

Открытый интерфейс «SmartCOM»

Версия от 14.10.2010

1. Оглавление

2. УСТАНОВКА И УДАЛЕНИЕ	2-3
3. ВЗАИМОДЕЙСТВИЕ С ОТКРЫТЫМ ИНТЕРФЕЙСОМ	3-3
3.1. Подключение к своему приложению	3-3
4. СПРАВОЧНИК ФУНКЦИЙ	4-4
5. СОБЫТИЯ	5-6
5.1. СОБЫТИЕ ADDBAR	5-6
5.2. СОБЫТИЕ ADDPORTFOLIO	5-7
5.3. СОБЫТИЕ ADDSYMBOL	5-7
5.4. СОБЫТИЕ ADDTICK	5-8
5.5. СОБЫТИЕ ADDTICKHISTORY	5-8
5.6. СОБЫТИЕ ADDTRADE	5-9
5.7. СОБЫТИЕ CONNECTED	5-9
5.8. СОБЫТИЕ DISCONNECTED	5-9
5.9. СОБЫТИЕ ORDERCANSELFALLED	5-10
5.10. СОБЫТИЕ ORDERCANSELSUCCEEDED	5-10
5.11. СОБЫТИЕ ORDERMOVEFAILED	5-10
5.12. СОБЫТИЕ ORDERMOVESUCCEEDED	5-11
5.13. СОБЫТИЕ ORDERFAILED	5-11
5.14. СОБЫТИЕ ORDERSUCCEEDED	5-11
5.1. СОБЫТИЕ SETMYCLOSEPOS	5-12
5.2. СОБЫТИЕ SETMYORDER	5-12
5.1. СОБЫТИЕ SETMYTRADES	5-13
5.2. СОБЫТИЕ SETPORTFOLIO	5-13
5.3. СОБЫТИЕ UPDATEBIDASK	5-14
5.4. СОБЫТИЕ UPDATEORDER	5-14
5.5. СОБЫТИЕ UPDATEPOSITION	5-15
5.6. СОБЫТИЕ UPDATEQUOTE	5-16
6. МЕТОДЫ	6-17
6.1. МЕТОД CANCELALLOORDERS	6-17
6.2. МЕТОД CANSELBIDASK	6-17
6.3. МЕТОД CANSELORDER	6-17
6.4. МЕТОД CANCELPORTFOLIO	6-18
6.5. МЕТОД CANCELQUOTES	6-18
6.6. МЕТОД CANCELTICKS	6-18
6.7. МЕТОД CONNECT	6-18
6.8. МЕТОД DISCONNECT	6-19
6.9. МЕТОД GETBARS	6-19
6.10. МЕТОД GETMYCLOSEPOS	6-20

6.11.	МЕТОД GETMYORDERS	6-20
6.1.	МЕТОД GETMYTRADE	6-20
6.1.	МЕТОД GETTRADES	6-20
6.2.	МЕТОД GETPRORTFOLIOLIST	6-21
6.3.	МЕТОД GETSYMBOLS	6-21
6.4.	МЕТОД ISCONNECTED	6-21
6.5.	МЕТОД LISTENBIDASKS.....	6-21
6.6.	МЕТОД LISTENPORTFOLIO	6-22
6.7.	МЕТОД LISTENQUOTES	6-22
6.8.	МЕТОД LISTENTICKS.....	6-22
6.9.	МЕТОД MOVEORDER	6-23
6.10.	МЕТОД PLACEORDER	6-23
7.	ПРИМЕР ИСПОЛЬЗОВАНИЯ	7-24

2. Установка и удаление.

Установка:

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

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

Далее следуйте указаниям мастера.

Удаление:

Для удаления интерфейса SmartCOM, зайдите: Пуск → Панель управления → Установка и удаление программ. Выберите из списка SmartCOM 2. Нажмите Удалить, и следуйте указаниям мастера удаления.

3. Взаимодействие с открытым интерфейсом

COM-модуль «SmartCOM» позволяет:

- Подавать заявки.
- Получать информацию о рынке.
- Подключаться/отключаться от сервера котировок ITinvest.

3.1. Подключение к своему приложению

В настоящее время, существует множество языков программирования позволяющих использовать COM технологии. Ниже будут рассмотрены варианты подключения открытого интерфейса к внешним приложениям, написанных на C#.

Для подключения COM модуля к проекту необходимо:

1. Нажать правой кнопкой мыши на References в Solution Explorer и выбрать пункт Add reference...
2. Переключиться на вкладку COM
3. Выбрать «**SmartTrade client library**» и нажать «Select»
4. После чего нажать ОК

Для использования в проекте необходимо подключить через **using** модуль **StClientLib**. Создание объекта выглядит следующим образом:

[C#]

```
StServer SmartCOM = new StClientLib.StServer();
```

4. Справочник функций

Метод	Описание	Событие	Описание
Соединение			
IsConnected	Возвращает текущее состояние		
Connect	Подключение к серверу IT Invest	Connected	При успешном подключении
Disconnect	Отключение от сервера	Disconnected	При отключении от сервера
Работа с приказами			
CancelAllOrders	Снимает все активные приказы		
CancelOrder	Снимает приказ по Id	OrderCancelFailed	Невозможно снять приказ
		OrderCancelSucceeded	Приказ будет снят
MoveOrder	Задаёт приказу новую цену, PTC	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	Получить справочник по инструментам	AddSymbol	Инструмент
ListenQuotes	Подписаться на получение котировок по инструменту	UpdateQuote	Текущая котировка
CancelQuotes	Отмена получения		
ListenTicks	Подписаться на получение всех сделок по инструменту	AddTick	Последняя сделка
CancelTicks	Отмена получения		
ListenBidAsks	Подписаться на получение стакана по инструменту	UpdateBidAsk	Строка стакана
CancelBidAsks	Отмена получения		
GetTrades	Запросить историю всех сделок по инструменту	AddTickHistory	
GetBars	Запросить историю по инструменту	AddBar	
Информация по счётам			
GetPrortfolioList	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		
Исторические данные за сессию			
GetMyTrades	Получить все сделки по счёту	SetMyTrade	
GetMyClosePos	Получить закрытые позиции	SetMyClosePos	
GetMyOrders	Получить приказы за сессию	SetMyOrder	
Для партнёров			
CheckSubscription	Проверить состояние подписки на сервис, для партнёров.	SetSubscriptionCheckReult	

5. События

Важное преимущество SmartCOM, данные поступают событиями, вам достаточно подписаться на их получение. Технология COM является не многопоточной, поэтому рекомендуется обработчики событий делать как можно более лёгкими. При отключении от сервера с причиной «Timeout detected. check your internet connectivity or event handler code», следует оптимизировать код обработчиков или изменить таймаут на их обработку. Также необходимо учитывать, что события приходят построчно.

Для изменения таймаута на обработку событий измените параметр в реестре windows

```
[HKEY_LOCAL_MACHINE\SOFTWARE\IT Invest\SmartCOM]
logLevel"=dword:000003e8"
```

5.1. Событие AddBar

Происходит при получении исторических данных по инструменту в интервале.

[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)
```

Наименование	Тип	Описание
Row	int	Номер бара в списке
NRows	int	Всего баров в списке
Symbol	String	Код ЦБ из таблицы котировок SmartTrade
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

5.2. Событие AddPortfolio

Происходит при обновлении списка счетов доступных по текущему аккаунту

[C#]

```
void AddPortfolio(int row, int nRows, string portfolioName, string portfolioExch)
```

Наименование	Тип	Описание
Row	int	Номер счета в списке
NRows	int	Всего счетов в списке
portfolioName	string	Наименование портфеля
portfolioExch	string	Площадка доступная для портфеля

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, System.DateTime expiryDate,
double days_before_expiry)
```

Наименование	Тип	Описание
Row	int	Номер инструмента в справочнике
NRows	int	Всего инструментов в справочнике
Symbol	string	Код ЦБ из таблицы котировок SmartTrade
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 ценных бумаг
punkt	double	Цена минимального шага
step	double	Минимальный шаг цены
sec_ext_id	string	ISN
sec_exch_name	string	Наименование площадки
expiryDate	DateTime	Дата экспирации
days_before_expiry	double	Дней до экспирации

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)
```

Наименование	Тип	Описание
Symbol	string	Код ЦБ из таблицы котировок SmartTrade
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 nrow, string symbol, System.DateTime datetime, double price, double volume,
string tradeno, StClientLib.StOrder Action action)
```

Наименование	Тип	Описание
Row	int	Номер сделки в списке
NRows	int	Всего сделок в списке
Symbol	string	Код ЦБ из таблицы котировок SmartTrade
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)

5.6. Событие AddTrade

По счёту произошла 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)
```

Наименование	Тип	Описание
Portfolio	string	Номер торгового счёта на торговой площадке.
Symbol	string	Код ЦБ из таблицы котировок SmartTrade
OrderNo	string	Номер заявки на сервере котировок
Price	double	Цена сделки
AMount	double	Объём сделки, если положительный покупка, отрицательный в случае продажи
Datetime	DateTime	Время сделки
TradeNo	string	Id сделки на рынке

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)
```

Наименование	Тип	Описание
Reason	string	Причина

5.9. Событие OrderCancelFailed

При подачи запроса на отмену заявки, произошла ошибка.

Примечание: Для получения этого события необходимо наблюдать за торговым счётом, метод [ListenPortfolio](#).

[C#]

```
void OrderCancelFailed(string OrderId)
```

Наименование	Тип	Описание
OrderId	string	Id заявки на сервере котировок

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)
```

Наименование	Тип	Описание
OrderId	string	Id заявки на сервере котировок

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(int cookie, string OrderId, string reason)
```

Наименование	Тип	Описание
OrderId	string	Id заявки на сервере котировок

5.12. Событие OrderMoveSucceeded

Запрос на перемещении заявки отправлен.

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

[C#]

```
void OrderSucceeded(int cookie, string OrderId)
```

Наименование	Тип	Описание
OrderId	string	Id заявки на сервере котировок

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)
```

Наименование	Тип	Описание
Cookie	int	Ваш уникальный номер заявки из PlaceOrder
OrderId	string	Id заявки на сервере котировок
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)
```

Наименование	Тип	Описание
Cookie	int	Ваш уникальный номер заявки из PlaceOrder
OrderId	string	Id заявки на сервере котировок

5.1. Событие SetMyClosePos

Закрытые позиции за сессию, по запросу *GetMyClosePos*

[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)
```

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 SmartTrade
OrderNo	string	Номер заявки на сервере котировок
AMount	double	Объём позиции
Price_Buy	double	Цена открытия
Price_Sell	double	Цена закрытия
Postime	DateTime	Время
Order_Open	string	Id приказа открытия позиции
Order_Close	string	Id приказа закрытия позиции

5.2. Событие SetMyOrder

Состояние заявок за сессию, по запросу *GetMyOrder*

[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)
```

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 SmartTrade
State	StOrder_State	Состояние приказа.
Action	StOrder_Action	Вид торговой операции.
Type	StOrder_Type	Тип приказа.
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
Filled	Double	Объём, оставшийся в приказе
Datetime	DateTime	Время последнего изменения приказа
OrderId	string	Id приказа на сервере котировок
OrderNo	string	Номер приказа на сервере котировок

5.1. Событие SetMyTrades

Состояние заявок за сессию, по запросу *GetMyTrade*

[C#]

```
void SetMyTrades(int row, int nrows, string portfolio, string symbol, System.DateTime datetime, double price, double volume, string tradeno, StClientLib.StOrder Action buysell, string orderno)
```

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 SmartTrade
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 на площадке
BuySell	StOrder_Action	Вид торговой операции.
OrderNo	string	Номер приказа на сервере котировок

5.2. 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)
```

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

5.3. Событие UpdateBidAsk

Изменилась очередь заявок по инструменту.

[C#]

```
void UpdateBidAsk(string symbol, int row, int nrow, double bid, double bsize, double ask, double asize)
```

Наименование	Тип	Описание
Symbol	string	Код ЦБ из таблицы котировок SmartTrade
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

5.4. 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 orderno, int status_mask)
```

Наименование	Тип	Описание
Portfolio	string	Номер торгового счёта на торговой площадке.
Symbol	string	Код ЦБ из таблицы котировок SmartTrade
State	StOrder_State	Состояние приказа. Принимает следующие значения: StOrder_State_ContragentReject=-1 StOrder_State_Submitted=0, Принят ТС 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 stOrder_Action_Short = 3, - Открыть «короткую позицию» StOrder_Action_Cover = 4 – Закрывать «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)
Amount	Double	Объем, ЦБ в приказе
Stop	Double	Цена СТОП для приказа типа Стоп и Стоп-Лимит
Filled	Double	Объем, оставшийся в приказе
Datetime	DateTime	Время последнего изменения приказа
OrderId	string	Id приказа на сервере котировок
OrderNo	string	Номер приказа на сервере котировок
Status_Mask	int	

5.5. 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)
```

Наименование	Тип	Описание
Portfolio	string	Номер торгового счёта на торговой площадке.
Symbol	string	Код ЦБ из таблицы котировок SmartTrade
AVPrice	double	Средневзвешенная цена
Amount	double	Объём сделки, если положительный Long, отрицательный в случае Short
Planned	double	Количество ЦБ с учетом выставленных заявок

5.6. 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 bidsize, double asksize, 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)
```

Наименование	Тип	Описание
Symbol	string	Код ЦБ из таблицы котировок SmartTrade
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)
High_Limit	double	Верхний лимит цены
Low_Limit	double	Нижний лимит цены
Trading_Status	int	Недоступно

6. Методы

6.1. Метод `CancelAllOrders`

Отменяет все выставленные приказы на всех площадках.

[C#]

```
void CancelAllOrders()
```

6.2. Метод `CancelBidAsk`

Отменяет получение очереди заявок по инструменту.

[C#]

```
void CancelBidAsks(string symbol)
```

Наименование	Тип	Описание
Symbol	string	Код ЦБ из таблицы котировок SmartTrade

6.3. Метод `CancelOrder`

Отменяет приказ выставленный на рынок методом *PalceOrder*.

[C#]

```
void CancelOrder(string portfolio, string symbol, string orderid)
```

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

6.4. Метод CancelPortfolio

Отменяет наблюдение за торговым счётом.

[C#]

```
void CancelPortfolio(string portfolio)
```

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

6.5. Метод CancelQuotes

Отменяет получение котировок по инструменту.

[C#]

```
void CancelQuotes(string symbol)
```

Наименование	Тип	Описание
Symbol	string	Код ЦБ из таблицы котировок SmartTrade

6.6. Метод CancelTicks

Отменяет получение всех сделок на рынке по инструменту.

[C#]

```
void CancelTicks(string symbol)
```

Наименование	Тип	Описание
Symbol	string	Код ЦБ из таблицы котировок SmartTrade

6.7. Метод Connect

Вызывает подключение к серверу котировок.

[C#]

```
void connect(string ip, short port, string login, string password)
```

Наименование	Тип	Описание
IP	string	IP адрес сервера котировок
Port	short	Порт сервера, 8090
Login	string	Логин для доступа к серверу
Password	string	Пароль для доступа

6.8. Метод Disconnect

Вызывает отключение от сервера котировок.

[C#]

```
void disconnect()
```

6.9. Метод GetBars

Запрашивает исторические данные по инструменту в интервале.

[C#]

```
int GetBars(string symbol, StClientLib.StBarInterval interval, System.DateTime since, int count)
```

Наименование	Тип	Описание
Symbol	string	Код ЦБ из таблицы котировок SmartTrade
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»

6.10. Метод GetMyClosePos

Запрашивает историю закрытых позиций за сессию.

[C#]

```
void GetMyClosePos(string portfolio)
```

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

6.11. Метод GetMyOrders

Запрашивает историю заявок за сессию.

[C#]

```
void GetMyOrders(int onlyActive, string portfolio)
```

Наименование	Тип	Описание
onlyActive	int	0 – все за сессию, 1 – только активные
Portfolio	String	Номер торгового счёта на торговой площадке.

6.1. Метод GetMyTrade

Запрашивает историю сделок за сессию.

[C#]

```
void GetMyTrades(string portfolio)
```

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

6.1. Метод GetTrades

Запрашивает тиковую историю сделок по инструменту.

[C#]

```
void GetTrades(string symbol, System.DateTime from, int count)
```

Наименование	Тип	Описание
Symbol	string	Код ЦБ из таблицы котировок SmartTrade
From	DateTime	Время
Count	int	Количество запрашиваемых сделок.

6.2. Метод GetPortfolioList

Запрашивает получение справочника счетов доступных текущему логину

[C#]

```
void GetPortfolioList()
```

6.3. Метод GetSymbols

Запрашивает получение справочника ЦБ

[C#]

```
void GetSymbols()
```

6.4. 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 являете не достоверным.

Смотри также методы [Connect](#), [Disconnect](#).

6.5. Метод ListenBidAsks

Позволяет получать очередь заявок по инструменту.

[C#]

```
void ListenBidAsks(string symbol)
```

Наименование	Тип	Описание
Symbol	string	Код ЦБ из таблицы котировок SmartTrade

6.6. Метод ListenPortfolio

Позволяет наблюдать за состоянием торгового счёта.

[C#]

```
void ListenPortfolio(string portfolio)
```

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

6.7. Метод ListenQuotes

Позволяет получать котировки по инструменту.

[C#]

```
void ListenQuotes(string symbol)
```

Наименование	Тип	Описание
Symbol	string	Код ЦБ из таблицы котировок SmartTrade

6.8. Метод ListenTicks

Позволяет получать все сделки на рынке по инструменту.

[C#]

```
void ListenTicks(string symbol)
```

Наименование	Тип	Описание
Symbol	string	Код ЦБ из таблицы котировок SmartTrade

6.9. Метод MoveOrder

Изменить цену в приказе

[C#]

```
void MoveOrder(string portfolio, string orderno, double targetPrice)
```

Наименование	Тип	Описание
Portfolio	String	Номер торгового счёта на торговой площадке.
OrderId	string	Id приказа на сервере котировок
TargetPrce	double	Новая цена для прказа

6.10. Метод PlaceOrder

Выставить приказ на рынок.

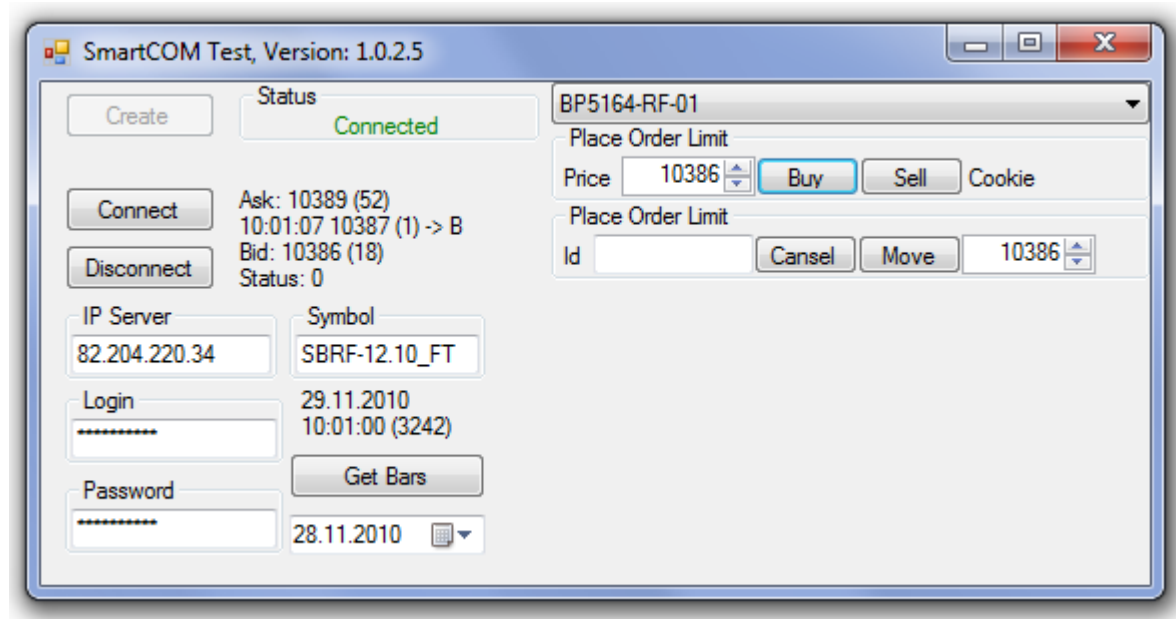
[C#]

```
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 SmartTrade
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 Для приказа: <i>По рынку</i> или <i>Стоп</i> , установите значение 0
Amount	Double	Объем, ЦБ в приказе
Stop	Double	Цена СТОП для приказа типа Стоп и Стоп-Лимит Для приказа: <i>По рынку</i> или <i>Лимит</i> , установите значение 0
Cookie	int	Ваш уникальный номер приказа, используется для определения Id приказа через событие OrderSucceeded, а также для определения ошибки при доставке приказа на сервер котировок, событием OrderFailed.

7. Пример использования

7.1. Программа TestConnect



Программа, тестирует все функции SmartCOM.

После запуска, нажмите кнопку Create, для создания экземпляра SmartCOM и назначения обработчиков. Если служба SmartCOM запущена и соединение установлено, то начнётся прослушка по инструменту указанному в поле Symbol, запросится список доступных портфелей, история сделок по инструменту (последние 100), а так же после получения списка счётов по ним запросится история сделок, приказов за сессию и подставятся на прослушку. В противном случае нужно указать логин и пароль в соответствующие поля и нажать кнопку Connect, произойдёт подключение к серверу IT Invest, указанному в поле IP Server. Для отключения от сервера нажмите Disconnect. Текущий статус подключения отображается в поле Status.

Программа показывает, как работать с приказами. Для выставления лимитного приказа: выберите счёт, укажите цену в поле Price (подставить текущую, можно кликнув мышкой по слову ask/bid) затем выберите операцию купить или продать, кнопки Buy/Sell соответственно. После успешной доставки приказа в поле Place Order Limit -> Id появится его Id. По нему мы можем отменить приказ, кнопка Cancel или переместить задав новую цену, кнопка Move.

Также реализован простой способ, запросить большое количество баров, в данном случае пакетами по 500 штук. Выберите дату с которой следует начать запрос, данные поступают начиная с текущего времени по выбранное, и нажмите кнопку Get Bars.

Подробно разобрать функции, можно в исходном коде программы.