются напрямую, сразу же после получения данных. Необходимо избегать долгих блокировок при обработке этих событий, использовать при необходимости очереди.
- Все методы интерфейса SmartCOM 4.0 потоко-безопасны. Нет необходимости использовать дополнительную синхронизацию.
- Поддерживается возможность вызова методов интерфейса непосредственно из обработчиков событий.

Подробнее про многопоточное окружение COM можно найти здесь:

[http://msdn.microsoft.com/ru-ru/library/windows/desktop/ms693421\(v=vs.85\).aspx](http://msdn.microsoft.com/ru-ru/library/windows/desktop/ms693421(v=vs.85).aspx)

1.2. Поддержка платформы x64 (для этого необходимо скачать отдельную сборку SmartCOM 4.0 для x64).

1.3. Изменения интерфейса:

- В событии UpdateQuote добавлено поле:
step_price – стоимость шага цены.
- В событии SetPortfolio добавлены следующие поля:
liquidationValue – ликвидационная стоимость портфеля;
initialMargin – размер гарантийного обеспечения (ГО);
totalAssets – всего активов.
- В событии AddTrade, SetMyTrade добавлены следующие поля:
value – объем сделки в валюте расчетов;
accruedint – накопленный купонный доход.
- Добавлена новая функция, которая позволяет получить описание ошибки постановки приказа:
`string OrderErrorDescription(string OrderId).`
- Поддержка единой денежной позиции:
Портфели, входящие в ЕДП, скрыты. Пример: есть ЕДП портфель MO-01, в который входят позиционные счета RF-01 и MS-01. Все заявки и сделки по счетам RF-01 и MS-01 будут приходиться на портфель MO-01. Постановка заявок возможна только по MO-01, приказы будут автоматически маршрутизироваться на соответствующую площадку согласно инструменту и позиционному счету (RF-01 или MS-01). Логика для позиционных счетов, не входящих в ЕДП, остается без изменений.

Открытый интерфейс “SmartCOM”

- Методы GetMyOrders, GetMyTrades и GetMyClosePos объединены в один вызов GetMyPortfolioData. При этом данные по прежнему приходят в событиях SetMyTrade, SetMyOrder, SetMyClosePos.
- В событии UpdateOrder добавлен новый параметр: description – описание ошибки, в случае если приказ был отвергнут биржей/системой.

2. Установка и удаление.

Установка:

Для установки интерфейса SmartCOM 4.0, скачайте и запустите мастер установки программного обеспечения, расположенный по адресу:

<https://iticapital.ru/software/download/#smartcom>

Далее следуйте указаниям мастера.

Удаление:

Для удаления интерфейса SmartCOM 4.0 зайдите: Пуск → Панель управления → Установка и удаление программ.

Выберите из списка SmartCOM 4.0.***. Нажмите Удалить, и следуйте указаниям мастера удаления.

3. Взаимодействие с открытым интерфейсом

Открытый интерфейс «SmartCOM» позволяет:

- Подключаться к торговому серверу;
- Подавать заявки в торговую систему;
- Получать рыночную информацию.

3.1. Подключение к своему приложению

В настоящее время существует множество языков программирования, позволяющих использовать COM технологии. Ниже будет рассмотрен вариант подключения открытого интерфейса к внешним приложениям, который написан на C#.

Для подключения COM модуля к проекту Microsoft Visual Studio необходимо:

1. Нажать правой кнопкой мыши на References в Solution Explorer и выбрать пункт Add reference...
2. Переключиться на вкладку COM.
3. Выбрать «**SmartCom4(jelly) Type Library**» и нажать ОК.
4. Добавить код **using = SmartCOM4Lib;**

Создание объекта выглядит следующим образом:

```
[C#]  
StServerClass SmartCOM = new SmartCOM4Lib.StServerClass();
```

Открытый интерфейс “SmartCOM”

3.2. Работа в торговой системе MatriX

Для клиентов, подключенных к торговой системе MatriX, выставление приказов, а также получение событий о заявках, сделках и позициях, необходимо осуществлять на счете единой денежной позиции (суффикс MO). Логика для позиционных счетов, не входящих в ЕДП, остается без изменений.

Подробнее о торговой системе MatriX можно узнать на сайте по следующей ссылке:

<http://www.itinvest.ru/software/matrix/>

3.3. Дополнительная настройка

Методы `ConfigureClient` и `ConfigureServer` используются для конфигурации клиентской и серверной стороны перед началом работы.

Значение переменной `paramsSet` принимает вид строки в виде "param1=value1; param2=value2;...". У каждого из параметров есть значения по умолчанию, поэтому вызов данных методов является необязательным.

Все параметры вступают в силу только при вызове метода `Connect`.

```
[C#]  
void ConfigureClient(string paramsSet)
```

Доступные параметры:

Наименование	Описание	Значение по умолчанию
logFilePath	полный путь к клиентскому лог файлу	C:\Users\{user}\AppData\Local\IT Invest где {user} – имя пользователя в системе
logLevel	уровень логирования (1-6)	2
CalcPlannedPos	опция расчета планируемой позиции. Если флаг установлен в "yes", то автоматически будет рассчитываться поле <code>planned</code> в событии <code>UpdatePosition</code> . При этом событие <code>UpdatePosition</code> будет вызываться при каждом изменении поля <code>planned</code> . Так же требует загрузки состояния всех приказов при старте (что может быть накладно). Если же флаг установлен в "no", поле <code>planned</code> всегда будет равно 0.	yes

Открытый интерфейс “SmartCOM”

Наименование	Описание	Значение по умолчанию
asyncSocketConnectionMode	Включение/отключение асинхронного режима соединения при выполнении методов PlaceOrder/CancelOrder/MoveOrder. При использовании асинхронного режима нет необходимости ожидать, пока будет установлено соединение с сервером для выполнения данных команд.	yes

[C#]

```
void ConfigureServer (string paramsSet)
```

Доступные параметры:

Наименование	Описание	Значение по умолчанию
logFilePath	полный путь к клиентскому лог файлу	C:\Users\{user}\AppData\Local\IT Invest где {user} – имя пользователя в системе
logLevel	уровень логирования (1-6)	2

Открытый интерфейс “SmartCOM”

4. Справочник функций

Метод	Описание	Событие	Описание
Соединение			
IsConnected	Возвращает текущее состояние		
Connect	Подключение к торговому серверу брокера	Connected	Успешное подключение
Disconnect	Отключение от торгового сервера брокера	Disconnected	Успешное отключение от сервера
Работа с приказами			
CancelOrder	Снимает приказ по Id	OrderCancelFailed	Невозможно снять приказ
		OrderCancelSucceeded	Приказ будет снят
MoveOrder	Задаёт приказу новую цену, РТС	OrderMoveFailed	Невозможно изменить цену
		OrderMoveSucceeded	Приказ будет перемещён
PlaceOrder	Выставляет на биржу новый приказ	OrderFailed	Невозможно выставить приказ
		OrderSucceeded	Приказ выставлен
Работа с инструментами			
GetSymbols	Запросить справочник по инструментам	AddSymbol	Инструмент

Открытый интерфейс “SmartCOM”

Метод	Описание	Событие	Описание
ListenQuotes	Подписаться на получение котировок по инструменту	UpdateQuote	Текущая котировка
CancelQuotes	Отмена получения котировок по инструменту		
ListenTicks	Подписаться на получение всех сделок по инструменту	AddTick	Последняя сделка
CancelTicks	Отмена получения всех сделок по инструменту		
ListenBidAsks	Подписаться на получение стакана по инструменту	UpdateBidAsk	Строка стакана
CancelBidAsks	Отмена получения стакана по инструменту		
GetTrades	Запросить историю всех сделок по инструменту	AddTickHistory	
GetBars	Запросить историю по инструменту	AddBar	
Информация по счетам			
GetPrortfolioList	Запросить список доступных счетов	AddPortfolio	
ListenPortfolio	Подписаться на получение информации по счёту	SetPortfolio	Информация по денежным средствам
		AddTrade	Новая сделка
		UpdateOrder	Изменился приказ
		UpdatePosition	Изменилась позиция
CancelPortfolio	Отмена получения информации по счету		
GetMoneyAccount	Возвращает счет МО(ЕДП) для указанного позиционного счета	Нет события	
Исторические данные за сессию			
GetMyPortfolioData	Получить все сделки по счёту	SetMyTrade	Все сделки за текущую сессию
	Получить закрытые позиции	SetMyClosePos	Все закрытые позиции за текущую сессию
	Получить приказы	SetMyOrder	Все приказы за текущую сессию

Открытый интерфейс “SmartCOM”

5. События

Основное преимущество SmartCOM 4.0 заключается в том, что данные поступают в виде событий, достаточно вызвать соответствующий метод. Необходимо учитывать, что события приходят построчно.

5.1. Событие AddBar

Происходит при получении исторических данных по инструменту в интервале времени.

```
[C#]  
void AddBar(int row, int nRows, string symbol, StClientLib.StBarInterval interval, System.DateTime  
datetime, double open, double high, double low, double close, double volume, double open_int)
```

Описание параметров:

Наименование	Тип	Описание
Row	int	Номер бара в списке
NRows	int	Всего баров в списке
Symbol	String	Код ЦБ из таблицы котировок TC Matrix
Interval	StBarInterval	Интервал времени. StBarInterval_1Min = 1, StBarInterval_5Min = 2, StBarInterval_10Min = 3, StBarInterval_15Min = 4, StBarInterval_30Min = 5, StBarInterval_60Min = 6, StBarInterval_2Hour = 7, StBarInterval_4Hour = 8, StBarInterval_Day = 9, StBarInterval_Week = 10, StBarInterval_Month = 11, StBarInterval_Quarter = 12, StBarInterval_Year = 13.
Datetime	DateTime	Дата и время интервала
Open	Double	Цена первой сделки после открытия в интервале
High	Double	Максимальная цена сделки в интервале
Low	Double	Минимальная цена сделки в интервале
Close	Double	Цена последней сделки в интервале
Volume	Double	Объём сделок в интервале
Open_Int	Double	Открытый интерес

Открытый интерфейс “SmartCOM”

5.2. Событие AddPortfolio

Происходит при обновлении списка счетов, доступных текущему аккаунту.

```
[C#]  
void AddPorfolio(int row, int nrows, string portfolioName, string portfolioExch, StPortfolioStatus  
portfolioStatus)
```

Описание параметров:

Наименование	Тип	Описание
Row	int	Индекс портфеля
NRows	int	Всего портфелей
portfolioName	string	Наименование портфеля
portfolioExch	string	Площадка доступная для портфеля
portfolioStatus	StPortfolioStatus	Статус счёта. Принимает следующие значения: StPortfolioStatus_Broker = 0 StPortfolioStatus_TrustedManagement = 1 StPortfolioStatus_ReadOnly = 2 StPortfolioStatus_Blocked = 3 StPortfolioStatus_Restricted = 4 StPortfolioStatus_AutoRestricted = 5 StPortfolioStatus_OrderNotSigned = 6

5.3. Событие AddSymbol

Происходит при обновлении справочника.

```
[C#]  
void AddSymbol(int row, int nrows, string symbol, string short_name, string long_name, string type,  
int decimals, int lot_size, double punkt, double step, string sec_ext_id, string sec_exch_name,  
DateTime expiry_date, double days_before_expiry, double strike)
```

Описание параметров:

Наименование	Тип	Описание
Row	int	Индекс символа
NRows	int	Всего символов
Symbol	string	Код ЦБ из таблицы котировок TC Matrix
Short_name	string	Краткое наименование
Long_name	string	Полное наименование
Type	string	Код типа из справочника SmartTrade
Decimals	int	Точность цены
Lot_size	int	Размер лота
punkt	double	Стоимость минимального шага цены
step	double	Минимальный шаг цены
sec_ext_id	string	ISIN
sec_exch_name	string	Наименование площадки
expiryDate	DateTime	Дата экспирации
days_before_expiry	double	Дней до экспирации
strike	double	Цена исполнения

Открытый интерфейс “SmartCOM”

5.4. Событие AddTick

Происходит при возникновении сделки на рынке по инструменту.

```
[C#]  
void AddTick(string symbol, System.DateTime datetime, double price, double volume, string tradeno, StClientLib.StOrder_Action action)
```

Описание параметров:

Наименование	Тип	Описание
Symbol	string	Код ЦБ из таблицы котировок TC Matrix
DateTime	DateTime	Время сделки
Price	double	Цена сделки
Volume	double	Объём сделки
TradeNo	string	Id сделки на рынке
Action	StOrder_Action	Направление сделки (Buy/Sell)

5.5. Событие AddTickHistory

Происходит при запросе истории сделок по инструменту.

```
[C#]  
void AddTickHistory (int row, int nRows, string symbol, System.DateTime datetime, double price, double volume, string tradeno, StClientLib.StOrder_Action action)
```

Описание параметров:

Наименование	Тип	Описание
Row	int	Номер сделки в списке
NRows	int	Всего сделок в списке
Symbol	string	Код ЦБ из таблицы котировок TC Matrix
DateTime	DateTime	Время сделки
Price	double	Цена сделки
Volume	double	Объём сделки
TradeNo	string	Id сделки на рынке
Action	StOrder_Action	Направление сделки (Buy/Sell)

Открытый интерфейс “SmartCOM”

5.6. Событие AddTrade

По счёту прошла сделка.

Примечание: Для получения этого события необходимо наблюдать за торговым счётом, метод *ListenPortfolio*.

```
[C#]  
void AddTrade(string portfolio, string symbol, string orderNo, double price, double amount,  
System.DateTime datetime, string tradeno, double value, double accruedint)
```

Описание параметров:

Наименование	Тип	Описание
Portfolio	string	Номер торгового счёта на торговой площадке.
Symbol	string	Код ЦБ из таблицы котировок TC Matrix
OrderNo	string	Номер заявки в ТС Биржи
Price	double	Цена сделки
AMount	double	Объём сделки, если положительный покупка, отрицательный в случае продажи
Datetime	dateTime	Время сделки
TradeNo	string	Номер сделки в ТС биржи
Value	double	Объем сделки в валюте расчетов
Accruedint	double	Накопленный купонный доход

5.7. Событие Connected

Связь с сервером котировок установлена.

```
[C#]  
void Connected()
```

5.8. Событие Disconnected

Связь с сервером котировок разорвана.

```
[C#]  
void Disconnected(string reason)
```

Описание параметров:

Наименование	Тип	Описание
Reason	string	Причина

5.9. Событие OrderCancelFailed

При подаче запроса на отмену заявки, произошла ошибка.

Примечание: Для получения этого события необходимо наблюдать за торговым счётом, метод *ListenPortfolio*.

```
[C#]  
void OrderCancelFailed (string OrderId)
```

Открытый интерфейс “SmartCOM”

Описание параметров:

Наименование	Тип	Описание
OrderId	string	Номер заявки в TC Matrix

5.10. Событие OrderCancelSucceeded

Запрос на отмену заявки доставлен.

Примечание: Для получения этого события необходимо наблюдать за торговым счётом, метод *ListenPortfolio*.

```
[C#]  
void OrderCancelSucceeded (string OrderId)
```

Описание параметров:

Наименование	Тип	Описание
OrderId	string	Номер заявки в TC Matrix

5.11. Событие OrderMoveFailed

При запросе на перемещение заявки произошла ошибка.

Примечание: Для получения этого события необходимо наблюдать за торговым счётом, метод *ListenPortfolio*.

```
[C#]  
void OrderMoveFailed (string OrderId)
```

Описание параметров:

Наименование	Тип	Описание
OrderId	string	Номер заявки в TC Matrix

5.12. Событие OrderMoveSucceeded

Запрос на перемещении заявки отправлен.

Примечание: Для получения этого события необходимо наблюдать за торговым счётом, метод *ListenPortfolio*.

```
[C#]  
void OrderMoveSucceeded(string OrderId)
```

Описание параметров:

Наименование	Тип	Описание
OrderId	string	Номер заявки в TC Matrix

Открытый интерфейс “SmartCOM”

5.13. Событие OrderFailed

При выставлении заявки произошла ошибка.

Примечание: Для получения этого события необходимо наблюдать за торговым счётом, метод *ListenPortfolio*.

```
[C#]  
void OrderFailed(int cookie, string OrderId, string reason)
```

Описание параметров:

Наименование	Тип	Описание
Cookie	int	Ваш уникальный номер заявки из PlaceOrder
OrderId	string	Номер заявки в TC Matrix
Reason	string	Причина

5.14. Событие OrderSucceeded

Заявка доставлена на сервер котировок.

Примечание: Для получения этого события необходимо наблюдать за торговым счётом, метод *ListenPortfolio*.

```
[C#]  
void OrderSucceeded(int cookie, string OrderId)
```

Описание параметров:

Наименование	Тип	Описание
Cookie	int	Ваш уникальный номер заявки из PlaceOrder
OrderId	string	Номер заявки в TC Matrix

5.15. Событие SetMyClosePos

Закрытые позиции за сессию после вызова метода *GetMyPortfolioData*.

```
[C#]  
void SetMyClosePos(int row, int nrows, string portfolio, string symbol, double amount, double price_buy, double price_sell, System.DateTime postime, string order_open, string order_close)
```


Открытый интерфейс “SmartCOM”

Описание параметров:

Наименование	Тип	Описание
Row	int	Номер в списке
NRows	int	Всего в списке
Portfolio	string	Номер торгового счёта на торговой площадке
Symbol	string	Код ЦБ из таблицы котировок TC Matrix
OrderNo	string	Номер заявки в TC Matrix
AMount	double	Объём позиции
Price_Buy	double	Цена Покупки
Price_Sell	double	Цена Продажи
Postime	DateTime	Время
Order_Open	string	Id приказа открытия позиции
Order_Close	string	Id приказа закрытия позиции

5.16. Событие SetMyOrder

Состояние собственных заявок за сессию после вызова метода **GetMyPortfolioData**.

```
[C#]
void SetMyOrder(int row, int nrows, string portfolio, string symbol, StClientLib.StOrder_State state,
StClientLib.StOrder_Action action, StClientLib.StOrder_Type type, StClientLib.StOrder_Validity
validity, double price, double amount, double stop, double filled, System.DateTime datetime, string
orderid, string orderno, int Cookie)
```

Описание параметров:

Наименование	Тип	Описание
Row	int	Номер в списке
NRows	int	Всего в списке
Portfolio	string	Номер торгового счёта на торговой площадке
Symbol	string	Код ЦБ из таблицы котировок TC Matrix
State	StOrder_State	Состояние приказа.
Action	StOrder_Action	Вид торговой операции. Принимает следующие значения: StOrder_Action_Buy = 1 StOrder_Action_Sell = 2 StOrder_Action_Short = 3 StOrder_Action_Cover = 4
Type	StOrder_Type	Тип приказа. Принимает следующие значения: StOrder_Type_Market = 1, - Маркет StOrder_Type_Limit = 2, - Лимит StOrder_Type_Stop = 3, - Стоп StOrder_Type_StopLimit = 4, Стоп-лимит
Validity	StOrder_Validity	Срок действия приказа.
Price	double	Цена Лимит, для заявок типа Лимит и Стоп-Лимит
Amount	Double	Объём, ЦБ в приказе
Stop	Double	Цена СТОП для приказа типа Стоп и Стоп-Лимит

Открытый интерфейс “SmartCOM”

Наименование	Тип	Описание
Filled	Double	Объем, исполненный в приказе
Datetime	DateTime	Время последнего изменения приказа
OrderId	string	Номер заявки в TC Matrix
OrderNo	string	Номер заявки в TC биржи
Cookie	int	Ваш уникальный номер приказа из PlaceOrder

5.17. Событие SetMyTrade

Список собственных сделок за сессию после вызова метода *GetMyPortfolioData*.

```
[C#]  
void SetMyTrade(int row, int nRows, string portfolio, string symbol, System.DateTime datetime, double price, double volume, string tradeno, StClientLib.StOrder_Action buySell, string orderno, double value, double accruedint)
```

Описание параметров:

Наименование	Тип	Описание
Row	int	Номер в списке
NRows	int	Всего в списке
Portfolio	string	Номер торгового счёта на торговой площадке
Symbol	string	Код ЦБ из таблицы котировок TC Matrix
Datetime	dateTime	Время сделки
Price	double	Цена Лимит, для заявок типа Лимит и Стоп-Лимит)
Volume	double	Объем, ЦБ в сделке
TradeNo	string	Номер сделки в TC биржи
BuySell	stOrder_Action	Вид торговой операции. Принимает следующие значения: StOrder_Action_Buy = 1 StOrder_Action_Sell = 2 StOrder_Action_Short = 3 StOrder_Action_Cover = 4
OrderNo	string	Номер заявки в TC Matrix
Value	double	Объем сделки в валюте расчетов
Accruedint	double	Накопленный купонный доход

5.18. Событие SetPortfolio

Изменился торговый счёт.

Примечание: Для получения этого события необходимо наблюдать за торговым счётом, метод *ListenPortfolio*.

```
[C#]  
void SetPortfolio(string portfolio, double cash, double leverage, double comission, double saldo, double liquidationValue, double initialMargin, double totalAssets)
```

Открытый интерфейс “SmartCOM”

Описание параметров:

Наименование	Тип	Описание
Portfolio	string	Номер торгового счёта на торговой площадке
Cash	double	Сумма доступных наличных средств на счёте
Leverage	double	Плечо для маржинальной торговли
Commision	double	Сумма биржевой комиссии за торговый день
Saldo	double	Сальдо торгового дня
LiquidationValue	double	Ликвидационная стоимость портфеля
InitialMargin	double	Начальная маржа
TotalAssets	double	Всего активов (деньги + стоимость маржинальных активов + вариационная маржа)

5.19. Событие UpdateBidAsk

Изменилась очередь заявок по инструменту.

[C#]

```
void UpdateBidAsk(string symbol, int row, int nrows, double bid, double bidsizes, double ask, double asksizes)
```

Описание параметров:

Наименование	Тип	Описание
Symbol	string	Код ЦБ из таблицы котировок TC Matrix
Row	int	Порядковый номер строки в очереди заявок
NRows	int	Общее количество строк в очереди заявок
Bid	double	Цена на покупку
BidSize	double	Объем ценных бумаг по цене на покупку
Ask	double	Цена на продажу
AskSize	double	Объем ценных бумаг по цене на продажу

5.20. Событие UpdatePosition

Изменилась позиция по инструменту на торговом счёте.

Примечание: Для получения этого события необходимо наблюдать за портфелем, метод *ListenPortfolio*.

[C#]

```
void UpdatePosition(string portfolio, string symbol, double avprice, double amount, double planned)
```

Открытый интерфейс “SmartCOM”

Описание параметров:

Наименование	Тип	Описание
Portfolio	string	Номер торгового счёта на торговой площадке.
Symbol	string	Код ЦБ из таблицы котировок TC Matrix
AVPrice	Double	Средневзвешенная цена
Amount	Double	Объем, ЦБ в приказе
Planned	Double	Количество ЦБ с учетом выставленных заявок

5.21. Событие UpdateOrder

Изменилось состояние приказа.

Примечание: Для получения этого события необходимо наблюдать за торговым счётом, метод **ListenPortfolio**.

[C#]

```
void UpdateOrder(string portfolio, string symbol, StClientLib.StOrder_State state, StClientLib.StOrder_Action action, StClientLib.StOrder_Type type, StClientLib.StOrder_Validity validity, double price, double amount, double stop, double filled, System.DateTime datetime, string orderid, string ordeno, int status_mask, int Cookie, string description)
```

Описание параметров:

Наименование	Тип	Описание
Portfolio	string	Номер торгового счёта на торговой площадке.
Symbol	string	Код ЦБ из таблицы котировок TC Matrix
State	StOrder_State	Состояние приказа. Принимает следующие значения: StOrder_State_Pending=1, Зарегистрирован в ТС StOrder_State_Open=2, Выведен на рынок StOrder_State_Expired=3, Снят по окончании торгового дня StOrder_State_Cancel=4, Отменён StOrder_State_Filled=5, Исполнен StOrder_State_Partial=6, Частично исполнен StOrder_State_ContragentCancel=7, Отклонён биржей StOrder_State_SystemReject=8, Отклонён биржей StOrder_State_SystemCancel=9, Отклонён биржей
Action	StOrder_Action	Вид торговой операции. Принимает следующие значения: StOrder_Action_Buy = 1, - Купить stOrder_Action_Sell = 2, - Продать
Type	StOrder_Type	Тип приказа. Принимает следующие значения: StOrder_Type_Market = 1, - Приказ по рынку StOrder_Type_Limit = 2, - Лимитированный приказ StOrder_Type_Stop = 3, - Стоп приказ StOrder_Type_StopLimit = 4 – приказ Стоп-Лимит

Открытый интерфейс “SmartCOM”

Наименование	Тип	Описание
Validity	StOrder_Validity	Срок действия приказа. Принимает следующие значения: StOrder_Validity_Day = 1, - День StOrder_Validity_Gtc = 2 – GTC (до отмены, макс. 30 дней)
Price	double	Цена Лимит, для заявок типа Лимит и Стоп-Лимит
Amount	Double	Объем, ЦБ в приказе
Stop	Double	Цена СТОП для приказа типа Стоп и Стоп-Лимит
Filled	Double	Объем, исполненный в приказе
Datetime	DateTime	Время последнего изменения приказа
OrderId	string	Номер заявки в ТС Matrix
OrderNo	string	Номер заявки в ТС биржи
Cookie	int	Ваш номер приказа из PlaceOrder
Description	string	Описание ошибки

5.22. Событие UpdateQuote

Изменение текущей котировки на рынке по инструменту.

[C#]

```
void UpdateQuote(string symbol, System.DateTime datetime, double open, double high, double low, double close, double last, double volume, double size, double bid, double ask, double bidsizes, double asksizes, double open_int, double go_buy, double go_sell, double go_base, double go_base_backed, double high_limit, double low_limit, int trading_status, double volat, double theor_price, double step_price)
```

Описание параметров:

Наименование	Тип	Описание
Symbol	string	Код ЦБ из таблицы котировок ТС Matrix
Datetime	DateTime	Время последней сделки
Open	double	Цена первой сделки в текущей сессии
High	double	Максимальная цена сделки в текущей сессии
Low	double	Минимальная цена сделки в текущей сессии
Close	double	Цена последней сделки предыдущей торговой сессии
Last	double	Цена последней сделки
Volume	double	Объем по ЦБ за текущую торговую сессию
Size	double	Объем последней сделки
Bid	double	Цена спроса
Ask	double	Цена предложения
BidSize	double	Общий объем ЦБ с требованием на покупку
AskSize	double	Общий объем ЦБ с требованием на продажу
Open_Int	double	Совокупная стоимость открытых позиций
Go_Buy	double	Гарантийное обеспечение покупки (фьючерсы)
Go_Sell	double	Гарантийное обеспечение продажи (фьючерсы)
Go_Base	double	Базовое ГО (гарантийное обеспечение продажи опционов)
Go_Base_Backed	double	Гарантийное обеспечение по синтетическим позициям (опционы)

Открытый интерфейс “SmartCOM”

Наименование	Тип	Описание
High_Limit	double	Верхний лимит цены
Low_Limit	double	Нижний лимит цены
Trading_Status	int	Торговый статус инструмента: 0 - Торгуемый 1 - Не торгуемый
volat	double	Волатильность
theor_price	double	Теоретическая цена
step_price	double	Стоимость шага цены

5.23. Событие OrderErrorDescription

Получение описания ошибки.

[C#]

```
void OrderErrorDescription(string orderid, string description)
```

Описание параметров:

Наименование	Тип	Описание
OrderId	string	Номер приказа
Description	string	Описание ошибки

6. Методы

6.1. Метод CancelBidAsk

Отменяет получение очереди заявок по инструменту.

```
[C#]  
void CancelBidAsks(string symbol)
```

Описание параметров:

Наименование	Тип	Описание
Symbol	string	Код ЦБ из таблицы котировок TC Matrix

6.2. Метод CancelOrder

Отменяет приказ, выставленный на рынок методом *PlaceOrder*.

```
[C#]  
void CancelOrder(string portfolio, string symbol, string orderid)
```

Описание параметров:

Наименование	Тип	Описание
Portfolio	string	Номер торгового счёта на торговой площадке.
Symbol	string	Код ЦБ из таблицы котировок TC Matrix
OrderId	string	Id приказа на сервере котировок

6.3. Метод CancelPortfolio

Отменяет наблюдение за торговым счётом.

```
[C#]  
void CancelPortfolio(string portfolio)
```

Описание параметров:

Наименование	Тип	Описание
Portfolio	string	Номер торгового счёта на торговой площадке.

Открытый интерфейс “SmartCOM”

6.4. Метод CancelQuotes

Отменяет получение котировок по инструменту.

```
[C#]  
void CancelQuotes(string symbol)
```

Описание параметров:

Наименование	Тип	Описание
Symbol	string	Код ЦБ из таблицы котировок TC Matrix

6.5. Метод CancelTicks

Отменяет получение всех сделок на рынке по инструменту.

```
[C#]  
void CancelTicks(string symbol)
```

Описание параметров:

Наименование	Тип	Описание
Symbol	string	Код ЦБ из таблицы котировок TC Matrix

6.6. Метод Connect

Вызывает подключение к серверу котировок. Доступные адреса подключения к серверам указаны на сайте ITinvest: <http://www.itinvest.ru/software/trade-servers/>

```
[C#]  
void connect(string ip, ushort port, string login, string password)
```

Описание параметров:

Наименование	Тип	Описание
IP	string	IP адрес сервера котировок
Port	ushort	Порт сервера (8443 – для торговой системы MatriX)
Login	string	Логин для доступа к серверу
Password	string	Пароль для доступа

6.7. Метод Disconnect

Вызывает отключение от сервера котировок.

```
[C#]  
void disconnect()
```


Открытый интерфейс “SmartCOM”

6.8. Метод GetBars

Заказать исторические данные по инструменту в заданном интервале.

```
[C#]  
Int GetBars(string symbol, StClientLib.StBarInterval interval, System.DateTime since, int count)
```

Описание параметров:

Наименование	Тип	Описание
Symbol	string	Код ЦБ из таблицы котировок TC Matrix
Interval	StBarInterval	Интервал времени. StBarInterval_1Min = 1, StBarInterval_5Min = 2, StBarInterval_10Min = 3, StBarInterval_15Min = 4, StBarInterval_30Min = 5, StBarInterval_60Min = 6, StBarInterval_2Hour = 7, StBarInterval_4Hour = 8, StBarInterval_Day = 9, StBarInterval_Week = 10, StBarInterval_Month = 11, StBarInterval_Quarter = 12, StBarInterval_Year = 13
Since	DateTime	Дата начала запрашиваемого интервала
Count	int	Количество запрашиваемых интервалов. Если количество запрашиваемых интервалов положительно, сбор идет «назад» по времени в прошлое от указанной даты; если отрицательно – то «вперед»

6.9. Метод GetMyPortfolioData

Заказать историю закрытых позиций, заявок и сделок за сессию.

```
[C#]  
void GetMyPortfolioData (StClientLib.StOrder_Mode mode, string portfolio)
```

Описание параметров:

Наименование	Тип	Описание
Mode	stOrder_Mode	stOrder_Active = 1 – только активные; stOrder_All = 2 – все за сессию.
Portfolio	string	Номер торгового счета на торговой площадке

Открытый интерфейс “SmartCOM”

6.10. Метод GetTrades

Заказать тиковую историю сделок по инструменту.

```
[C#]  
void GetTrades (string symbol, System.DateTime from, int count)
```

Описание параметров:

Наименование	Тип	Описание
Symbol	string	Код ЦБ из таблицы котировок TC Matrix
From	DateTime	Время
Count	int	Количество запрашиваемых тиков.

6.11. Метод GetPrortfolioList

Заказать справочник доступных счетов.

```
[C#]  
void GetPrortfolioList ()
```

6.12. Метод GetSymbols

Заказать справочник ЦБ.

```
[C#]  
void GetSymbols()
```

6.13. Метод IsConnected

Возвращает текущее состояние соединения с сервером котировок.

```
[C#]  
bool IsConnected()
```

Возвращает **true** если соединение установлено, и **false** если связь отсутствует.

Примечание: метод реализован по принципу флага, то есть запоминает последнее состояние событий connected/disconnected. В связи с чем является недостоверным.

Смотри также методы **Connect**, **Disconnect**.

6.14. Метод ListenBidAsks

Заказать очередь заявок по инструменту.

```
[C#]  
void ListenBidAsks (string symbol)
```

Открытый интерфейс “SmartCOM”

Описание параметров:

Наименование	Тип	Описание
Symbol	string	Код ЦБ из таблицы котировок TC Matrix

6.15. Метод ListenPortfolio

Позволяет наблюдать за состоянием торгового счёта.

```
[C#]  
void ListenPortfolio (string portfolio)
```

Описание параметров:

Наименование	Тип	Описание
Portfolio	string	Номер торгового счёта на торговой площадке

6.16. Метод ListenQuotes

Заказать котировки по инструменту.

```
[C#]  
void ListenQuotes (string symbol)
```

Описание параметров:

Наименование	Тип	Описание
Symbol	string	Код ЦБ из таблицы котировок TC Matrix

6.17. Метод ListenTicks

Заказать все сделки на рынке по инструменту.

```
[C#]  
void ListenTicks(string symbol)
```

Описание параметров:

Наименование	Тип	Описание
Symbol	string	Код ЦБ из таблицы котировок TC Matrix

6.18. Метод MoveOrder

Изменить цену в приказе.

```
[C#]  
void MoveOrder(string portfolio, string symbol, string orderid, double targetprice)
```

Описание параметров:

Открытый интерфейс “SmartCOM”

Наименование	Тип	Описание
Portfolio	string	Номер торгового счёта на торговой площадке.
Symbol	string	Код ЦБ из таблицы котировок SmartTrade
OrderId	string	Номер заявки в TC Matrix
TargetPrice	double	Новая цена приказа

6.19. Метод PlaceOrder

Выставить приказ.

[C#]

```
void PlaceOrder(string portfolio, string symbol, StClientLib.StOrder_Action action,
StClientLib.StOrder_Type type, StClientLib.StOrder_Validity validity, double price, double amount,
double stop, int cookie)
```

Наименование	Тип	Описание
Portfolio	string	Номер торгового счёта на торговой площадке.
Symbol	string	Код ЦБ из таблицы котировок TC Matrix
Action	StOrder_Action	Вид торговой операции. Принимает следующие значения: StOrder_Action_Buy = 1, - Купить stOrder_Action_Sell = 2, - Продать stOrder_Action_Short = 3, - Открыть «короткую позицию» StOrder_Action_Cover = 4 – Закрыть «короткую» позицию
Type	StOrder_Type	Тип приказа. Принимает следующие значения: StOrder_Type_Market = 1, - Приказ по рынку StOrder_Type_Limit = 2, - Лимитированный приказ StOrder_Type_Stop = 3, - Стоп приказ StOrder_Type_StopLimit = 4, – приказ Стоп-Лимит
Validity	StOrder_Validity	Срок действия приказа. Принимает следующие значения: StOrder_Validity_Day = 1, - День StOrder_Validity_Gtc = 2, – GTC (до отмены, макс. 30 дней)
Price	double	Цена Лимит - для заявок типа Лимит и Стоп-Лимит 0 - для приказа: По рынку или Стоп
Amount	Double	Объем, ЦБ в приказе
Stop	Double	Цена СТОП для приказа типа Стоп и Стоп-Лимит 0 - для приказа: По рынку или Лимит
Cookie	int	Ваш уникальный номер приказа, используется для определения Id приказа через события OrderSucceeded/OrderFailed и UpdateOrders

Открытый интерфейс “SmartCOM”

7. Пример использования

7.1. Программа TestConnect

Данная программа тестирует функции SmartCOM. Исходный код программы можно скачать с официального сайта компании АО “ИК” Ай Ти Инвест” -

<https://iticapital.ru/software/download/#smartcom>

Как работает программа:

1. После компиляции и запуска, нажмите кнопку Create (для создания экземпляра SmartCOM и назначения обработчиков).
2. Заполните поля IP server (указав адрес сервера и порт через двоеточие), Login, Password и Symbol.
3. Нажмите кнопку Connect. После установления соединения:
 - 3.1. Начнётся обновление данных по инструменту, указанному в поле Symbol.
 - 3.2. Программа запросит историю сделок по инструменту (последние 100).
 - 3.3. Программа запросит список доступных портфелей.
 - 3.4. Программа запросит историю сделок и приказов по всем доступным портфелям за сессию.
 - 3.5. Начнется обновление данных по всем доступным портфелям.
4. Для отключения от сервера нажмите Disconnect.
5. Текущий статус подключения отображается в поле Status.
6. Для выставления лимитного приказа: выберите портфель, укажите цену в поле Price (подставить текущую, можно кликнув мышкой по слову ask/bid), выберите операцию (купить/продать), нажмите кнопку Buy/Sell соответственно. После успешной доставки приказа в поле Place Order Limit -> Id появится его номер.
7. Для отмены лимитного приказа: нажмите кнопку Cancel.
8. Для перемещения лимитного приказа: задав новую цену, нажмите кнопку Move.
9. Реализован способ запросить большое количество баров (пакетами по 500 штук). Укажите дату, с которой следует начать запрос, нажмите кнопку Get Bars. Данные поступают, начиная с текущего времени по указанное.

Подробно разобрать функции, можно в исходном коде программы.