

# Java интерфейс доступа к торговому терминалу MT4

## Функциональное Описание

### Содержание

Java интерфейс доступа к торговому терминалу MT4	1
Функциональное Описание	1
Терминология	1
Обзор	3
Диаграммы последовательностей	4
Терминальный Сервер (Terminal Server)	7

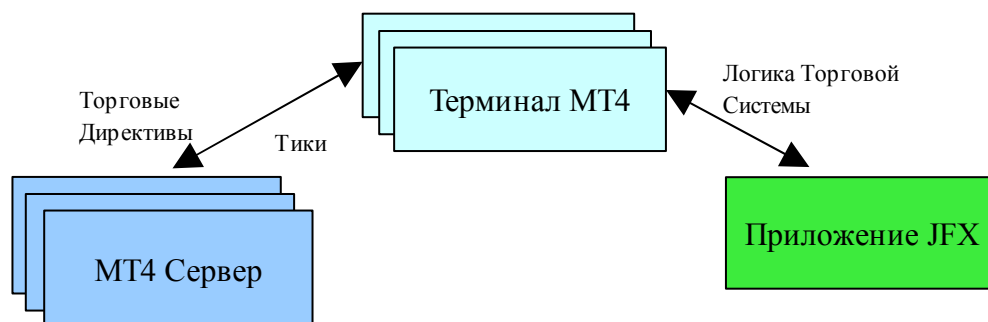
### Терминология

Термин	Описание
Советник (ЕА)	“Советник” (Expert Advisor) это механическая торговая система (МТС), имеющая привязку к определенному графику.
JFX	Java интерфейс доступа к терминалу MT4
JFX Сервер	Сокет-сервер, часть Java интерфейса доступа к терминалу MT4, который принимает соединение со стороны MT4 терминала и инициализирует JFX Стратегии
JFX Стратегия	Торговая стратегия, реализованная с помощью JFX API. Обычно представлена java-классом, унаследованным от <i>com.jfx.strategy.Strategy</i> .
MT4 Сервер	Программное обеспечение торгового сервера фирмы MetaQuotes Software Corp. Сервер получает все торговые ордера и другие запросы от торговых терминалов и исполняет их.
Терминал MT4	Клиентский терминал MetaTrader 4 является рабочим местом трейдера и позволяет работать на финансовых рынках Forex, CFD и Futures.

## Обзор

Java интерфейс доступа к терминалу MT4 (JFX) был разработан с целью предоставления доступа к торговым системам MetaQuotes с помощью стандартного терминала MetaTrader 4. Также была цель полностью исключить необходимость использования языка MQL4 при разработке механических торговых систем (советников).

Роль терминала MT4 сводится к посредничеству между MT4 Сервером и приложением на языке Java, которое использует библиотеку JFX:



Одно приложение JFX может работать со многими терминалами MT4 одновременно.

JFX состоит из трех частей:

- JFX MQL4 Советник (Expert Advisor) (часть терминала MT4)
- Коммуникационная DLL (часть терминала MT4)
- Библиотека Java классов (JFX, часть приложения на языке Java)

Для того, чтобы начать использовать библиотеку JFX, нужно создать свой собственный java класс, который наследуется от *com.jfx.strategy.Strategy* и переопределяет его метод *coordinate()*, например:

```
public class MyStrategy extends com.jfx.strategy.Strategy {
    public void init(String symbol, int period, StrategyRunner strategyRunner) {
        super.init(symbol, period, strategyRunner);
        //
        // загрузка существующих ордеров, восстановление после предыдущего перезапуска
        //
    }
    public void deinit() {
        // освобождение ресурсов (напр. открытых файлов) при выходе из советника
    }
    public void coordinate() {
        // здесь размещена логика механической торговой системы — реакция на новый тик
        /* можно использовать любые из методов API: accountBalance, accountCompany, accountCredit, accountCurrency,
        accountEquity, accountFreeMargin, accountMargin, accountName, accountNumber, accountProfit, comment, day, dayOfWeek,
        dayOfYear, getLastError, getTickCount, hour, iAC, iAD, iADX, iAlligator, iAO, iATR, iBands, iBars, iBarShift, iBearsPower,
        iBullsPower, iBWMFI, iCCI, iClose, iCustom, iDeMarker, iEnvelopes, iForce, iFractals, iGator, iHigh, iHighest, iLow, iLowest,
```

```

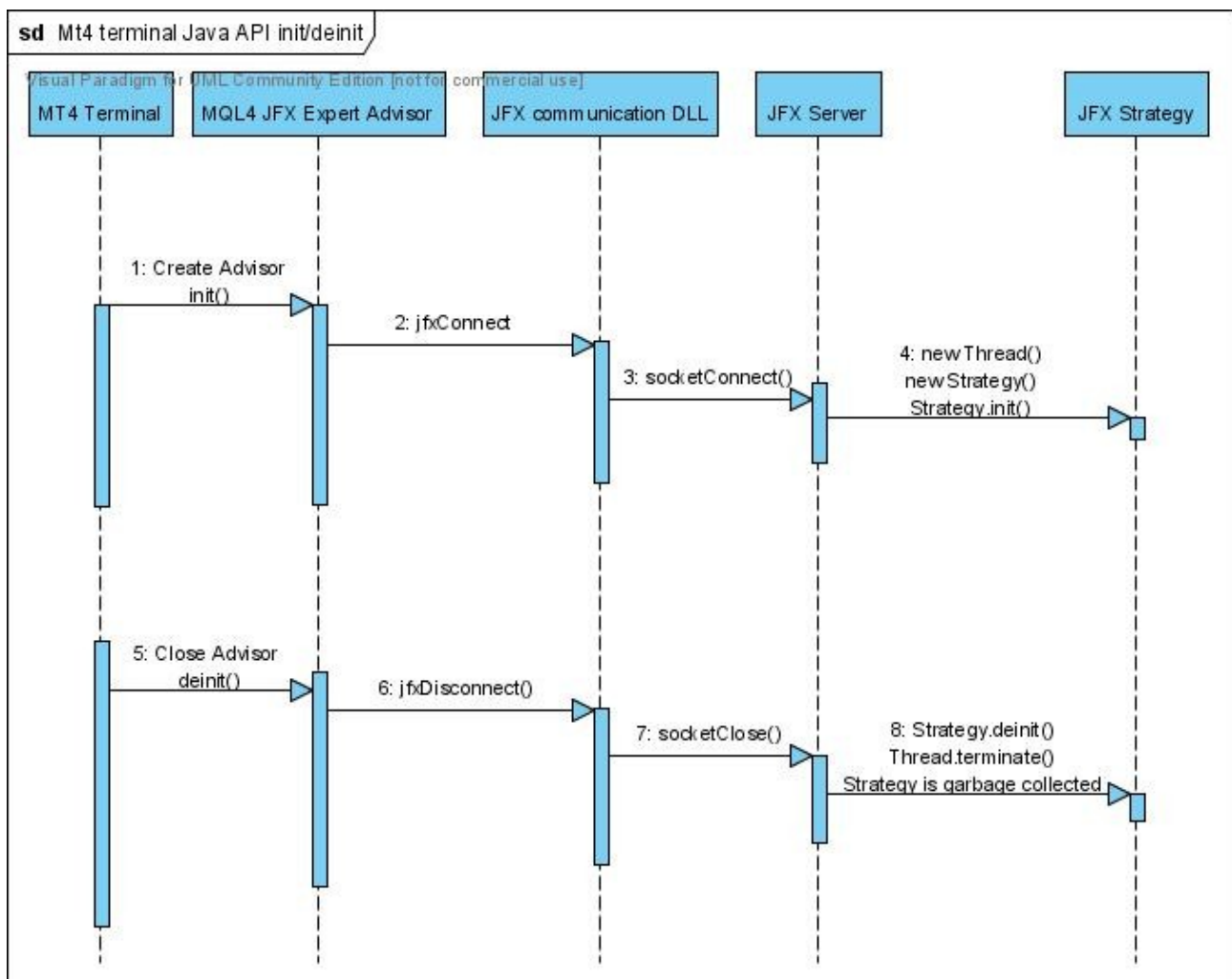
iMA, iMACD, iMFI, iMomentum, iOBV, iOpen, iOsMA, iRSI, iRVI, iSAR, isConnected, isDemo, iStdDev, isTesting, iStochastic,
isTradeContextBusy, isVisualMode, iTime, iVolume, iWPR, marketInfo, minute, month, objectCreate, objectCreate, objectCreate,
objectDelete, objectGet, objectGetFiboDescription, objectSet, objectSetFiboDescription, objectSetText, objectsTotal, objectType,
orderClose, orderCloseBy, orderClosePrice, orderCloseTime, orderComment, orderCommission, orderDelete, orderExpiration,
orderLots, orderMagicNumber, orderModify, orderOpenPrice, orderOpenTime, orderPrint, orderProfit, orderSelect, orderSend,
ordersHistoryTotal, orderStopLoss, ordersTotal, orderSwap, orderSymbol, orderTakeProfit, orderTicket, orderType, print,
refreshRates, seconds, timeCurrent, year
*/
}
}

```

## Диаграммы последовательностей

Название java класса советника передается в приложение JFX из терминала MT4. Библиотека JFX автоматически создает новый экземпляр этого класса и начинает посылку тиков.

Ниже приводится пример инициализации/завершения работы приложения JFX:



### Инициализация приложения JFX

- 1: Create Advisor – Создание Советника — при создании обычного советника JFX MQL4, терминал MT4 вызывает его метод init()
- 2: jfxConnect – метод советника init() подключается к Java приложению JFX с помощью библиотеки MT4IF.DLL
- 3: socketConnect – библиотека MT4IF.DLL устанавливает сокет-соединение с приложением JFX и возвращает id сессии советнику
- 4: new Thread/Strategy – во время соединения, JFX Сервер создает новый экземпляр java класса MTC и выделяет для его работы отдельный thread.

### Завершение работы приложения JFX

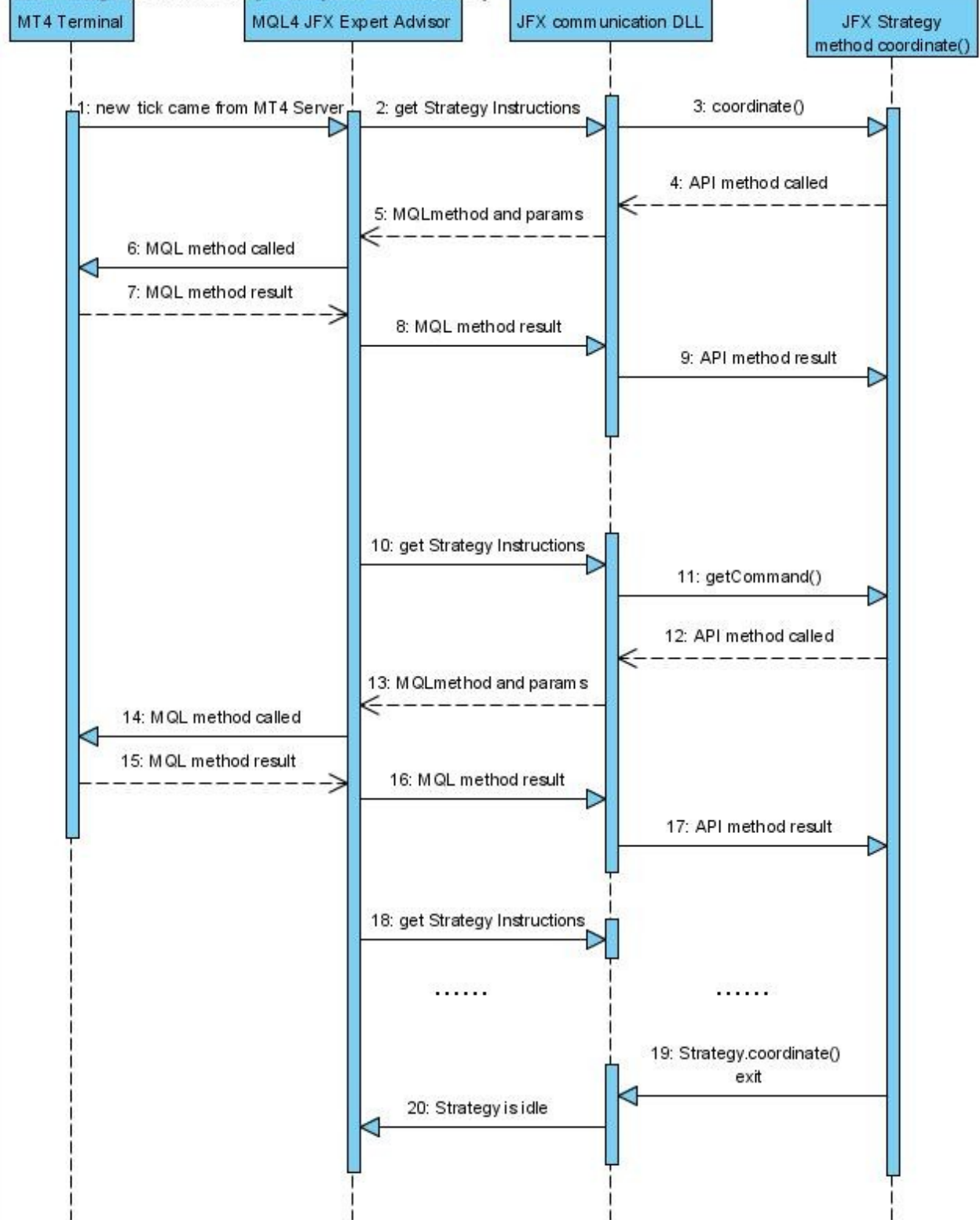
- 5: Close Advisor – Закрытие Советника — когда терминал MT4 удаляет советник JFX из графика выбранного инструмента, вызывается специальный метод советника deinit()
- 6: jfxDisconnect() - внутри метода deinit() происходит закрытие соединения с приложением JFX с помощью библиотеки MT4IF.DLL
- 7: socketClose() - библиотека MT4IF.DLL закрывает сокет-соединение с приложением JFX Application и освобождает выделенный ранее id сессии.
- 8: Strategy.deinit() - экземпляр java класса освобождается, Thread закрывается.

Завершение работы приложения JFX может также произойти в случае нарушения сетевого соединения с терминалом MT4. В этом случае библиотека MT4IF.DLL пытается возобновить сетевое соединение автоматически. Возобновление соединения для приложения JFX будет выглядеть как обычная инициализация.

После инициализации приложения JFX, вся работа происходит при обработке тиков от MT4 Сервера:

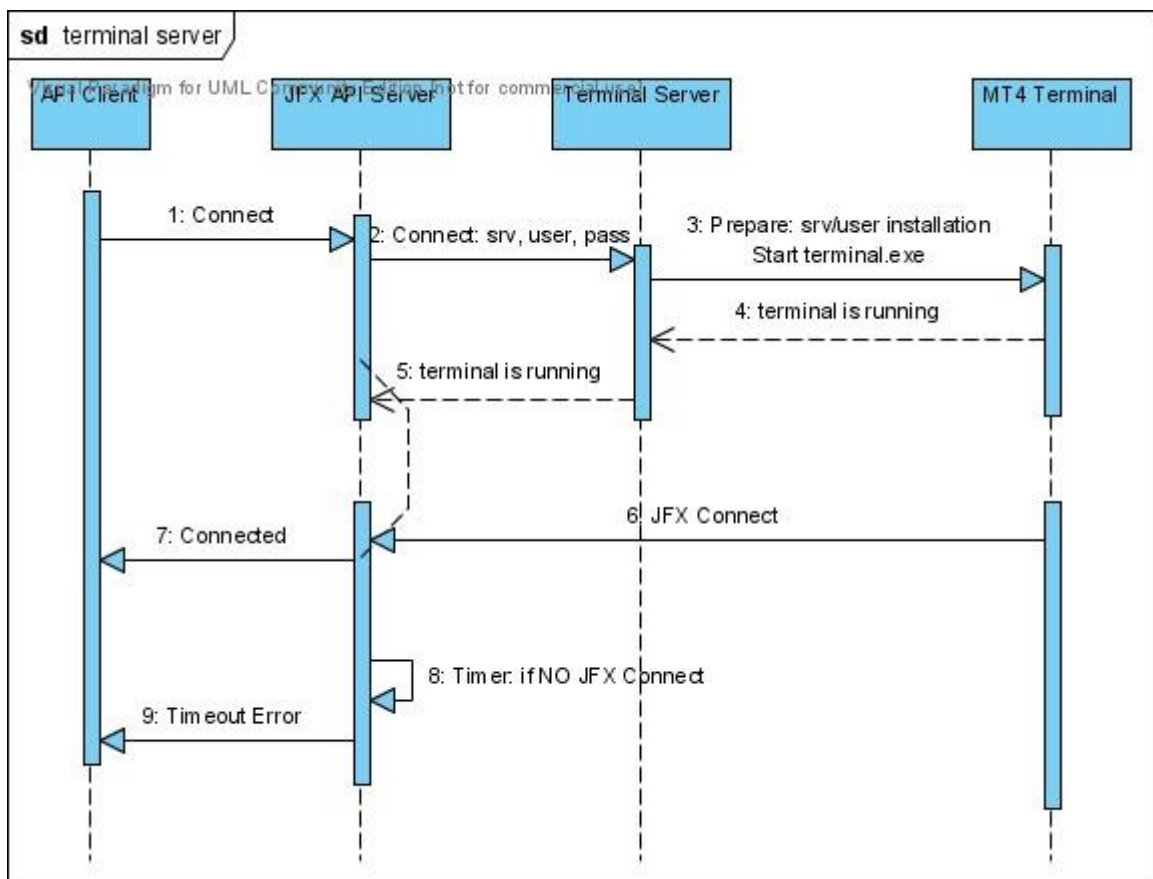
# sd Tick processing

Visual Paradigm for UML Community Edition [not for commercial use]



## Терминальный Сервер (Terminal Server)

Терминальный сервер - это Windows-приложение, которое облегчает запуск нескольких терминалов MT4, т. е. делает возможным для Java-приложения JFX создавать параллельно несколько соединений с различными MT4 серверами.



С использованием терминального сервера, java-приложение может самостоятельно инициализировать соединение с терминалом MT4 с помощью метода *connect*.

```
public class JFXExample extends Strategy {
    ...
    connect("127.0.0.1" /*адрес терминального сервера*/, 7788 /*порт*/,
        Broker.AlpariUK_Demo /*брокер*/, "1107623" /*пользователь*/, "w4ylog" /*пароль*/);
}
```