Interactive Brokers API

A Brief Overview
by
Stergios Marinopoulos
for
Hacker Dojo Meetup
9/26/2011

DISCLAIMER

Nothing in this talk should be construed to be Investment Advice or a Solicitation.

The information does not constitute investment advice or an offer to invest or to provide management services and is subject to correction, completion and amendment without notice.

It is not my intention to state, indicate or imply in any manner that current or past results are indicative of future results or expectations.

As with all investments, there are associated risks and you can and probably will lose money investing. Prior to making any investment, a prospective investor should consult with its own investment, accounting, legal and tax advisers to evaluate independently the risks, consequences and suitability of that investment.

Stergios Marinopoulos stergios@gmail.com

IB API Overview

- www.interactivebrokers.com
  - Global Offerings, Many Products
  - 19 Countries, 90 Markets
  - stocks, options, futures, forex, bonds, CFDs
- The beginning: Build "5" 8/7/2002
- Live, Paper, and Demo account
- What do you get in the API Download?
  1. The API Source (C++, Java)
  2. Sample Apps in C++, Java, Excel
  3. Test Client App!!!

Common Programs / Tasks

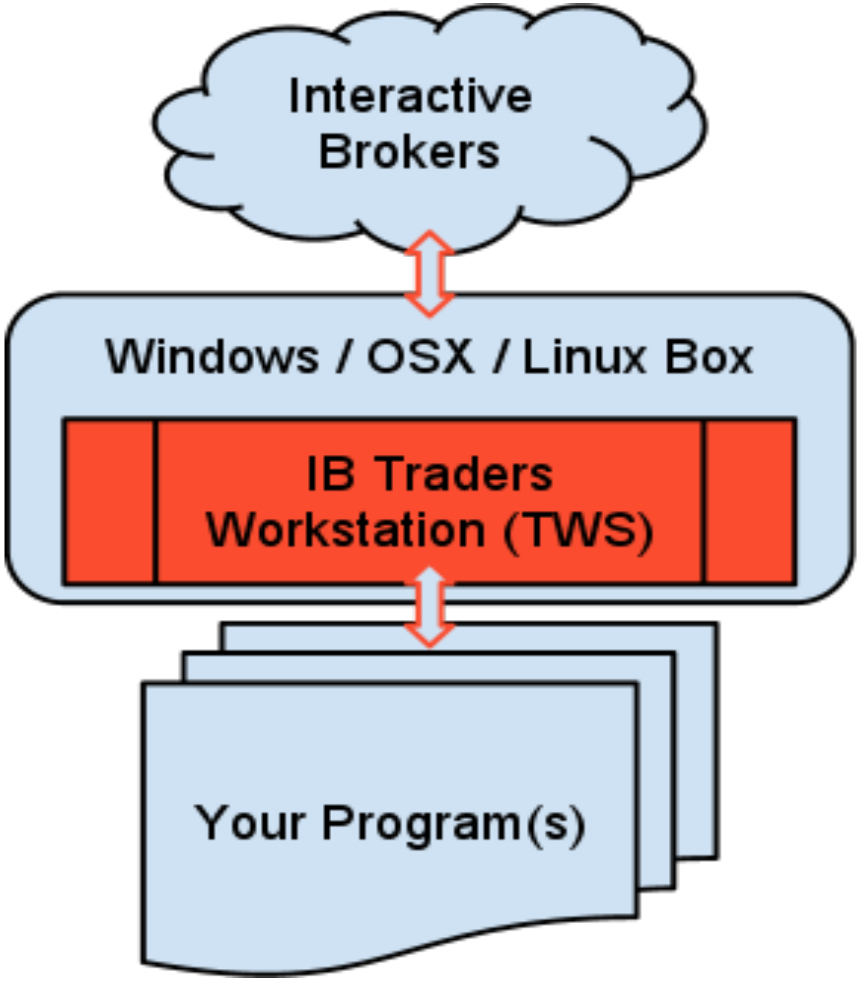
- Automate manual activities in TWS
- Subscribe to Market Data and Market Depth
- Retrieve Historical Data
- Execute algos & trading stratigies
- Access account / portfolio information
- Monitor open orders, statuses & executions

Stergios Marinopoulos stergios@gmail.com

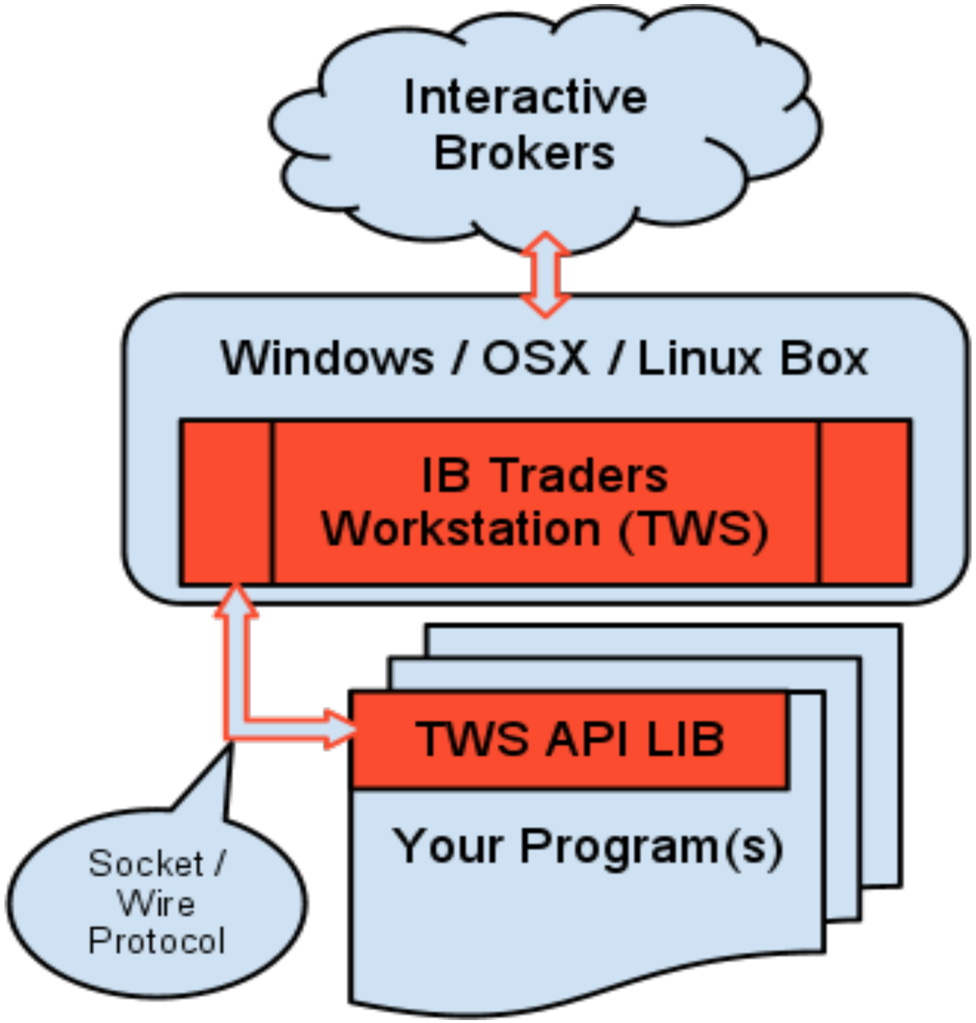
Common Programs / Tasks

- Automate manual activities in TWS
- ***Subscribe to Market Data*** and Market Depth
- Retrieve Historical Data
- ***Execute algos & trading stratigies***
- Access account / portfolio information
- ***Monitor open orders, statuses & executions***

Architecture - Markitecture A

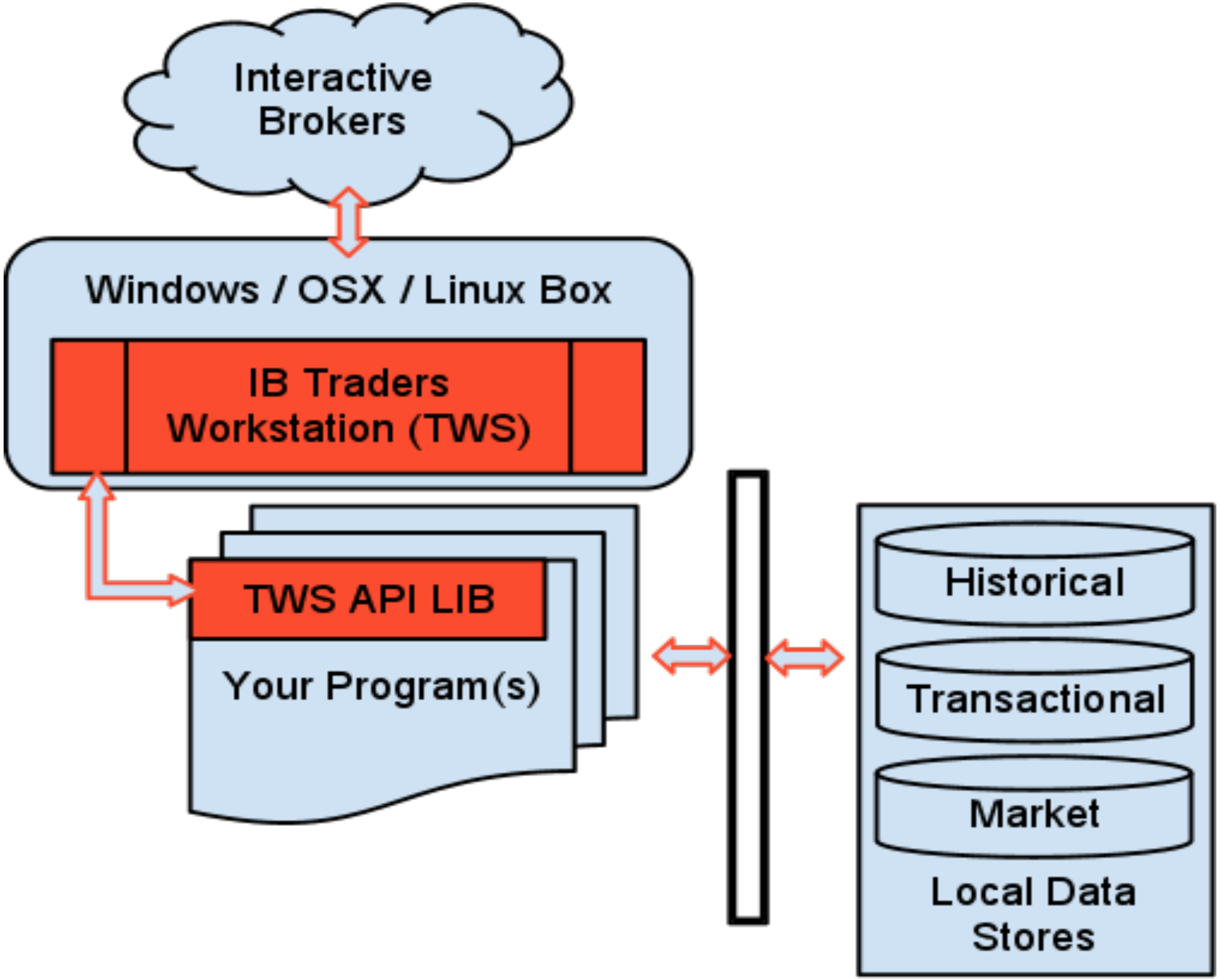


Architecture - Markitecture B



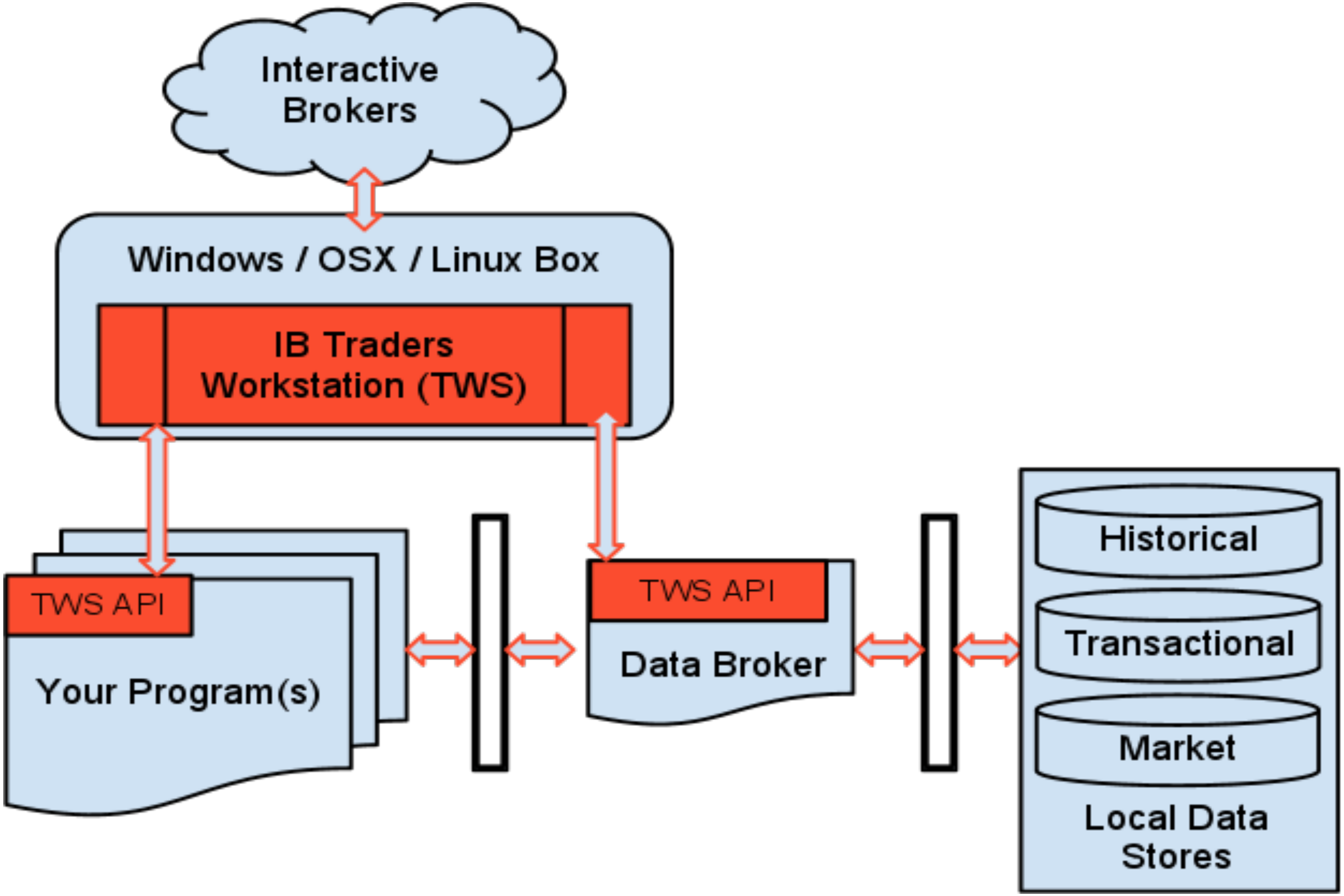
Stergios Marinopoulos stergios@gmail.com

Architecture - Markitecture C



Stergios Marinopoulos stergios@gmail.com

Architecture - Markitecture D



Stergios Marinopoulos stergios@gmail.com

Technology - Platform Choices

- IB API via TWS
- IB API Gateway
- IB via FIX
- Many Languages
- Many OS'es

Technology - Language Choices

ActiveX	Delphi, Excel-VBA, Matlab, .Net, VB, etc, etc
C++	Interactive Brokers - References MFC out of the box TwsApiC++ - Unix/Cygwin/OSX ready TWSAPI - autoconf ready - github.com/rudimeier/twsapi
C	TWS-C-API - tws-c-api.sourceforge.net
C#	IBCSharp - ibcsharp.codeplex.com Dinosaur Technology - www.dinosaurtech.com
DDE	Excel (pure excel)
Java	Interactive Brokers
Perl	Finance-InteractiveBrokers-TWS-v0.0.8
Python	ibpy - code.google.com/p/ibpy profitpy - code.google.com/p/profitpy
R	IBrokers - cran.r-project.org/web/packages/IBrokers qsiblive - censix.com
Ruby	ib-ruby.rubyforge.org
Misc / Platforms	Trading Shim - www.trading-shim.org NinjaTrader - www.ninjatrader.com JBookTrader - code.google.com/p/jbooktrader IBTrader - sourceforge.net/ibtrader.html JSystemTrader - jsystemtrader.sourceforge.net

Stergios Marinopoulos stergios@gmail.com

Technology
Complexity & Consequences

- C++/C#/C/Java
  - Two thread minimum
  - Full API exposure
  - You have the library source
- ActiveX
  - No (obvious) threading at all
  - Almost any Win32 Development Environment
  - Limited to the actual ActiveX implementation
- Excel DDE
  - What's a thread?
  - Familiar modeling environment

Architecture
Your "*Conceptual*" Program Loop

1. Connect to TWS
2. Listen for real time data events
3. Feed data into your model
4. Update model
5. Makes buy/sell/hold decision(s)
6. Take Action ? Execute decision(s)
7. Listen for execution result(s)
8. GOTO "3"

Stergios Marinopoulos stergios@gmail.com

But 1st
Two Types, Two Threads

EClientSocket

- Class
- Methods **SEND** messages to TWS
- Main program thread (Usually)

EWrapper

- Interface
- Defines methods that **RECEIVE** messages from TWS
- Call back thread
- Store data and return ASAP

TWS Java API package

- com.ib.client

Example Snippets
for Conceptual Program

- Connect to TWS
- Create a Contract
- Request Tick Stream
- Request Real Time Bars
- Place a Buy order
- Place a Sell order
- Place a Bracket Order

Example Snippets for Conceptual Program

- **Connect to TWS**
- Create a Contract
- Request Tick Stream
- Request Real Time Bars
- Place a Buy order
- Place a Sell order
- Place a Bracket Order

Stergios Marinopoulos stergios@gmail.com

Example Snippet - Connect to TWS

```
EClientSocket eClientSocket = new EClientSocket(this);
eClientSocket.eConnect(tws_host, tws_port, tws_clientId);

// no feedback on success!

// Later on check
eClientSocket.isConnected() ;


// Finally
eClientSocket.eDisconnect();
```

Example Snippets for Conceptual Program

- Connect to TWS
- **Create a Contract**
- Request Tick Stream
- Request Real Time Bars
- Place a Buy order
- Place a Sell order
- Place a Bracket Order

Example Snippet - Contract
Stock: GOOG

```
  Contract contract = null ;
  int cdRequestId = 123456789 ;

  try {
    contract              = new Contract();
    contract.m_symbol     = "GOOG" ;
    contract.m_secType    = "STK"
    contract.m_exchange   = "SMART" ;
    contract.m_currency   = "USD" ;
    eClientSocket.reqContractDetails( cdRequestId , contract );
  } catch (Throwable t) {
    // handle error
  }

////////////////////////////////////////////////////////////////////

 @Override // on EWrapper interface, in callback thread
 public void contractDetails(int reqId, ContractDetails contractDetails)
 {
  contractId = contractDetails.m_summary.m_conId ;
  // push data on FIFO queue, return
 }
```

Stergios Marinopoulos   stergios@gmail.com

Example Snippet - Contract Index: S&P 500 (SPX)

```
Contract contract = null ;
int cdRequestId = 123456789 ;

try {
  contract                 = new Contract();
  contract.m_symbol        = "SPX" ;
  contract.m_secType       = "IND"
  contract.m_exchange      = "CBOE" ;
  contract.m_currency      = "USD" ;
  eClientSocket.reqContractDetails( cdRequestId , contract );
} catch (Throwable t) {
  // handle error
}
```

Example Snippet - Contract Futures: S&P 500 (ES)

```
Contract contract = null ;
int cdRequestId = 123456789 ;

try {
  contract                  = new Contract();
  contract.m_symbol         = "ES" ;
  contract.m_secType        = "FUT"
  contract.m_expiry         = "20111216" ;
  contract.m_exchange       = "GLOBEX" ;
  contract.m_currency       = "USD" ;
  contract.m_includeExpired = true ;      // useful for historical data
  eClientSocket.reqContractDetails( cdRequestId , contract );
} catch (Throwable t) {
  // handle error
}

//  ES 20110617 Contract Id: 73462897
//  ES 20110916 Contract Id: 76308824
//  ES 20111216 Contract Id: 79028739
//  ES 20120316 Contract Id: 81918684
```

Stergios Marinopoulos   stergios@gmail.com

Example Snippet - Contract
FX: EUR.USD

```
Contract contract = null ;
int cdRequestId = 123456789 ;

try {
  contract                    = new Contract();
  contract.m_symbol         = "EUR" ;
  contract.m_secType        = "CASH"
  contract.m_exchange       = "IDEALPRO" ;
  contract.m_currency       = "USD" ;
  eClientSocket.reqContractDetails( cdRequestId , contract );
} catch (Throwable t) {
  // handle error
}

// Euro Swiss: "EUR", "CHF"
// Dollar Yen: "USD", "JPY"
// Pound Yen: "GBY", "JPY"
```

Stergios Marinopoulos   stergios@gmail.com

Example Snippet - Contract Option: GOOG Dec 550 Call

```
Contract contract = null ;
int cdRequestId = 123456789 ;

try {
  contract                  = new Contract();
  contract.m_symbol         = "GOOG" ;
  contract.m_secType        = "OPT"
  contract.m_exchange       = "SMART" ;
  contract.m_expiry         = "20111216";
  contract.m_strike         = 550.0 ;
  contract.m_right          = "C" ;
  contract.m_currency       = "USD" ;
  eClientSocket.reqContractDetails( cdRequestId , contract );
} catch (Throwable t) {
  // handle error
}
```

Stergios Marinopoulos stergios@gmail.com

Example Snippet - Contract Combo: Calender Spread on CL

```
try {
   ComboLeg leg1 = new ComboLeg(); // for the first leg
   ComboLeg leg2 = new ComboLeg(); // for the second leg
   Vector addAllLegs = new Vector();

   leg1.m_conId    = conIdDec11; leg2.m_conId    = conIdJune12;
   leg1.m_ratio    = 1;          leg2.m_ratio    = 1;
   leg1.m_action   = "BUY";      leg2.m_action   = "SELL";
   leg1.m_exchange = "NYMEX";    leg2.m_exchange = "NYMEX";

   addAllLegs.add(leg1);         addAllLegs.add(leg2);

   Contract contract    = new Contract();
   contract.m_symbol    = "CL";
   contract.m_secType   = "BAG";
   contract.m_exchange  = "NYMEX";
   contract.m_currency  = "USD";
   contract.m_comboLegs = addAllLegs;
   eClientSocket.reqContractDetails( cdRequestId , contract );
} catch (Throwable t) {
   // handle error
}
```

Stergios Marinopoulos   stergios@gmail.com

Example Snippets for Conceptual Program

- Connect to TWS
- Create a Contract
- **Request Tick Stream (A)**
- Request Real Time Bars
- Place a Buy order
- Place a Sell order
- Place a Bracket Order

Stergios Marinopoulos stergios@gmail.com

Example Snippet - Tick Stream

// Call this method to request market data.

eClientSocket.**reqMktData**(..., **ticktype**, ...)

// The market data will be collected on 1 or more
// of EWrappers methods:

tickPrice(...)
tickSize(...)
tickOptionComputation(...)
tickGeneric(...)
tickString(...)
tickEFP(...)

Example Snippet
Tick Types

Tick Type ID	**Type, Resulting Tick Value**
100,	Option Volume (stocks), 29 30
101,	Option Open Interest (stocks), 27 28
104,	Historical Volatility (stocks), 23
106,	Option Implied Volatility (stocks), 24
162,	Index Future Premium, 31
165,	Miscellaneous Stats, 15 16 17 18 19 20 21
221,	Mark Price (used in TWS P&L computations), 37
225,	Auction values (volume, price and imb), 34, 35, 36
233,	RTVolume, 48
236,	Shortable, 46
256,	Inventory,
258,	Fundamental Ratios, 47

Stergios Marinopoulos stergios@gmail.com

Example Snippet
Tick Types

Tick Type ID	**Type, Resulting Tick Value**
100,	Option Volume (stocks), 29 30
101,	Option Open Interest (stocks), 27 28
104,	Historical Volatility (stocks), 23
106,	Option Implied Volatility (stocks), 24
162,	Index Future Premium, 31
165,	***Miscellaneous Stats, 15 16 17 18 19 20 21***
221,	Mark Price (used in TWS P&L computations), 37
225,	Auction values (volume, price and imb), 34, 35, 36
233,	***RTVolume, 48***
236,	Shortable, 46
256,	Inventory,
258,	Fundamental Ratios, 47

Stergios Marinopoulos stergios@gmail.com

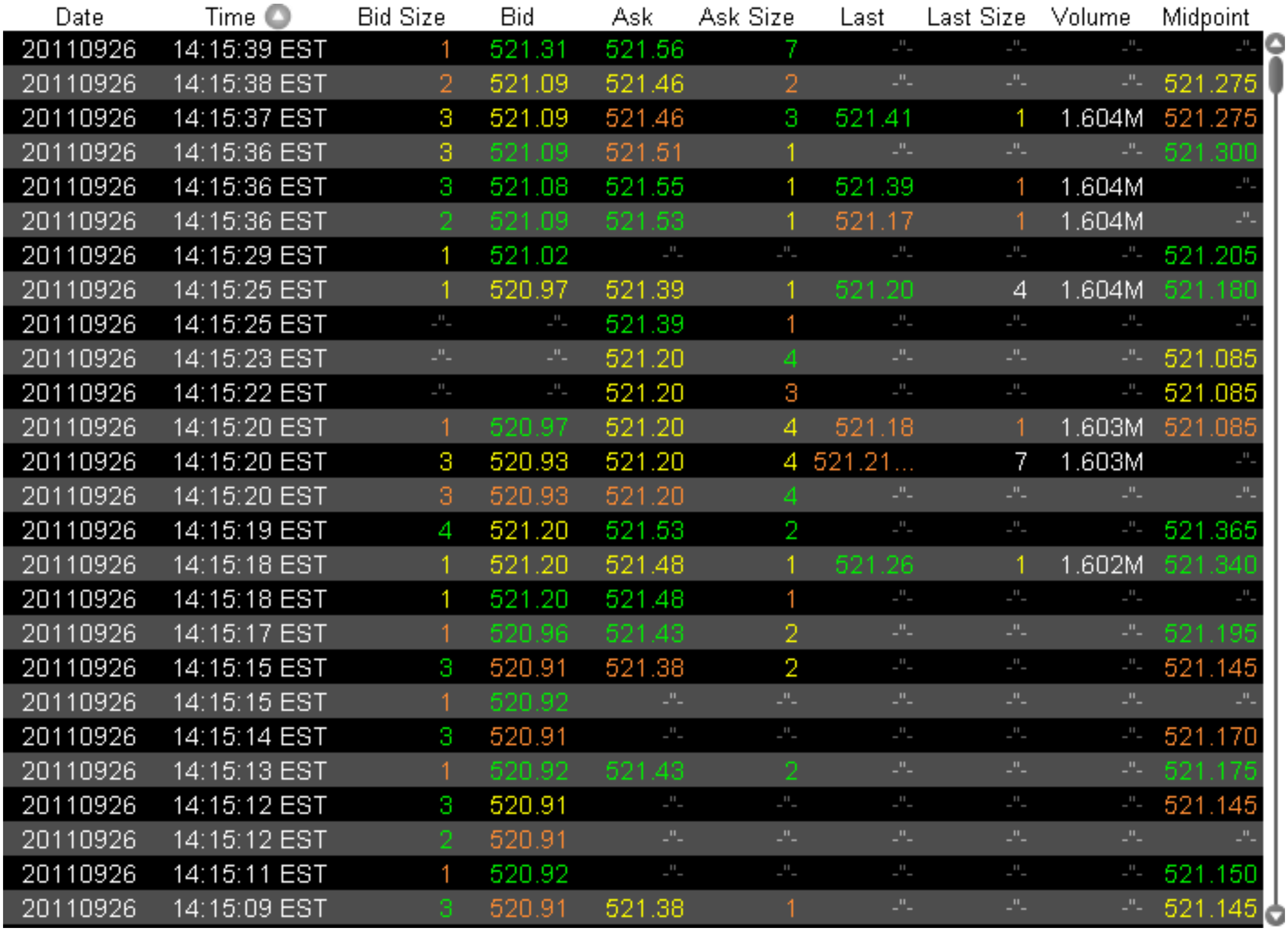
Example Snippet - Tick Stream TickType 165 "Misc Stats"

```
tickType = "165" ;
eClientSocket.reqMktData(reqId, contract, "165", snapshot) ;
////////////////////////////////////////////////////
// The Misc Stats data is delivered via EWrappers:
```

CALL BACK	TICK-VALUE
tickGeneric()	halted
tickPrice()	askPrice
tickPrice()	bidPrice
tickPrice()	close
tickPrice()	high
tickPrice()	lastPrice
tickPrice()	low
tickPrice()	open
tickSize()	askSize
tickSize()	bidSize
tickSize()	lastSize
tickSize()	volume
tickString()	lastTimestamp

Stergios Marinopoulos stergios@gmail.com

TickType 165 "Misc Stats" - Visualized



Date	Time	Bid Size	Bid	Ask	Ask Size	Last	Last Size	Volume	Midpoint
20110926	14:15:39 EST	1	521.31	521.56	7	-"-	-"-	-"-	-"-
20110926	14:15:38 EST	2	521.09	521.46	2	-"-	-"-	-"-	521.275
20110926	14:15:37 EST	3	521.09	521.46	3	521.41	1	1.604M	521.275
20110926	14:15:36 EST	3	521.09	521.51	1	-"-	-"-	-"-	521.300
20110926	14:15:36 EST	3	521.08	521.55	1	521.39	1	1.604M	-"-
20110926	14:15:36 EST	2	521.09	521.53	1	521.17	1	1.604M	-"-
20110926	14:15:29 EST	1	521.02	-"-	-"-	-"-	-"-	-"-	521.205
20110926	14:15:25 EST	1	520.97	521.39	1	521.20	4	1.604M	521.180
20110926	14:15:25 EST	-"-	-"-	521.39	1	-"-	-"-	-"-	-"-
20110926	14:15:23 EST	-"-	-"-	521.20	4	-"-	-"-	-"-	521.085
20110926	14:15:22 EST	-"-	-"-	521.20	3	-"-	-"-	-"-	521.085
20110926	14:15:20 EST	1	520.97	521.20	4	521.18	1	1.603M	521.085
20110926	14:15:20 EST	3	520.93	521.20	4	521.21...	7	1.603M	-"-
20110926	14:15:20 EST	3	520.93	521.20	4	-"-	-"-	-"-	-"-
20110926	14:15:19 EST	4	521.20	521.53	2	-"-	-"-	-"-	521.365
20110926	14:15:18 EST	1	521.20	521.48	1	521.26	1	1.602M	521.340
20110926	14:15:18 EST	1	521.20	521.48	1	-"-	-"-	-"-	-"-
20110926	14:15:17 EST	1	520.96	521.43	2	-"-	-"-	-"-	521.195
20110926	14:15:15 EST	3	520.91	521.38	2	-"-	-"-	-"-	521.145
20110926	14:15:15 EST	1	520.92	-"-	-"-	-"-	-"-	-"-	-"-
20110926	14:15:14 EST	3	520.91	-"-	-"-	-"-	-"-	-"-	521.170
20110926	14:15:13 EST	1	520.92	521.43	2	-"-	-"-	-"-	521.175
20110926	14:15:12 EST	3	520.91	-"-	-"-	-"-	-"-	-"-	521.145
20110926	14:15:12 EST	2	520.91	-"-	-"-	-"-	-"-	-"-	-"-
20110926	14:15:11 EST	1	520.92	-"-	-"-	-"-	-"-	-"-	521.150
20110926	14:15:09 EST	3	520.91	521.38	1	-"-	-"-	-"-	521.145

Stergios Marinopoulos
stergios@gmail.com

Example Snippets for Conceptual Program

- Connect to TWS
- Create a Contract
- **Request Tick Stream (B)**
- Request Real Time Bars
- Place a Buy order
- Place a Sell order
- Place a Bracket Order

Stergios Marinopoulos stergios@gmail.com

Example Snippet - Tick Stream TickType 233 "RTVolume"

```
tickList = "233,mdoff" ;
eClientSocket.reqMktData(reqId, contract, tickList, snapshot) ;



//////////////////////////////////////////////

// The RTVolume data is delivered via EWrappers:
tickString(requestId, field, value)
{
    if ( field == 48 )
    {
        // store data in a FIFO queue and return
    }
}
```

Example Snippet - Tick Stream TickType 233 "RTVolume"

"last trade price";"last trade size";"last trade time";"total volume";"vwap";" single trade flag"

```
1127.25;219;1316698383578;850035;1134.4753169;false
1127.25;25;1316698383906;850060;1134.47510117;false
1127.25;252;1316698384672;850312;1134.47295728;false
1127.25;149;1316698385000;850461;1134.47169035;false
1127.25;73;1316698385547;850534;1134.47107053;false
1127.00;65;1316698386203;850599;1134.47050343;false
1127.00;4;1316698386547;850603;1134.4704683;false
1127.25;17;1316698387188;850620;1134.47032194;false
```

Example Snippets for Conceptual Program

- Connect to TWS
- Create a Contract
- Request Tick Stream (B)
- **Request Real Time Bars**
- Place a Buy order
- Place a Sell order
- Place a Bracket Order

Stergios Marinopoulos stergios@gmail.com

Example Snippet - Real Time Bars

```
// Call this method to request market data.

eClientSocket.reqRealTimeBars(..., barSize, whatToShow, useRTH) ;

// The real time bars are collected on EWrappers:

realtimeBar(...)
{
  // store data in a FIFO queue and return
}
```

Stergios Marinopoulos stergios@gmail.com

Example Snippet - Real Time Bars

```
barSize    = 5 ; // Only 5 second bars are supported
whatToShow = "TRADES" ; // "BID", "ASK", "MIDPOINT", "BID_ASK"
useRTH     = false ;    // ...   OIV, OV, HV
eClientSocket.reqRealTimeBars(reqId, contract, barSize, whatToShow, useRTH) ;

//////////////////////////////////////////////////

// The real time bars are collected on EWrapper:
public void realtimeBar(int reqId,
    long time, // start of the bar!
    double open, double high, double low, double close,
    long volume, double wap, int count)
{
  // store data in a FIFO queue and return
}
```

Stergios Marinopoulos stergios@gmail.com

Data Streams - Limitations

- 50 Messages / sec
- 100 tickers simultaneously
- 3 Levell II Feeds simultaneously
- Sampled Tick Stream
- Differential Tick Stream

- Aggregate higher time frames yourself
- No technical indicators
- Everyone has to reinvent the wheel

Example Snippets for Conceptual Program

- Connect to TWS
- Create a Contract
- Request Tick Stream (B)
- Request Real Time Bars
- **Place a Buy order**
- Place a Sell order
- Place a Bracket Order

Stergios Marinopoulos stergios@gmail.com

Example Snippet - Buy Order

```
// Call this method to place a buy order.

eClientSocket.placeOrder(orderId, contract, order);


// Listen for order executions on EWrappers

void orderStatus(...);

// and/or

void execDetails(orderId, contract, execution);
```

Example Snippet - Buy Order Market Order

```
// Create order
Order order          = new Order ;
order.m_action       = "BUY";  // or "SELL"
order.m_totalQuantity = 100 ;
order.m_orderType     = "MKT" ;
order.m_transmit      = true;
order.m_orderRef      = "note to self" ;
orderEntry.m_goodTillDate   = "GTC" ;

eClientSocket.placeOrder(orderId, contract, order);

////////////////////////////////////////

void execDetails(orderId, contract, execution)
{
  price = execution.m_avgPrice ;  // put in FIFO
  // signal other thread we have an execution
}
```

Stergios Marinopoulos stergios@gmail.com

Example Snippet - Buy Order Limit Order

```
// Create order
Order order             = new Order ;
order.m_action          = "BUY";  // or "SELL"
order.m_totalQuantity = 100 ;
order.m_orderType      = "LMT" ;
order.m_lmtPrice       = 50.0 ;
order.m_transmit       = true;
order.m_orderRef       = "note to self" ;
orderEntry.m_goodTillDate   = "DAY" ;

eClientSocket.placeOrder(orderId, contract, order);

////////////////////////////////////////

void execDetails(orderId, contract, execution)
{
  price = execution.m_avgPrice ;  // put in FIFO
  // signal other thread we have an execution
}
```

Stergios Marinopoulos   stergios@gmail.com

Example Snippet - Buy Order
Stop Order

```
// Create order
Order order          = new Order ;
order.m_action        = "BUY"; // or "BUY"
order.m_totalQuantity = 100 ;
order.m_orderType     = "STP" ;
order.m_auxPrice      = 55.0 ;
order.m_transmit      = true;
order.m_orderRef      = "note to self" ;
orderEntry.m_goodTillDate   = "DAY" ;
eClientSocket.placeOrder(orderId, contract, order);

////////////////////////////////////////

void execDetails(orderId, contract, execution)
{
  price = execution.m_avgPrice ;  // put in FIFO
  // signal other thread we have an execution
}
```

Example Snippets for Conceptual Program

- Connect to TWS
- Create a Contract
- Request Tick Stream (B)
- Request Real Time Bars
- Request Scanner
- Place a Buy order
- **Place a Sell order**
- Place a Bracket Order

Stergios Marinopoulos stergios@gmail.com

Example Snippet - Sell Order

Replace "**BUY**" with "**SELL**" in the previous examples. We had:

```
order.m_action       = "BUY";

// Now we want:

order.m_action       = "SELL";
```

Stergios Marinopoulos   stergios@gmail.com

Example Snippets for Conceptual Program

- Connect to TWS
- Create a Contract
- Request Tick Stream (B)
- Request Real Time Bars
- Request Scanner
- Place a Buy order
- Place a Sell order
- **Place a Bracket Order**

Stergios Marinopoulos stergios@gmail.com

Example Snippet - Bracket Order

```
orderEntry.m_action         = "BUY" ;  // use orderId = 10
orderEntry.m_totalQuantity  = 100 ;
orderEntry.m_orderType      = "MKT" ;  // assume current price is $55.00
orderEntry.m_transmit       = false;

orderTarget.m_action        = "SELL" ; // use orderId = 11
orderTarget.m_totalQuantity = 100 ;
orderTarget.m_orderType     = "LMT" ;
orderTarget.m_lmtPrice      = 60.0 ;
orderTarget.m_transmit      = false;
orderTarget.m_parentId      = 10 ;

orderStop.m_action          = "SELL" ; // use orderId = 12
orderStop.m_totalQuantity   = 100 ;
orderStop.m_orderType       = "STP" ;
orderStop.m_auxPrice        = 50.0 ;
orderStop.m_transmit        = true;
orderStop.m_parentId        = 10 ;

eClientSocket.placeOrder(10, contract, orderEntry);
eClientSocket.placeOrder(11, contract, orderTarget);
eClientSocket.placeOrder(12, contract, orderStop);
```

What Data Topics did I skip?

- Live Market Data
  - Order Book / Level II
    - reqMktDepth(), updateMktDepth(), updateMktdepthL2()
  - Option Computations, Exercise
  - Snapshots
  - Markets Scanners
    - reqScannerSubscription(), scannerData()
  - Many, many tick types
- Bonds
- Historical Data
  - reqHistoricalData(), historicalData()
- Account & Portfolio Details
- News Bulletins
- Fundamental Data

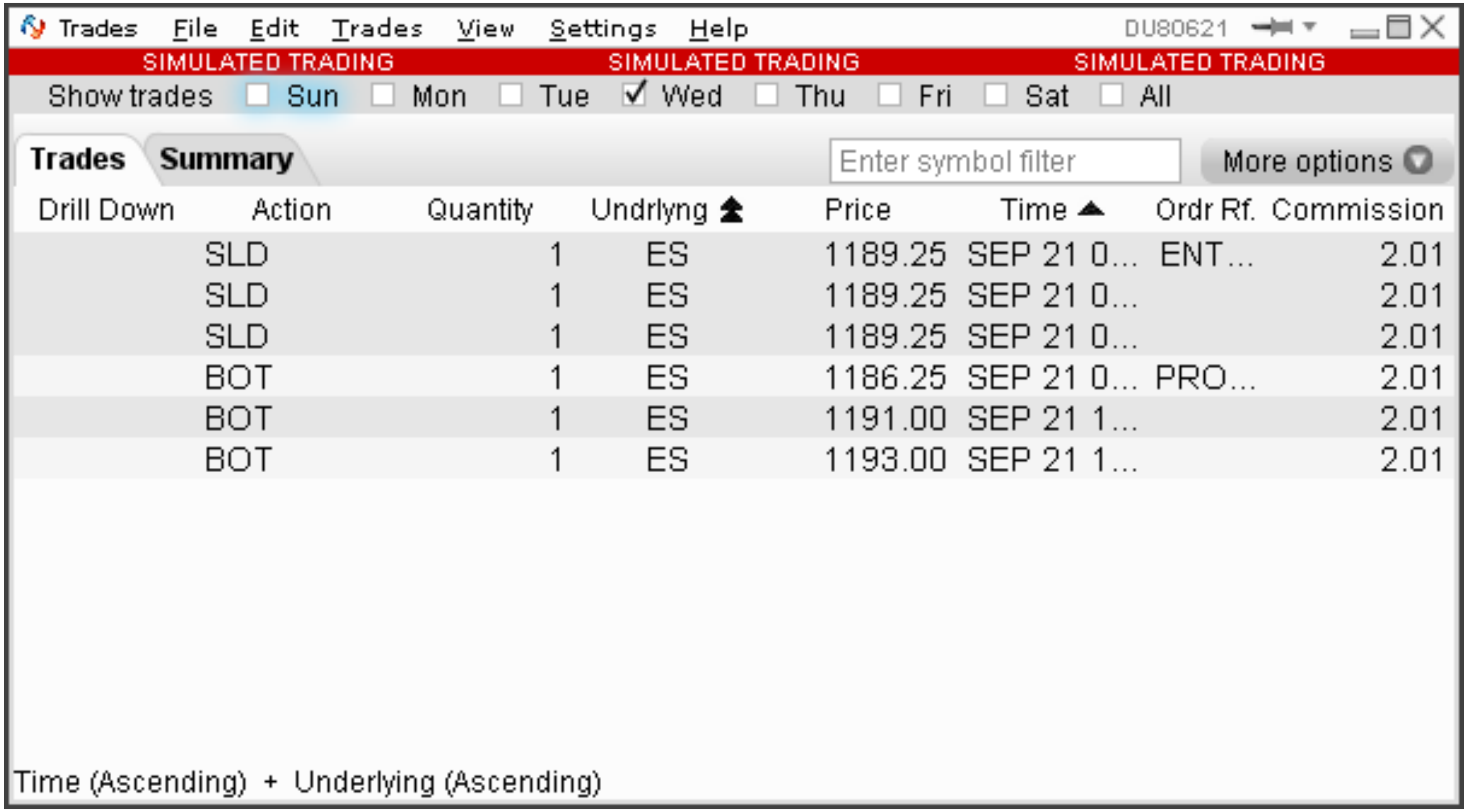
Gotchas

1. GUI Check-boxes from TWS pollute API behavior
2. Throttling WRT historical data (60 reqs/10 min)
3. Unhanded Exceptions / Errors (not uniform)
4. Daily Server "Reboot"
5. Daily TWS “Reboot”
6. Security Device / Dongle
7. Paper Trade vs Live Trade machine restriction
8. Data Issues

Stergios Marinopoulos stergios@gmail.com

Gotchas - GUI Check - Boxes

GUI Check-boxes pollute API behavior


Trades File Edit Trades View Settings Help
DU80621
SIMULATED TRADING SIMULATED TRADING SIMULATED TRADING
Trades
Summary
Enter symbol filter
More options

Show trades □ Sun □ Mon □ Tue ☑ Wed □ Thu □ Fri □ Sat □ All

Drill Down	Action	Quantity	Undrlyng	Price	Time	Ordr Rf.	Commission
	SLD	1	ES	1189.25	SEP 21 0...	ENT...	2.01
	SLD	1	ES	1189.25	SEP 21 0...		2.01
	SLD	1	ES	1189.25	SEP 21 0...		2.01
	BOT	1	ES	1186.25	SEP 21 0...	PRO...	2.01
	BOT	1	ES	1191.00	SEP 21 1...		2.01
	BOT	1	ES	1193.00	SEP 21 1...		2.01

Time (Ascending) + Underlying (Ascending)

Gotchas - Pacing Violations

Throttling WRT historical data (60 reqs/10 min)

- Limit yourself to 1 request every 10 seconds, or
- Keep a queue of request times and manage for flexibility to allow bursting

Gotchas - Exceptions / Errors

- Unhanded Exceptions
- Error Callbacks
  - eWrapper.error1()
  - eWrapper.error2()
  - eWrapper.error3()
- Why so many?

Ok, so you got an error, now what?

1. Keep Buying?
2. Keep Selling?
3. Close all Positions?
4. Panic?
5. Send yourself an SMS?

Gotchas - Daily Reboots

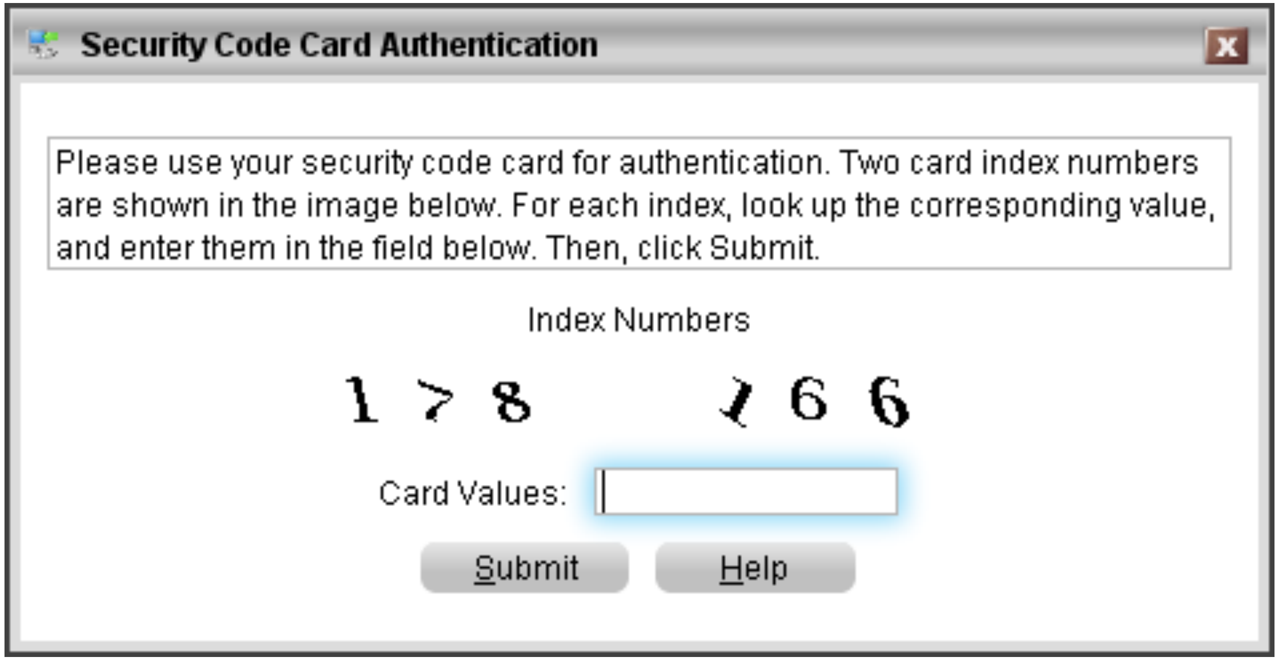
- Daily Server "Reboot"
- Daily TWS “Reboot”

- IBController can help

Gotchas - Security

Login Security

- ○ Dongle
- ○ Card - Two Factor Auth and Ask



Security Code Card Authentication

Please use your security code card for authentication. Two card index numbers are shown in the image below. For each index, look up the corresponding value, and enter them in the field below. Then, click Submit.

Index Numbers

1 7 8 1 6 6

Card Values:

Stergios Marinopoulos stergios@gmail.com

Gotchas - Paper v.s. Live

- Restrictions
  - both must be on same physical computer to run simultaneously
  - combined they are under the same data throttles
  - another reason to create a data broker
- Paper uses only Top of the Book to simulate orders

Stergios Marinopoulos stergios@gmail.com

Gotchas - Data Issues

Strategy for dealing with bad data

- Spikes?
- Bid / Ask Crossed?
- What does DanBot do?

- Clock Drift
- Flash Crash

Best Practices

1. Use ContractId’s
2. Server Logging Level
3. IBController - keep TWS running
4. Redundancy, Redundancy, Redundancy
5. More Redundancy
6. Yahoo Group: TWSAPI community where API programmers talk

Stergios Marinopoulos stergios@gmail.com

Best Practices - ContractIds

- Use ContractId’s when creating Contracts to eliminate several sources of common error
- Prefetch and record locally

Don't	Do

contract = new Contract();
contract.m_symbol   = "GOOG" ;
contract.m_secType = "OPT" ;
contract.m_exchange = "SMART" ;
contract.m_expiry   = "20111216";
contract.m_strike   = 550.0 ;
contract.m_right   = "C" ;
contract.m_currency = "USD"
	contract = new Contract();
	contract.conId = 123456789 ;

Stergios Marinopoulos   stergios@gmail.com

Best Practices - Server Logging

- TWS has a log file "log.txt"
- It can log API details
- eClientSocket.setServerLoglevel(5) ;

Log levels:
1 = SYSTEM (least detailed)
2 = ERROR (default, if no level is specified)
3 = WARNING
4 = INFORMATION
5 = DETAIL (most detailed, maybe performance hit)

Best Practices - IBController

- Keeps TWS running forever after you supply credentials
- If you opted-out of security, it is even more useful
    - Do you really want to opt-out? Think hard about this!

Best Practices - Redundancey, Redundancey, Redundancey

Q: How to do you keep portfolio state?
A: As many ways as possible!

1. eWrapper.orderStatus()
2. eWrapper.execDetails()
3. eClient.reqAllOpenOrders()
4. eClient.reqExecutions()
5. eClient.reqAccountUpdates()
6. Executions.txt file
7. IB's FLEX WebService to retrieve account balance, transaction fees, cash transactions, NAV, etc

Best Practices - More Redundancy

- Two internet connections at home?
- Two computers at home?
- Have brokers telephone # on speed dial
    - hope they answer!
- Know how to use Broker's web site
- Host in the Cloud?
    - AWS Eastern instances reduce latency
    - Virtual X Server - Xvfb
- Create a web based status/reporting GUI
- Push notices via SMS

Remember, it's your money on the line!

Best Practices - Community

- Yahoo! Group: TWSAPI community where API programmers talk
  - Read this religiously
- Interactive Brokers BB - Traders Chat, API
  - (All the hard questions get reposted Yahoo! TWSAPI, so you might as well start there.)
- Interactive Brokers Webinars - Live, recorded
  - Best way to start
- Book - Java - Concurrency in Practice by Goetz
  - Become one with Concurrency

Questions?

Interactive Brokers API Overview
by
Stergios Marinopoulos

Stergios Marinopoulos stergios@gmail.com

Appendix - Technology - Misc.

Good ideas

- Brush up on concurrency in general
- In particular java Thread cooperation
    - Message Queues (Blocking)
        - LinkedBlockingQueue
    - Producer / Consumer Pattern
    - CountdownLatch
    - CyclicBarrier