

# API Reference Guide

of alternatives for connectivity directly with  
ary Application Program Interface (API)  
ader Workstation and does not require  
dedicated FIX server. This solution is  
l Basic. In addition, we offer an  
s already supporting a FIX  
re. Both solutions can  
ne. See the sub-level tabs  
tion types.

er the Internet. It is  
and IB. The  
sing the same  
need to  
'Users'  
and the



**Interactive Brokers**

*The Professional's Gateway to the World's Markets*

## API参考指南

2013年4月

API发行版本9.69

© 2013 Interactive Brokers LLC。保留所有权力。

Sun、Sun Microsystems、the Sun Logo和Java 是 Sun Microsystems, Inc在美国和其它国家的商标或注册商标。Excel、Windows和Visual Basic (VB)是Microsoft Corporation在美国和/或其它国家的商标或注册商标。

指南页面中显示的任何代码仅用于说明，不具有任何推荐意图。

# 目录

目录 .....	i
概论 .....	23
关于 APIs .....	24
通过 TWS 运行 API .....	25
通过 IB 网关运行 API .....	26
建议 .....	28
API 记录 .....	29
记录输入举例 .....	29
API 请求/服务器回应信息标识符 .....	30
历史数据限制 .....	31
违反请求间隔 .....	31
历史数据请求的有限长度和柱尺寸设置 .....	31
API 定单和 TWS 预防设置 .....	33
API 定单代号 .....	35
新定单举例 .....	35
修改定单举例 .....	35
获部分执行的定单状态 .....	36
交易平台 API 设置 .....	37
一般情况 .....	37
可信任的 IP 地址 .....	38
报价请求( RFQs) .....	39
使用 API 提交 RFQs .....	39
Delta-中立 RFQs .....	39
RFQ 举例 .....	39
请求实时指数溢价数据 .....	40

在PC计算机上卸载和重新安装TWS API软件 .....	41
故障排除常见问题 .....	42
Excel的DDE .....	43
Excel的DDE API入门 .....	44
下载API组件和表格 .....	45
配置交易平台支持API组件 .....	46
打开样本表格 .....	47
使用Excel的DDE样本表格 .....	48
代码页面 .....	49
使用代码页面 .....	49
代码页面工具栏按键 .....	51
基本定单页面 .....	52
下达定单 .....	53
下达组合定单 .....	54
支持的定单类型 .....	55
基本定单页面工具栏按钮 .....	55
扩展定单属性页面 .....	56
手动设置扩展定单属性 .....	57
应用扩展定单属性到单个定单或组定单 .....	57
扩展定单属性 .....	57
条件定单页面 .....	62
设置条件定单 .....	62
条件定单举例 .....	64
如果获执行的定单 .....	64
价格改变定单 .....	64
条件定单页面工具栏按钮 .....	65
高级定单页面 .....	66

下达括号定单 .....	66
下达波动率定单 .....	67
下达追踪止损限价单 .....	68
下达分段定单 .....	69
下达相对定单 .....	69
高级定单页面工具栏按钮 .....	70
开放定单页面 .....	70
查看开放定单 .....	71
开放定单标签工具栏按钮 .....	71
执行页面 .....	72
查看执行 .....	72
执行页面工具栏按钮 .....	73
执行报告页面 .....	73
运行执行报告 .....	74
账户页面 .....	74
使用账户页面 .....	75
账户页面工具栏按钮 .....	76
账户页面值 .....	76
投资组合页面 .....	80
查看你的投资组合 .....	81
投资组合页面工具栏按钮 .....	81
历史数据页面 .....	82
查看历史数据 .....	82
历史数据页面工具栏按钮 .....	84
历史数据页面寻求详细区域 .....	84
扫描仪页面 .....	86
开始市场扫描仪订阅 .....	87

市场扫描仪参数 .....	87
市场扫描仪页面工具栏按钮 .....	89
可用的市场扫描仪 .....	89
合约详细页面 .....	94
请求合约详细 .....	94
合约详细页面工具栏按钮 .....	94
债券合约详细页面 .....	95
请求债券合约详细 .....	96
债券合约详细页面工具栏按钮 .....	96
市场深度页面 .....	96
使用市场深度页面 .....	97
市场深度页面工具栏按钮 .....	98
顾问页面 .....	98
分配股票到一个单一管理的账户 .....	99
使用金融顾问( FA) 账户组和方法下达定单 .....	100
使用分配方案下达定单 .....	100
顾问页面工具栏按钮 .....	100
Excel的DDE API参考 .....	102
查看编码 .....	102
模块 .....	103
宏程序 .....	103
命名范围 .....	104
Excel的DDE句法 .....	105
Active X .....	117
使用ActiveX连接应用程序 .....	118
注册第三方ActiveX控制 .....	119
在64位Windows XP系统下运行ActiveX API .....	120

使用 Visual Basic 样本程序 .....	121
ActiveX 方法 .....	122
connect() .....	123
disconnect() .....	123
reqCurrentTime() .....	123
setServerLogLevel() .....	123
reqMktDataEx() .....	124
cancelMktData() .....	124
calculateImpliedVolatility() .....	124
cancelCalculateImpliedVolatility() .....	124
calculateOptionPrice() .....	125
cancelCalculateOptionPrice() .....	125
reqMarketDataType() .....	125
placeOrderEx() .....	125
cancelOrder() .....	126
reqOpenOrders() .....	126
reqAllOpenOrders() .....	126
reqAutoOpenOrders() .....	126
reqIds() .....	127
exerciseOptionsEx() .....	127
reqExecutionsEx() .....	127
reqContractDetailsEx() .....	128
reqMktDepthEx() .....	128
cancelMktDepth() .....	128
reqAccountUpdates() .....	128
reqNewsBulletins() .....	129
cancelNewsBulletins() .....	129

reqManagedAccts()	129
requestFA()	129
replaceFA()	129
reqAccountSummary()	130
cancelAccountSummary()	132
reqPositions()	132
cancelPositions()	132
reqHistoricalDataEx()	132
cancelHistoricalData()	134
reqScannerParameters()	134
reqScannerSubscriptionEx()	134
cancelScannerSubscription()	135
reqRealTimeBarsEx()	135
cancelRealTimeBars()	135
createComboLegList()	136
createContract()	136
createExecutionFilter()	136
createOrder()	136
createScannerSubscription()	136
createTagValueList	136
createUnderComp()	137
reqFundamentalData()	137
cancelFundamentalData()	137
ActiveX行动	138
connectionClosed()	139
currentTime()	139
errMsg()	139



tickPrice()	139
tickSize()	140
tickOptionComputation()	140
tickGeneric()	141
tickString()	141
tickEFP()	142
tickSnapshotEnd()	142
marketDataType()	143
orderStatus()	143
openOrderEx()	145
nextValidId()	145
permId()	145
updateAccountValue()	145
updatePortfolioEx()	148
updateAccountTime()	149
updateNewsBulletin()	149
contractDetailsEx()	150
contractDetailsEnd()	150
bondContractDetails()	150
execDetailsEx()	152
execDetailsEnd()	152
commissionReport()	152
updateMktDepth()	152
updateMktDepthL2()	153
managedAccounts()	154
receiveFA()	154
accountSummary()	154

accountSummaryEnd .....	156
position() .....	156
positionEnd() .....	157
historicalData() .....	157
scannerParameters() .....	157
scannerDataEx() .....	157
scannerDataEnd() .....	158
realtimeBar() .....	158
fundamentalData() .....	159
ActiveX COM对象 .....	160
IExecution .....	160
IExecutionFilter .....	161
ICommissionReport .....	162
IContract .....	162
IContractDetails .....	163
IComboLeg .....	165
IComboLegList .....	166
IOrder .....	166
OrderComboLeg .....	173
IOrderState .....	173
IScannerSubscription .....	174
ITagValueList .....	175
ITagValue .....	175
IUnderComp .....	175
ActiveX属性 .....	176
下达组合定单 .....	177
举例 .....	177

C++ .....	181
使用 TwsSocketClient.dll连接 TWS .....	182
使用 C++ TestSocketClient样本程序 .....	187
运行预先创建的样本程序 .....	187
从 Microsoft Visual Studio 2008运行 TestSocketClient程序 .....	187
Class EClientSocket功能 .....	188
EClientSocket() .....	189
eConnect() .....	189
eDisconnect() .....	189
isConnected() .....	189
reqCurrentTime() .....	189
serverVersion() .....	189
setLogLevel() .....	190
TwsConnectionTime() .....	190
checkMessages() .....	190
reqMktData() .....	190
cancelMktData() .....	191
calculateImpliedVolatility() .....	191
cancelCalculateImpliedVolatility() .....	191
calculateOptionPrice() .....	191
cancelCalculateOptionPrice() .....	192
reqMarketDataType() .....	192
placeOrder() .....	192
cancelOrder() .....	192
reqOpenOrders() .....	193
reqAllOpenOrders() .....	193
reqAutoOpenOrders() .....	193

reqIDs()	193
exerciseOptions()	193
reqAccountUpdates()	194
reqExecutions()	194
reqContractDetails()	194
reqMktDepth()	195
cancelMktDepth()	195
reqNewsBulletins()	195
cancelNewsBulletins()	195
reqManagedAccts()	195
requestFA()	196
replaceFA()	196
reqAccountSummary()	196
cancelAccountSummary()	199
reqPositions()	199
cancelPositions()	199
reqHistoricalData()	199
cancelHistoricalData()	201
reqScannerParameters()	201
reqScannerSubscription()	201
cancelScannerSubscription()	201
reqRealTimeBars()	202
cancelRealTimeBars()	202
reqFundamentalData()	202
cancelFundamentalData()	203
Class EWrapper功能	204
winError()	205

error()	205
connectionClosed()	205
currentTime()	205
tickPrice()	205
tickSize()	206
tickOptionComputation()	206
tickGeneric()	207
tickString()	207
tickEFP()	208
tickSnapshotEnd()	208
marketDataType()	209
orderStatus()	209
openOrder()	211
nextValidId()	211
updateAccountValue()	212
updatePortfolio()	212
updateAccountTime()	213
updateNewsBulletin()	213
contractDetails()	214
contractDetailsEnd()	214
bondContractDetails()	214
execDetails()	214
execDetailsEnd()	215
commissionReport	215
0updateMktDepth()	215
updateMktDepthL2()	215
managedAccounts()	216

receiveFA() .....	216
accountSummary() .....	217
accountSummaryEnd .....	219
position() .....	219
positionEnd() .....	220
historicalData() .....	220
scannerParameters() .....	220
scannerData() .....	220
scannerDataEnd() .....	221
realtimeBar() .....	221
fundamentalData() .....	222
SocketClient资源文件 .....	223
Execution .....	223
ExecutionFilter .....	224
Contract .....	224
ContractDetails .....	226
ComboLeg .....	228
Order .....	228
OrderState .....	235
ScannerSubscription .....	236
UnderComp .....	237
CommissionReport .....	237
下达一个组合定单 .....	239
举例 .....	239
Java .....	241
使用Java API连接TWS .....	242
运行Java测试客户端样本程序 .....	246

运行Java测试客户程序和Eclipse .....	250
Java测试客户概括 .....	252
Package .....	252
TestJavaClient Classes .....	252
Java API概括 .....	253
Java EClientSocket方法 .....	255
EClientSocket() .....	256
eConnect() .....	256
eDisconnect() .....	256
isConnected() .....	256
setServerLogLevel() .....	256
reqCurrentTime() .....	257
serverVersion() .....	257
TwsConnectionTime() .....	257
reqMktData() .....	257
cancelMktData() .....	257
calculateImpliedVolatility() .....	258
cancelCalculateImpliedVolatility() .....	258
calculateOptionPrice() .....	258
cancelCalculateOptionPrice() .....	258
reqMarketDataType() .....	259
placeOrder() .....	259
cancelOrder() .....	259
reqOpenOrders() .....	259
reqAllOpenOrders .....	260
reqAutoOpenOrders() .....	260
reqIDs() .....	260

exerciseOptions()	260
reqAccountUpdates()	261
reqExecutions()	261
reqContractDetails()	261
reqMktDepth()	262
cancelMktDepth()	262
reqNewsBulletins()	262
cancelNewsBulletins()	262
reqManagedAccts()	262
requestFA()	263
replaceFA()	263
reqAccountSummary()	263
cancelAccountSummary()	266
reqPositions()	266
cancelPositions()	266
reqScannerParameters()	266
reqScannerSubscription()	266
cancelScannerSubscription()	267
reqHistoricalData()	267
cancelHistoricalData()	269
reqRealTimeBars()	269
cancelRealTimeBars()	269
reqFundamentalData()	270
cancelFundamentalData()	270
Java EWrapper 方法	271
currentTime()	271
error()	272



connectionClosed()	272
tickPrice()	272
tickSize()	273
tickOptionComputation()	273
tickGeneric()	274
tickString()	274
tickEFP()	275
tickSnapshotEnd()	275
marketDataType()	276
orderStatus()	276
openOrder()	278
nextValidId()	278
updateAccountValue()	278
updatePortfolio()	279
updateAccountTime()	279
contractDetails()	280
contractDetailsEnd()	280
bondContractDetails()	280
execDetails()	280
execDetailsEnd()	281
commissionReport()	281
updateMktDepth()	281
updateMktDepthL2()	282
updateNewsBulletin()	282
managedAccounts()	283
receiveFA()	283
accountSummary()	283

accountSummaryEnd .....	286
position() .....	286
positionEnd() .....	287
historicalData() .....	287
scannerParameters() .....	287
scannerData() .....	287
scannerDataEnd() .....	288
realtimeBar() .....	288
fundamentalData() .....	289
Java SocketClient资源文件 .....	290
Execution .....	290
ExecutionFilter .....	291
CommissionReport .....	291
Contract .....	292
ContractDetails .....	293
ComboLeg .....	295
OrderComboLeg .....	296
Order .....	296
OrderState .....	303
ScannerSubscription .....	304
UnderComp .....	305
下达组合定单 .....	306
举例 .....	306
Java编码样本：合约参数 .....	309
如何确定期权合约 .....	309
如何确定期货合约 .....	310
如何确定股票 .....	310

顾问 .....	313
金融顾问定单和账户配置 .....	314
支持Excel DDE .....	315
其它API技术的支持 .....	316
改进的金融顾问执行报告 .....	317
账户组的分配方法 .....	318
EqualQuantity Method .....	318
NetLiq Method .....	318
AvailableEquity Method .....	318
PctChange Method .....	318
金融顾问API定单的Java编码样本 .....	320
为单一管理的账户下达定单 .....	320
为一个分配方案下达定单 .....	321
为账户组下达定单 .....	321
改变/更新分配信息 .....	322
ActiveX for Excel( Excel的ActiveX) .....	325
开始使用ActiveX for Excel API .....	326
下载API组件和表格 .....	326
在64位Windows XP系统下运行ActiveX for Excel API .....	326
打开样本表格 .....	327
使用ActiveX for Excel样本表格 .....	328
一般页面 .....	329
一般页面工具栏按钮 .....	331
代码页面 .....	331
使用代码页面 .....	332
代码页面工具栏按钮 .....	333
公告页面 .....	334

公告页面工具栏按钮 .....	335
市场深度页面 .....	335
使用市场深度页面 .....	336
市场深度页面工具栏按钮 .....	336
基本订单页面 .....	337
下达订单 .....	338
下达组合订单 .....	338
支持的订单类型 .....	340
基本订单页面工具栏按钮 .....	340
条件订单页面 .....	341
设置条件订单 .....	342
条件订单举例 .....	343
如果获执行的订单 .....	343
价格改变订单 .....	344
条件订单页面工具栏按钮 .....	345
高级订单页面 .....	345
下达括号单 .....	347
下达波动率订单 .....	347
下达跟踪止损限价单 .....	348
下达分段订单 .....	349
下达相对订单 .....	350
高级订单页面工具栏按钮 .....	350
扩展订单属性页面 .....	351
手动编程扩展订单属性 .....	352
对单个订单和订单组应用扩展订单属性 .....	352
开放订单页面 .....	353
查看开放订单 .....	354

开放定单标签工具栏 .....	355
账户页面 .....	355
使用账户页面 .....	356
账户页面工具栏按钮 .....	357
投资组合页面 .....	358
查看你的投资组合 .....	358
行使期权 .....	359
投资组合页面工具栏按钮 .....	359
执行页面 .....	359
查看执行 .....	360
执行页面工具栏按钮 .....	361
历史数据页面 .....	361
查看历史数据 .....	362
历史数据页面查询说明区域 .....	363
历史数据页面工具栏按钮 .....	365
合约详细页面 .....	365
请求合约详细 .....	366
合约详细页面工具栏按钮 .....	367
债券合约详细页面 .....	367
请求债券合约详细 .....	368
债券合约详细页面工具栏按钮 .....	369
实时柱页面 .....	369
实时柱页面工具栏按钮 .....	370
市场扫描仪页面 .....	371
开始市场扫描仪订阅 .....	372
市场扫描仪参数 .....	372
市场扫描仪页面工具栏按钮 .....	373

基本面页面 .....	373
基本面页面工具栏按钮 .....	375
顾问页面 .....	375
分配股票到一个单个账户 .....	376
使用FA账户组和方法下达定单 .....	377
使用一个分配方案下达定单 .....	377
顾问页面工具栏按钮 .....	378
记录页面 .....	378
POSIX .....	381
在 Windows 计算机上运行 POSIX 客户端 .....	382
参考表 .....	383
API消息编码 .....	384
出错编码 .....	384
系统信息编码。 .....	395
警告信息编码 .....	395
跳动类型 .....	397
通用跳动类型 .....	400
使用 SHORTABLE Tick( 可卖空跳动) .....	401
FUNDAMENTAL_RATIOS tickType( 基本面比率跳动类型) 的标签值 .....	402
IBDividends跳动举例 .....	408
举例 .....	408
RTVolume .....	408
定单类型和IBAlgos( IB算法) .....	410
支持的定单类型 .....	410
IBAlgo参数 .....	412
到达价格(ArrivalPx) .....	413
黑冰(DarkIce) .....	414

交易量百分比(PctVol) .....	415
时间加权平均价格(Twap) .....	416
交易量加权平均价格(Vwap) .....	416
平衡影响和风险(BalanceImpactRisk) .....	416
最小化影响(MinImpact) .....	418
积累/分配(AD) .....	418
CSFB算法参数 .....	420
Crossfinder (CROS) .....	421
Crossfinder (CROS) Java编码举例 .....	421
Float (FLT) .....	423
Float (FLT) Java编码举例 .....	423
Guerilla (GRRL) .....	424
Guerilla (GRRL) Java编码举例 .....	424
Work It IW (INIW) .....	425
Work It IW (INIW) Java编码举例 .....	426
Work It (INLN) .....	427
Work It (INLN) Java编码举例 .....	427
Pathfinder (PTHF) .....	428
Pathfinder (PTHF) Java编码举例 .....	428
Reserve (RSRV) .....	430
Reserve (RSRV) Java编码举例 .....	430
Strike (SNPR) .....	431
Strike (SNPR) Java编码举例 .....	431
10B 18 (TENB) Java编码举例 .....	432
10B 18 (TENB) Java编码举例 .....	432
Tex (TEX) .....	433
Tex (TEX) Java编码举例 .....	434

TWAP (TWAP) .....	435
TWAP (TWAP) Java编码举例 .....	435
VWAP (VWAP) .....	436
VWAP (VWAP) Java编码举例 .....	436
扩展定单属性 .....	439
可用的市场扫描仪 .....	443
市场扫描仪的产品和地点代码 .....	447
支持的时区 .....	448
智能组合传递 .....	449
索引 .....	451



## 概论

这一章提供APIs( 应用程序界面) 的概述，包括以下题目：

- [关于 APIs](#)
- [通过交易平台\( TWS\) 运行API](#)
- [通过盈透网关\( IB Gateway\) 运行API](#)
- [建议](#)
- [API记录](#)
- [历史数据限制](#)
- [API定单和TWS预防设置](#)
- [API定单代号](#)
- [获部分执行的定单状态](#)
- [交易平台API设置](#)
- [报价请求\( RFQs\)](#)
- [请求实时指数溢价数据](#)
- [在PC计算机上删除和重新安装交易平台API软件](#)

## 关于 APIs

你可以选择我们提供的几种 APIs 与你的系统连接来交易你的 IB 账户。API 允许你通过 TWS 或者 IB 网关进行连接。通过 TWS 连接要求你运行应用程序，也可以让你测试并确认你的 API 定单工作是否正常。通过 IB 网关连接允许你使用 API 而无需运行一个大容量的图形用户界面 (GUI) 应用程序，但不提供用于测试和确认 API 活动的界面。

不管你使用那种连接方式，API 允许你完成：

查看特定功能的句法，请参阅本指南的 [Excel 的 DDE](#)，[ActiveX](#)，[C++](#) 或 [Java](#) 标题。不具备程序设计知识的客户应该从 Excel 的 DDE 部分开始，你可以使用常用的 Excel® 表格通过 API 和 TWS 连接。

**Note:** API 内容是为具有经验的程序员撰写，对非技术用户提供了有限的指导。

设计和测试你的 API 程序，我们建议你使用举例应用程序与 TWS 进行连接。一旦 API 工作的如你设计的一样令你满意，如果你希望的话，你可以使用低容量 GUI 的 IB 网关来连接。

### 使用 API 组件和查看样本源代码和表格文件

1. 通过 IB 网站安装或更新最新的 API 和样本文件：
  - 从 IB 网站菜单，点击 **交易技术 > API 方案**。
  - 点击 **盈透 API**，然后点击 **API 软件** 按钮。
  - 在 API 软件窗口中，点击你希望安装的相应的正式版或 Beta 版 API 版本按键。安装程序即被下载到你的计算机中。
  - 运行下载的安装程序来安装 API 软件到你的计算机中。
2. 登录进入交易平台 (TWS) 并配置用于支持 API。配置是通过 TWS 的全局配置 > API > 设置来进行的。
3. 使用样本应用程序来学习如何寻求市场数据、提交定单，等。
4. 自定义样本应用程序来适合你的需求，或使用描述的句法和功能创建你自己的应用程序。

## 通过TWS运行API

通过TWS运行API，你必须使系统一直处于运行状态，必须具有可使用任何API组件的配置。

### 启动通过TWS连接API

1. 登录TWS。
2. 在**编辑**菜单上，选择**全局配置**。
3. 在左侧面板中选择**API**，然后点击**设置**。
4. 在右侧面板中，点击**启动ActiveX和套接字客户端( Socket Clients)**（ActiveX、C++和Java API连接），和/或**启动DDE客户端**（仅用于Excel的DDE API连接）为相应的API连接配置TWS。你必须使这些设置得到启动后方才可以**通过TWS连接API**。

**Note:** 具有不同客户代码的多个API客户可以通过相同的计算机进入TWS的单个实例。除了Excel的DDE外，API应用程序不需要像TWS一样在同一个计算机上运行。

有关所有交易平台的API设置的完整描述，请参阅[TWS用户指南](#)。

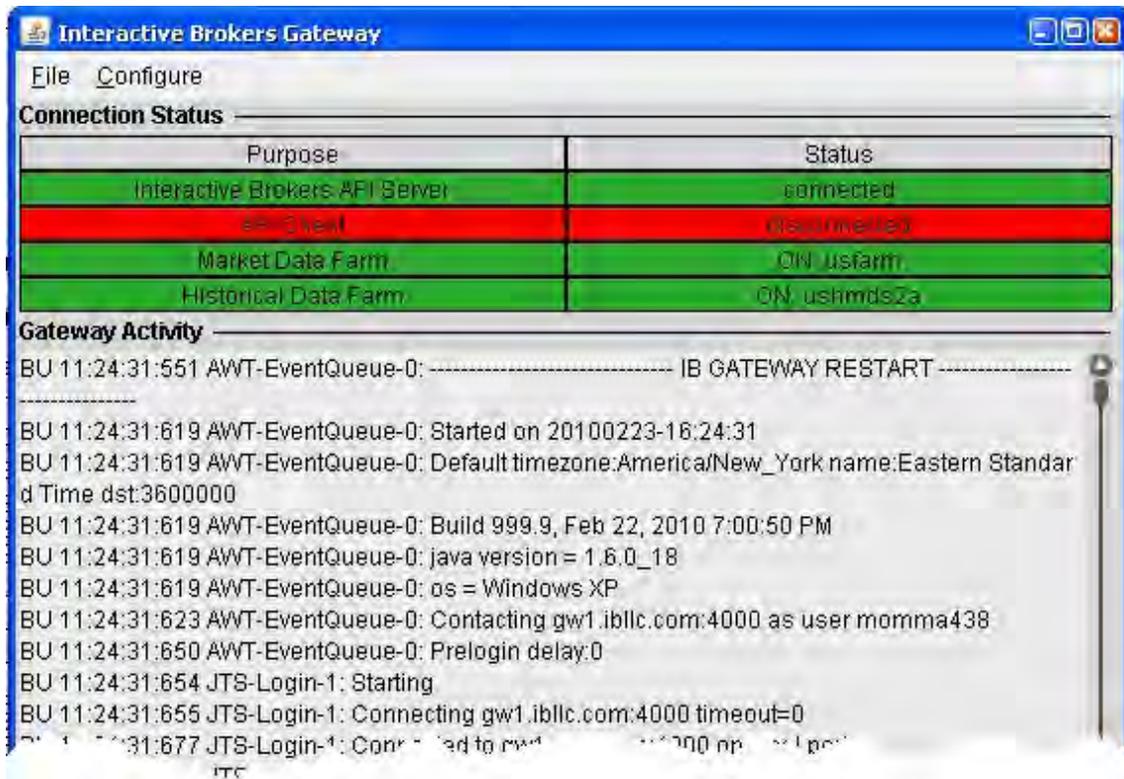
## 通过IB网关运行API

做为通过TWS的一种替代，IB网关提供了低容量要求连接API到IB交易系统。网关使用大约比TWS低40%的系统资源。但是，网关的GUI资源比较低，所以你不能够像运行TWS一样查看API活动。



### 登录进入IB网关

1. 从IB网站的**登录**菜单，选择**IB网关**。
2. 选择API按钮。
3. 使用你的IB用户名和密码登录，就如同你登录TWS一样。
4. 点击**登录**。盈透证券网关框打开，显示连接状态和网关活动。



在与API连接时，你必须使IB网关处于运行状态。

## 建议

在你使用我们的TWS API创建你自己客户化的交易应用程序之前，你应该考虑以下重要的事项：

- 用Conid下达定单 — 当你用conid下单时，你必须提供 conid和交易所。如果你在用conid下单时提供了额外的区域，定单也许不会工作。
- 定单代码 — 你下达的每个定单必须拥有一个独特的定单代码。我们建议你按递增的方式创建自己的定单代码以避免与从你API应用程序下达的定单发生冲突。解决定单代码的问题，请点击API上的**重新设置API定单代码顺序**按钮 — TWS全局配置的设置面板。
- 请使用IB模拟交易账户测试你API应用程序以发现和避免任何错误。你可用通过账户管理申请一个模拟交易账户。
- 尽管API支持使用同一个登录在单一TWS运行下的同时8个API连接，我们还是建议你避免此种情形。如果可能的话，对应用程序使用单一API连接以避免超负荷运行。

## API记录

当处理客户端的请求时(包括系统和API客户)，将会记录某些信息到位于安装目录中的'log.txt'记录文件。该文件的目的是通过提供程序的状态信息，从而在问题发生前帮助解决问题。

API客户可以通过设定记录级别来指定记录输入的详细程度。记录级别是：

- 1 = 系统
- 2 = 出错
- 3 = 警告
- 4 = 信息
- 5 = 详细

**Note:** 设置记录级别为5将增加系统的运行负荷。你仅在试图解决某个问题时使用记录级别5。

API请求的记录输入的格式：

**[ClientID:ClientVersion:ServerVersion:ClientType:Request:Response:Version:LogEntryType]**

这里：

- **ClientID** 是连接时使用的clientId。
- **ClientVersion** 确认客户的请求流量(内部使用)。
- **ServerVersion** 确认服务器的响应流量(内部使用)。
- **ClientType** 是API连接类型：DDE = 0、Socket = 1。
- **Request:** 如果大于0，表示记录输入是API客户请求的结果。显示的数字是在下列的“Outgoing Request Identifiers(发出请求标识符)”部分中的请求代号。
- **Response:** 如果大于0，表示记录输入是服务器对API的响应的结果。显示的数字是在下列的“Incoming Request Identifiers(接收请求标识符)”部分中的响应代号。
- **Version** 确认请求或响应信息的版本。当信息格式变化时，版本也改变。
- **LogEntryLevel** 确认记录输入的类型(例如，上面列出的记录级别)

### 记录输入举例

```
[0:9:9:1:1:0:3:DET]Socket request -
[3;52;IBM;STK>null;0.0;2;SMART>null>null]
```

从该举例中，我们可以知道一个具有clientId=0连接的套接字客户端提交了一个市场数据的请求。市场数据请求的版本(版本3)表示数据应该已经被发送了。

**API请求/服务器回应信息标识符**

发出请求标识符	传入回应标识符
1 = 请求市场数据	代码价格
2 = 取消市场数据	2 = 代码尺寸
3 = 下定单	3 = 定单状态
4 = 取消定单	4 = 错误信息
5 = 请求开放定单	5 = 开放定单
6 = 请求账户数据	6 = 账户值
7 = 请求执行报告	7 = 投资组合值
8 = 请求下一个定单代号	8 = 账户更新时间
9 = 请求合约详细	9 = 下一个有效定单代号
10 = 请求市场深度	10 = 合约详细
11 = 取消市场深度	11 = 执行报告详细
12 = 请求新闻公告	12 = NYSE Open Book行输入
13 = 取消新闻公告	13 = 二级报价行输入
14 = 设置服务器记录级别	14 = 新闻公告

**Note:** 上述信息，以及各种请求/回应信息版本，可以在API安装时提供的EClientSocket实施文件中找到。



## 历史数据限制

请求历史数据有以下限制：

- 请求的历史数据是每30秒或低于的柱尺寸只能包含过去6个月的数据。
- 每个请求返回的数据被限制在2000个柱以内的长度和柱尺寸值( 每次请求为2000个柱) 。
- 根据实时市场数据线的数量，历史数据请求可以为一个日历年或更多的数据。

市场数据线数量	历史数据请求限制
低于 499	一年
500 - 749	两年
750 - 999	三年
1000	四年

市场数据线可以根据月佣金额、股权额和报价助推器的订阅而增加。

- 关于市场数据是如何受佣金和股权影响的信息，请参阅我们网站上的[市场数据](#)页面底部的倒数第二个注解( 带有举例的长注解) 。
- 有关报价助推器的详细信息，请参阅我网站上的[报价助推器](#)页面。你通过账户管理中的市场数据订阅页面订阅报价助推器。

### 违反请求间隔

所有的API技术均支持历史数据请求。但是，在短时间内请求同样的历史数据可以引起后端运行的超额负载，从而引起违反请求间隔。表示出现间隔违反的错误码和信息是：

162 - 历史市场数据服务错误信息：违反历史数据请求间隔

以下条件可以引起违反请求间隔：

- 在15秒钟内请求相同的历史数据；
- 在两秒钟内对相同合约、交易所和Tick类型提出6次或以上的历史数据请求。

还有，请查看当请求历史数据时下列限制：

- 不要在任何10分钟区间内提出超过60次历史数据请求。

**Note:** 关于历史数据请求的更多信息，请查看Excel的DDE章节的[查看历史数据](#)，和ActiveX章节的[reqHistoricalDataEx\(\)](#)，C++章节的[reqHistoricalData\(\)](#)，以及Java章节的[reqHistoricalData\(\)](#)。

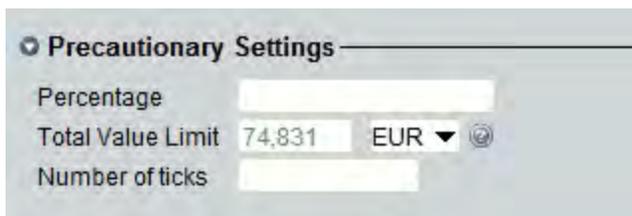
### 历史数据请求的有限长度和柱尺寸设置

下面表格列出API历史数据请求的有效长度和柱尺寸设置。请注意这些仅为参考。

长度	柱尺寸
1年	1天
6个月	1天
3个月	1天
1个月	1天, 1小时
1周	1天, 1小时, 30分钟, 15分钟
2天	1小时, 30分钟, 15分钟, 3分钟, 2分钟, 1分钟
1天	1小时, 30分钟, 15分钟, 5分钟, 3分钟, 2分钟, 1分钟, 30秒
14400秒( 4小时)	1小时, 30分钟, 15分钟, 5分钟, 3分钟, 2分钟, 1分钟, 30秒, 15秒
7200秒( 2小时)	1小时, 30分钟, 15分钟, 5分钟, 3分钟, 2分钟, 1分钟, 30秒, 15秒, 5秒
3600秒( 1小时)	15分钟, 5分钟, 3分钟, 2分钟, 1分钟, 30秒, 15秒, 5秒
1800秒( 30分钟)	15分钟, 5分钟, 3分钟, 2分钟, 1分钟, 30秒, 15秒, 5秒, 1秒
960秒( 15分钟)	5分钟, 3分钟, 2分钟, 1分钟, 30秒, 15秒, 5秒, 1秒
300秒( 5分钟)	3分钟, 2分钟, 1分钟, 30秒, 15秒, 5秒, 1秒
60秒( 1分钟)	30秒, 15秒, 5秒, 1秒

## API定单和TWS预防设置

按默认状态，交易平台包括的预防设置为TWS配置页面上定单预设的一部分。预防设置是一种安全检查，包括百分比、尺寸限制。总价值和跳动数目( number of ticks) 。预防设置可以在TWS中针对大部分交易产品类型( 股票、期权、或其它) 或特定代码进行修改。

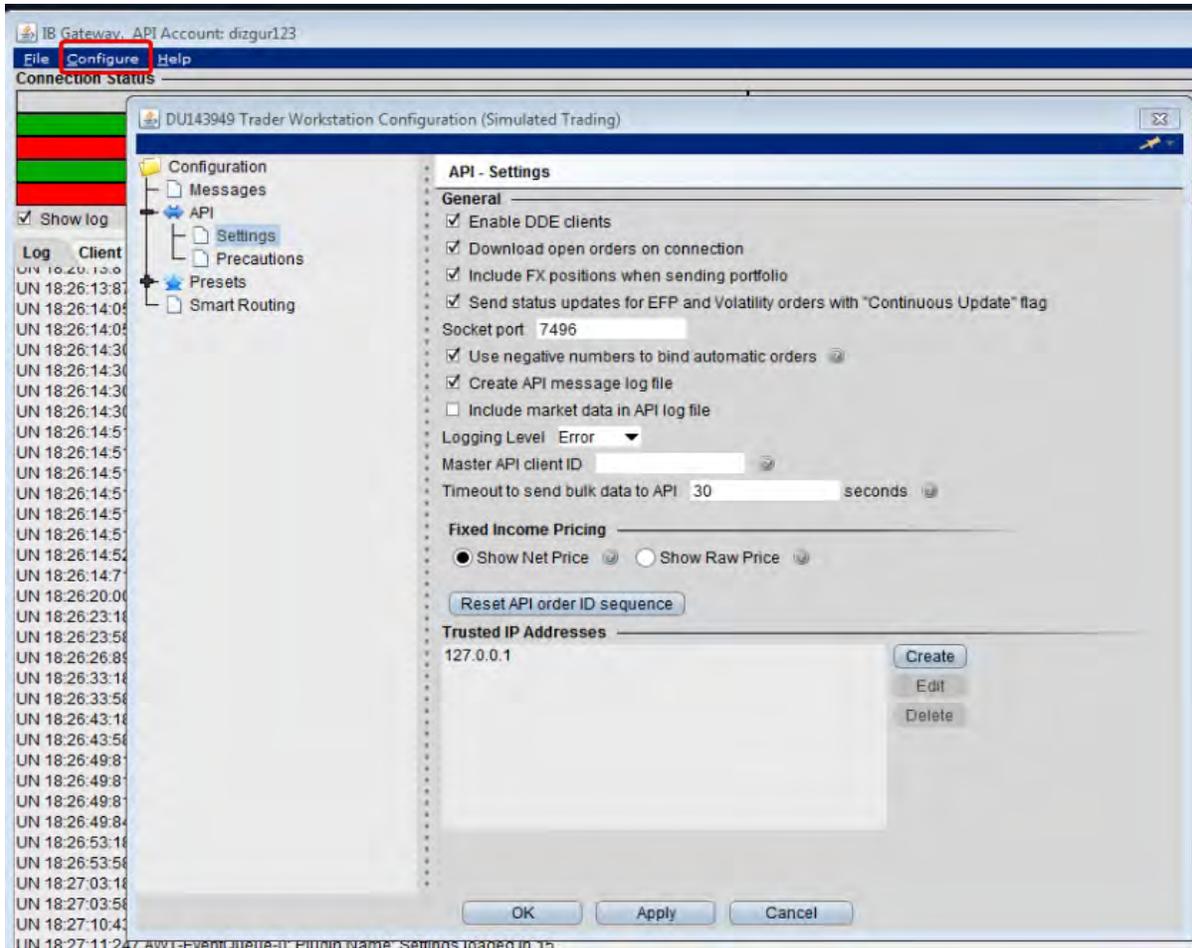


如果你的API定单违反了这些设置，你将收到一个错误信息。例如，定单尺寸的默认预防设置是500。如果你下达的定单为1000股，你将收到一个错误信息指明特定的尺寸违反了默认定单设置规定的约束。TWS预防设置适用于所有通过API技术下达的API定单。

你可以通过下列操作之一来取代预防设置：

在TWS中：

- 在配置菜单中，选择API，然后选择所有API设置。选择 *Bypass Order Precautions for API Orders*( 绕过API定单的定单预防) 的选择框。并点击 **OK**。所有你的API定单将忽略TWS中的预防设置。
- 在定单预设中，对希望交易的产品类型和代码输入较高的预防设置限制。在配置菜单中，选择定单，然后选择定单预设。在左侧中选择产品类型或代码，在页面的预防设置部分输入希望的限制，然后点击 **OK**。



在IB网关中：

- 从配置菜单中，选择设置。选择 *Bypass Order Precautions for API Orders* ( 绕过API定单的定单预防) 的选择框，并点击 **OK**。所有你的API定单将忽略你在TWS中设置的预防设置。

## API定单代号

当你使用API下达一个新的定单时，定单代号数字必须大于前一个使用的数字。例如，如果你用定单代号11下达一个定单，则下一个定单的定单代号应该至少为12。所以当你下达一个新定单时，定代号必须大于之前使用的定单代号数字。

### 新定单举例

在这个举例中，用户将对IBM股票下达两个定单。第一个定单是买进200股的买单，设定限价价格为\$85.25。第二个定单将为限价价格设定为\$84.25的100股的卖单。

在这个举例中，第一个定单的定单代号设定为1：

```
.placeOrder(1, IBM, BUY, $85.25, 200...)
```

现在，用户可以下达第二个定单。分配给这个定单的定单代号是2：

```
.placeOrder(2, IBM, SELL, $84.25, 100...)
```

### 修改定单举例

使用API修改定单，需要使用相同的定单代号重新提交你希望修改的定单，但使用依据需要修改的价格或数量。仅某些区域，比如价格或数量可以使用这个方法来更改。如果你想改变定单类型或行动，你将需要取消定单然后重新提交一个新的定单。

在这个例子中，用户初始决定买进100股，并设置限价价格为\$85.25。然后，客户决定修改相同的定单，将限价价格改为\$86.25。注意，第一个定单分配的定单代号是3：

```
.placeOrder(3, IBM, BUY, $85.25, 100...)
```

通过召回相同的.placeOrder方法和使用相同的定单代号3，你现在可以将这个定单的限价价格修改为\$86.25

```
.placeOrder(3, IBM, BUY, $86.25, 100...)
```

## 获部分执行的定单状态

以下举例说明当一个定单获部分执行时，`orderStatus()` (定单状态) 将如何表现。

### 部分执行举例

你下达一份1000股的XYZ股票定单。在1000股被全部执行之前共有四次不同的执行。第一个部分执行为200股，然后第二个部分执行为200股。第三个部分执行为另一个200股，最终将最后部分执行400股。

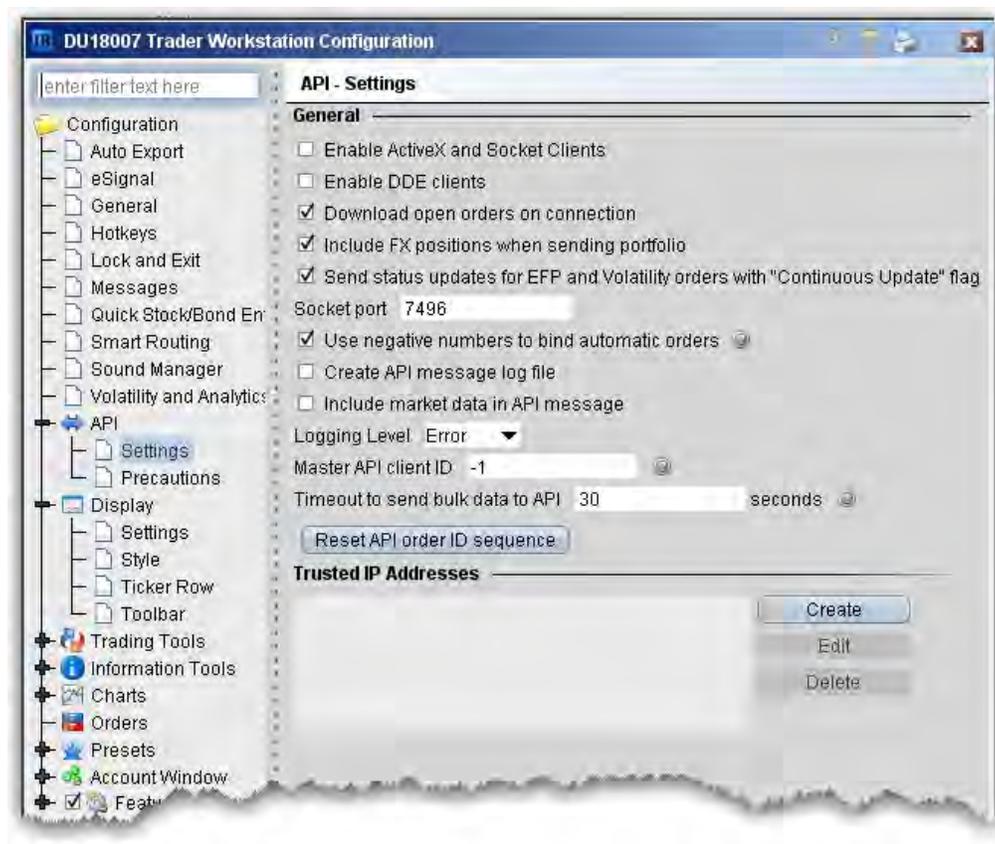
- 第一执行： 200
- 第二执行： 200
- 第三执行： 200
- 第四执行： 400

在`orderStatus()`活动中，根据这个举例API将连续收到状态信息为：

执行	状态信息
第一个执行	状态 = 提交的执行数量 = 200 剩余数量 = 800
第二个执行	状态 = 提交的执行数量 = 400 剩余数量 = 600
第三个执行	状态 = 提交的执行数量 = 600 剩余数量 = 400
第四次执行	状态 = 提交的执行数量 = 1000 剩余数量 = 0

## 交易平台 API 设置

除了配置交易平台( TWS) 与API的接通外, 你还可以在TWS中配置多个与API有关的设置。



### 在TWS中配置API设置

1. 在TWS中, 选择**编辑**菜单, 然后选择**全局配置**。
2. 点击左侧面板中的**API**, 然后选择**设置**。
3. 按要求配置API设置。有关描述如下。

**Note:** 除了DDE以外, API应用程序不需要在运行应用程序的同一个计算机上运行。

### 一般情况

- **启动 Active X 和 Socket Clients( 套接字客户端)** — 选取启用整合使用 ActiveX 或 socket clients 包括 Java 和 C++。
- **启用 DDE 用户端** - 选取启用整合 TWS 与通过 DDE 的 TWS。
- **连接时下载开放定单** — 如果你不希望在你与你的 API 连接时下载所有开放定单, 可取消选取 (uncheck)。

- **当发送投资组合时包括外汇头寸** — 如果你将包括外汇头寸功能启动了，当给API客户发送投资组合更新时，将包括所有外汇头寸。取消这个框的选取，如果你不希望在投资组合更新发送时将外汇头寸发给API客户。
- **发送EFP( 期货换现货) 和波动率定单的状态更新并带有”连续更新“标签** — 如果你启动了连续更新EFP或波动率定单，所有更新将按默认发送给API客户。取消选取如果你不希望这些更新从TWS发送给API客户。
- **使用负数绑定自动定单** — 如果选取了，所有通过reqOpenOrdersif或reqAutoOpenOrders指令，或通过系统生成的定单( 比如 波动率对冲定单) 与API客户绑定的自动定单将被分配给负的API定单代号。否则，这些定单将被分配给递增的API定单代号。在可能情况下，波动率对冲定单的定单代号将为”母API定单代号+1“。
- **创建API信息记录文件** — 选取来创建一个信息记录文件。使用记录级别选择器来定义记录中的详细程度。
- **在API信息中包括市场数据** — 在API记录文件中显示市场数据。
- **套接字端口( Socket port)** — 输入一个允许你进入多个TWS实例或在一个单一主机运行的IB网关的套接字端口数字。通过给每个TWS或IB网关实例指定一个独特的套接字端口数字，一个单一ActiveX或套接字API客户将可以进入每一个实例。这个不适用于DDE客户端。
- **记录级别** — 为API文字记录设定记录详细级别。系统将给出最一般的记录级别；详细的将给出最详细的级别。注意，详细记录级别使用更多的计算资源并可能引起运行能力的降低。
- **主API客户代号** — 具有特定客户代号的API客户将会收到所有的定单，即使是那些由其它API客户下达的定单。这和具有客户代号为“0”的将收到来自TWS图形用户界面(GUI)所有定单是不同的。
- **发送批量数据到API的时间限制** — 如果数据不能在足够快的时间内发送时，切断连接API客户之前的以秒为单位的TWS等待时间。

### 可信任的IP地址

如果你通过信任的IP地址与API连接，连接即不会有问题。否则，你将收到一个确认的信息询问是否你希望执行连接。

- 点击**创建**添加一个新的信任的IP地址到列表中。
- 点击**编辑**来修改选择的地址。
- 点击**删除**来取消选择的地址。



## 报价请求 ( RFQs )

从IB期权交易服务台发出RFQs允许你对来自IB附属公司Timber Hill的大额定单获得报价。报价包括美国股票和指数期权、和主要欧洲和亚洲指数期权和组合。有关完整的列表，请联系[盈透期权交易服务台](#)。

从盈透期权交易服务台发出RFQs仅适用于那些可使用这些特定区域的用户。如果你感兴趣参加，请联系盈透期权交易服务台。

### 使用API提交RFQs

通过发送定单类型为QUOTE的定单提交RFQ。相对应的，具有和RFQ的orderId( 定单代号) 相匹配的tickerId( 股票代码) 的tickPrice()/tickSize() 即被调用。使用具有相对较高数字的orderId以避免冲突。非RFQ的tickerIds需要额外的空间。

收到RFQ的市场数据，直到用户取消RFQ或者RFQ被服务器取消为止。服务器通常在RFQ过时后( 大约1分钟) 将其取消，或如果在RFQ请求无效和/或为不支持的交易产品时将其取消。

### Delta-中立RFQs

创建组合定单来提交Delta-中立RFQs，即使是单一合约也必须进行对冲，填写和附加一个UnderComp结构到合约的underComp区域。

在UnderComp结构中，你必须指定对冲合约的conId。价格和delta区域可以是空的(0)。

在接受Delta-中立DN RFQ后，服务器将发出一个具有UnderComp结构的deltaNeutralValidation() 信息。如果原始请求的delta和价格区域是空的，确认将包括服务器中当前的值。这些值在处理RFQ时是被锁定的，并在RFQ被取消前一致保持锁定。

### RFQ举例

欲了解更多关于使用交易平台API提交RFQs，请参阅API软件发行要点9.6版本中的RFQ举例。举例位于你的API软件下载文件夹的举例/rfq文件夹中。TSampleRFQ.java举例运行一个简单程序过程演示如何为下列提交RFQ's:

- 欧洲股票
- 美国期货
- 美国股票期权
- 欧洲股票期权
- 指数期权的日历差价( Delta-中立)
- 美国股票期权( Delta-中立)
- 美国指数期权( Delta-中立)
- 欧洲指数期权( Delta-中立)

## 请求实时指数溢价数据

你可以使用以下 APIs 和 API 样本应用程序请求实时指数溢价市场数据：

- ActiveX ( 包括 ActiveX API 样本应用程序)
- C++ ( 包括 C++ API 样本应用程序)
- Java ( 包括 Java API 样本应用程序)
- 表格的 ActiveX

请求实时指数溢价数据，你必须完成以下内容：

- 指定代码、证券类型和交易所。
- 例如，INDU、IND 和 NYSE 将给出道琼工业平均值的指数溢价数据。
- 交易所必须匹配你请求数据的指数。
- 你必须使用通用的 Tick 类型 162 ( 用于指数期货溢价) 。

## 在PC计算机上卸载和重新安装TWS API软件

如果你在PC平台下运行TWS API软件时遇到问题，你可以卸载和重新安装API软件。

**Note:** 这个过程通常只在解决最极端的API问题时才使用。

### 在PC计算机上卸载和重新安装TWS API软件

1. 打开窗口控制面板，然后打开添加或删除程序。
2. 从已安装程序列表中选择TWS互通性组件，然后点击更改/删除。
3. 选择自动，然后点击下一个来卸载TWS API软件。
4. 在微软浏览器( Windows Explorer) 中，从Windows\system32文件夹中删除文件TwsSocketClient.dll。
5. 程序启动你的计算机。
6. 重新安装TWS API软件。

## 故障排除常见问题

以下是一些常见的API故障/问题和方案/答案。

### 我接到的错误信息是“用于表格整合的DDEDLL.dll文件缺失或过时。” - (表格的DDE API)

这个错误信息是在当你试图在一个64位的PC计算机上使用32位Java执行来启动交易平台(TWS)时出现的。解决这个问题，使用64位的Java可执行文件启动TWS。

### 使用64位的Java可执行文件启动独立TWS

1. 在你的PC桌面上，右键点击TWS的快捷链接并从弹出的菜单中选择特性( *Properties* )。
2. 在特性对话框中，点击快捷链接( *Shortcut* ) 标签，然后点击第一个目标区域。
3. 通常目标区域出现的文字显示一个以“C:\Windows\system32\javaw.exe ...”开始的长的路径。改变路径中的某个部分，比如将system32改为sysWOW64。The sysWOW64文件夹包括64位Java可执行文件(javaw.exe)。当你完成后，点击OK。
4. 如果目标路径已经包括“sysWOW64，”。将其改变为system32，然后点击采用。然后再将system32改变回sysWOW64并点击采用，然后点击OK。
5. 启动TWS。

基于浏览器的TWS可以通过使用64位Java可执行文件浏览器启动，Internet Explorer是64位浏览器( 当此API参考指南版本发布时，Firefox和Google Chrome不具备64位版本)。某些情形下，当两种Java可执行文件都被安装的情况下( 32位和64位)，64位浏览器可能仍使用32位Java。此时，有必要卸载Java32位可执行文件，并甚至可重新安装Java64位可执行文件。

### 是否可在没订阅市场数据的情况下获得历史数据？

不可以。但是，某些市场数据订阅是免费并通过默认启动的，你可以获得这些免费订阅市场数据的历史数据。

### 是否可以通过TWS中创建的API定单获取数据？

可以。

### 是否可以修改通过TWS创建的API定单？

可以，包括Active X、Java和C++ APIs ( Excel的DDE API不可以)。如修改这些定单，你必须使用orderID，而非permID。如果定单是通过TWS创建的，orderID将为负数。

### 当使用Excel的DDE API时，为什么我收到的市场数据单元中的数据均为零？

这是因为Excel DDE表格还没有和TWS链接好。解决的方法是重新启动Excel和TWS。

### Excel的DDE API是与PC机的64位Excel兼容吗？

不是。

### API可以使用条件定单吗？

从条件定单将被盈透证券的服务器持有并监控的角度讲，从API发出的条件定单是不可用的。交易人可以在自己的机器上监控条件并在条件满足时为执行发送定单。

TWS确实提供通过盈透服务器持有并监控的条件定单，尽管条件的范围仅限为几种变量，如价格和交易量。

## Excel的DDE

这一章描述Excel的DDE API，包括以下内容：

- [Excel的DDE API入门](#)
- [使用Excel的DDE样本表格](#)
- [Excel的DDE API参考](#)

DDE是Dynamic DataExchange(动态数据交换)的缩写，是微软创建的通讯方式，允许多个应用程序同时运行来交换数据和指令。我们使用这个规则将Excel与你运行的TWS版本或IB网关相连接，从而允许你使用Excel表格查看实时市场数据(包括市场深度数据)管理订单、监控你的执行和账户信息。

下面图示显示Excel DDE API样本表格中的代码页面。

The screenshot shows an Excel spreadsheet with a table of market data. The table has the following columns: Symbol, Type, Expiry, Strike, Vol, Manager, Exchange, Primary Exchange, Currency, Open Leg, Ch, Bid Imp, Vol, Bid Dots, Bid Size, Bid Price, Ask Price, Ask Size. The data is categorized into Stocks, Options, Index Futures, Interest Rate Futures, and Energy Futures.

Symbol	Type	Expiry	Strike	Vol	Manager	Exchange	Primary Exchange	Currency	Open Leg	Ch	Bid Imp	Vol	Bid Dots	Bid Size	Bid Price	Ask Price	Ask Size
<b>Stocks</b>																	
MSFT	STK					SMART	ISLAND	USD			0		99	26.62	26.68	260	
YHOO2	STK					SMART	ISLAND	USD			0		184	20.12	20.13	135	
GE	STK					SMART	ARCA	USD			0		107	28.17	28.18	16	
GOOGL	STK					SMART	ISLAND	USD			0		-1	482.47	482.76	6	
GOOGL	STK					SMART	ARCA	USD			0		794	45.21	45.22	545	
IBM	STK					NYSE	ARCA	USD			0		1	127.28	127.37	1	
MOT	STK					SMART		USD			0		1	7.41	7.42	164	
<b>Options</b>																	
GOOGL	OPT	200812	273	C	-33	SMART		USD			0	0.3755722	0.998132	40	215.8	217.8	85
GOOGL	OPT	200812	273	P	-33	SMART		USD			0	0.4994851	0.3174134	59	1.85	1.84	18
IBM	OPT	200810	135	P	-33	NYSE		USD			0	0.2359493	0.936139	181	10.3	10.5	156
IBM	OPT	200810	79	C	-33	SMART		USD			0	-	0	447	58.8	58.2	454
MSFT	OPT	200810	29	C	-33	SMART		USD			0	0.2538875	0.8891857	8396	6.65	6.25	4745
MSFT	OPT	200810	29	P	-33	SMART		USD			0	0.3934492	0.9539312	9447	0.15	0.15	5264
<b>Index Futures</b>																	
Z	FUT	200812					IBFF	GBP									
SPX	FUT	200812				252	CME	USD									
SPX	FUT	200812				252	GLOBAL	USD									
<b>Interest Rate Futures</b>																	
ZB	FUT	200812				-933	CBOT	USD									
ZB	FUT	200812				-933	ECBOT	USD									
<b>Energy Futures</b>																	

## Excel的DDE API入门

我们创建了一个DDE连接Excel表格的样本，TwsDde.xls，你可以使用这个样本和你的TWS创建一个自定义的Excel应用程序。使用Excel的DDE API非常容易：

- [下载API组件](#)和Excel表格样本。
- 确定DDE用户端已经启动，或是通过交易平台或IB网关，[这里](#)描述的是使用TWS，或IB网关处于运行状态中。
- [打开表格](#)并开始使用Excel的DDE API。

当前的样本表格由几个包含数据和活动按键的页面组成，可以让你方便地获得市场数据、发送定单和查看你的活动。

## 下载API组件和表格

建议使用我们提供的样本Excel表格做为起点创建你自己的Excel的Excel API。遵循以下步骤下载样本表格。

### 安装样本DDE表格

1. 在[IB网站](#)上，从[交易技术](#)菜单选择API方案。
2. 点击IB API图标，在API软件页面上，找到适合你使用的操作系统的相应栏，点击下载最新的版本。

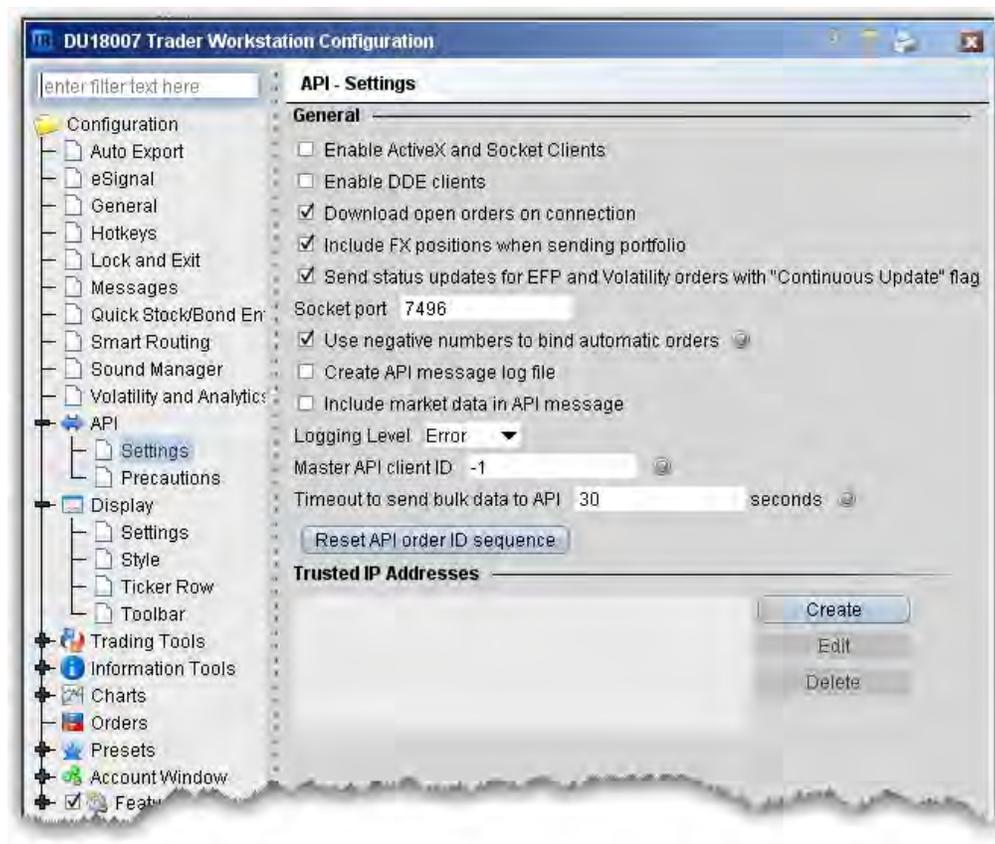
**Note:** Windows用户可以下载API的Beta测试版本，或选择下载前一版本。

3. 保存安装程序到你的计算机中，依据需要选择不同的路径。点击**保存**。注意，API安装文件的命名是使用API的版本；例如，*InstallAX\_960*。
4. 关闭你正在运行的任何版本的TWS、IB网关和Excel。
5. 在你的计算机中找出你刚刚保存的API安装程序，然后双击文件开始API安装。
6. 跟随安装向导指令。按默认设置，样本DDE表格被保存在C:\Jts\Excel\TwsDde.xls。

**Note:** 在你[使用表格](#)之前，你必须使TWS处于运行状态，并配置好支持DDE API。你还可以通过[IB网关](#)运行样本，但我们建议你以运行TWS开始。

## 配置交易平台支持API组件

你必须使你的系统处于运行状态才能使用任何API组件。



### 配置应用程序支持通过API进入其功能

1. 在编辑菜单上选择全局配置。
2. 点击左侧面板中的API，然后选择设置。
3. 在右侧面板中，选取Enable DDE clients( 启动DDE用户端) 以使TWS与通过DDE的TWS整合。从IB网站中的软件页面下载样本程序。
4. 按要求设置其它API参数。有关详细内容，请参阅[交易平台应用程序接口设置](#)。

**Note:** 每次只能同时运行一个API应用程序。DDE除外，API应用程序不需要在应用程序运行的同一个计算机上运行。



## 打开样本表格

在你完成了下载样本表格和配置程序使Excel的DDE API可以与之连接后，可打开表格和保存做为你的个人文件。

### 打开样本表格

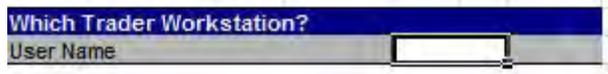
1. 进入含有Excel API样本表格的API安装文件夹(通常为 C:\Jts\Excel\TwsDde.xls) ，并双击 **TwsDde.xls**。
2. 在宏 ( macro) 警告信息框中，点击**Enable Macros( 启动宏程序)**。如果你收到一个询问你是否希望连接到另一个表格中的信息，点击**Yes( 是)**。

**Note:** 使用表格宏程序，你的Excel宏程序安全性必须设置为中等或低。如果你不能打开表格或宏指令不工作，你需要修改你的宏安全性级别。

在微软Excel 2007中，点击Microsoft Office 按钮，点击**Excel选择**，然后点击Excel选择窗口中的**可信中心**。在可信中心中，点击宏设置，然后按要求更改你的设置。

对之前版本的Excel，从**工具**菜单中选择宏，然后选择安全性。将安全性设置为中等或低。

3. 在*Which Trader Workstation?( 哪一个交易平台?)* 部分的*User Name(用户名)*区域，输入你账户的用户名。注意为取得正确的连接，你必须在每个表格页面输入你的用户名。



我们建议使用这个表格做为你的API应用程序的起点。这意味着当新的功能被添加时，你将需要复制你Excel表格中的信息到新发布的样本表格中，然后用不同的文件名保存应用程序。

## 使用 Excel的 DDE 样本表格

Excel的DDE API样本表格，TwsDde.xls，包括下列页面( 标签)：

页面	描述
<a href="#"><u>Tickers( 代码)</u></a>	让你设置你的代码行和请求市场数据。你可以查看包括期货转现货( EFPs) 和组合定单的所有资产类型的市场数据。
<a href="#"><u>Basic Orders( 基本定单)</u></a>	让你发送和修改定单，和设置组合定单和EFPs。
<a href="#"><u>Extended Order Attributes( 扩展的定单属性)</u></a>	和基本定单、高级定单、条件定单和顾问页面一起配合使用，通过这个页面你可以改变有效时间、创建隐藏或冰山定单和运用许多其它的定单属性。
<a href="#"><u>Conditional Orders( 条件定单)</u></a>	让你创建一个定单其发送是基于其它设定的条件，比如根据之前执行情况的定单。
<a href="#"><u>Open Orders( 开放定单)</u></a>	显示你已经发送的但还在处理中的定单，包括已被IB系统接收和仍在交易所中等待的定单。
<a href="#"><u>Advanced Orders( 高级定单)</u></a>	让你发送和修改高级的定单类型，需要使用扩展的定单属性，比如括号、分段和跟踪止损限价定单。
<a href="#"><u>Executions( 执行)</u></a>	让你查看所有执行报告，以及包括一个你可以用来限制你查看结果的过滤框。
<a href="#"><u>Executions Reporting( 执行报告)</u></a>	链接到执行页面，这个页面让你运行四种不同类型的执行报告。
<a href="#"><u>Account( 账户)</u></a>	提供最新的账户信息
<a href="#"><u>Portfolio( 投资组合)</u></a>	显示你当前的所有头寸。
<a href="#"><u>Historical Data( 历史数据)</u></a>	根据你在询问中输入的数据请求产品的历史数据。
<a href="#"><u>Market Scanner( 市场扫描仪)</u></a>	订阅TWS市场扫描仪。
<a href="#"><u>Contract Details( 合约详细)</u></a>	让你收集为进行它交易活动需要的合约特定信息，包括conid和某个合约可用的定单类型。
<a href="#"><u>Bond Contract Details( 债券合约详细)</u></a>	让你收集为进行它交易活动需要的债券合约特定信息，包括债券利率和到期日。

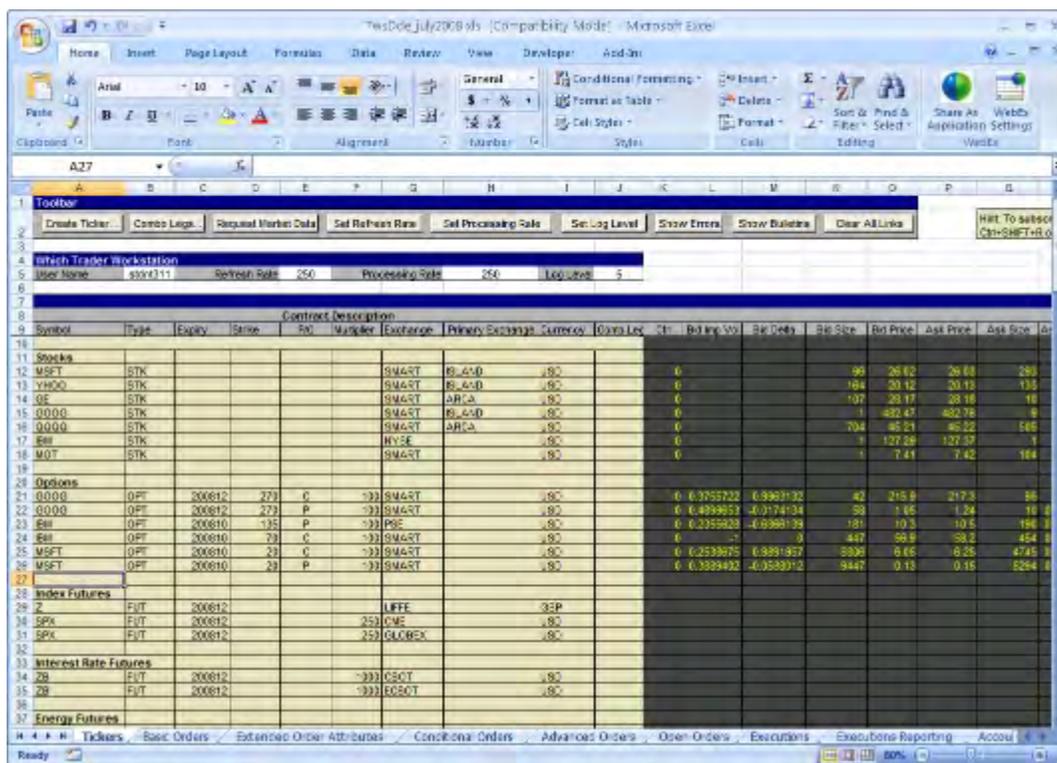
<a href="#">Market Depth( 市场深度)</a>	让你查看选取报价的市场深度。
<a href="#">Advisors( 顾问)</a>	让金融顾问发送和修改金融顾问定单。

**Note:** 两个另外的页面，旧版的执行和旧版的账户投资组合页面，代表了已被表格中的其它页面( 执行、账户和投资组合页面) 代替的功能。尽管这些旧版页面仍然包括在TswDde.xls样本表格中，该API用户指南中并没有介绍，你也不应该使用它们。

### 代码页面

使用代码页面完成：

- 创建市场数据( 代码) 行。
- 请求市场数据。



### 使用代码页面

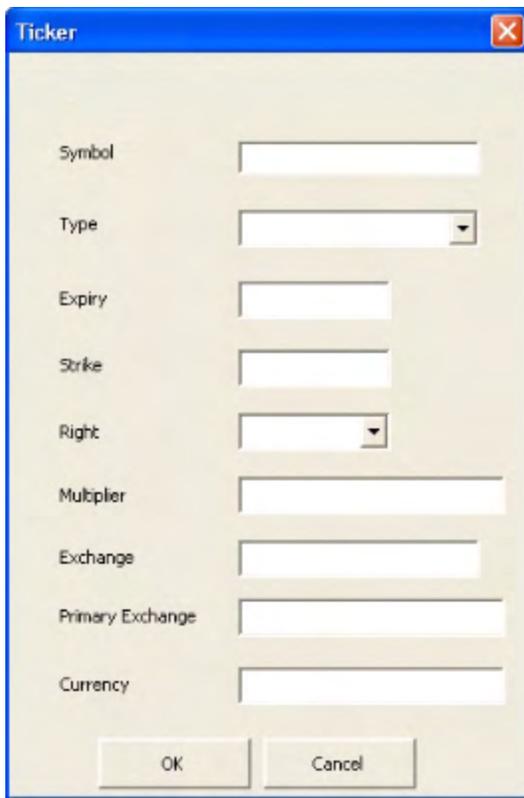
**Note:** 确定TWS处于运行状态，并且你已经在哪一个交易平台？部分的用户名区域输入了你的用户名，包括Excel表格中所有页面，才能保证与TWS的连接。

### 使用创建代码按钮创建代码

1. 点击表格底部的代码标签。
2. 点击一个空白行左侧的行号码选取该行。你必须选择一个空白行才能创建一个代码行。

3. 点击工具栏上的**创建动漫**按钮，并在代码框中输入信息。
4. 点击**OK**。

对股票，你仅需要指定**股票代码**、**类型**、**交易所**(通常为智能传递)、和**货币**。



### 在表格上创建代码

1. 在**代码**栏内输入一个代码。
2. 敲击所有合约描述区域并依据需要输入数据，例如如果你输入一个股票代码，你即不需要在**过期日**、**执行价**、**P/C**和**乘数**区域输入任何值。

**交易所**区域接受以下的值：**SMART**(智能传递定单)，和任何有效的交易所缩写。

### 请求代码的市场数据

1. 点击行号码选择你希望请求市场数据的代码行。
2. 按下**Ctrl+R**，或点击工具栏上的**请求市场数据**。

获取一组代码的市场数据，按住**Shift**键选择多个代码行，然后多次点击**请求市场数据**直到所有行出现数据。

### 设定更新频率

更新频率决定DDE与TWS连接的更新次数。

按默认设置，TWS市场数据每300毫秒更新一次，所以将更新频率设置为250将会将每一次价格的跳动传递到表格中。

### 设置处理频率

服务器的处理频率将影响DDE处理TWS和表格之间请求的数度。

允许的范围是100毫秒 - 2000毫秒。

### 设置记录API客户请求的详细程度

1. 在那一个交易平台？中的记录级别区域输入希望的记录级别值( 1=系统、2=出错、3=警告。4=信息、5=详细)。
2. 将你的光标从记录级别区域移开，然后点击**设置记录级别**按键。

### 取消所有DDE与TWS的连接

位于代码页面上的**取消所有连接**按钮可以取消从TwsDde.xls到TWS的所有DDE连接，这里表格可能会创建提供的应用程序的Visual Basic(VBA)编码。当你准备保存表格会通常使用这个按钮。

点击这个按钮取消所有市场数据、历史数据、市场扫描仪订阅、和其它数据请求。如果你添加你自己的连接到现有的或新的页面，也需要更新 *clearAllLinks* 宏指令来清除这些连接。表格中的每个页面包含其自己的 *clearLinks*宏指令；均称之为 *clearAllLinks* 宏指令。

**Note:** 清除所有的连接并不取消定单。

### 代码页面工具栏按键

代码页面上工具栏包括的按钮描述如下：

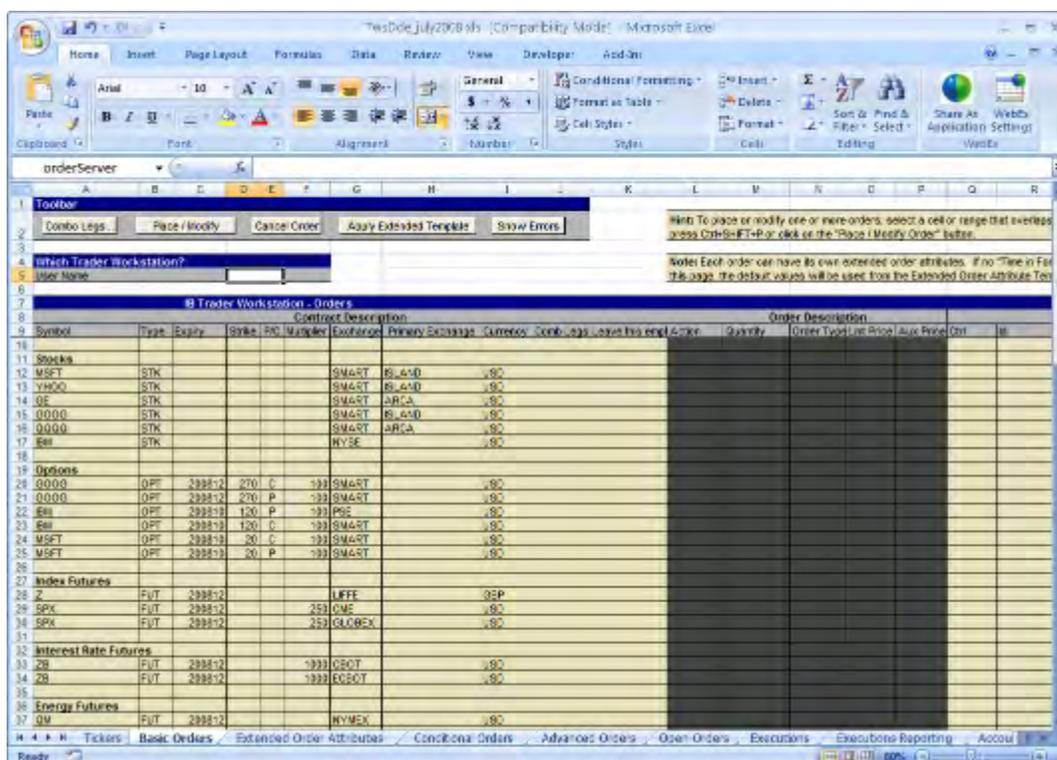
按钮	描述
创建代码	打开代码框。输入信息创建市场数据行。
组合边	打开组合边框。输入合约信息 创建组合定单的每一边。
请求市场数据	选择一个行并点击获取选取合约的市场数据。
设置更新频率	更新频率值为毫秒，确定DDE与TWS连接的更新频率。默认的更新频率为1000( 每一秒钟更新一次)， 允许的频率范围为100毫秒到2000毫秒。 注意， TWS市场数据每300毫秒更新一次。 意味着默认的“每1秒钟”更新频率将仅显示30%的数据变化。更新频率为250将使得每个变化均传递到表格中。
设置处理频率	设定TWS/DDE服务器信息处理频率( 也为毫秒) 将影响DDE处理表格和TWS之间的请求处理速度。 允许的范围是100毫秒到2000毫秒。

<p><b>设定记录级别</b></p>	<p>指定用于在处理API请求时记录输入的详细级别。有效的值包括：                  1 = 系统                  2 = 出错                  3 = 警告                  4 = 信息                  5 = 详细</p>
<p><b>显示错误</b></p>	<p>跳至错误编码区域并显示最新的错误编码。</p>
<p><b>显示公告</b></p>	<p>打开新闻公告信息。如果你订阅了公告，新闻将出现在表格右上角的红色框中。</p>
<p><b>清除所有连接</b></p>	<p>清除所有DDE和TWS的连接。</p>

**基本定单页面**

使用基本定单页面进行以下操作：

- 创建定单。
- 创建“括号”定单。
- 修改和取消定单。
- 创建组合定单。



## 下达定单

这里介绍如何在定单页面上下达下列类型的定单：

- 简单定单
- 篮子定单
- 修改的定单

**Note:** 确定TWS处于运行状态，并且你已经在哪一个交易平台？部分的用户名区域输入了你的用户名，包括Excel表格中所有页面，才能保证与TWS的连接。

## 下达定单

1. 点击表格底部的**基本定单**标签。
2. 在空白代码区域输入代码指定合约，然后在相应的合约描述区域输入信息。
3. 选择一个合约并使用定单描述区域设定定单。  
你必须设定活动(买、卖或做空)，数量，定单类型，限价(除非是市价单)，并如果定单类型需要的辅助价格。
4. 如果需要，点击工具栏上的**应用扩展的模版**按钮使用扩展的定单属性。这适用于你已经在[扩展的定单属性](#)页面上定义的所有属性。
5. 点击页面上工具栏部分的下达/修改定单按钮。

## 下达“篮子”定单

1. 点击表格底部的**基本定单**标签。
2. 在空白代码区域输入代码指定合约，然后在相应的合约描述区域输入信息。
3. 选择一个合约并使用定单描述部分中的区域设定定单。
4. 对其它的定单重复第一步和第二步。
5. 选择一组定单。
  - 选择一组邻近定单的方式是首先选中第一个定单，按下**Shift**键，然后点击组中最后的定单。
  - 选择一组不在一起的定单，按住**Ctrl**键同时选取每一个定单。
6. 点击**下达/修改定单**按钮。

## 修改一个定单(或一组定单)

1. 在基本定单页面上，改变定单或定单组中需要改动的参数。
2. 选择定单或一组定单。
  - 选择一组邻近定单的方式是首先选中第一个定单，按下**Shift**键，然后点击组中最后的定单。
  - 选择一组不在一起的定单，按住**Ctrl**键同时选取每一个定单。
3. 点击**下达/修改定单**按钮。

**Note:** 你不能够在Excel的DDE API中修改从TWS提交的定单。这是因为Excel的DDE API使用其自己创建的定单代号来修改定单，并非全局客户定单代号。所有在TWS

中创建的定单具有一个内部定单代号0。如果你试图在Excel的DDE API中修改具有代号="id0"的定单，你将收到错误信息“重复定单代号”。

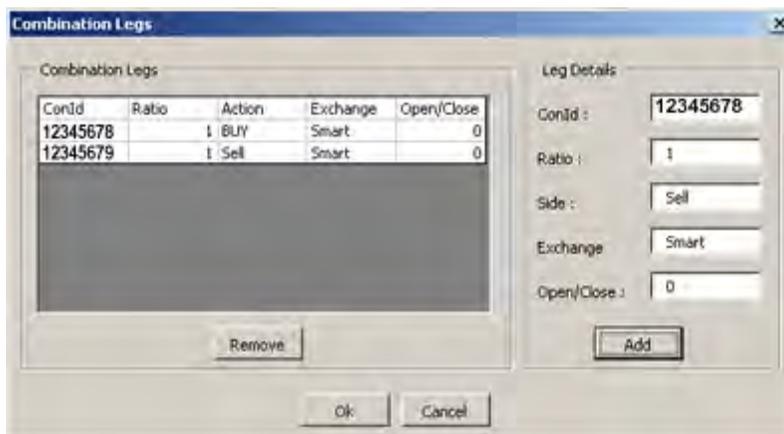
### 下达组合定单

组合定单是一种特殊定单，由许多独立的定单边组成，但按一个单一定单来执行。

例如，买进一个日历差价，你需要：

- 买 1 OPT JUL03 17.5 CALL (100)
- 卖 1 OPT AUG03 17.5 CALL (100)

下面的例子用一个假想下达ISE交易所上XYZ股票的日历差价定单来帮助你了解组合定单的过程。



### 创建日历差价定单

1. 使用 [合约详细](#) 页面找出两边定义的合约代码。
  - XYZ期权( JUL08 17.5 CALL on ISE ) 的conid是 "12345678"。
  - XYZ期权( AUG08 17.5 CALL on ISE ) 的conid是 "12345679"。
2. 点击 **基本定单** 标签创建组合边定义。点击基本定单页面工具栏上的 **组合边** 按钮，输入边信息。你的边信息被转换成以下格式：

[CMBLGS]\_[NumOfLegs]\_[Combo Leg Definitions]\_[CMBLGS]

这里：

- [CMBLGS] 是用于确定边定义开始和截止的分隔符
- [NumOfLegs] 是边定义的数目
- [Combo Leg Definitions] 确定N边的定义，每个边的定义包括[conid]\_[ratio]\_[action]\_[exchange]\_[openClose]，所以组合子字符串的结果是：

CMBLGS\_2\_17496957\_1\_BUY\_EMPTY\_0\_15910089\_1\_SELL\_EMPTY\_0\_CMBLGS

3. 组合边的定义必须在扩展定单属性之前完成。完整下达DDE请求定单字符串将为：



=acctName|ord!id12345?place?BUY\_1\_XYZ\_BAG\_ISE\_LMT\_1\_CMBLGS\_2\_12345678\_1\_BUY\_EMPTY\_0\_12345679\_1\_SELL\_EMPTY\_0\_CMBLGS\_DAY\_EMPTY\_0\_O\_0\_EMPTY\_0\_EMPTY\_0\_0EMPTY\_0\_0

如果定单边没有构成一个有效的组合，将出现以下的错误信息之一：

- 312 = 组合详细无效。
- 313 = '<leg number>' 的组合详细无效。
- 314 = 证券类型'BAG'要求组合边详细。
- 315 = 股票组合边只限于智能传递交易所。

**Note:** 1.交易所上的边定义必须与组合定单中的匹配。STK边定义除外，其必须指定用智能传递交易所。  
2.对零售账户情况， openClose边定义值永远是“相同”的(例如0)。对机构账户，该值可以是以下的任何之一(相同、开盘、收盘)。

### 支持的定单类型

目前通过Excel的DDE API支持的定单类型：

- 限价单(LMT)
- 市价单(MKT)
- 触及限价(LIT)
- 触及市价(MIT)
- 收盘市价(MOC)
- 收盘限价(LOC)
- 挂钩市价(PEGMKT)
- 相对(REL)
- 止损(STP)
- 止损限价(STPLMT)
- 跟踪止损(TRAIL)
- 跟踪止损限价(TRAILLIMIT)
- 交易量加权平均价格(VWAP)
- 波动率定单(VOL)

### 基本定单页面工具栏按钮

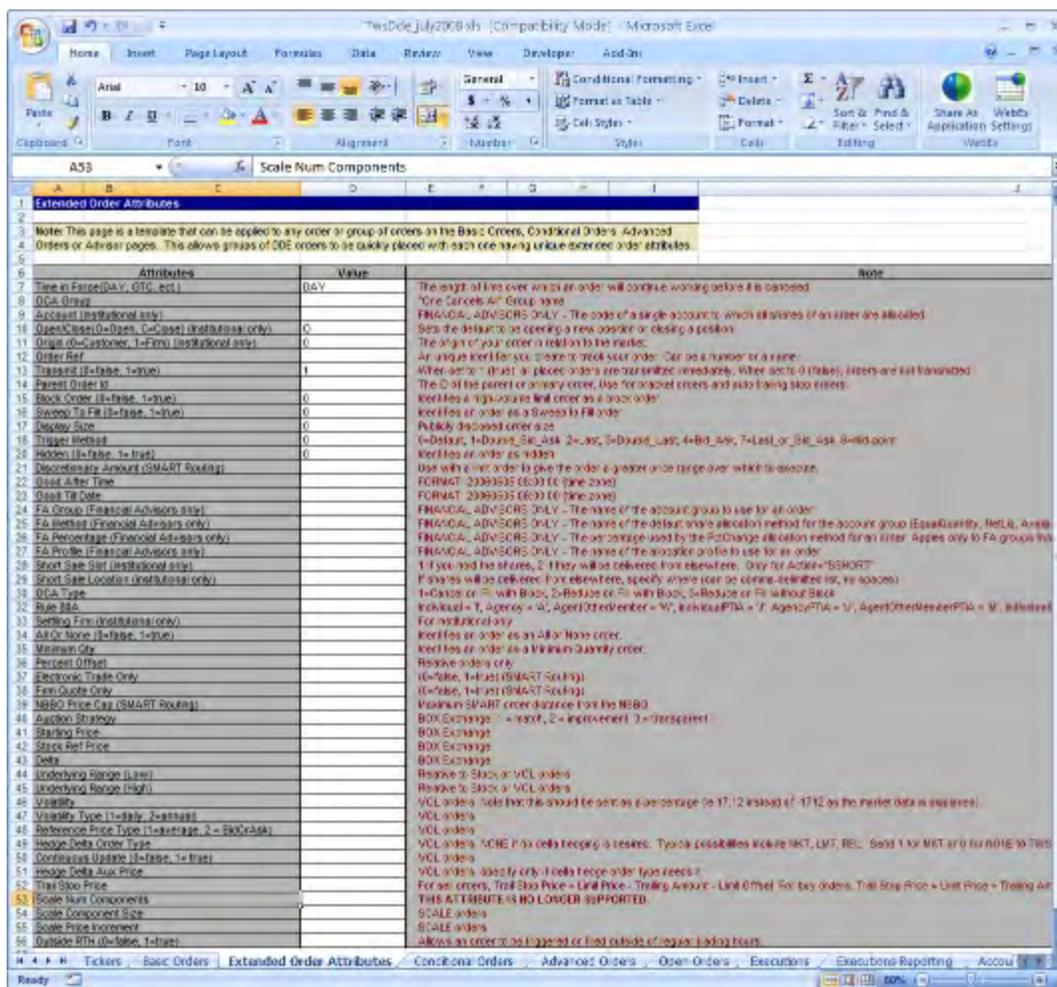
基本定单页面上的按钮包括下列：

按钮	描述
组合边	打开组合边框。输入合约详细创建组合定单的每个边。

下达/修改定单	完成了定单描述区域后，定义了任何扩展的属性后，点击创建选取合约的定单。
取消定单	这个按钮取消你选中的定单。
应用扩展模版	应用扩展定单属性页面上的当前值到选中的定单行。
显示错误	跳至错误编码区域和显示错误编码。

### 扩展定单属性页面

扩展定单属性页面包括所有下列情况下你可以使用的属性，发送定单时、设定显示尺寸创建冰山定单时、添加定单到一取消全(OCA)组中和为到时后有效定单设定发送日期时。一旦你在这个页面上确定了属性，你即可以使用**应用扩展模版**按钮将它们应用到一个单个定单或选取的一组定单中，该按钮出现在定单页面和条件定单页面。属性将填充扩展定单属性区域，包括从定单状态区域一直到页面的最右面。





timeInForce ( 有效时间)	DAY( 日内) GTC( 取消前有效) OPG( 开盘市价或限价) IOC( 立即或取消) GTD( 直到日期/时间有效) FOK( 全部执行或取消) DTC( 日内直到取消)
ocaGroup( 一取消全组)	确定OCA( 一取消全) 组的字符串
account( 账户)	字符串( 用于机构)
open/close( 开盘/收盘)	O, C ( 用于机构)
origin( 起始)	0, 1 ( 用于机构)
orderRef( 订单参考)	字符串
transmit( 发送)	说明订单是被立即发送( 设定为1) 或不是( 设定为0) 。  这个参数对篮子订单非常有用。首先，准备一个篮子订单( 未发送的)，当准备好后，设置每个订单的发送参数值为1来为执行发送篮子订单。
parentId( 母代号)	字符串( 用于母订单、括号单和自动跟踪止损单的订单代号)
blockOrder( 批量订单)	0 ( 非批量订单) 1 ( 代表批量订单)
sweepToFill( 一扫光)	0( 非一扫光订单) 1( 一扫光订单)
displaySize( 显示尺寸)	公开披露冰山订单的订单尺寸。这个值应该为保存为字符串的一个数字。

<p>triggerMethod( 触发方法)</p>	<p>说明模拟的止损、止损限价和跟踪止损定单是如何被触发的：</p> <ul style="list-style-type: none"> <li>• 0 - 默认值。将对场外( OTC) 股票和美国期权的定单使用“两次买价/卖价”方法。所有其它定单将使用“最后”方法。</li> <li>• 1 - 使用“两次买价/卖价”方法，这里止损定单是基于两个连续买价或卖价价格被触发的。</li> <li>• 2 - “最后”方法，这里止损定单将基于最后价格触发。</li> <li>• 3 - “两次最后”方法，这里止损定单的触发将基于最后的两个价格。</li> <li>• 4 - “买价 - 卖价”方法。对买单，单个发生的买价价格必须等于或大于触发价格。对于卖单，单个发生的卖价价格必须等于或小于触发价格。</li> <li>• 7 - “最后或买价 - 卖价”方法。对买单，单个买价价格或最后价格必须等于或大于触发价格。对卖单，单个卖价价格或最后价格必须等于或小于触发价格。</li> <li>• 8 - “中点”方法，这里中点必须等于或大于( 对买单) 或者等用于或小于( 对卖单) 触发价，买卖价之间的价差必须小于中点的0.1%。</li> </ul> <p>有关触发方法的完整描述，请参阅交易平台用户指南中的<a href="#">修改触发方式</a>章节。</p>
<p>hidden( 隐藏)</p>	<p>0 1( 在查看市场深度时定单为不可视的)</p>
<p>Discretionary Amount (SMART Routing)( 委托额) ( 智能传递)</p>	<p>和限价单一起使用，使定单的执行价格具有更大的范围。</p>
<p>Good After Time( 到时有有效)</p>	<p>输入定单将成为活跃定单的日期和时间。使用的格式是YYYYMMDD hh:mm:ss TMZ，这里TMZ为可选用的三个字母的时区代号。允许使用的时区在<a href="#">这里</a>列出。</p>
<p>Good 'Till Date( 直到日期有效)</p>	<p>定单继续工作到你输入日期的收盘时为止。使用的格式是YYYYMMDD。设定定单关闭日的时间，请使用HH:MM:SS格式。使用有效的三位字母缩写 指定时区。</p>
<p>FA Group( 金融顾问团组)</p>	<p>仅用于顾问账户。是将交易分配给的金融顾问团组的名称。如果不适用，请使用空字符串。</p>

FA Method( 金融顾问方法)	仅用于顾问账户。 共享的分配方法。  <ul style="list-style-type: none"> <li>• EqualQuantity( 等量)</li> <li>• NetLiq( 净流动性)</li> <li>• AvailableEquity( 可用股权)</li> <li>• PctChange( 百分比变化)</li> </ul>
FA Percentage( 金融顾问百分比)	仅用于顾问账户。 共享分配百分比。
FA Profile( 金融顾问规则)	仅用于顾问账户。 共享分配规则的名称。
Short Sale Slot( 卖空来源)	仅对机构账户。 有效值为1( 经纪商持有股票) 和2( 股票来自其它地方) 。
卖空地点	仅机构账户。
OCA Type( 一取消全类型)	1 = 执行后批量取消。 2 = 执行后批量降低。 3 = 执行后无批量降低
Rule 80A( 法规80A)	<ul style="list-style-type: none"> <li>• Individual = 'I'</li> <li>• Agency = 'A',</li> <li>• AgentOtherMember = 'W'</li> <li>• IndividualPTIA = 'J'</li> <li>• AgencyPTIA = 'U'</li> <li>• AgentOtherMemberPTIA = 'M'</li> <li>• IndividualPT = 'K'</li> <li>• AgencyPT = 'Y'</li> <li>• AgentOtherMemberPT = 'N'</li> </ul>
Settling Firm( 结算公司)	仅用于机构账户
All or None( 所有或无)	0 = 假 1 = 真
Minimum Qty( 最低数量)	指定订单为最低数量订单。
Percent Offset( 百分比抵消)	相对订单的百分比抵消。
Electronic Trade Only( 仅电子交易)	0 = 假 1 = 真

Firm Quote Only( 仅不变报价)	0 = 假 1 = 真
NBBO Price Cap( 全国最佳买卖价价格限额)	与全国最佳买卖价的智能传递定单的最大差额。
Auction Strategy( 拍卖策略)	相等 = 1 改善 = 2 透明 = 3 仅用于BOX交易所。
Starting Price( 起始价)	起始价。 仅用于BOX交易所定单。
Stock Ref Price( 股票参考价格)	用于波动率定单来计算发给交易所的限价价格( 不论是否使用连续更新) ， 和监控的价格范围。还用于期权定单的价格改善。
Delta	股票 delta。 仅用于BOX交易所定单。
Underlying Range (Low)( 底层证券范围) ( 低)	可接受的底层股票价格范围的低端值。 对BOX交易所的期权定单价格改善和可动态管理的波动率定单。
Underlying Range (High)( 底层证券范围) ( 高)	可接受的底层股票价格范围的高端值。 对BOX交易所的期权定单价格改善和可动态管理的波动率定单。
Volatility( 波动率)	通过TWS期权分析计算的波动率中的期权价格。 这个值的表示为一个百分比， 并被用于计算发送给交易所的限价价格。
Volatility Type( 波动率类型)	1 = 日 2 = 年
Reference Price Type( 参考价格类型)	1 = 平均 2 = 买价或卖价
Hedge Delta Order Type( 对冲Delta定单类型)	TWS版本859之前， 使用“1”来发送市价单，“0”不发送定单。 TWS版本859之后， 输入一个可接受的定单类型， 比如： MKT、 LMT、 REL或MTL。
Continuous Update( 连续更新)	0 = 假 1 = 真
Hedge Delta Aux Price( 对冲Delta辅助价格)	为对冲Delta定单类型输入要求的辅助价格。
Trail Stop Price( 跟踪止损价格)	仅用于跟踪止损限价单。 这是TRAILLMT定单的止损触发价格。





- 在蓝色的条件陈述区域，使用陈述区域来设定必须满足才能触发定单的条件。当陈述 = 真，你的定单将被发送。

样本表格中包括一对定单，其中第二个定单的发送是基于第一个定单被完全执行后。这个例子中，陈述区域的触发条件是单元格T10中的值(执行的区域) 必须等于单元格M10(定单数量区域)。

- 在添加/修改(ADD/MOD) 区域输入**添加(ADD)**，因为你在创建一次性定单。
- 如同你在定单描述区域中完成的一样定义其它定单参数。

IB Trader Workstation - Conditional Orders													
Option			Order Description						Order				
Primary Exchange	Currency	Comb Legs	Leave this empty	Action	Quantity	Order Type	Lmt Price	Aux Price	Ctrl	Id	St	Filled	Remainin
ARCA	USD			BUY	10	MKT			0	id	Fi	10	0
ISLAND	USD			SELL	10	MKT			0	id	Fi	10	0
	usd			buy	200	lmt	81.00						

- 根据下列表格中的句法完成**条件定单**页面中的相关区域。

区域	描述
Statement(陈述)	Excel的一个功能，结果为真或假。为真时，定单将被发送；为假时，什么都不发生。
ADD/MOD(添加/修改)	对一次性定单使用ADD。使用MOD来连续查看和修改定单，直到定单被完全执行。这个区域是启动条件定单的区域，定单将仅在具有“ADD”或“MOD”标签的情况下才能被启动。
Action(活动)	买 卖
Quantity(数量)	输入定单的数量。
Order Type(定单类型)	请参考 <a href="#">支持的定单类型列表</a> 。
Lmt Price(限价)	限价单和止损限价单的限价。
Aux. Price(辅助价格)	止损和止损限价单的选择止损价格，或相对定单的抵消价格。

以上描述的所有区域可能会是其它单元格的变量，所以可能会创建任何类型的条件定单。

### 条件定单举例

#### 如果获执行的定单

如果获执行的定单是如果先前的定单被执行后即执行的定单。使用条件“如果一个买单被全部执行，输入限价价格为\$50.00的限价卖单”创建如果获执行的定单：

区域	值
<i>Statement</i> ( 陈述)	执行的单元格 = 100
<i>ADD/MOD</i> ( 添加/修改)	ADD
<i>Action</i> ( 活动)	卖
<i>Quantity</i> ( 数量)	100
<i>Order Type</i> ( 定单类型)	限价
<i>Lmt Price</i> ( 限价价格)	50
<i>Aux. Price</i> ( 辅助价格)	空

#### 价格改变定单

如果一个特定的买价或卖价大于、小于或等于一个特定的价格，则价格改变定单将被触发。创建价格改变定单用的条件是“如果买价低于81.20，发送一个限价价格为\$81.10的买进200股限价单：

区域	值
<i>Statement</i> ( 陈述)	<p>在代码页面上将光标指向你希望使用的买价区域，然后复制表格顶部公式栏中的值(“=”输入区域)。这个值应该显示为：</p> <p><code>=username tik!id4?bid</code></p> <p>这里“4”代表特定合约的买价。</p> <p>将这个值粘贴到公式栏中(“=”输入区域) 用作陈述，并添加你的条件，“&gt;” or “&lt;”，接下来为价格。这个例子中，公式应该为：</p> <p><code>=username tik!id4?bid&lt;81.20</code></p>

<i>ADD/MOD</i> ( 添加/修改)	ADD
<i>Action</i> ( 活 动)	买
<i>Quantity</i> ( 数量)	200
<i>Order Type</i> ( 订单类型)	限价
<i>Lmt Price</i> ( 限价价格)	81.10
<i>Aux. Price</i> ( 辅助价格)	该举例中不需要。

### 修改订单( 或篮子订单)

1. 选择订单或一组订单。
  - 选择一组相邻的订单，选中第一个订单，按住 **Shift** 键，然后选中组中最后一个订单。
  - 选择一组不相邻的订单，按住 **Ctrl** 键，然后选择每个订单。
1. 点击 **Place/Modify**( 下达/修改) 订单按钮。
2. 更改任何需要的参数，然后点击 **Place/Modify** 订单按钮。

### 条件订单页面工具栏按钮

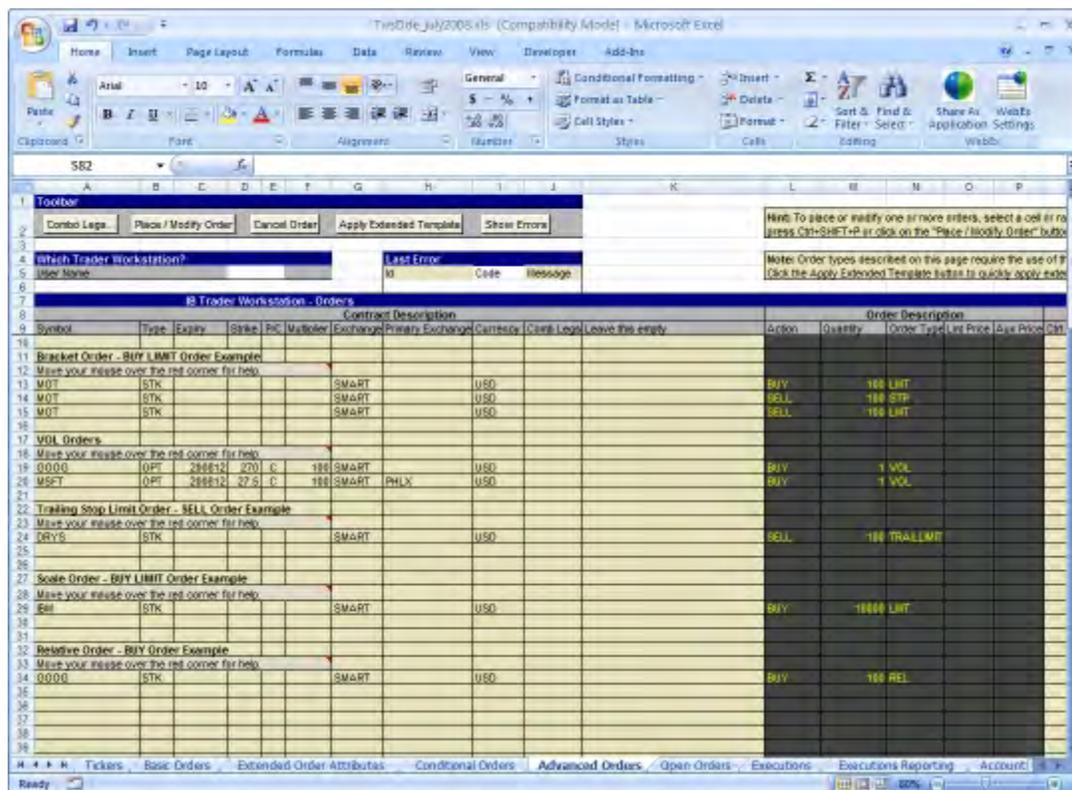
条件订单页面上的工具栏包括下列按钮：

按钮	描述
<b>Combo Legs</b> ( 组合边)	打开组合边框。输入合约详细创建组合订单的每个边。
<b>Place/Modify Order</b> ( 下达/修改订单)	在你完成了订单描述区域，定义了任何扩展的属性，点击创建选取合约的订单。
<b>Apply Extended Template</b> ( 应用扩展模版)	将所有扩展订单属性页面上的属性应用到选取的订单。
<b>Cancel Order</b> ( 取消订单)	这个按钮取消你已选中的订单。
<b>Show Errors</b> ( 显示错误)	跳至错误编码区域并显示错误编码。

### 高级定单页面

使用高级定单页面创建需要使用扩展定单属性的复杂定单，包括：

- 括号定单
- 波动率定单
- 跟踪止损限价定单
- 分段定单
- 相对定单



有关对单个定单或组定单使用扩展定单属性的详细信息，请参阅[应用扩展定单属性到单个定单和组定单](#)

### 下达括号定单

Excel的DDE样本表格中的括号定单要求使用扩展定单属性发送和母定单代号。你必须在定单被完全设定好之前将发送关闭，并且你必须确定括号中的第一个定单做母定单。

### 下达买进限价括号定单

1. 点击表格底部的高级定单标签。
2. 对三个连续行上的所有三个定单输入合约描述和定单描述：

- 第一个定单应该是买限价单。
  - 第二个定单应该是卖止损单。
  - 第三个定单应该是卖限价单。
2. 点击**扩展定单属性**标签。改变发送的值为**0**（扩展定单属性页面上第13行）。  
这将保证你的定单不会在你完成定单设置之前被发送。
  3. 点击**高级定单**标签，选中括号定单中的第一个定单，然后点击**下达/修改定单**按钮。  
定单没有被执行，但系统生成了一个定单代号。
  4. 为第一个定单复制定单代号，除去前缀“id( 代号) ”，然后点击**扩展定单属性**标签并将定单代号粘贴到**值**区域用作**母定单代号**( 14行)。这个值将被用于所有后续定单，直到你将其从扩展定单属性页面中删除。  
括号定单的第一个定单现在为主定单。
  5. 点击**高级定单**标签，选中第二个定单，然后点击**下达/修改定单**按钮。  
定单没有被执行，但现在通过母定单代号扩展定单属性与主定单相连在一起。
  6. 点击**扩展定单属性**标签，将发送的值还原为**1**（第13行）。
  7. 点击**高级定单**标签，选中括号定单的第三个定单，然后点击**下达/修改定单**按钮。整个括号定单即被发送。
  8. 当你完成了下达你的括号定单后，进入**扩展定单属性**页面并删除你输入的**母定单代号**值。如果你不将其删除，这个值将被应用于所有你的表格中下达的后续定单。

### 下达波动率定单

在Excel的DDE样本表格中，你可以通过输入下面扩展定单属性的值来下达波动率( VOL) 定单：

- 波动率
- 波动率类型
- 参考价格类型
- 连续更新
- 底层证券范围( 低) —可选
- 底层证券范围( 高) —可选
- 对冲Delta定单类型—可选
- 对冲Delta辅助价格—可选

### 下达一个VOL定单

1. 点击表格底部的**高级定单**标签。
2. 在一个空白代码区域输入一个代码来定义合约，然后在相应的合约描述区域内输入信息。
3. 选择一个合约并使用**定单描述**区域设置定单。
  - 在**定单类型**区域内输入**VOL**。

4. 点击**扩展定单属性**标签。在*值*区域内为下列扩展定单属性输入值：
  - **波动率** - 这个值代表用于计算期权限价价格的波动率。以百分比形式输入这个值，而非显示的市场数据。例如，输入17.12而不是.1712。
  - **波动率类型** - 输入1代表日波动率或2代表年波动率。
  - **参考价格类型** - 这个值被用来计算发给交易所的限价价格和股票范围价格监控。使用平均的最佳买价和卖价输入1；或输入2使用全国最佳买价（当买进看涨或卖出看跌时），或全国最佳卖价（当卖出看涨或买进看跌时）。
  - **连续更新** - 输入1在底层股票价格（或期货价格。用于指数期权时）变化时自动更新期权价格。输入0，如果你不希望使用这个功能。
5. 在**扩展定单属性**页面，在*值*区域内对下列可选的扩展定单属性输入值：
  - **底层证券范围(低)** - 输入一个相对于选取期权定单的可接受的低端股票价格。如果底层证券的价格下跌至低端股票范围价格以外，期权定单将被取消。
  - **底层证券范围(高)** - 输入一个相对于选取期权定单的可接受的高端股票价格。如果底层证券的价格上涨超出高端股票范围价格，期权定单将被取消。
  - **对冲Delta定单类型** - 输入LMT(限价单)、MKT(市价单)或REL(相对单)。输入NONE(无)，如果你不希望使用delta对冲。
  - **对冲Delta辅助价格** - 如果你已经输入了LMT或REL做为对冲Delta定单类型，请输入价格做为这个属性的值。
6. 点击**高级定单**标签，然后选中定单行。
7. 点击**应用扩展模版**按钮。你输入的扩展定单属性的值被应用于定单行，并显示在页面的**扩展定单属性**部分。
8. 选中定单行后，点击**下达/修改定单(Place/Modify Order)**按钮。
9. 当你完成了下达VOL定单，进入**扩展定单属性**页面并删除你输入的VOL定单值。如果你不将其删除，这些值将被应用到所有你将在表格中下达的后续定单。

### 下达追踪止损限价单

在TWS中，有四个值可以组成一个追踪止损限价单：

- 追踪额
- 止损价
- 限价
- 限价抵消

你在Excel的DDE API表格中输入追踪额、止损价和限价。这里没有用于限价抵消值的区域或扩展定单属性。你必须在止损价中包括限价抵消（**追踪止损价扩展定单属性**）。

### 创建追踪止损限价单

1. 点击表格底部的**高级定单**标签。
2. 在一个空白代码区域输入一个代码来定义合约，然后在相应的合约描述区域内输入信息。
3. 选择一个合约并使用**定单描述**区域设置定单。

- 在 *活动区域* 输入 **买或卖**。
  - 在 *限价价格 ( Lmt Price)* 区域输入限价价格。
  - 在 *定单类型 ( Order Type)* 区域输入 **追踪限价 ( TRAILLIMIT)** 。
  - 在 *辅助价格 ( Aux Price)* 区域输入追踪额。
4. 点击 **扩展定单属性** 页面标签。指定追踪止损价做为扩展定单属性。将这个值输入到追踪止损价值区域。
- 追踪止损价值必须包括限价抵消。
- 对卖单：  
**追踪止损价 = 限价 - 追踪额 - 限价抵消**
  - 对买单：  
**追踪止损价 = 限价 + 追踪额 + 限价抵消**
5. 在 **高级定单** 页面上，选择定单行并点击 **用于扩展模版** 按钮。追踪止损价值被应用到选取的定单并显示在页面 *扩展定单属性* 部分的 *追踪止损价* 区域中。
6. 点击 **下达/修改定单** 按钮。
7. 当你完成了下达定单后，进入 **扩展定单属性** 页面并删除你输入的 *追踪止损价* 值。如果你不将其删除，这些值将被应用到所有你将在表格中下达的后续定单。

### 下达分段定单

在Excel的DDE样本表格中，你可以通过输入下面扩展定单属性的值来下达分段定单：

#### 下达分段定单

1. 点击表格底部的 **高级定单** 标签。
2. 在一个空白代码区域输入一个代码来定义合约，然后在相应的合约描述区域内输入信息。
3. 选择一个合约并使用 *定单描述* 区域设置定单。定单类型应该为 **LMT** 或 **REL**。
4. 点击 **扩展定单属性** 标签。在 *值* 区域内为下列扩展定单属性输入值：
  - **分段组尺寸** - 输入第一组，或起始组、定单组，的尺寸。例如，如果你发送一个10,000股的定单，分段组尺寸值为1000，第一组将为1000股。
  - **分段价格增量** - 输入用于计算分段阶梯中每个组的单位价格的数额。不可以是负数。

**Note:** API发行要点9.41之后，不在支持分段数字组。

3. 在 **高级定单** 页面上，选择定单行并点击 **应用扩展模版** 按钮。分段定单值将被应用到选取的定单并显示在页面的 *扩展定单属性* 部分。
4. 点击 **下达/修改定单** 按钮。
5. 当你完成了下达定单后，进入 **扩展定单属性** 页面并删除你输入的分段定单值。如果你不将其删除，这些值将被应用到所有你将在表格中下达的后续定单。

### 下达相对定单

在Excel的DDE样本表格中，输入一个 *百分比抵消* 扩展定单属性值来下达相对定单。

#### 下达相对定单

1. 点击表格底部的**高级定单**标签。
2. 在一个空白代码区域输入一个代码来定义合约，然后在相应的合约描述区域内输入信息。
3. 选择一个合约并使用**定单描述**区域设置定单。
  - 输入**REL** 做为定单类型。
  - 在**限价价格( Lmt Price)** 单元中输入价格上限。
4. 点击**扩展定单属性**标签。在**值**区域以十进制形式输入一个百分比用作**百分比抵消( Percent Offset)** 扩展定单属性。
5. 在**高级定单**页面上，选择定单行并点击**用于扩展模版**按钮。百分比抵消值将被应用到选取的定单并显示在页面的**扩展定单属性**部分。
6. 点击**下达/修改 定单**按钮。
7. 当你完成了下达定单后，进入**扩展定单属性**页面并删除你输入的**百分比抵消值**。如果你不将其删除，这些值将被应用到所有你将在表格中下达的后续定单。

### 高级定单页面工具栏按钮

高级定单页面工具栏上包括下列按钮：

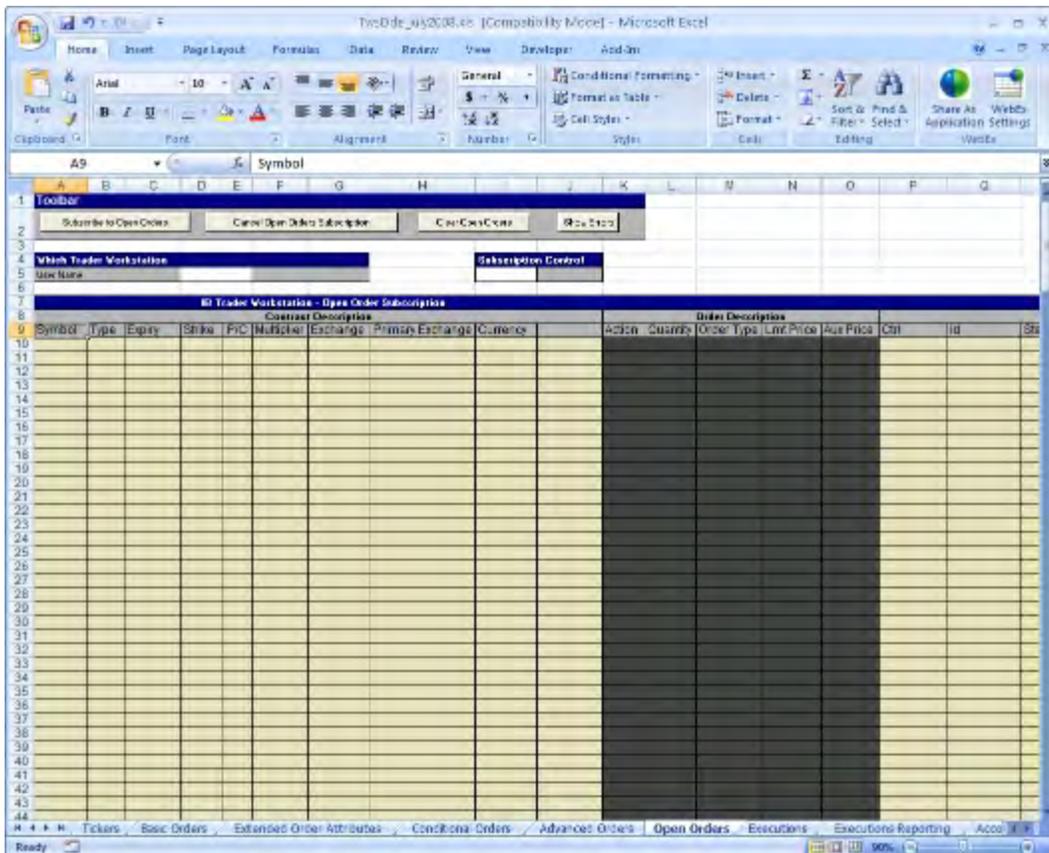
按钮	描述
<b>Combo Legs( 组合边)</b>	打开组合边框。输入合约详细创建组合定单的每个边。
<b>Place/Modify Orders( 下达/修改定单)</b>	在你完成了定单描述区域，定义了任何扩展属性后，点击该按钮为选择的合约创建定单。
<b>Cancel Order( 取消定单)</b>	这个按钮取消你选中的定单。
<b>Apply Extended Template( 应用扩展模版)</b>	在扩展定单属性页面中选中的定单行上应用当前的值。
<b>Show Errors( 显示错误)</b>	跳至错误编码区域并显示错误编码。

### 开放定单页面

开放定单页面显示所有你已经发送的定单，包括已经被IB系统接受的定单，以及仍然在交易所中工作的定单。一旦你订阅了后，该页面将在你每次发送一个新的定单时更新，或是通过API或TWS。

一旦某个定单被执行，它将在开放定单页面上保留30秒，其状态值改变为已执行。然后已执行的定单即被清除，你可以在执行页面上( 如果你订阅了实时执行) 看到这个定单。





### 查看开放定单

**Note:** 确认TWS处于运行状态，并且你已经在哪个交易平台？的用户名区域输入了你的用户名，包括Excel表格中所有页面，才能保证与TWS的连接。

### 查看开放定单：

1. 点击表格底部的**开放定单**标签。
2. 点击工具栏上的**订阅开放定单**。

所有你的开放定单即显示在页面中，包括你在Excel API表格和TWS输入的定单。

定单被执行后在页面上保留30秒，状态区域显示为**执行**。

### 清除开放定单

1. 点击工具栏上的**取消开放定单订阅**按钮。
2. 点击**清除开放定单**按钮。

### 开放定单标签工具栏按钮

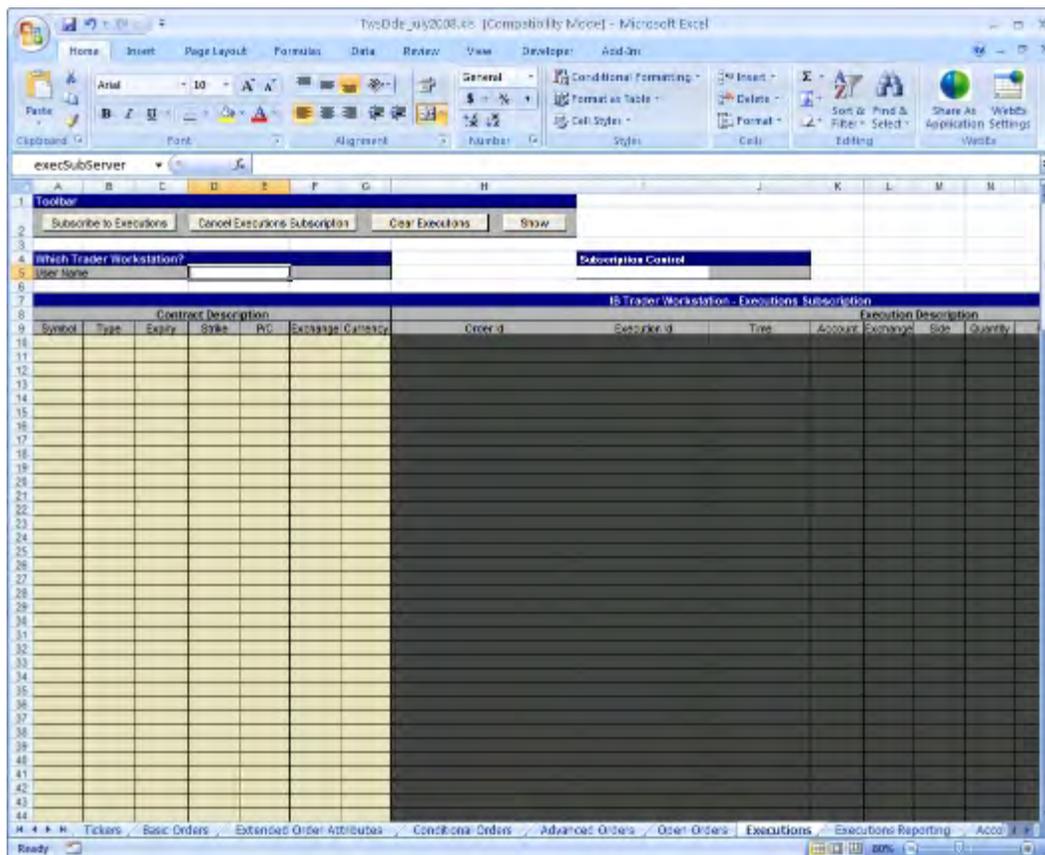
开放定单页面上的工具栏包括以下按钮：

按钮	描述
----	----

<b>Subscribe to Open Orders( 订阅开放定单)</b>	在你输入了有效的用户名之后，点击这个按钮查询TWS并返回所有的开放定单。一旦你订阅了开放定单，这个页面在每次有开放定单时即进行更新。
<b>Cancel Open Orders Subscription( 取消开放定单订阅)</b>	取消开放定单订阅。这个页面将不在显示你的开放定单。
<b>Clear Open Orders( 清除开放定单)</b>	删除页面上所有开放定单。
<b>Show Errors( 显示错误)</b>	跳至错误编码区域并显示错误编码。

**执行页面**

当你订阅执行时，执行页面显示关于所有完成交易的信息( 也称之为“执行报告”)。



**查看执行**

**Note:** 确认TWS处于运行状态，并你已经并且你已经在哪个交易平台？哪个交易平台？的用户名用户名区域输入了你的用户名，包括Excel表格中所有页面，才能保证与TWS的连接。

### 查看执行

1. 点击表格底部的执行标签。
2. 点击工具栏上的 **Subscribe to Executions( 订阅执行)** 按钮。

### 删除执行报告

1. 点击工具栏上的 **Cancel Executions Subscription( 取消执行订阅)** 按钮。
2. 点击 **Clear Executions( 清除执行)** 按钮。

### 执行页面工具栏按钮

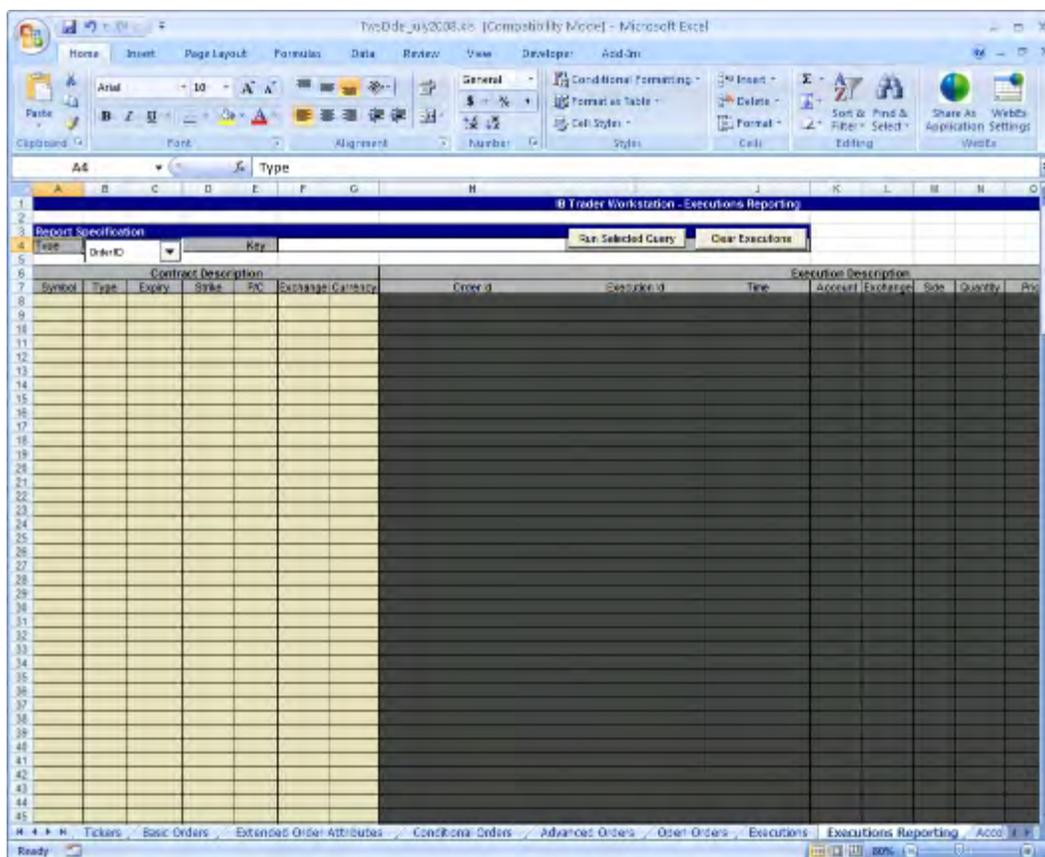
执行页面的工具栏包括下列按钮：

按钮	描述
<b>Subscribe to Executions( 订阅执行)</b>	在你输入了有效的用户名后，这个按钮查询TWS并返回关于索引有效执行的信息。你订阅了执行后，这个页面将在每次有定单执行时获得更新。
<b>Cancel Executions Subscription( 取消执行订阅)</b>	点击取消执行订阅。
<b>Clear Executions( 清除执行)</b>	从页面上删除所有执行报告。
<b>Show Errors( 显示错误)</b>	跳至错误编码区域并显示错误编码。

### 执行报告页面

一旦你订阅了执行页面上的执行后，你可以使用执行报告页面运行基于定单代号、定单参考号、VOL定单签、或策略的报告。

从程序设计的角度看，执行报告页面是一个你怎样从[命名区域](#)收到数据时提取批量订阅数据，以及这些数据是如何在你客户化的Excel的DDE API应用程序中应用的一个实际例子。



### 运行执行报告

**Note:** 确认TWS处于运行状态，并且你以及在哪一个交易平台？中的用户名区域中输入了你的用户名，包括Excel表格中所有页面，才能保证与TWS的连接。

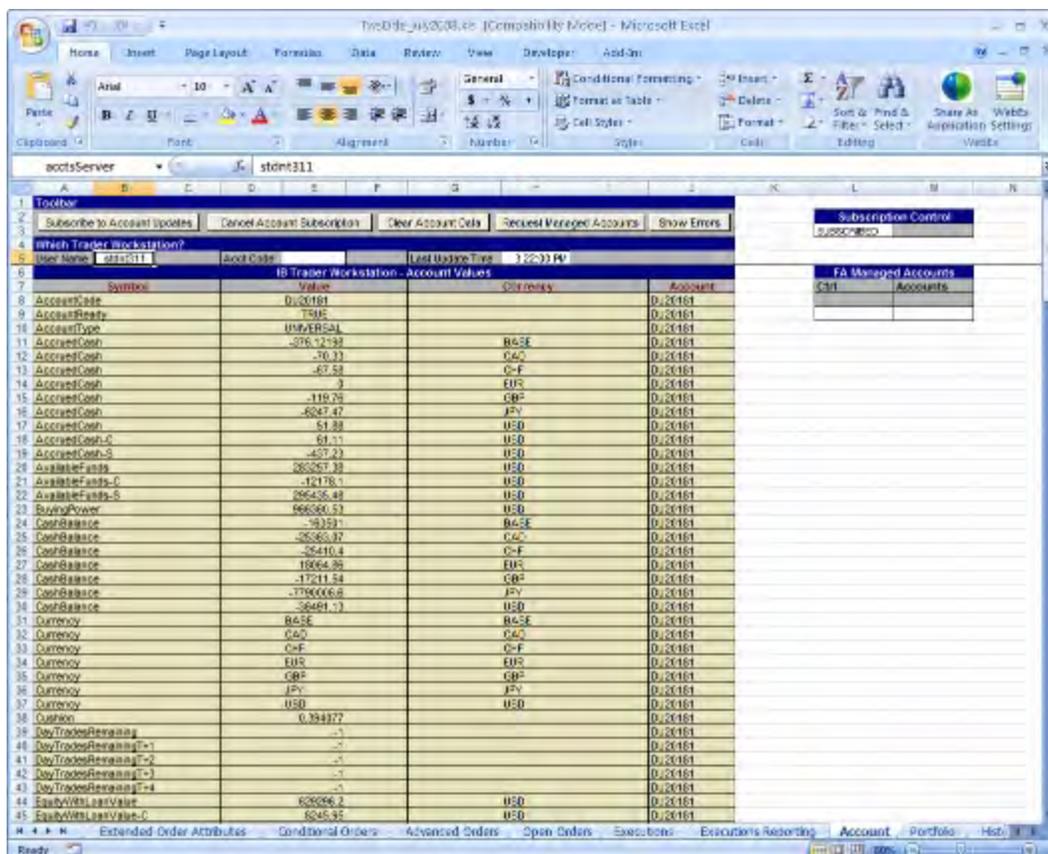
### 运行执行报告

1. 在执行页面上，点击工具栏上的**订阅执行**按钮。
2. 点击工作表底部的**执行报告**标签。
3. 在类型区域选择：
  - *Order ID*( 定单代号) - 找出所有具有特定PermID定单的执行。
  - *Order Ref*( 定单参考) - 找出所有具有定单参考定单的执行；例如一个特定篮子定单的执行。
  - *VOL order*( 波动率定单) - 找出所有特定波动率定单的执行，包括任何对冲delta执行。
  - *Strategy*( 策略) - 在标签区域，输入一个值来定义你选择的类型。例如，如果你选择了*Order ID*做为类型，在标签区域输入一个特定的定单代号。

### 账户页面

使用账户页面来完成：

- 查看包括你当前含借贷价值的股权和可用资金的详细信息。
- 查看顾问管理的账户代码的列表。
- 查看你当前的投资组合。



### 使用账户页面

**Note:** 确认TWS处于运行状态，并且你以及在哪个交易平台？中的用户名区域中输入了你的用户名包括Excel表格中所有页面，才能保证与TWS的连接。

### 查看账户信息

1. 点击表格底部的**账户**标签。
2. 点击工具栏上的**订阅账户更新**按钮。

### 取消账户信息

1. 点击**取消账户订阅**按钮。
2. 点击**清除账户数据**按钮。

### 请求金融顾问(FA)管理的账户代码列表

1. 点击表格底部的**账户**标签。
2. 点击**请求管理的账户**按钮。

### 请求金融顾问(FA)管理的账户的详细内容

1. 点击表格底部的**账户**标签。
2. 在哪个交易平台? 区域中的**账户代码**区域输入你希望查看详细的账户代码。
3. 点击**请求管理的账户**按钮。

### 账户页面工具栏按钮

账户页面上工具栏包括下列按钮。

按钮	描述
<b>Subscribe to Account Updates( 订阅账户更新)</b>	每次点击该按钮给出特定账户值的数据。账户投资组合部分之前的所有空白行将显示数据。连续点击直到所有行都充满数据。
<b>Cancel Account Subscription( 取消账户订阅)</b>	对你持有的每个头寸点击一次这个按钮。当你看到一个行内出现“0”时，即所有当前的头寸已被全部下载。这些值继续被实时更新。
<b>Clear Account Data( 清除账户数据)</b>	从页面上清除所有信息。在你可以清除数据之前你必须先要取消你的订阅。
<b>Request Managed Accounts( 请求管理的账户)</b>	针对顾问账户，对每个账户点击一次这个按钮。
<b>Show Errors( 显示错误)</b>	跳至错误编码并显示错误编码。

### 账户页面值

账户页面显示下列值：

区域	描述	注释
<i>Account Code</i> ( 账户代码)	账户号码。	
<i>Account Ready</i> ( 账户准备)	仅内部使用。	
<i>Account Type</i> ( 帐户类型)	确认IB帐户类型。	

<i>Accrued Cash</i> ( 账户现金)	反映当前月份直至今日累计的借贷和信用利息，并每日更新。	每个月的开始时，上一个月累计的数字被加到现金余额中，这个区域被归零。
<i>Available Funds</i> ( 可用资金)	对股票账户： 含借贷股权－初始保证金 对商品账户： 净清算值－初始保证金	
<i>Buying Power</i> ( 交易能力)	现金账户： ( 最小( 含借贷值股权，前一天含借贷值股权) －初始保证金) 标准保证金账户： 可用资金*4	
<i>Cash Balance</i> ( 现金余额)	对股票账户： 结算的现金＋交易时交易值 对商品账户： 结算的现金＋交易时交易值＋期货PNL	
<i>Currency</i> ( 货币)	显示市值区域部分内列出的货币类型。	
<i>Cushion</i> ( 缓冲)	显示你当前保证金缓冲。	
<i>Day Trades Remaining</i> ( 日交易剩余时间)	日交易人模式期间剩余的交易日天数。	
<i>Day Trades Remaining T+1, T+2, T+3, T+4</i> ( 日交易剩余时间 T+1、T+2、T+3、T+4)	4天日交易人模式中所剩余的交易日天数。	

<i>Equity With Loan Value</i> ( 含借贷值股权)	<p>对股票账户： 现金账户： 结算的现金</p> <p>保证金账户： 总现金值+股票值+债券值+( 非美国 &amp;加拿大股票期权值)</p> <p>对商品账户： 现金账户： 总现金值+商品期权值- 期货维持保证金要求+最小( 0, 期 货PNL)</p> <p>保证金账户： 总现金值+商品期权值-期货维持保 证金要求</p>	
<i>Excess Liquidity</i> ( 多余流 动性)	含借贷值股权 - 维持保证金	
<i>Exchange Rate</i> ( 汇率)	相对于你基础货币的货币汇率。	
<i>Full Available Funds</i> ( 全 部可用资金)	<p>对股票账户： 含借贷值股权 - 初始保证金</p> <p>对商品账户： 净清算值 - 初始保证金</p>	
<i>Full Excess Liquidity</i> ( 全 部多余流动性)	含借贷值股权 - 维持保证金	
<i>Full Init Margin Req</i> ( 全 部初始保证金要求)	按账户基础货币计算的隔夜初始保证 金要求	
<i>Full Maint Margin Req</i> ( 全部维持保证金要求)	做为下一个区间内按账户基础货币计 算的保证金变化的维持保证金要求。	
<i>Future Option Value</i> ( 期 货期权值)	期货期权的实时市值。	
<i>Futures PNL</i> ( 期 货PNL)	自最后结算的期货值的实时改变。	
<i>Gross Position Value</i> ( 总 头寸值)	做多股票值+做空股票值+做多期权 值+做空期权值。	
<i>Init Margin Req</i> ( 初始保 证金要求)	按账户基础货币计算的初始保证金要 求。	

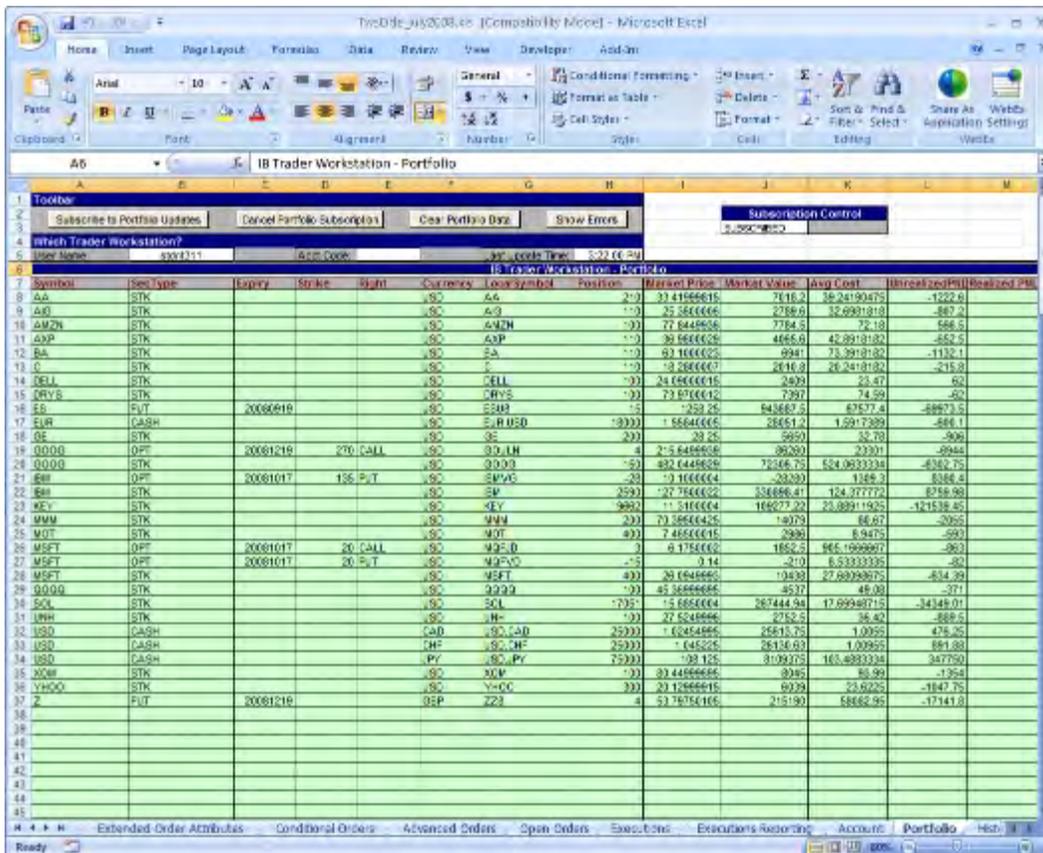


<i>Leverage</i> ( 杠杆)	对股票账户： 总头寸值/净清算值  对商品账户： 净清算值－初始保证金	
<i>Look Ahead Available Funds</i> ( 预期可用资金)	对股票账户： 含借贷值股权－预期初始保证金。  对商品账户： 净清算值－预期初始保证金。	
<i>Look Ahead Excess Liquidity</i> ( 预期多余流动性)	含借贷值股权－预期维持保证金。	
<i>Look Ahead Init Margin Req</i> ( 预期初始保证金要求)	做为下一个区间内按账户基础货币计算的保证金变化的初始保证金要求。	
<i>Look Ahead Maint Margin Req</i> ( 预期维持保证金要求)	按账户的基础货币计算的下一个期间保证金变化的维持保证金要求。	
<i>Maint Margin Req</i> ( 维持保证金要求)	按账户基础货币计算的维持保证金要求。	
<i>Net Liquidation</i> ( 净清算)	对股票账户： 总现金值+股票值+证券期权值+债券值  对商品账户： 总现金值+商品期权值	
<i>Net Liquidation by Currency</i> ( 按货币的净清算)	对单个货币如上。	
<i>Option Market Value</i> ( 期权市值)	股票期权的实时市值。	
<i>PNL</i>	你的未平仓的当前市值和平均成本的差，或值－平均成本。	
<i>Previous Day Equity with Loan Value</i> ( 前一天的含借贷值股权)	前一天东部时间16:00的可用于保证金的含借贷值股权，仅用于股票。	

<i>Realized PnL</i> ( 可实现的PnL)	显示已平仓头寸的利润，是你的买进执行成本和卖出执行成本的差，或(开仓的执行价格+佣金) -(平仓的执行价格+佣金)。	
<i>Reg T Equity</i> ( Reg T股票)	按美国法规T规则计算的初始保证金要求。	
<i>Reg T Margin</i> ( Reg T保证金)	对股票账户： 现金账户： 结算的现金 保证金账户： 总现金值+股票值+证券值+(非美国 &加拿大股票期权值)  对商品账户： 现金账户： 总现金值+商品期权值-期货维持保证金要求+最小( 0, 期货PNL) 保证金账户： 总现金值-期货维持保证金要求	
<i>SMA</i>	最大( ( EWL - 美国初始保证金要求) *, ( 前一天SMA +/- 日内现金的变化 +/- 美国初始保证金要求** 对日内所进行的交易。) ) *在交易日结束时按美国股票规则结算的，不论交易的国家。 **在交易发生时	仅用于股票。
<i>Stock Market Value</i> ( 股票市值)	股票的实时市值	
<i>Total Cash Balance</i> ( 总现金余额)	交易时确认的现金+期货PNL	
<i>Total Cash Value</i> ( 总现金值)	总的股票、商品和证券的现金值	

### 投资组合页面

投资组合页面显示所有你当前的头寸。这个页面于TWS接通并每三分钟将值更新一次，你可以在页面的哪个交易平台？部分的最后更新时间区域内看到最后更新的时间。



查看你的投资组合

**Note:** 确认TWS处于运行状态，并你已经在Excel表格中所有页面的哪个交易平台？部分的用户名区域输入了你的用户名，以实现和TWS准确连接。

查看你的投资组合

1. 点击表格底部的投资组合标签。
2. 点击订阅投资组合更新按钮。

删除投资组合信息

1. 点击取消投资组合订阅按钮。
2. 点击清除投资组合数据按钮。

投资组合页面工具栏按钮

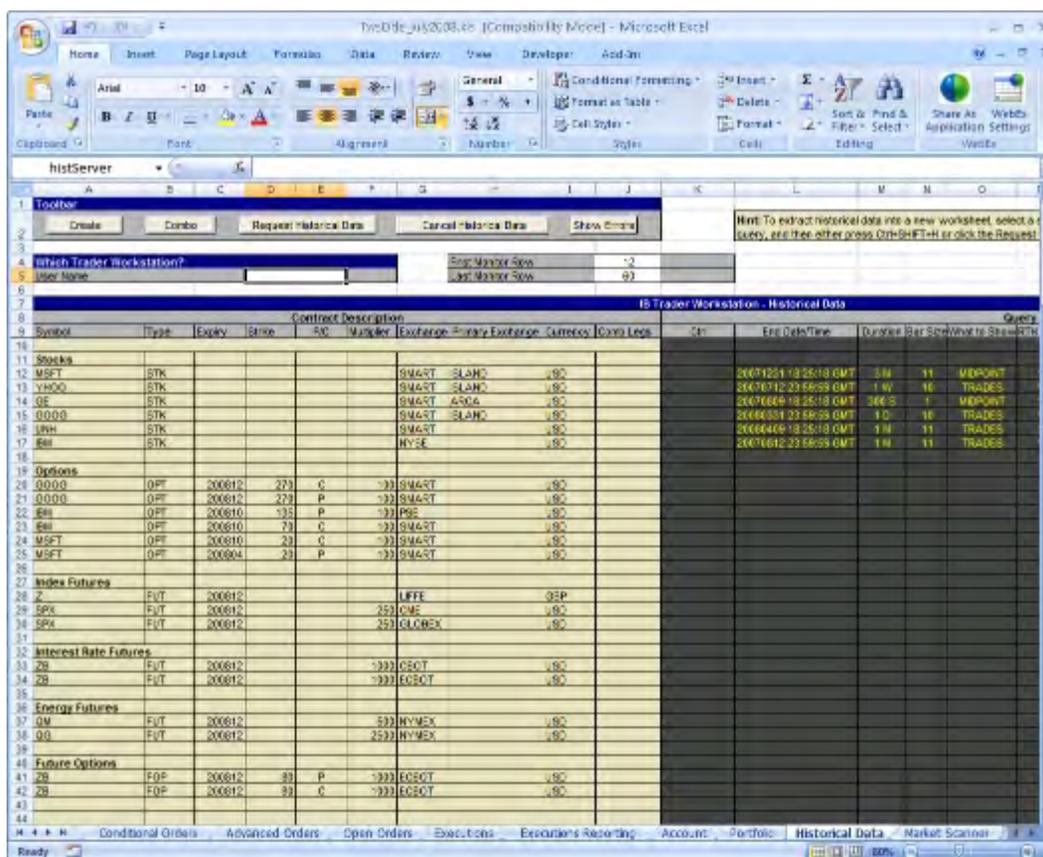
投资组合页面上的工具栏包括下列按钮。

按钮	描述
Subscribe to Portfolio Updates( 订阅投资组合更新)	点击查看所有当前投资组合数据。

<b>Cancel Portfolio Subscription( 取消投资组合订阅)</b>	取消和更新你投资组合信息的TWS的连接。
<b>Clear Portfolio Data( 清除投资组合数据)</b>	从页面中删除所有数据。你必须在可以清除所有数据请取消你的订阅。
<b>Show Errors( 显示错误)</b>	跳至错误编码区域和显示错误编码。

### 历史数据页面

使用历史数据页面请求交易产品的历史数据( 根据寻求区域中你输入的数据)。寻求的结果显示在另外个工作页面中, 在该页面当前不存在的情况下将为结果创建一个新的页面。注意因为查询的结果为在一个命名的单元格范围内, 你可以编制一个VBA宏程序对其进行计算, 你还可以绘制图表和在Excel中排列数据。



**Note:** 有关历史数据寻求限制的信息, 请参阅[历史数据限制](#)章节。

### 查看历史数据

**Note:** 确认TWS处于运行状态, 并你已经在Excel表格中所有页面的哪个交易平台? 部分的用户名区域输入了你的用户名, 以实现和TWS准确连接。

### 请求历史数据

1. 点击表格底部的**历史数据**标签。
2. 通过填写页面的**合约描述**区域创建代码，或点击工具栏上的**创建代码**按钮和在代码框中输入要求的信息。
3. 在**寻求详述**区域输入你的寻求参数。包括完成寻求区域的描述，历史数据页面寻求详细区域。
4. 选择行，然后点击**请求历史数据**按钮。当**Ctrl(控制)**区域显示“Finished(完成)”，时，结果即显示在特定的页面上。

### 请求已过期合约的历史数据

1. 在历史数据页面上，通过填写页面的**合约描述**区域创建代码，或点击工具栏上的**创建代码**按钮和在代码框中输入要求的信息。
2. 在**寻求详述**区域输入你的寻求参数。
3. 在寻求详细部分的**过期**区域中，输入**TRUE(真)**。
4. 选择行，然后点击**请求历史数据**按钮。当**Ctrl(控制)**区域显示“Finished(完成)”，时，结果即显示在特定的页面上。

**Note:** 对过期合约的历史数据寻求被限制在合约期间的最后一年。

下图显示一个典型的历史数据结果页面。

	A	B	C	D	E	F	G	H	I	J	K	L	M
1													
2		DATE/TIME: 20060609	OPEN	HIGH	LOW	CLOSE	VOLUME	WAP	HAS GAPS				
3		14:20:18	77.89	77.89	77.89	77.89	-1	-1	FALSE				
4		14:20:19	77.89	77.89	77.89	77.89	-1	-1	FALSE				
5		14:20:20	77.89	77.89	77.89	77.89	-1	-1	FALSE				
6		14:20:21	77.89	77.89	77.89	77.89	-1	-1	FALSE				
7		14:20:22	77.89	77.89	77.89	77.89	-1	-1	FALSE				
8		14:20:23	77.89	77.89	77.89	77.89	-1	-1	FALSE				
9		14:20:24	77.89	77.89	77.89	77.89	-1	-1	FALSE				
10		14:20:25	77.89	77.89	77.89	77.89	-1	-1	FALSE				
11		14:20:26	77.89	77.89	77.89	77.89	-1	-1	FALSE				
12		14:20:27	77.89	77.89	77.89	77.89	-1	-1	FALSE				
13		14:20:28	77.89	77.89	77.89	77.89	-1	-1	FALSE				
14		14:20:29	77.89	77.89	77.89	77.89	-1	-1	FALSE				
15		14:20:30	77.89	77.89	77.89	77.89	-1	-1	FALSE				
16		14:20:31	77.89	77.89	77.89	77.89	-1	-1	FALSE				
17		14:20:32	77.89	77.89	77.89	77.89	-1	-1	FALSE				
18		14:20:33	77.89	77.89	77.89	77.89	-1	-1	FALSE				
19		14:20:34	77.89	77.89	77.89	77.89	-1	-1	FALSE				
20		14:20:35	77.89	77.89	77.89	77.89	-1	-1	FALSE				
21		14:20:36	77.89	77.89	77.89	77.89	-1	-1	FALSE				
22		14:20:37	77.89	77.89	77.89	77.89	-1	-1	FALSE				
23		14:20:38	77.89	77.89	77.89	77.89	-1	-1	FALSE				
24		14:20:39	77.89	77.89	77.89	77.89	-1	-1	FALSE				
25		14:20:40	77.89	77.89	77.89	77.89	-1	-1	FALSE				
26		14:20:41	77.89	77.89	77.89	77.89	-1	-1	FALSE				
27		14:20:42	77.89	77.89	77.89	77.89	-1	-1	FALSE				
28		14:20:43	77.89	77.89	77.89	77.89	-1	-1	FALSE				
29		14:20:44	77.89	77.89	77.89	77.89	-1	-1	FALSE				
30		14:20:45	77.89	77.89	77.89	77.89	-1	-1	FALSE				
31		14:20:46	77.89	77.89	77.89	77.89	-1	-1	FALSE				
32		14:20:47	77.89	77.89	77.89	77.89	-1	-1	FALSE				
33		14:20:48	77.89	77.89	77.89	77.89	-1	-1	FALSE				

### 历史数据页面工具栏按钮

历史数据页面上的工具栏包括以下按钮。

按钮	描述
<b>Create Ticker( 创建代码)</b>	打开代码框。输入信息创建一个市场数据行。
<b>Combo Legs( 组合边)</b>	打开组合边框。输入合约详细 创建每个组合定单边。
<b>Request Historical Data( 请求历史数据)</b>	对TWS提交你的历史数据寻求和在不同的工作页面上显示结果。
<b>Cancel Historical Data( 取消历史数据)</b>	取消历史数据请求。
<b>Show Errors( 显示错误)</b>	跳至错误编码区域和显示错误编码。

### 历史数据页面寻求详细区域

历史数据寻求包括下列区域：

参数	描述
<b>End Date/Time( 截止日期/时间)</b>	使用格式 <code>yyyymmdd {空格}hh:mm:ss{空格}时区</code> 在时区为允许使用时( 可选) ， 在空格之后的结尾处。
<b>Duration( 区间)</b>	<p>这是请求将覆盖的时间跨度，使用特定的格式整数 {空格} 单位，有效的单位包括：</p> <ul style="list-style-type: none"> <li>• S ( 秒)</li> <li>• D ( 日)</li> <li>• W ( 星期)</li> <li>• Y ( 年)</li> </ul> <p>单位目前限制为一个。如果没有指定单位，将使用秒。</p>

<p>Bar Size( 柱尺寸)</p>	<p>指定结果显示的柱的尺寸。可以使用下列柱尺寸，并使用特定的参数值：</p> <p><b>柱尺寸表示 - 整数值</b></p> <ul style="list-style-type: none"> <li>• 1 秒 - 1</li> <li>• 5 秒 - 2</li> <li>• 15 秒 - 3</li> <li>• 30 秒 - 4</li> <li>• 1 分钟 - 5</li> <li>• 2 分钟 - 6</li> <li>• 3 分钟 - 16</li> <li>• 5 分钟 - 7</li> <li>• 15 分钟 - 8</li> <li>• 30 分钟 - 9</li> <li>• 1 小时 - 10</li> <li>• 1 天 - 11</li> </ul> <p>在寻求结果页面上，每个“柱”由表格中的一条线表示。如果你指定的区间为300秒，和柱尺寸为“1”( 1秒) ，寻求的结果将包括300线，每个线的值等于1秒，或是一个1秒的柱。注意，你可以使用柱尺寸表中的整数值，或整数值来定义柱尺寸。</p>
<p>What to Show( 显示内容)</p>	<p>决定提取数据的性质。有效的值包括：</p> <ul style="list-style-type: none"> <li>• 交易</li> <li>• 中点</li> <li>• 买价</li> <li>• 卖价</li> <li>• 买_卖价</li> <li>• 历史波动率</li> <li>• 期权隐含波动率</li> </ul> <p>所有数据中只有买价/卖价数据包括开始时间、开盘、高、低、收盘、交易量和寻求区间的加权平均价。</p> <p>对买价/卖价寻求，开盘和收盘值分别是时间加权平均值买价和时间加权平均值卖价。这些柱和TWS图表的烛柱图中的柱是相同的。</p>

<p>RTH Only( 仅正常交易时段)</p>	<p>仅正常交易时段。有效的值包括：</p> <ul style="list-style-type: none"> <li>• 0 - 结果包括寻求期间的所有可用数据，包括时间区间在寻求市场的正常交易时段以外的区段。</li> <li>• 1 - 结果仅包括寻求产品在正常交易时段的数据，不论时间区间是部分或全部位于正常时段以外的情况下。</li> </ul>
<p>Date Format Style( 日期格式)</p>	<p>有效值包括：</p> <ul style="list-style-type: none"> <li>• 1 - 用于柱的日期的结果的格式为yyyyymmdd{空格}{空格}hh:mm:dd ( 相同的格式也用于报告执行时) 。</li> <li>• 2 - 日期的结果为整数，以自格林威治标准时间1/1/1970的秒的整数。</li> </ul>
<p>Page Name( 页面名称)</p>	<p>结果页面的名称。这个名称出现在工作表格底部的结果标签上。</p>
<p>Expired( 过期的)</p>	<p>有效值：TRUE( 真) 、FALSE( 假)                  如果为TRUE，数据寻求可以包括过期的期货合约，，限制为合约期间的最后一年。</p>

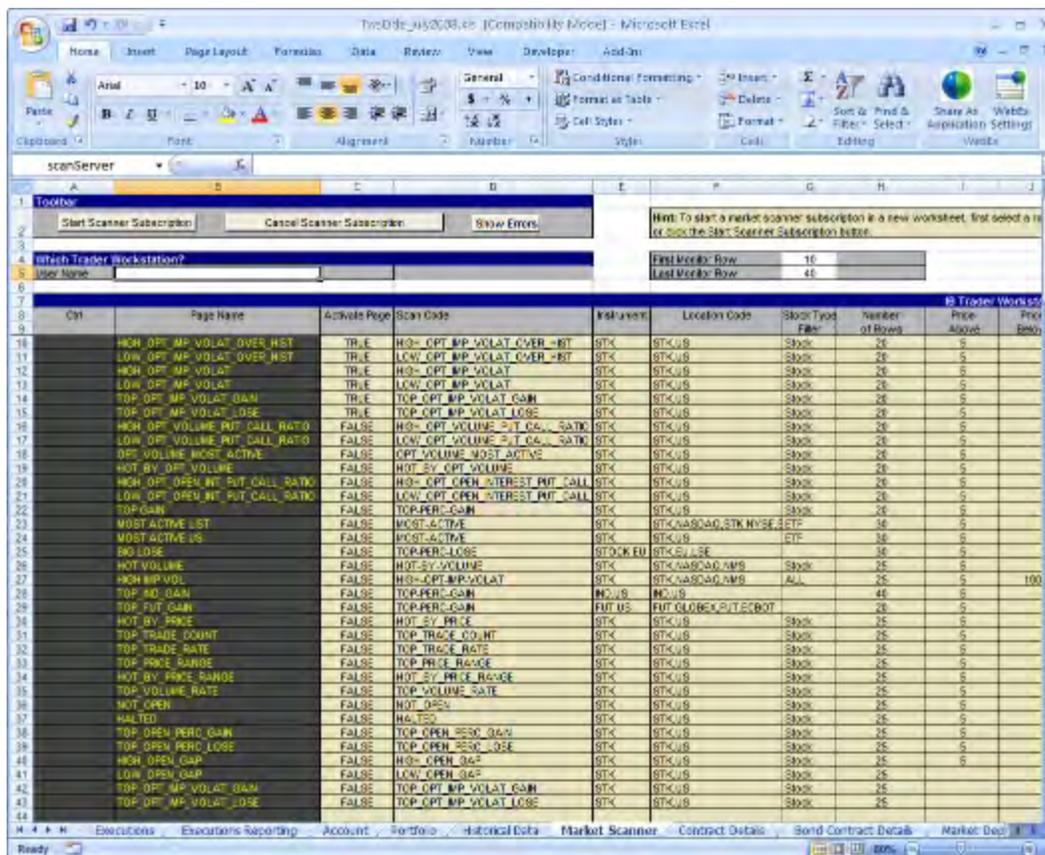
对请求的区段为300秒和1秒的柱范围内，寻求的结果显示如( 页面右侧的滚动条允许你下拉并查看所有300个柱) 。

注意，新的页面将被添加到工作表格现有标签的右侧。

### 扫描仪页面

使用市场扫描仪页面订阅TWS市场扫描仪。这些扫描仪允许你定义条件和设定过滤器，以取得满足所有扫描条件的x数量的底层证券。扫描是连续实时更新的。





### 开始市场扫描仪订阅

**Note:** 确认TWS处于运行状态，并你已经在Excel表格中所有页面的哪个交易平台？部分的用户名区域输入了你的用户名，以实现和TWS准确连接。

### 开始扫描仪订阅

1. 点击表格底部的市场扫描仪标签。
2. 选中一个现有的扫描仪行，或输入一个不同市场扫描仪的信息：
  - a. 在页面名称区域输入扫描结果页面的名称。
  - b. 在启动页面区域输入**TRUE**或**FALSE**。

设定这些区域为TRUE将限定扫描结果在每次更新时出现在你的应用程序的前面。如想结束这个设置，将这个区域的值设为FALSE。

- c. 对页面的阴影区域的其它的扫描参数输入值。
3. 点击工具栏上的**开始扫描仪订阅**按钮。一个新的扫描仪页面即被创建完成，并在订阅被处理后显示。

### 市场扫描仪参数

下面的表格描述组成扫描仪订阅的市场扫描仪参数。

参数	描述
<i>Page name</i> ( 页面名称)	使用扫描仪数据创建的新的页面的名称。
<i>Activate Page?</i> ( 启动页面?)	如果设置为 <b>true</b> ( 真) ，则新的扫描仪页面将在每次扫描结果更新时显示在工作表格的上面。这可能会每分钟发生一次。
<i>Scan Code</i> ( 扫描代码)	扫描类型。
<i>Instrument</i> ( 产品)	扫描中使用的产品类型。
<i>Location code</i> ( 市场代码)	扫描应用的市场( 比如， 美国股票) 。
<i>Stock type filter</i> ( 股票类型过滤器)	允许你指定只用于股票、只用于交易所指数基金( ETFs) ， 或两者。
<i>Number of rows</i> ( 行的数目)	结果显示的数据行的数目。
<i>Price Above</i> ( 价格之上)	过滤掉价格低于指定价格的结果。可能会没有结果。
<i>Price Below</i> ( 低于价格)	过滤掉价格高于指定价格的结果。可能会没有结果。
<i>Average volume above</i> ( 平均交易量之上)	过滤掉低于指定价格的平均交易量结果。可能会没有结果。
<i>Average Option Volume Above</i> ( 平均期权交易量之上)	过滤掉低于指定价格的平均期权交易量的结果。可能会没有结果。
<i>Market Cap Above</i> ( 市值之上)	过滤掉低于指定价格的市值结果。可能会没有结果。
<i>Market Cap Below</i> ( 市值以下)	过滤掉高于指定价格的市值结果。可能会没有结果。
<i>Moody Rating Above</i> ( 穆迪评级之上)	过滤掉低于指定价格的穆迪评级结果。可能会没有结果。
<i>Moody Rating Below</i> ( 穆迪评级以下)	过滤掉高于指定价格的穆迪评级结果。可能会没有结果。
<i>S &amp; P Rating Above</i> ( 标普评级之上)	过滤掉低于指定价格的标普评级结果。可能会没有结果。

<i>S &amp; P Rating Below</i> ( 标普评级以下)	过滤掉高于指定价格的标普评级结果。可能会没有结果。
<i>Maturity Date Above</i> ( 到期日期之上)	过滤掉低于指定价格的到期日期结果。可能会没有结果。
<i>Maturity Date Below</i> ( 到期日期之下)	过滤掉高于指定价格的到期日期结果。可能会没有结果。
<i>Coupon Rate Above</i> ( 利率之上)	过滤掉低于指定价格的利率结果。可能会没有结果。
<i>Coupon Rate Below</i> ( 利率以下)	过滤掉高于指定价格的利率结果。可能会没有结果。
<i>Exclude Convertible</i> ( 不包括可转换债券)	过滤掉可转换债券。可能会没有结果。
<i>Scanner Settings Pairs</i> ( 扫描仪设置对)	例如，用于扫描最高期权隐含波动率百分比赢家的“Annual/True(年/真)”将指示扫描返回年度波动率结果。由斜线分隔设置对。

### 市场扫描仪页面工具栏按钮

市场扫描仪页面上的工具栏包括下列按钮。

按钮	描述
<b>Start Scanner Subscription</b> ( 开始扫描仪订阅)	创建和显示选取市场扫描仪扫描结果的新页面。
<b>Cancel Scanner Subscription</b> ( 取消扫描仪订阅)	取消市场扫描仪。
<b>Show Errors</b> ( 显示错误)	跳至错误编码区域和显示错误编码。

### 可用的市场扫描仪

下面表格中列出Excel的DDE API表格中可用的市场扫描仪。

**Note:** 如想获得比Excel的DDE API表格的扫描仪更详细的市场扫描结果，请在TWS中运行市场扫描仪。

市场扫描仪 (扫描代码)	描述
-----------------	----

Low Opt Volume P/C Ratio( 低期权交易量看跌/看涨比率) (LOW_OPT_VOL_PUT_CALL_RATIO)*	看跌期权交易量除以看涨期权交易量，显示具有最低比率的底层代码。
High Option Imp Vol Over Historical( 区间内的高期权隐含波动率) (HIGH_OPT_IMP_VOLAT_OVER_HIST)*	显示隐含和历史波动率之间最大差异的底层合约( 股票或指数) 。
Low Option Imp Vol Over Historical( 区间内的低期权隐含波动率) (LOW_OPT_IMP_VOLAT_OVER_HIST)*	显示隐含和历史波动率之间最小差异的底层合约( 股票或指数) 。
Highest Option Imp Vol( 最高期权隐含波动率) (HIGH_OPT_IMP_VOLAT)*	显示下两个月内到期的近市价执行价期权的具有最高vega-加权隐含波动率的底层合约( 股票或指数) 。
Top Option Imp Vol % Gainers( 期权隐含波动率百分比的赢家) (TOP_OPT_IMP_VOLAT_GAIN)*	显示当前隐含波动率和昨天收盘前15分钟的平均隐含波动率值之间的最大百分比赢家的底层合约( 股票或指数) 。
Top Option Imp Vol % Losers( 期权隐含波动率波动率输家) (TOP_OPT_IMP_VOLAT_LOSE)*	显示当前隐含波动率和昨天收盘前15分钟的平均隐含波动率值之间的最大百分比输家的底层合约( 股票或指数) 。
High Opt Volume P/C Ratio( 高期权交易量看跌/看涨比率) (HIGH_OPT_VOLUME_PUT_CALL_RATIO)	看跌期权交易量除以看涨期权交易量和显示具有最高比率的底层代码。
Low Opt Volume P/C Ratio( 低期权交易量看跌/看涨比率) (LOW_OPT_VOLUME_PUT_CALL_RATIO)	看跌期权交易量除以看涨期权交易量，显示具有最低比率的底层代码。
Most Active by Opt Volume( 按期权交易量计算的最活跃合约) (OPT_VOLUME_MOST_ACTIVE)	按期权交易量降序排列显示最活跃合约。
Hot by Option Volume( 按期权交易量计算的最热门合约) (HOT_BY_OPT_VOLUME)	显示过去10天内平均最高期权交易量的底层合约。
High Option Open Interest P/C Ratio( 高期权未平仓合约看跌/看涨比率) (HIGH_OPT_OPEN_INTEREST_PUT_CALL_RATIO)	显示50个具有 <b>最高</b> 看跌/看涨比率的未平仓合约。

<p>Low Option Open Interest P/C Ratio( 低期权未平仓合约看跌/看涨比率) (LOW_OPT_OPEN_INTEREST_PUT_CALL_RATIO)</p>	<p>给出50个具有<b>最低</b>看跌/看涨比率的未平仓合约。</p>
<p>Top % Gainers( 最高百分比赢家) (TOP_PERC_GAIN)</p>	<p>合约的最后交易价和前一天收盘价比较具有最高百分比变化的合约。</p>
<p>Most Active( 最活跃合约) (MOST_ACTIVE)</p>	<p>基于TWS使用的单位( 美国股票的Lots; 衍生品和非美国股票的合约) 计算的今天具有最高交易量的合约。  样本表格的两个最活跃扫描: 最活跃列表, 用来显示NASDAQ、NYSE和AMEX市场上的最活跃合约; 最活跃美国, 用来显示美国的最活跃股票。</p>
<p>Top % Losers( 最高百分比输家) (TOP_PERC_LOSE)</p>	<p>合约的最后交易价和前一天收盘价比较具有最低百分比变化的合约。</p>
<p>Hot Contracts by Volume( 按交易量计算的最热门合约) (HOT_BY_VOLUME)</p>	<p>合约为:</p> <ul style="list-style-type: none"> <li>• 今天的交易量/平均日交易量为最高。</li> <li>• 平均日交易量( avgDailyVolume) 是合约日交易量的30天日平均值。</li> </ul>
<p>Top % Futures Gainers( 最高百分比期货赢家) (TOP_PERC_GAIN)</p>	<p>期货的最后交易价和前一天收盘价比较具有最高百分比增加的合约。</p>
<p>Hot Contracts by Price( 按价格衡量的最热门合约) (HOT_BY_PRICE)</p>	<p>合约为:</p> <ul style="list-style-type: none"> <li>• ( 最后交易价格 - 前一天收盘价) / 平均日变化的绝对值( 正或负) 最高的合约。</li> <li>• 平均日变化( avgDailyChange) 的定义为合约的合约的指数移动平均( 日收盘 - 日开盘)</li> </ul>
<p>Top Trade Count( 最多交易数) (TOP_TRADE_COUNT)</p>	<p>日内的最多交易数。</p>
<p>Top Trade Rate( 最高交易比率) (TOP_TRADE_RATE)</p>	<p>过去60秒内具有最高交易次数的合约( 不考虑交易的尺寸) 。</p>
<p>Top Price Range( 最大价格范围) (TOP_PRICE_RANGE)</p>	<p>今天的高和低( 或昨天的收盘价, 如果超出今天的范围) 的最大差。</p>

Hot by Price Range( 按价格范围计算的最热门合约) (HOT_BY_PRICE_RANGE)	最大的价格波动范围( 从最大价格范围计算) 。
Top Volume Rate( 最高交易量比率) (TOP_VOLUME_RATE)	每分钟的最高交易量比率。
Lowest Option Imp Vol( 最低期权隐含波动率) (LOW_OPT_IMP_VOLAT)	显示下两个月内到期的具有最低vega加权隐含波动率的近市价执行价期权的底层合约( 股票或指数) 。
Most Active by Opt Open Interest( 最活跃未平仓期权) (OPT_OPEN_INTEREST_MOST_ACTIVE)	显示头50个( 最高数目的未平仓看涨合约) +( 最高数目的未平仓看跌合约) 的底层合约。
Not Open( 无交易合约) (NOT_OPEN)	今天还没有交易的合约。
Halted( 暂停) (HALTED)	被暂停交易的合约。
Top % Gainers Since Open( 自开盘最高百分比赢家) (TOP_OPEN_PERC_GAIN)	显示最后交易和开盘价之间具有最高百分比价格增加的合约。
Top % Losers Since Open( 自开盘最大百分比输家) (TOP_OPEN_PERC_LOSE)	显示最后交易和开盘价之间具有最高百分比价格减少的合约。
Top Close-to-Open % Gainers( 最高收盘-开盘百分比赢家) (HIGH_OPEN_GAP)	显示前一天收盘价和今天开盘价之间具有最高百分比价格增加的合约。
Top Close-to-Open % Losers( 最高收盘-开盘百分比输家) (LOW_OPEN_GAP)	显示前一天收盘价和今天开盘价之间具有最低百分比价格减少的合约。
Lowest Option Imp Vol( 最低期权隐含波动率) (LOW_OPT_IMP_VOLAT)*	显示下两个月内到期的具有最低vega加权隐含波动率的近市价执行价期权的底层合约( 股票或指数) 。
Top Option Imp Vol % Gainers( 期权隐含波动率百分比的赢家) (TOP_OPT_IMP_VOLAT_GAIN)*	显示当前隐含波动率和昨天收盘值的15分钟的评级隐含波动率之间的最大百分比赢家的底层合约( 股票或指数) 。

Top Option Imp Vol % Losers( 期权隐含波动率波动率输家) (TOP_OPT_IMP_VOLAT_LOSE)*	显示当前隐含波动率和昨天收盘前15分钟的平均隐含波动率值之间的最大百分比输家的底层合约( 股票或指数) 。
13-Week High( 13个星期最高) (HIGH_VS_13W_HL)	过去13个星期内的最高价格。
13-Week Low( 13个星期最低) (LOW_VS_13W_HL)	过去13个星期内的最低价格。
26-Week High( 26个星期最高) (HIGH_VS_26W_HL)	过去26个星期内的最高价格。
26-Week Low( 26个星期最低) (LOW_VS_26W_HL)	过去26个星期内的最低价格。
52-Week High( 52个星期最高) (HIGH_VS_52W_HL)	过去52个星期内的最高价格。
52-Week Low( 52个星期最低) (LOW_VS_52W_HL)	过去52个星期内的最低价格。
<b>EFP - High Synth Bid Rev Yield( 期货换现货—最高合成投标收益)</b> (HIGH_SYNTH_BID_REV_NAT_YIELD)	突出可用的最高合成EFP利率。这些利率的计算是使用SSF( 单一股票期货) 和底层股票之间的价格差, 以及扣除股息来计算SSF合约期间的年度合成隐含利率。最高利率可能表示存在一个投资机会。
<b>EFP - Low Synth Bid Rev Yield( EFP—最低合成投标收益)</b> (LOW_SYNTH_BID_REV_NAT_YIELD)	选中可用的最低合成EFP利率。这些利率的计算是使用SSF( 单一股票期货) 和底层股票之间的价格差, 以及扣除股息来计算SSF合约期间的年度合成隐含利率。低利率可能代表一个借贷机会。

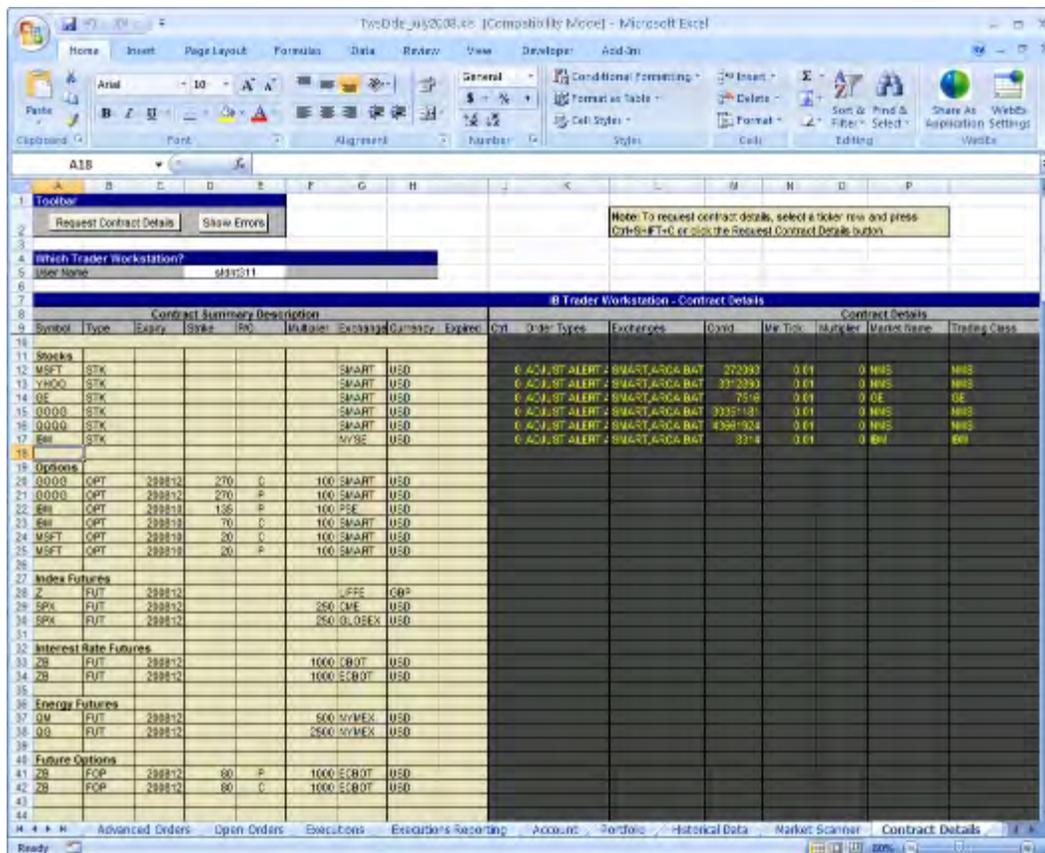
\* 30天(V30)隐含波动率:

隐含波动率的计算是对美国类型期权使用100步二叉树( 100-step binary tree) , 对欧洲类型期权使用Black-Scholes模型。利率的计算是使用当日欧元期货合约的结算价, 股息是基于历史支付信息。

盈透的30天波动率是对自当前交易未来30个日历日到期的市场波动率预期。它是基于两个相邻过期月的期权价格。第一个过期月至少还有8个日历日。隐含波动率是对8个期权的每个过期的4个最接近市场执行价的估计。隐含波动率是与做为每个过期执行价功能的抛物线相吻合。一个过期的市场隐含波动率被用作过期的预期未来价格的抛物线的值。计算基于市场波动率平方30天方差的一个线性内差( 或外推, 依据要求) 。V30是估计方差的平方根。如果60个日历天内不存在第一过期月, 我们就不计算V30。

### 合约详细页面

使用合约详细页面寻求合约特定的详细，比如支持的定单类型、有效的交易所、合约代码，等等。



### 请求合约详细

**Note:** 确认TWS处于运行状态，并且你以及在哪一个交易平台？中的用户名区域中输入了你的用户名包括Excel表格中所有页面，才能保证与TWS的连接。

### 请求一个合约的详细信息

1. 点击表格底部的**合约详细**标签打开合约详细页面。
2. 选择或输入你希望请求合约详细的合约代码。
3. 如请求一个过期合约的合约详细，在过期的区域输入**TRUE**。
4. 点击工具栏上的**请求合约详细**按钮。

### 合约详细页面工具栏按钮

合约详细页面的工具栏包括下列按钮：

按钮	描述
----	----



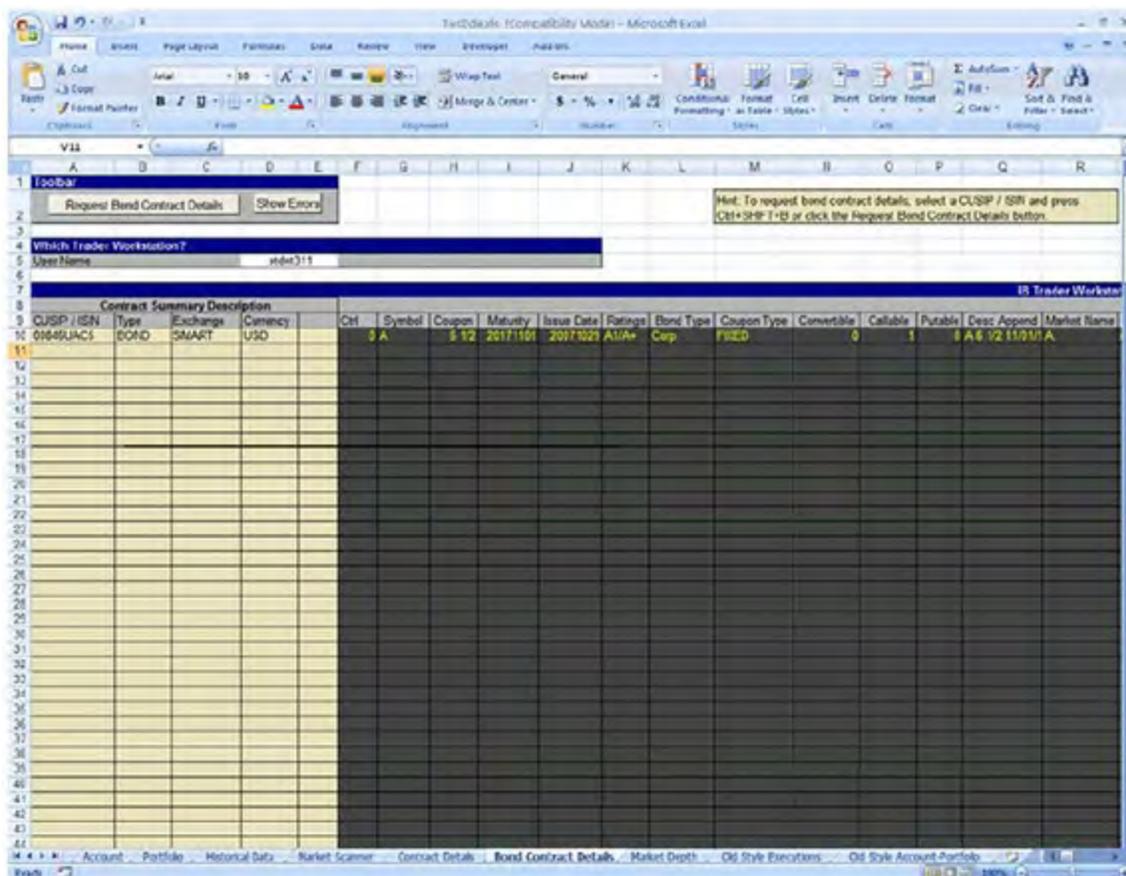
<b>Request Contract Details(请求合约详细)</b>	给出选取合约的信息。
<b>Show Errors(显示错误)</b>	跳至错误编码区域和显示错误编码。

债券合约详细页面

使用债券合约详细页面请求债券的合约特定信息，包括利率、评级、债券类型、到期日，等等。

**Note:** 从TWS版本921开始，某些债券合约数据将被限制，将不能通过API获得。所有债券合约数据将继续可以通过交易平台获得，但仅以下债券合约数据可通过API获得：

- 合约代码
- 最低Tick
- CUSIP(如果你已经订阅了CUSIP服务)
- 评级(如果你已经订阅了评级)



### 请求债券合约详细

**Note:** 确认TWS处于运行状态，并且你以及在哪一个交易平台？中的用户名区域中输入了你的用户名包括Excel表格中所有页面，以保证与TWS的连接。

### 请求债券合约的详细信息

1. 点击表格底部的**债券合约详细**标签。
2. 输入你希望请求合约详细的合约代码。
3. 点击工具栏上的**请求债券合约详细**按钮。

**Note:** 从TWS版本921开始，某些债券合约数据将被限制，将不能通过API获得。All bond contract data will continue to be 所有债券合约数据将继续可以通过交易平台获得，但仅以下债券合约数据可通过API获得：

- 合约代码
- 最低Tick
- CUSIP( 如果你已经订阅了CUSIP服务)
- 评级( 如果你已经订阅了评级)

### 债券合约详细页面工具栏按钮

债券合约详细页面上的工具栏包括下列按钮：

按钮	描述
<b>Request Bond Contract Details( 请求债券合约详细)</b>	获得选取合约的债券信息数据。
<b>Show Errors( 显示错误)</b>	跳至错误编码区域和显示错误编码。

### 市场深度页面

使用市场深度页面查看选取合约的市场深度。你还可以通过开放在册市场数据订阅查看NYSE挂牌产品的市场深度，和通过TotalView市场数据订阅查看NASDAQ挂牌产品的市场深度，如果你已经签署订阅了这些服务。



1. 从市场深度页面，按住 **Alt+F11**。

Visual Basic 编辑器(VBE)打开并显示市场深度页面的编码。

2. 在页面的编码窗口顶部的声明部分，将数字值 **numDisplayRows = 10**改为更高/低的价值，然后点击VBE工具栏上的**保存**按钮。
3. 关闭Visual Basic编辑器。

### 市场深度页面工具栏按钮

市场深度页面的工具栏包括下列按钮：

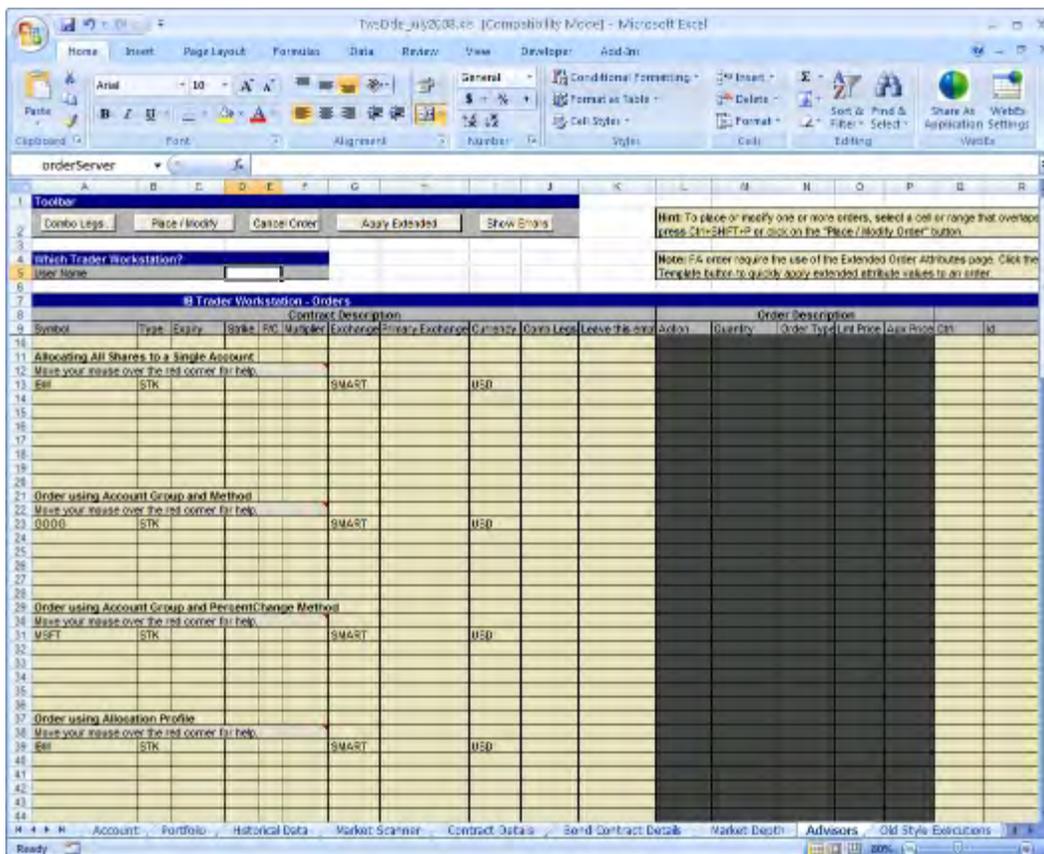
按钮	描述
<b>Request Market Depth( 请求市场深度)</b>	查看选取合约的买/卖深度价格。
<b>Cancel Market Depth(取消市场深度)</b>	取消选取合约的市场深度。
<b>Set Refresh Rate( 设置更新频率)</b>	重新设置市场数据的更新频率( 以毫秒为单位) 。
<b>Show Errors( 显示错误)</b>	跳至错误编码区域和显示错误编码。

### 顾问页面

如果你是一个金融顾问并管理多个账户，使用金融顾问页面创建FA定单：

- 分配股票到一个单一管理的账户
- 使用FA账户组和方法

- 使用分配方案



**Note:** 在你可以通过Xecel的DDE API样本表格下达FA定单前，你必须在TWS中设置你的管理账户、账户组、方法和分配方案。

### 分配股票到一个单一管理的账户

你可以使用顾问页面设置一个定单和将定单中的所有股票分配给一个单一账户。

### 分配股票到一个单一账户：

1. 在TWS中创建一个账户组。
2. 点击表格底部的顾问标签。
3. 在合约描述单元内输入合约信息，然后在定单描述单元内输入定单信息。
4. 点击扩展定单属性标签。为账户(仅机构)的扩展定单属性在值单元格内输入账户代码。
5. 点击顾问标签。
6. 选中定单行，然后点击应用扩展的按钮将账户定单属性值应用到定单中。账户值即被应用到选取的定单并显示在页面的扩展定单属性部分中。
7. 点击下载/修改定单按钮。

8. 当你完成了给账户分配定单后，从**扩展定单属性**页面中删除账户值。如果你没有将该值删除，它将被应用到所有未来从Excel的DDE API表格下达的定单。

#### 使用金融顾问(FA) 账户组和方法下达定单

你还可以通过**顾问**页面使用FA账户组和FA方法设置定单。

#### 使用FA账户组和FA方法下达定单：

1. 在TWS中创建FA账户组和FA方法。
2. 点击表格底部的**顾问**标签。
3. 在**合约描述**单元格内输入合约信息，然后在**定单描述**单元格内输入定单信息。
4. 点击**扩展定单属性**标签。为下列扩展定单属性输入值：
  - FA组 - 输入账户组名称。
  - FA方法 - 输入用于这个定单的分配方法名称。
  - FA百分比 - 输入用于这个定单的百分比变化分配方法的百分比。这个属性仅用于使用这个方法  
的FA组。
5. 点击**顾问**标签。
6. 高亮选中定单行，然后点击**应用扩展**按钮将扩展定单属性值用于定单。*FA组*、*FA方法*和 *FA百分比*的值被用于选取的定单并在页面的**扩展定单属性百分**中显示。
7. 点击**下达/修改定单**按钮。
8. 当你完成了给账户分配定单后，从**扩展定单属性**页面中删除你输入的值。如果你没有将该值删除，它将被应用到所有未来从Excel的DDE API表格下达的定单。

#### 使用分配方案下达定单

你可以使用**顾问**页面使用FA分配方案设置定单。

#### 使用FA分配方案下达定单：

1. 在TWS中创建FA分配方案。
2. 点击表格底部的**顾问**标签。
3. 在**合约描述**单元格内输入合约信息，然后在**定单描述**单元格内输入定单信息。
4. 点击**扩展定单属性**标签。在**值区域**为FA方案扩展定单属性输入分配方案名称。
5. 点击**顾问**标签。
6. 高亮选中定单行，然后点击**应用扩展**按钮将扩展定单属性值用于定单FA方案的值被应用到选取的定单并在页面的**扩展定单属性百分**中显示。
7. 点击**下达/修改定单**按钮。
8. 当你完成了给账户分配定单后，在**扩展定单属性**页面中删除你输入的FA方案值。如果你没有将该值删除，它将被应用到所有未来从Excel的DDE API表格下达的定单。

#### 顾问页面工具栏按钮

基本定单页面上的工具栏包括下列按钮：

按钮	描述
<b>Combo Legs( 组合边)</b>	打开组合边框。输入合约详细创建每个组合定单边。
<b>Place/Modify Orders( 下达修改定单)</b>	在你完成了定单描述区域和定义了任何扩展属性后，点击创建选取合约的定单。
<b>Cancel Order( 取消定单)</b>	这个按钮取消你高亮选中的定单。
<b>Apply Extended( 应用扩展)</b>	应用扩展定单属性上当前的值到选中的定单行。
<b>Show Errors( 显示错误)</b>	跳至错误编码区域和显示错误编码。

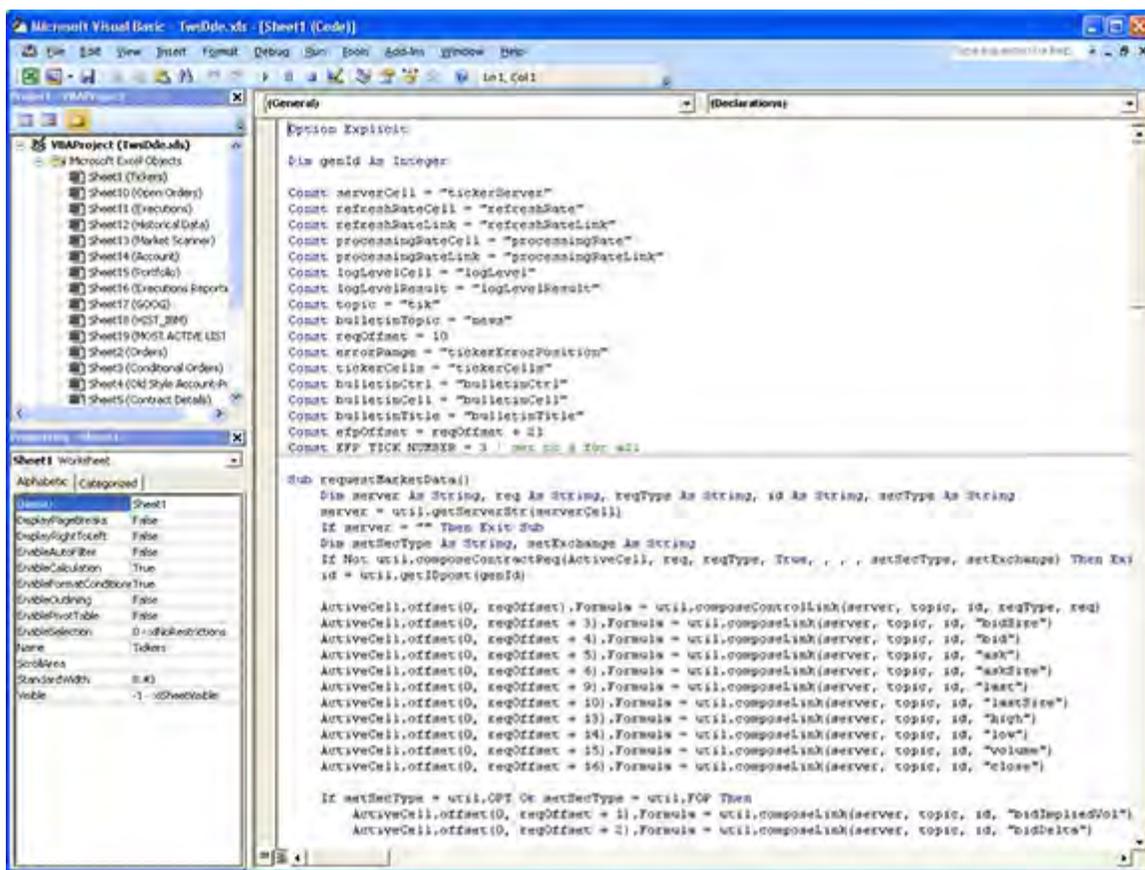
## Excel的DDE API参考

这一部分提供了关于Excel的DDE API的各种参考信息，包括下列内容：

- [查看编码](#)
- [模块](#)
- [命名范围](#)
- [宏程序](#)
- [Excel的DDE句法](#)

### 查看编码

查看用于Excel的DDE API的Visual Basic编码，在任一页面上按住 **Alt+F11** 键。Visual Basic编辑器打开：



Visual Basic编辑器包括三个主要内容：

- 项目资源管理
- 属性窗口
- 编码窗口



项目资源管理包括使用在表格中的目标列表。这些目标与表格中的页面相对应；查看特定页面的编码，双击项目资源管理中页面的相对应目标。

## 模块

Visual Basic编码包括下列模块(在VBE项目资源管理中可见)：

- ArrayQueries(数组查询)
- ErrorDisplay(错误显示)
- Orders(定单)
- util(应用工具)

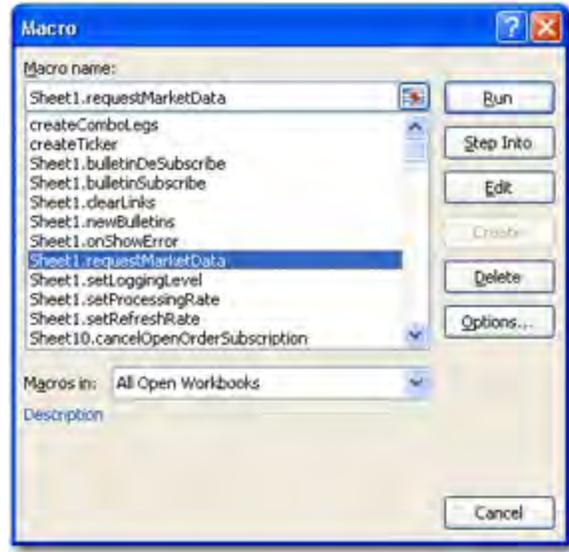
util模块包括多个预先定义的常量，你可在编制你自己的DDE API应用程序时使用。使用这些常量替代硬编码值将使你的应用程序更强大和更容易维护。特别是，下列工具功能对创建新的Visual Basic功能特别有用：

- composeLink – 组成一个接收数据的链接，比如市场数据买价尺寸、请求模型波动率、或执行代号。
- composeControlLink – 组成一个启动操作的链接，比如订阅市场数据、下达定单、或定义市场扫描仪。
- composeContractReq – 从诸如代码或定单页面值读取合约描述，和创建DDE字符串加以代表。

## 宏程序

DDE API样本表格广泛的使用微软Excel宏程序。当你点击表格中每个页面上的每个工具栏的按钮时均运行一个宏程序。例如，当你点击代码页面的寻求市场数据按钮时，你实际上是在运行一个叫做requestMarketData的宏程序。

你可以通过打开Excel宏对话框查看所有在样本表格中使用的宏程序。从这里，你可以选择编辑宏程序，其将在Visual Basic编辑器中打开宏编码。



**Note:** 你必须在你开启Excel时启动宏程序，否则表格中的宏程序将不会工作。

有关记录、编辑和查看宏程序，请参考你的微软Excel文档。

### 命名范围

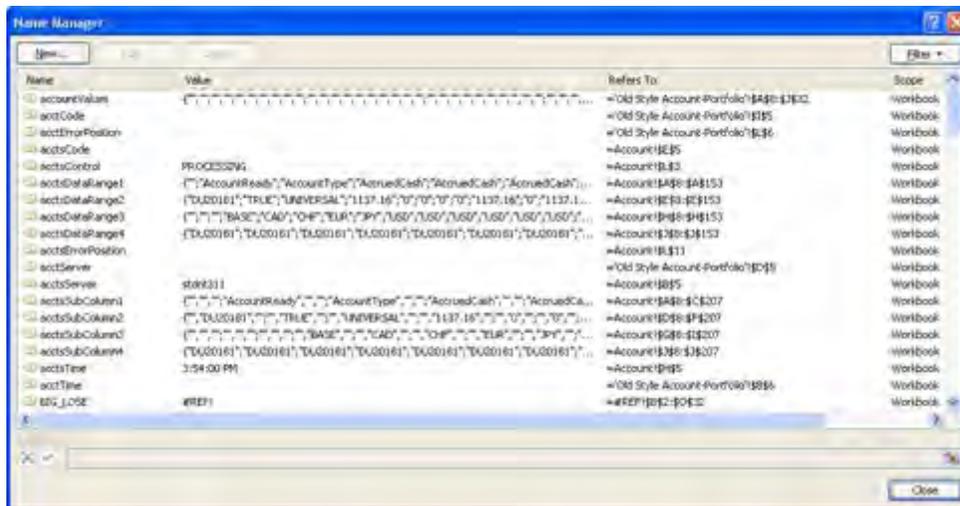
命名范围是微软Excel的一个功能，让你可以在微软Excel中给一个单元格或一组单元格指定有意义的名字。TwsDde.xls样本表格即广泛的使用命名范围。

命名范围帮助你在工作表格中移动数据而不会打乱现有的逻辑。它还可以使重要的数据被应用到你自定义的宏程序和工作表格中。

你可以查看表格中使用的所有命名范围，根据你使用的Excel版本，你可以进行如想操作：

- 对Excel 2007版本，你可以看到用于表格的一个完整的所有命名范围列表，做法是点击**Formulas(公式)**，然后点击**Name Manager(命名管理)**。命名管理显示表格中使用的每个命名范围、范围的值、和页面以及范围覆盖的单元格范围。
- 对早期的Excel版本，你可以通过选择工具栏菜单上的**Name > Define**查看命名范围。你还可以从微软下载一个免费的命名管理，从而为这些早期Excel版本添加额外的功能。

下面的屏幕显示的是Excel的DDE API样本表格的命名管理：



### Excel的DDE句法

下面表格定义用于支持DDE功能的可能的单元格值。当你将光标指向一个单元格时，出现在Excel公式栏中的基本句法(“=”表格上方可输入区域)是：

**=server|topic!id?reqType?field2**

或者

**=server|error!error** (对将显示错误的可选标签)

这里：

描述	Server(服务器)	Topic(主题)	id(代号)	reqType(寻求类型)	field2(区域2)
Place an order(下达订单)	server	ord	idn	place(下达)	orderDescription(订单描述)
Modify an order(修改订单)	server	ord	idn	modify(修改)	orderModification(订单修改)
Cancel an order(取消订单)	server	ord	idn	cancel(取消)	
Check order status(查看订单状态)	server	ord	idn	status(状态)	
Request open orders(请求开放订单)	server	ord	idn	open(开放)	

Request executions( 请求执行)	server	ord	idn	executed( 执行的)	
Check shares filled in order( 查看定单中执行的股份)	server	ord	idn	sharesFilled( 执行的股份)	
Check shares remaining in order( 查看定单中剩余股份)	server	ord	idn	sharesRemaining( 剩余股份)	
Execution (average) price( 平均执行价)	server	ord	idn	price( 价格)	
Underlying( 底层证券)	server	ord	idn	symbol( 代码)	
Security type( 证券类型)	server	ord	idn	secType( 证券类型)	Refer to Note 6( 参考注释6)
Expiry( 过期)	server	ord	idn	expiry( 过期)	Refer to Note 7( 参考注释7)
Strike( 执行价)	server	ord	idn	strike( 执行价)	Refer to Note 8( 参考注释8)
Right( 行使权)	server	ord	idn	right( 行使权)	Refer to Note 8( 参考注释8)
Specify contract multiplier for options and futures( 为期权和期货指定合约乘数)	server	ord	idn	multiplier( 乘数)	Refer to Note 8( 参考注释8)
Order destination( 定单目的地)	server	ord	idn	exchange( 交易所)	
Currency( 货币)	server	ord	idn	currency( 货币)	
Order side( 定单边)	server	ord	idn	side( 边)	
Order quantity( 定单数量)	server	ord	idn	size( 尺寸)	
Order type( 定单类型)	server	ord	idn	orderType( 定单类型)	

Limit price( 限价)	server	ord	idn	limitPrice( 限价)	
Auxiliary price( 辅助价格)	server	ord	idn	auxPrice( 辅助价格)	
Local symbol( 本地代码)	server	ord	idn	localSymbol( 本地代码)	
Last fill price( 最后执行价)	server	ord	idn	lastFillPrice( 最后执行价)	
Create ticker( 创建代码)	server	tik	idn	req( 请求)	
Bid implied volatility( 买价隐含波动率)	server	tik	idn	bidImpliedVol( 买价隐含波动率)	
Bid delta( 买价 delta)	server	tik	idn	bidDelta( 买价 delta)	
Request bid size( 请求买价尺寸)	server	tik	idn	bidSize( 买价尺寸)	
Request bid price( 请求买价价格)	server	tik	idn	bid( 买价)	
Request ask price( 请求卖价价格)	server	tik	idn	ask( 卖价)	
Request ask size( 请求卖价尺寸)	server	tik	idn	askSize( 卖价尺寸)	
Ask implied volatility( 卖价隐含波动率)	server	tik	idn	askImpliedVol( 卖价隐含波动率)	
Ask delta( 卖价 delta)	server	tik	idn	askDelta( 卖价 delta)	
Request last price( 请求最后价格)	server	tik	idn	last( 最后)	
Request last size( 请求最后尺寸)	server	tik	idn	lastSize( 最后尺寸)	
Last implied volatility( 最后隐含波动率)	server	tik	idn	lastImpliedVol( 最后隐含波动率)	
Last delta( 最后 delta)	server	tik	idn	lastDelta( 最后 delta)	

Request today's high price( 请求今天的最高价格)	server	tik	idn	high( 高)	
Request today's low price( 请求今天的最低价格)	server	tik	idn	low( 低)	
Request today's volume size( 请求今天的交易量尺寸)	server	tik	idn	volume( 交易量)	
Request last close price( 请求最后收盘价)	server	tik	idn	close( 收盘)	
Request implied volatility calculated by the TWS option modeler( 请求由TWS期权模型计算的隐含波动率)	server	tik	idn	modelVolatility( 模型波动率)	
Request option delta calculated by the TWS option modeler( 请求由TWS期权模型计算的期权delta)	server	tik	idn	modelDelta( 模型Delta)	
Request the model price( 请求模型价格)	server	tik	idn	modelPrice( 模型价格)	
Request present value of dividends expected on the options underlier( 请求期权底层资产的预期股息的现值)	server	tik	idn	pvDividend	
Request number of hold days until the expiry of the EFP( 请求EFP过期日之前持有日期)	server	tik	idn	holdDays( 持有日期)	

Request expiration date of the single stock future( 请求单一股票期货的过期日)	server	tik	idn	futureExpiry( 期货过期)	
Request dividends expected until the expiration of the single stock future( 请求单一股票期货过期前的预计股息)	server	tik	idn	dividendsToExpiry( 过期前股息)	
Request annualized basis points( 请求年率基点)	server	tik	idn	basisPoints( 基点)	
Request annualized basis points in percentage form( 请求百分比形式的年率基点)	server	tik	idn	formattedBasisPoints( 格式化基点)	
Request implied futures price( 请求隐含期货价格)	server	tik	idn	impliedFuture( 隐含期货)	
Request the dividend impact on the annualized basis points interest rate( 请求股息对年率基点利率的影响)	server	tik	idn	dividendImpact( 股息影响)	
Account statement control key( 账户报表控制键)	server	acct	idn	acctv	Account code (for Advisor-managed accounts only)( 仅用于顾问管理的账户的账户代码)
Request one account value string( 请求一个账户值字符串)	server	acct	idn	key	
Account value( 账户值)	server	acct	idn	value( 值)	
Account currency( 账户货币)	server	acct	idn	keyCurrency( 关键货币)	

Account portfolio control key( 账户投资组合控制键)	server	acct	idn	acctp	Account code (for Advisor-managed accounts only)( 仅用于顾问管理的账户的账户代码)
Account portfolio underlying symbol( 账户投资组合底层代码)	server	acct	idn	symbol( 代码)	
Account portfolio security type( 账户投资组合证券类型)	server	acct	idn	secType( 证券类型)	
Account portfolio expiry( 账户投资组合过期)	server	acct	idn	expiry( 过期)	
Account portfolio strike price( 账户投资组合执行价)	server	acct	idn	strike( 执行价)	
Account portfolio right( 账户投资组合权力)	server	acct	idn	right( 行使权)	
Account portfolio currency( 账户投资组合货币)	server	acct	idn	currency( 货币)	
Account portfolio local symbol( 账户投资组合本地代码)	server	acct	idn	localSymbol( 本地代码)	
Account portfolio market price( 账户投资组合市场价值)	server	acct	idn	marketPrice( 市价)	
Account portfolio market value( 账户投资组合市场价值)	server	acct	idn	marketValue( 市价)	
Account portfolio average cost( 账户投资组合平均成本)	server	acct	idn	avgCost( 平均成本)	



Account portfolio realized PNL( 账户投资者实现的PNL)	server	acct	idn	realizedPNL( 实现的PNL)	
Account portfolio unrealized PNL( 账户投资组合未实现的PNL)	server	acct	idn	unrealizedPNL( 未实现的PNL)	
Request contract details( 请求合约详细)	server	contract	idn	req( 请求)	contractDescription( 合约描述)
Valid order types( 有效订单类型)	server	contract	idn	orderTypes( 订单类型)	
Valid exchanges( 有效交易所)	server	contract	idn	validExchanges( 有效交易所)	
Contract identifier( 合约代号)	server	contract	idn	conid	
Minimum tick( 最小跳动)	server	contract	idn	minTick	
Order multiplier( 订单乘数)	server	contract	idn	multiplier( 乘数)	
Market name( 市场名称)	server	contract	idn	marketName( 市场名称)	
Trading class( 交易类别)	server	contract	idn	tradingClass( 交易类别)	
Execution order id( 执行订单代号)	server	exec	idn	orderId( 订单代号)	
Underlying( 底层)	server	exec	idn	symbol( 代码)	
Security type( 证券类型)	server	exec	idn	secType( 证券类型)	
Expiry( 过期)	server	exec	idn	expiry( 过期)	
Strike( 执行价)	server	exec	idn	strike( 执行价)	
Right( 行使权)	server	exec	idn	right( 行使权)	
Order destination( 订单目的地)	server	exec	idn	exchange( 交易所)	

Currency( 货币)	server	exec	idn	currency( 货币)	
Local symbol( 本地代码)	server	exec	idn	localSymbol( 本地代码)	
Execution id( 执行代号)	server	exec	idn	execId	
Execution time( 执行时间)	server	exec	idn	time( 时间)	
Account number( 账户号)	server	exec	idn	acctnNumber( 账户号)	
Exchange where executed( 交易执行的交易所)	server	exec	idn	eExchange	
Side( 边)	server	exec	idn	side	
Number of shares filled in order( 定单中获执行股票数量)	server	exec	idn	shares( 股份)	
Execution (average) price( 平均执行价)	server	exec	idn	price( 价格)	
Order ID( 定单代号)	server	exec	idn	permId	
Identifies position as one to be liquidated last( 确定最后清算头寸)	server	exec	idn	liquidation( 清算)	
Request execution details( 请求执行详细信息)	server	exec	idn	Req( 请求)	executionFilter( 执行过滤器)
Request list of Advisor-managed accounts( 请求顾问管理账户列表)	server	FAaccts( 顾问账户)	idn	Req( 请求)	
List of Advisor-managed accounts( 顾问管理账户列表)	server	FAaccts( 顾问账户)	idn	Value( 值)	

Request market depth( 请求市场深度)	server	mktDepth( 市场深度)	idn	req( 请求)	contractDescripton?( 合约描述? ) num_display_rows 参考下面注释(1)。
Market maker( 做市)	server	mktDepth( 市场深度)	idn	mktMaker( 做市)	rowId_side 参考下面注释(2)。
Order price( 订单价格)	server	mktDepth( 市场深度)	idn	price( 价格)	rowId_side 参考下面注释(2)。
Order size( 订单尺寸)	server	mktDepth( 市场深度)	idn	size( 尺寸)	rowId_side 参考下面注释(2)。
Market data refresh rate( 市场数据更新频率)	server	refreshRate	idn	millisec( 毫秒)	Number of milliseconds( 毫秒数目)
Subscribe to news bulletins( 订阅新闻公告)	server	news	sub	0	参考下面注释3
News bulletin message ID( 新闻公告信息代号)	server	news	newsID( 新闻代号)		
News bulletin message type( 新闻公告信息类型)	server	news	newsType( 新闻类型)		参考下面注释4
News bulletin message text( 新闻公告信息文字)	server	news	msg		
Exchange from which news bulletins originated( 发布新闻公告的原始交易所)	server	news	exchange( 交易所)		
Set the server log level( 设置服务器记录级别)	server	logLevel	<log_level>		参考下面注释5

这里：

```
orderDescription( 定单描述) = side_quantity_symbol_secType_exp_strike_right_exchange_orderType_lmtPrice_
auxPrice
{_timeInForce_ocaGroup_account_open/close_origin_orderRef_transmit_parentId_blockOrder
_sweepToFill_displaySize_triggerMethod_ignoreRth_hidden_clientId_accountCode_goodAfterTime_
goodTillDate_faGroup_faMethod_faPercentage_faProfile_shortSaleSlot_
PRIMARYEXCHANGE
_shortSaleLocation_ocaType_rthOnly_rule80A_settlingFirm_allOrNone_minimumQty_percentOffset_
electronicTradeOnly_firmQuoteOnly_nbboPriceCap_auctionStrategy_startingPrice_stockRefPrice_
delta_stockRangeLower_stockRangeUpper_volatility_volatilityType_referencePriceType_hedgeDelta_
continuousUpdate}
```

**注:**括号中的属性是扩展定单属性，在[扩展定单属性](#)章节中有描述。

- contractDescription - symbol\_secType\_exp\_strike\_right\_exchange
- tickerDescription( 定单描述) = symbol\_secType\_exp\_strike\_right\_exchange
- ExecutionFilter( 执行过滤器) = clientId\_accountCode\_date\_time\_symbol\_secType\_exchange\_side

**注 1:** 当请求市场深度时，你可以指定显示行的数目。如果没有提供，默认的行数目是5行。这个参数可以被用来优化表现，因为少的显示数目要求的CPU负荷比较低。

例如：=edemo|mktDepth!id0?req?MSFT\_STK\_SMART?10 将要求10行市场深度定单。

**注 2:** 市场深度区域2‘价格’和‘尺寸’请求类型包含其它的信息来定义数据语言的定单行和边。市场深度定单被分为两边，买和卖，定单输入从抵消额为0开始。

例如：

=edemo|mktDepth!id0?price?0\_ASK 将要求定单的卖价输入0。

=edemo|mktDepth!id0?size?2\_BID 将要求定单的买价尺寸输入为2。

**注 3:** 当订阅新闻公告时，请求的格式应该为：

=edemo|news!sub?0

这里请求类型为零。取消订阅只需清空订阅单元格。

**注 4:** 有效的新闻公告类型是：

- 1 = 普通新闻公告
- 2 = 交易所不再用于交易
- 3 = 交易所可用于交易

**注 5:** 有效的记录级别是：

- 1 = 系统( 最不详细)
- 2 = 错误( 默认的，如果指定级别)
- 3 = 警告

4 = 信息

5 = 详细(最详细的)

**注6:** 如果是组合定单, 组合边是在辅助价格之后插入的。这是一个三边的IBM组合描述:

CMBLGS\_3\_8314\_100\_BUY\_SMART\_0\_36930759\_1\_BUY\_SMART\_0\_36930816\_1\_SELL\_SMART\_0\_CMBLGS。

每个边包括: 合约代号、合约数目、边、交易所、和开盘/收盘(0或1)。个人可以永远指定为0, 机构需要指定为1, 如果被执行的定单将平仓头寸。

**注7:** 如果定单是一个期权或期货, 过期日是插入在证券类型之后的。

**注8:** 如果定单是期权, 执行价和行使权(看跌或看涨)将被插入到过期日之后。如果指定了乘数, 它将位于执行价和行使价之后。



## Active X

这一章描述ActiveX API，包括下面题目：

- [使用ActiveX连接应用程序](#)
- [注册第三方ActiveX控制](#)
- [在64位Windows XP系统下运行ActiveX API](#)
- [使用Visual Basic样本程序](#)
- [ActiveX功能](#)
- [ActiveX事件](#)
- [ActiveX COM目的](#)
- [ActiveX特性](#)
- [下达组合定单](#)

API软件还包括Excel的ActiveX样本表格，该表格和Excel的DDE样本表格的大部分功能是一样的，但是基于ActiveX控制，Tws.ocx。请参考[Excel的ActiveX](#)了解详细内容。

## 使用ActiveX连接应用程序

在你可以使用第三方的ActiveX控制，你必须[用Visual Basic进行注册](#)。

### 使用ActiveX控制和VB、VBA或C++进行连接

1. 将应用程序控制拖放到表格获对话框。
2. 启动下列方法：
  - 启动connect()方法连接运行的应用程序。
  - 启动任何你需要执行操作的方法，比如请求市场数据的reqMktData()方法。
3. 启动下达定单的placeOrder()方法。使用扩展属性的定单需要先设置ActiveX属性来代表它们。
4. 处理下列事件：
  - 处理nextValidId()事件以接收下一个可用的有效定单代号。连续定单的代号增量为1。
  - 处理tickPrice()和tickSize()事件接收市场数据。
  - 处理orderStatus()事件接收定单状态信息。
  - 处理error()事件接收出错信息。
  - 处理connectionClosed()事件以在应用程序和ActiveX控制终止连接时接到通知。



## 注册第三方 ActiveX 控制

在 Visual Basic 中使用第三方 ActiveX 控制首先必需要注册。

### 使用 Visual Basic 注册 ActiveX 控制的步骤：

1. 在你 VB 项目的 **组件 (Components)** 菜单中选择交易平台 ActiveX 控制(Tws.ocx)。
2. 点击 **应用**。
3. 检查 TWS 控制出现在工具栏上，并带有所有的标准控制。

## 在64位 Windows XP系统下运行 ActiveX API

在64位 Windows XP系统下运行 ActiveX API，请遵循以下步骤：

1. 安装 Microsoft Visual C++ 2005 SP1再发行组件包(x86)。
2. 安装 Microsoft Visual J# 2.0 再发行组件包。
3. 下载和安装API软件。

## 使用 Visual Basic 样本程序

你可以使用 Visual Basic 样本应用程序通过 ActiveX 接口金融服务器。运行样本必须完成：

- 安装 API 样本应用程序
- 配置应用程序支持 API 组件
- 再你的计算机中安装 MS Visual Studio (Visual Basic 6.0 或更高版本)。

Visual Basic 程序是一个演示使用 TWS ActiveX 控制从一个 Visual Basic 应用程序中连接服务器的样本程序。

### 打开 VB\_API\_样本程序：

1. 从 MS Visual Basic 文件菜单，选取 **打开程序( Open Project)** 。
2. 在你的 API 下载文件夹中找到 \TestActiveXClient\_VB 目录，并选取 *VB\_API\_sample.vbproj*。X\_XX 代表 API 版本号；例如，9.60。

如果你使用的是 Visual Studio 2008，你将需要在打开其之前更新你的项目。Visual Basic 提供一个升级向导来指引你完成这个简单的操作。

3. 在 **项目( Projects)** 菜单中，选取 *组件( Components)* 。
4. 在组件对话框中，选取 *TWS ActiveXControl* 模块并点击 **OK**。
5. 按住 **Ctrl+F5** 编辑和运行项目。

**Note:** 如果你在使样本程序运行时遇到困难，可尝试重新注册 TWS ActiveX 组件，Tws.ocx。

## ActiveX方法

ActiveX方法让你的应用程序启动功能和从TWS请求信息。

<p>连接和服务器</p> <ul style="list-style-type: none"> <li><a href="#">connect()</a></li> <li><a href="#">disconnect()</a></li> <li><a href="#">reqCurrentTime()</a></li> <li><a href="#">setServerLogLevel()</a></li> </ul> <p>市场数据</p> <ul style="list-style-type: none"> <li><a href="#">reqMktDataEx()</a></li> <li><a href="#">cancelMktData()</a></li> <li><a href="#">calculateImpliedVolatility()</a></li> <li><a href="#">cancelCalculateImpliedVolatility()</a></li> <li><a href="#">calculateOptionPrice()</a></li> <li><a href="#">cancelCalculateOptionPrice()</a></li> <li><a href="#">reqMarketDataType()</a></li> </ul> <p>定单</p> <ul style="list-style-type: none"> <li><a href="#">placeOrderEx()</a></li> <li><a href="#">cancelOrder()</a></li> <li><a href="#">reqOpenOrders()</a></li> <li><a href="#">reqAllOpenOrders()</a></li> <li><a href="#">reqAutoOpenOrders()</a></li> <li><a href="#">reqIds()</a></li> <li><a href="#">exerciseOptionsEx()</a></li> </ul> <p>执行</p> <ul style="list-style-type: none"> <li><a href="#">reqExecutionsEx()</a></li> </ul> <p><b>Contract Details</b></p> <ul style="list-style-type: none"> <li><a href="#">reqContractDetailsEx()</a></li> </ul> <p>市场深度</p> <ul style="list-style-type: none"> <li><a href="#">reqMktDepthEx()</a></li> <li><a href="#">cancelMktDepth()</a></li> </ul> <p>账户</p> <ul style="list-style-type: none"> <li><a href="#">reqAccountUpdates()</a></li> </ul>	<p>新闻公告</p> <ul style="list-style-type: none"> <li><a href="#">reqNewsBulletins()</a></li> <li><a href="#">cancelNewsBulletins()</a></li> </ul> <p>金融顾问</p> <ul style="list-style-type: none"> <li><a href="#">reqManagedAccts()</a></li> <li><a href="#">requestFA()</a></li> <li><a href="#">replaceFA()</a></li> <li><a href="#">reqAccountSummary()</a></li> <li><a href="#">cancelAccountSummary()</a></li> <li><a href="#">reqPositions()</a></li> <li><a href="#">cancelPositions()</a></li> </ul> <p>历史数据</p> <ul style="list-style-type: none"> <li><a href="#">reqHistoricalDataEx()</a></li> <li><a href="#">cancelHistoricalData()</a></li> </ul> <p>市场扫描仪</p> <ul style="list-style-type: none"> <li><a href="#">reqScannerParameters()</a></li> <li><a href="#">reqScannerSubscriptionEx()</a></li> <li><a href="#">cancelScannerSubscription()</a></li> </ul> <p>实时柱</p> <ul style="list-style-type: none"> <li><a href="#">reqRealTimeBarsEx()</a></li> <li><a href="#">cancelRealTimeBars()</a></li> </ul> <p>制作方法</p> <ul style="list-style-type: none"> <li><a href="#">createComboLegList()</a></li> <li><a href="#">createContract()</a></li> <li><a href="#">createExecutionFilter()</a></li> <li><a href="#">createOrder()</a></li> <li><a href="#">createScannerSubscription()</a></li> <li><a href="#">createTagValueList()</a></li> <li><a href="#">createUnderComp()</a></li> </ul> <p>基本面数据</p> <ul style="list-style-type: none"> <li><a href="#">reqFundamentalData()</a></li> <li><a href="#">cancelFundamentalData()</a></li> </ul>
--	---

### connect()

启动这个方法与主机应用程序连接。

可公共重写子连接( **ByVal host as String, ByVal port as Integer, ByVal clientID as Integer**

参数	描述
主机( host)	主机名称或运行TWS机器的IP地址。 留为空白与本地主机连接。
端口( port)	必须与配置>API>插口端口区域中指定的端口匹配。
客户代码( clientID)	用于代表这个客户连接的数字。所有从这个客户下达的/修改的订单将与这个客户代码相连。  注：每个客户必须使用一个独特的clientId进行连接。

### disconnect()

启动这个方法终止主机应用程序连接。启动这个方法并不会取消已经发送的订单。

**Public Overridable Sub disconnect()**

### reqCurrentTime()

返回到服务器的当前系统时间。

**Public Overridable Sub reqCurrentTime()**

### setServerLogLevel()

**Public Overridable Sub setServerLogLevel(ByVal logLevel As Integer)**

参数	描述
记录级别( logLevel)	在处理API请求时指定服务器使用的记录输入的详细级别。有效的值包括： <ul style="list-style-type: none"> <li>• 1 = 系统</li> <li>• 2 = 错误</li> <li>• 3 = 警告</li> <li>• 4 = 信息</li> <li>• 5 = 详细</li> </ul>

默认级别是错误。参阅 [API记录](#) 了解详细信息。

### reqMktDataEx()

启动这个方法请求市场数据。市场数据的结果为 [tickPrice\(\)](#), [tickSize\(\)](#), [tickOptionComputation\(\)](#), [tickGeneric\(\)](#), [tickString\(\)](#)和 [tickEFP\(\)](#) events in `disinterface_DTwsEvents`。

**Public Overridable Sub reqMktDataEx(ByVal tickerId As Integer, ByVal contract As TWSLib.IContract, ByVal genericTicks As String, ByVal snapshot As Integer)**

参数	描述
代码代号 ( tickerId)	代码代号 必须是一个特定值。 When the market data returns, 当市场数据返回时将通过这个标签代表。它也被用于取消市场数据。
合约 ( contract)	这一项包括请求的市场数据的合约描述。
通用跳动 ( genericTicks)	由逗号分隔的一组通用跳动类型。有关跳动类型的信息内容, 请参阅 <a href="#">通用跳动类型</a> 。
截图 ( snapshot)	选择返回市场数据的一个单一截图, 和取消市场数据订阅。 如果你使用截图, 不要输入任何 genericTicklist值。

### cancelMktData()

启动这个方法后, 特定代号的的市场数据将停止输送。

**Public Overridable Sub cancelMktData(ByVal id As Integer)**

参数	描述
代号 ( id)	启动 reqMktData()中指定的代号。

### calculateImpliedVolatility()

启动这个功能来计算提供的期权价格和底层价格的波动率。

**Public Overridable Sub calculateImpliedVolatility(ByVal reqId As Integer, ByVal contract As TWSLib.IContract, ByVal optionPrice As Double, ByVal underPrice As Double)**

参数	描述
reqId	代码代号。
contract	描述合约。
optionPrice	期权价格。
underPrice	底层证券价格。

### cancelCalculateImpliedVolatility()

启动这个功能来取消计算提供的期权价格和底层价格的波动率的请求。

**Public Overridable Sub calculateImpliedVolatility(ByVal reqId As Integer)**

参数	描述
reqId	代码代号。

**calculateOptionPrice()**

启动这个功能来计算提供的波动率和底层证券价格的期权价格和希腊值。

**Public Overridable Sub calculateOptionPrice(ByVal reqId As Integer, ByVal contract As TWSLib.IContract, ByVal volatility As Double, ByVal underPrice As Double)**

参数	描述
reqId	代码代号。
contract	合约描述。
volatility	波动率。
underPrice	底层证券价格。

**cancelCalculateOptionPrice()**

启动这个功能来取消对提供的波动率和底层价格的计算期权价格和希腊值的请求。

**Public Overridable Sub calculateOptionPrice(ByVal reqId As Integer)**

参数	描述
reqId	代码代号。

**reqMarketDataType()**

API可以接收来自交易平台的冻结市场数据。冻结市场数据是我们系统中记录的最后的数据。在正常交易时间内，API接收实时市场数据。如果你使用这个功能，你在指示TWS收盘后自动切换到冻结市场数据。然后，在下一个交易日开盘前，市场数据将自动切换回实时市场数据。

**Public Overridable Sub reqMarketDataType(type As Integer)**

参数	描述
类型 ( type)	1为实时市场数据流，2为冻结市场数据。

**placeOrderEx()**

启动这个方法来自下达订单。订单状态将由disinterface\_DTwsEvents中的[orderStatus\(\)](#)结果返回。

**Public Overridable Sub placeOrderEx(ByVal orderId As Integer, ByVal contract As TWSLib.IContract, ByVal order As TWSLib.IOrder)**

参数	描述
orderId	订单代号。你必须指定一个特殊的值。当订单状态返回时将由这个标签代表。当取消订单时也使用这个标签。
contract	包含用于描述合约的属性。
order	包含订单的详细。注：每个客户必须使用一个独特的clientId进行连接。

### cancelOrder()

启动这个方法取消订单。

#### Public Overridable Sub cancelOrder(ByVal id As Integer)

参数	描述
id	之前启动placeOrder()中指定的订单代号。

### reqOpenOrders()

启动这个方法请求由这个客户下达的开放订单。每个开放订单将通过[openOrderEx\(\)](#)结果反馈。

**Note:** 具有clientId为0的客户还将收到应用程序拥有的开放订单。这些订单将与客户相连，并将生成一个新的orderId。这种相连将存在于多个API和应用程序过程中。

#### Public Overridable Sub reqOpenOrders()

### reqAllOpenOrders()

启动这个方法请求由所有客户以及应用程序下达的开放订单。每个开放订单将通过[orderStatus\(\)](#)结果反馈。

**Note:** 反馈的订单和请求客户之间没有相连。

#### Public Overridable Sub reqAllOpenOrders()

### reqAutoOpenOrders()

启动这个方法请求新近创建的应用程序订单与客户隐式关联。当一个新应用程序订单被创建时，订单将和客户相连，并通过[orderStatus\(\)](#)结果反馈。

**Note:** 这个请求只可以用于clientId为0的客户。

#### Public Overridable Sub reqAutoOpenOrders(ByVal bAutoBind As Integer)

参数	描述
bAutoBind	如果设为真( TRUE) ，新创建的应用程序订单将与客户隐式关联。如果设置为假( FALSE) ，则不生成关联。



### reqIds()

启动这个功能来请求在下一个下达订单时可使用的有效代号。在启动这个方法之后，[nextValidId\(\)](#)行动将被触发，并给出下一个有效代号。该代号将反映任何已发生的自动绑定(生成新的代号和下一个有效代号中的增量)。

#### Public Overridable Sub reqIds(ByVal numIds As Integer)

参数	描述
numIds	设置为1。

### exerciseOptionsEx()

启动这个方法来执行期权。

**Note:** [exerciseOptionsEx\(\)](#)功能不允许智能传递交易所，并且TWS提出对由API起始的执行或还是起始失败时的头寸的请求。

#### Public Overridable Sub exerciseOptionsEx(ByVal tickerId As Integer, ByVal contract As TWSLib.IContract, ByVal exerciseAction As Integer, ByVal exerciseQuantity As Integer, ByVal account As String, ByVal override As Integer)

参数	描述
tickerId	执行请求的代码。
contract	该结构包括被请求市场数据的合约的描述。
exerciseAction	可以有两个值：1=执行；2=失效
exerciseQuantity	被执行的合约数目
account	对机构订单。指定IB账户。
override	指定是否你的设置将取代系统的自动行动。例如，如果你的活动是“执行”，且期权不在价内，系统的自动行动将不会执行。如果你已将取代设置为“是”，自动行动将被取代，且价外期权将被执行。值为：0=不取代；1=取代

### reqExecutionsEx()

当启动这个方法时，满足过滤条件的执行报告即通过dispinterface\_DTwsEvents中的execDetailsEx()行动下载给客户。查看过去24小时以外的执行，打开TWS中的交易记录，当交易记录显示后，从API再次请求执行。

#### Public Overridable Sub reqExecutionsEx(ByVal reqId As Integer, ByVal filter As TWSLib.IExecutionFilter)

参数	描述
filter	用于确定返回的执行报告的过滤条件。

### reqContractDetailsEx()

启动这个方法下载某个特定合约所有详细。合约详细将通过 `dispinterface_DTwsEvents` 中的 [contractDetailsEx\(\)](#) 回调收到。

**Public Overridable Sub reqContractDetailsEx(ByVal reqId As Integer, ByVal contract As TWSLib.IContract)**

参数	描述
reqId	数据请求的代号。如果同时处理几个请求时，保证结果与请求匹配。
contract	这个目的包含请求市场数据的合约的描述。

### reqMktDepthEx()

启动这个方法请求特定合约的市场深度。市场深度将通过 [updateMktDepth\(\)](#) 和 [updateMktDepthL2\(\)](#) 行动返回。

**Public Overridable Sub reqMktDepthEx(ByVal tickerId As Integer, ByVal contract As TWSLib.IContract, ByVal numRows As Integer)**

参数	描述
tickerId	代码代号。必须为独特的值。当市场深度数据返回时，将由这个标签代表。也被用于在取消市场深度请求时。
contract	这个目的包含请求市场数据的合约的描述。
numRows	指定返回的市场深度行的数目。

### cancelMktDepth()

调用这个方法之后，特定代码的市场深度数据将停止输送。

**Public Overridable Sub cancelMktDepth(ByVal id As Integer)**

参数	描述
id	启动 <a href="#">reqMktDepth()</a> 或 <a href="#">reqMktDepth2()</a> 中指定的代号。

### reqAccountUpdates()

启动这个方法请求账户更新。账户数据将由 [updateAccountTime\(\)](#)，[updateAccountValue\(\)](#) 和 [updatePortfolioEx\(\)](#) 行动反馈提供。

**Public Overridable Sub reqAccountUpdates(ByVal subscribe As Integer, ByVal acctCode As String)**

参数	描述
subscribe	如果设置为1，客户将开始接收账户和投资组合更新。如果设置为0，客户将停止接收这个信息。
acctCode	接收账户和投资组合更新的账户代码。

### reqNewsBulletins()

启动这个方法开始接收新闻公告。每个公告将由 [updateNewsBulletin\(\)](#) 行动反馈。

#### Public Overridable Sub reqNewsBulletins(ByVal allDaysMsgs As Integer)

参数	描述
allDaysMsgs	如果设为真( TRUE) ，返回当天所有现有的公告和任何新发布的公告。如果设为假( FALSE) ，将只返回新的公告。

### cancelNewsBulletins()

启动这个方法停止接收新闻公告。

#### Public Overridable Sub cancelNewsBulletins()

### reqManagedAccts()

启动这个方法请求受管理的账户列表。列表将由通过 [managedAccounts\(\)](#) 行动返回。

**Note:** 这个请求仅在与金融顾问账户连接时可以发出。

#### Public Overridable Sub reqManagedAccts()

### requestFA()

启动这个方法请求从服务器中调出金融顾问配置信息。数据通过 [receiveFA\(\)](#) 行动以XML数据流形式返回。

#### Public Overridable Sub requestFA(ByVal faDataType As Integer)

参数	描述
faDataType	指定请求的金融顾问配置数据的类型。有效的值包括： <ul style="list-style-type: none"> <li>• 1 = 组</li> <li>• 2 = 概况</li> <li>• 3 = 账户别名</li> </ul>

### replaceFA()

启动这个方法通过API修改金融顾问配置信息。注意，也能通过手动完成。

**Public Overridable Sub replaceFA(ByVal faDataType As Integer, ByVal cxml As String)**

参数	描述
<b>faDataType</b>	指定通过API修改的金融顾问配置数据的类型。有效值包括：
<b>cxml</b>	XML数据串包含新的金融顾问配置信息。

**reqAccountSummary()**

启动这个方法请求TWS账户窗口概括标签上的数据并保持实时更新。数据通过[accountSummary\(\)](#)行动返回。

**Note:** 只有在与金融顾问管理的账户连接时才能发出这个请求。

**Public Overridable Sub reqAccountSummary(ByVal messageType As Integer, ByVal version As Integer, ByVal reqId As Integer, ByVal groupName As String, tags As String)**

参数	类型	描述
messageType	Integer(整数)	设置为 62。
version	Integer(整数)	设置为 1。
reqId	Integer(整数)	
groupName	String(串)	设为 <i>All(所有)</i> 返回所有账户的账户概括数据，或设为一个在TWS全局配置中已创建的特定顾问账户组名。

参数	类型	描述
tags	String(串)	<p>由逗号分隔的账户标签列表。 可用的标签是：</p> <ul style="list-style-type: none"> <li>• <i>AccountType</i></li> <li>• <i>NetLiquidation</i></li> <li>• <i>TotalCashValue</i> — 包括pnl的所有现金</li> <li>• <i>SettledCash</i> — 对现金账户，这个与TotalCashValue相同</li> <li>• <i>AccruedCash</i> — 净累计利息</li> <li>• <i>BuyingPower</i> — 账户可以买进的保证金美国股票的最大额</li> <li>• <i>EquityWithLoanValue</i> — 现金 + 股票 + 债券 + 共同基金</li> <li>• <i>PreviousEquityWithLoanValue</i></li> <li>• <i>GrossPositionValue</i> — 所有股票和股票期权头寸的绝对值总合</li> <li>• <i>RegTEquity</i></li> <li>• <i>RegTMargin</i></li> <li>• <i>SMA</i> — 特殊备忘录账户</li> <li>• <i>InitMarginReq</i></li> <li>• <i>MaintMarginReq</i></li> <li>• <i>AvailableFunds</i></li> <li>• <i>ExcessLiquidity</i></li> <li>• <i>Cushion</i> — 做为净清算值百分比的多余流动性</li> <li>• <i>FullInitMarginReq</i></li> <li>• <i>FullMaintMarginReq</i></li> <li>• <i>FullAvailableFunds</i></li> <li>• <i>FullExcessLiquidity</i></li> <li>• <i>LookAheadNextChange</i> — 当前瞻值生效的时间</li> <li>• <i>LookAheadInitMarginReq</i></li> <li>• <i>LookAheadMaintMarginReq</i></li> <li>• <i>LookAheadAvailableFunds</i></li> <li>• <i>LookAheadExcessLiquidity</i></li> <li>• <i>HighestSeverity</i> — 衡量账户接近被清算的尺度</li> <li>• <i>DayTradesRemaining</i> — 在日交易模式被测试出之前，一个用户可以开仓/平仓交易的数目。值为“-1”表示用户可</li> </ul>

参数	类型	描述
		以进行无限制的日交易。 • <i>Leverage</i> — GrossPositionValue / NetLiquidation

### cancelAccountSummary()

取消账户窗口概括标签数据的请求。

**Note:** 只能在和FA管理的账户连接时才能提出这个请求。

**Public Overridable Sub cancelAccountSummary(ByVal messageId As Integer, ByVal version As Integer, ByVal reqId As Integer)**

参数	类型	描述
messageId	Integer	设为 63。
version	Integer	设为 1。
reqId	Integer	被取消请求的数据代码。

### reqPositions()

请求所有账户的实时头寸数据。

**Note:** 只能在和FA管理的账户连接时才能提出这个请求。

**Public Overridable Sub reqPositions(ByVal messageId As Integer, ByVal version As Integer)**

参数	类型	描述
messageId	Integer	设为 62
version	Integer	设为 1。

### cancelPositions()

取消实时头寸更新。

**Note:** 只有在和FA管理的账户连接时才能提出这个请求。

**Public Overridable Sub cancelPositions(ByVal messageId As Integer, ByVal version As Integer)**

参数	类型	描述
messageId	Integer	设为 64。
version	Integer	设为 1。

### reqHistoricalDataEx()

启动这个方法开始通过 historicalData() 行动接收 历史数据结果。

**Note:** 关于历史数据请求的限制，请参阅 [历史数据限制](#)。

**Public Overridable Sub reqHistoricalDataEx(ByVal tickerId As Integer, ByVal contract As TWSLib.IContract, ByVal endDateTime As String, ByVal duration As String, ByVal barSize As String, ByVal whatToShow As String, ByVal useRTH As Integer, ByVal formatDate As Integer)**

参数	描述
tickerId	请求的代码。必须是唯一的。当收到数据时，将通过这个代码确认。也被用来在取消历史数据请求时。
contract	这个结构包含请求市场数据的合约的描述。
endDateTime	使用格式yyyyymmdd hh:mm:ss tmz，时区的使用( 可选) 可以在空格后的结尾处。
durationStr	<p>这是请求将要覆盖的时间区域，并使用格式： &lt;integer&gt; &lt;unit&gt;指定，例如，1 D，这里有效的单位是：</p> <ul style="list-style-type: none"> <li>• S (秒)</li> <li>• D (天)</li> <li>• W (星期)</li> <li>• M (月)</li> <li>• Y (年)</li> </ul> <p><b>注：</b> 如果没有指定单位，秒将被使用。另外，“年”的使用目前只限制为一年。</p>
barSize	<p>返回的柱的尺寸( 在IB/TWS限制范围内) 。有效的值包括：</p> <p><b>柱尺寸</b></p> <ul style="list-style-type: none"> <li>• 1 秒</li> <li>• 5 秒</li> <li>• 15 秒</li> <li>• 30 秒</li> <li>• 1 分</li> <li>• 2 分</li> <li>• 3 分</li> <li>• 5 分</li> <li>• 15 分</li> <li>• 30 分</li> <li>• 1 小时</li> <li>• 1 天</li> </ul>

<p>whatToShow</p>	<p>决定被提取数据的性质。有效的值包括：</p> <ul style="list-style-type: none"> <li>• TRADES</li> <li>• MIDPOINT</li> <li>• BID</li> <li>• ASK</li> <li>• BID_ASK</li> <li>• HISTORICAL_VOLATILITY</li> <li>• OPTION_IMPLIED_VOLATILITY</li> </ul>
<p>useRTH</p>	<p>决定是否返回请求时间范围内的所有的可用数据，或仅返回处于正常交易时间内的数据。有效的值包括：</p> <ul style="list-style-type: none"> <li>• 0 - 返回所有数据，即使请求的市场处于正常交易时段以外。</li> <li>• 1 - 仅返回正常交易时段内的数据，尽管请求的时间区域部分或全部在正常交易时段以外。</li> </ul>
<p>formatDate</p>	<p>确定用于返回柱的日期格式。有效的值包括：</p> <ul style="list-style-type: none"> <li>• 1 - 返回的柱的日期格式是： yyymmdd {space} {space} hh:mm:dd</li> <li>• 2 - 返回的日期是自格林威治时间 1/1/1970 秒数的长整数形式。</li> </ul>

**cancelHistoricalData()**

在网路失去连接时或在查询的结果延迟时使用，以及应用程序已对接收数据不感兴趣时。

**Public Overridable Sub cancelHistoricalData(ByVal tickerId As Integer)**

参数	描述
tickerId	代码代号。必须是独特的。

**reqScannerParameters()**

请求描述所有可能的扫描仪查询的XML串。

**Public Overridable Sub reqScannerParameters()**

**reqScannerSubscriptionEx()**

启动 reqScannerSubscriptionEX() 方法开始通过 [scannerDataEx\(\)](#) 行动接收市场扫描仪结果。

**Public Overridable Sub reqScannerSubscriptionEx(ByVal tickerId As Integer, ByVal subscription As TWSLib.IScannerSubscription)**



参数	描述
tickerId	订阅的代号。必需是独特的值。当收到订阅的数据时，将通过这个代号确认。这个代号也被用于取消扫描仪时。
subscription	包括过滤器的扫描仪订阅参数的概括。

### cancelScannerSubscription()

Public Overridable Sub cancelScannerSubscription(ByVal tickerId As Integer)

参数	描述
tickerId	代码代号。必须是独特的值。

### reqRealTimeBarsEx()

启动 reqRealTimeBarsEx()方法开始通过 [realtimeBar\(\)](#) 行动接收实时柱结果。

Public Overridable Sub reqRealTimeBarsEx(ByVal tickerId As Integer, ByVal contract As TWSLib.IContract, ByVal barSize As Integer, ByVal whatToShow As String, ByVal useRTH As Integer)

参数	描述
tickerId	请求的代号。必须是独特的值。当接收数据时，通过这个代号确认。当取消历史数据请求时也使用这个代号。
contract	该结构包括被请求市场数据的合约的描述。
barSize	目前仅支持5秒的柱，如果使用任何其它的值，将会出现异常。
whatToShow	确认提取数据的性质。有效的值包括： <ul style="list-style-type: none"> <li>• TRADES</li> <li>• BID</li> <li>• ASK</li> <li>• MIDPOINT</li> </ul>
useRTH	仅用于正常交易时段。有效的值包括： <ul style="list-style-type: none"> <li>• 0 = 返回查询期间内所有可用的数据，包括查询市场是处于正常交易时段外的时间间隔。</li> <li>• 1 = 仅返回查询产品正常交易时段内的数据，不论查询时间是部分或全部处于正常交易时段之外。</li> </ul>

### cancelRealTimeBars()

在网路失去连接时或在查询的结果延迟时使用，以及应用程序已对接收数据不感兴趣时。

**Public Overridable Sub cancelRealTimeBars(ByVal tickerId As Integer)**

参数	描述
tickerId	代码代号。必须为独特的值。

**createComboLegList()**

这个制造的方法被用来创建一个 [IComboLegList](#) COM目标。

**Public Overridable Sub createComboLegList() As TWSLib.IComboLegList**

本节中，你必须使用制作“创建”方法来创建COM目标。例如，createComboLegList()方法创建一个IComboLeg目标。IComboLeg目标包括边定义的列表。

**createContract()**

这个制作的方法被用于创建一个 [IContract](#) COM目标。

**Public Overridable Sub createContract() As TWSLib.IContract**

你必须使用本节中描述的制作“创建”方法来创建COM目标。createContract()方法创建一个IContract目标，包括被请求市场数据的合约的描述。

**createExecutionFilter()**

这个制作的方法被用于创建一个 [IExecutionFilter](#) COM目标。

**Public Overridable Sub createExecutionFilter() As TWSLib.IExecutionFilter**

你必须使用本节中描述的制作“创建”方法来创建COM目标。createExecutionFilter()方法创建一个IExecutionFilter目标，包括被用于确认返回的执行报告的过滤条件。

**createOrder()**

这个制作的方法被用于创建一个 [IOrder](#) COM目标。

**Public Overridable Sub createOrder() As TWSLib.IOrder**

你必须使用本节中描述的制作“创建”方法来创建COM目标。createOrder()方法创建一个IOrder目标，包括定单的详细。

**createScannerSubscription()**

这个制作的方法被用于创建一个 [IScannerSubscription](#) COM目标。

**Public Overridable Sub createScannerSubscription() As TWSLib.IScannerSubscription**

你必须使用本节中描述的制作“创建”方法来创建COM目标。createScannerSubscription()方法创建一个IScannerSubscription目标，包括扫描仪订阅参数的概括。

**createTagValueList**

这个制作的方法被用于创建 [ITagValueList](#)和 [ITagValue](#)目标。

**Public Overridable Function createTagValueList() As TWSLib.ITagValueList**

你必须使用本节中描述的制作“创建”方法来创建COM目标。

**createUnderComp()**

这个制作的方法被用于创建一个 [IUnderComp](#) COM目标。

**Public Overridable Sub createUnderComp() As TWSLib.IScannerSubscription**

你必须使用本节中描述的制作“创建”方法来创建COM目标。 createUnderComp()方法创建一个 IUnderComp 目标，用于定义Delta-中性组合合约。

**reqFundamentalData()**

启动这个方法接收路透社全球基本面数据。在你接收这个数据之前，你必须通过账户管理订阅路透社基本面。

**Public Overridable Sub reqFundamentalData(ByVal reqId As Integer, ByVal contract As TWSLib.IContract, ByVal reportType As String)**

参数	描述
reqId	请求数据的代号。
contract	这个结构包括请求的路透社基本面数据的合约描述。
reportType	确认报告类型，包含以下其一： <ul style="list-style-type: none"> <li>• 估计</li> <li>• 财务报表</li> <li>• 概括</li> </ul>

**cancelFundamentalData()**

启动这个方法停止接收路透社全球基本面数据。

**Public Overridable Sub cancelFundamentalData(ByVal reqId As Integer)**

参数	描述
reqId	请求数据的代号。

## ActiveX行动

ActiveX行动从系统中接收信息，并可被用于应用程序。这一节定义通过DTwsEvents接口接收的ActiveX行动。

<p><b>连接和服务</b></p> <ul style="list-style-type: none"> <li><a href="#"><u>connectionClosed()</u></a></li> <li><a href="#"><u>currentTime()</u></a></li> <li><a href="#"><u>errMsg()</u></a></li> </ul> <p><b>市场数据</b></p> <ul style="list-style-type: none"> <li><a href="#"><u>tickPrice()</u></a></li> <li><a href="#"><u>tickSize()</u></a></li> <li><a href="#"><u>tickOptionComputation()</u></a></li> <li><a href="#"><u>tickGeneric()</u></a></li> <li><a href="#"><u>tickString()</u></a></li> <li><a href="#"><u>tickEFP()</u></a></li> <li><a href="#"><u>tickSnapshotEnd()</u></a></li> <li><a href="#"><u>marketDataType()</u></a></li> </ul> <p><b>定单</b></p> <ul style="list-style-type: none"> <li><a href="#"><u>orderStatus()</u></a></li> <li><a href="#"><u>openOrderEx()</u></a></li> <li><a href="#"><u>nextValidId()</u></a></li> <li><a href="#"><u>permId()</u></a></li> </ul> <p><b>账户和投资组合</b></p> <ul style="list-style-type: none"> <li><a href="#"><u>updateAccountValue()</u></a></li> <li><a href="#"><u>updatePortfolioEx()</u></a></li> <li><a href="#"><u>updateAccountTime()</u></a></li> </ul> <p><b>新闻公告</b></p> <ul style="list-style-type: none"> <li><a href="#"><u>updateNewsBulletin()</u></a></li> </ul>	<p><b>合约详细</b></p> <ul style="list-style-type: none"> <li><a href="#"><u>contractDetailsEx()</u></a></li> <li><a href="#"><u>contractDetailsEnd()</u></a></li> <li><a href="#"><u>bondContractDetails()</u></a></li> </ul> <p><b>执行</b></p> <ul style="list-style-type: none"> <li><a href="#"><u>execDetailsEx()</u></a></li> <li><a href="#"><u>execDetailsEnd()</u></a></li> <li><a href="#"><u>commissionReport()</u></a></li> </ul> <p><b>市场深度</b></p> <ul style="list-style-type: none"> <li><a href="#"><u>updateMktDepth()</u></a></li> <li><a href="#"><u>updateMktDepthL2()</u></a></li> </ul> <p><b>金融顾问</b></p> <ul style="list-style-type: none"> <li><a href="#"><u>managedAccounts()</u></a></li> <li><a href="#"><u>receiveFA()</u></a></li> <li><a href="#"><u>accountSummary()</u></a></li> <li><a href="#"><u>accountSummaryEnd()</u></a></li> <li><a href="#"><u>position()</u></a></li> <li><a href="#"><u>positionEnd()</u></a></li> </ul> <p><b>Historical Data</b></p> <ul style="list-style-type: none"> <li><a href="#"><u>historicalData()</u></a></li> </ul> <p><b>市场扫描仪</b></p> <ul style="list-style-type: none"> <li><a href="#"><u>scannerParameters()</u></a></li> <li><a href="#"><u>scannerDataEx()</u></a></li> <li><a href="#"><u>scannerDataEnd()</u></a></li> </ul> <p><b>实时柱</b></p> <ul style="list-style-type: none"> <li><a href="#"><u>realtimebar()</u></a></li> </ul> <p><b>基本面数据</b></p> <ul style="list-style-type: none"> <li><a href="#"><u>fundamentalData()</u></a></li> </ul>
---	--

**connectionClosed()**

当TWS终止插口与ActiveX控制的连接时启动这个行动，或者当TWS被关闭时。

**Sub connectionClosed()**

**currentTime()**

这个方法接收服务器方面的当前系统时间。

**Sub currentTime(ByVal time As Integer)**

参数	描述
time	服务器的当前系统时间。

**errMsg()**

当通信中出现错误时或当TWS要给客户发送信息时启动这个行动。

**Sub errMsg(ByVal id As Integer, ByVal errorCode As Integer, ByVal errorMsg As String)**

参数	描述
id	这是发生错误的orderId或tickerId的请求。
errorCode	错误编码在错误编码章节有介绍。
errorMsg	这是错误的文字描述，也包括在错误编码章节中。

**tickPrice()**

当市场数据变化时启动这个功能。价格的更新是实时的，没有延迟。

**Sub tickPrice(ByVal id As Integer, ByVal tickType As Integer, ByVal price As Double, ByVal canAutoExecute As Integer)**

参数	描述
id	代码代号在之前的启动reqMktData()中有说明。

tickType	指定价格类型。可能的值包括： <ul style="list-style-type: none"> <li>• 1 = 买价</li> <li>• 2 = 卖价</li> <li>• 4 = 最后价</li> <li>• 6 = 最高价</li> <li>• 7 = 最低价</li> <li>• 9 = 收盘价</li> </ul>
price	根据tickType的值，可以是买价、卖价或最后价，日内高价、日内低价或日内最后收盘价。
canAutoExecute	指定是否价格跳动可引起自动执行。可能的值包括： <ul style="list-style-type: none"> <li>• 0 = 不可以自动执行</li> <li>• 1 = 可以自动执行</li> </ul>

### tickSize()

当市场数据变化时启动这个功能。尺寸的更新是实时的，没有延迟。

#### Sub tickSize(ByVal id As Integer, ByVal tickType As Integer, ByVal size As Integer)

参数	描述
id	代码代号在之前的启动reqMktData()中有说明。
tickType	指定价格类型可能的值包括：
size	根据tickType的值，可以是买价尺寸、卖价尺寸、最后价尺寸或交易量。

### tickOptionComputation()

Sub tickOptionComputation(ByVal id As Integer, ByVal tickType As Integer, ByVal impliedVol As Double, ByVal delta As Double, ByVal optPrice As Double, ByVal pvDividend As Double, ByVal gamma As Double, ByVal vega As Double, ByVal theta As Double, ByVal undPrice As Double)

参数	描述
id	代码代号在之前的启动reqMktData()中有说明。

tickType	指定跳动类型。可能的值包括： <ul style="list-style-type: none"> <li>• 10 = 买价</li> <li>• 11 = 卖价</li> <li>• 12 = 最后价</li> </ul>
ImpliedVol	由TWS模型计算的隐含波动率，使用特定的ticktype值。
delta	由TWS期权模型计算的期权 delta。
optPrice	期权价格。
pvDividend	期权底层证券的预期股息的当前值。
gamma	期权 gamma 值。
vega	期权 vega 值。
theta	期权 theta 值。
undPrice	底层证券的价格。

**tickGeneric()**

当市场数据变化时启动这个功能。值的更新是实时的，没有延迟。

**Sub tickGeneric(ByVal id As Integer, ByVal tickType As Integer, ByVal value As Double)**

参数	描述
tickerId	代码代号在之前的启动reqMktData()中有说明。
tickType	指定跳动类型。 将区域中的值输入到TickType.getField(int tickType)中获取区域描述。例如，区域值46将代表卖空，等等。
value	特定区域的值。

**tickString()**

当市场数据变化时启动这个功能。值的更新是实时的，没有延迟。

**Sub tickString(ByVal id As Integer, ByVal tickType As Integer, ByVal value As String)**

参数	描述
tickerId	代码代号在之前的启动reqMktData()中有说明。

tickType	指定跳动类型。 将区域中的值输入到TickType.getField(int tickType)中获取区域描述。例如，区域值45代表lastTimestamp，等。
value	特定区域的值。

**tickEFP()**

当市场数据变化时启动这个功能。值的更新是实时的，没有延迟。

**Sub tickEFP(ByVal tickerId As Integer, ByVal field As Integer, ByVal basisPoints As Double, ByVal formattedBasisPoints As String, ByVal totalDividends As Double, ByVal holdDays As Integer, ByVal futureExpiry As String, ByVal dividendImpact As Double, ByVal dividendsToExpiry As Double)**

参数	描述
tickerId	代码代号在之前的启动reqMktData()中有说明。
field	指定价格类型。 将区域中的值输入到TickType.getField(int tickType)中获取区域描述。例如，区域中38代表bidEFP，等。
basisPoints	年率基点，代表可以直接和经纪商利率进行比较的融资利率。
formattedBasisPoints	格式化串的年率基点，以百分比形式表示。
totalDividends	全部预期股息。
holdDays	期货换现货到期之前的持有天数。
futureExpiry	单一股票期货的过期日。
dividendImpact	股息对年率基点利率的影响。
dividendsToExpiry	单一股票期货过期之前的预期股息。

**tickSnapshotEnd()**

当快照市场数据订阅已被完成并没有任何需要等待的操作时启动这个功能。这也适用于过时情形。

**Sub tickSnapshotEnd(ByVal reqId As Integer)**

参数	描述
reqID	数据请求的代号。



### marketDataType()

TWS发送一个marketDataType(类型) 回调给API, 这里类型的设置为冻结( Frozen) 或实时( RealTime) , 来宣布市场数据已经在冻结和实时之间进行转换了。 这个通知仅当市场数据在实时和冻结之间转换时才发生。 marketDataType()回调接受reqId参数, 和按每个订阅发送, 因为不同的合约可以通常可以以不同的时间表进行交易。

#### Sub marketDataType(ByVal reqId As Integer, type As Integer)

参数	描述
reqId	数据请求的代号。
type	1为实时流市场数据, 或2为冻结市场数据。

### orderStatus()

当订单状态改变时启动这个行动。如果客户有任何开放订单时, 在重新连接时也启动这个行动。

#### Sub orderStatus(ByVal id As Integer, ByVal status As String, ByVal filled As Integer, ByVal remaining As Integer, ByVal avgFillPrice As Double, ByVal permId As Integer, ByVal parentId As Integer, ByVal lastFillPrice As Double, ByVal clientId As Integer, ByVal whyHeld As String)

**Note:** 也有可能orderStatus()返回重复的信息。对信息进行相应的过滤是必须的。

参数	描述
id	订单代号在之前的启动placeOrder()中已经指定。

<p>status</p>	<p>定单状态。可能的值包括：</p> <ul style="list-style-type: none"> <li>• <b>PendingSubmit</b> - 表示你以及传递了定单，但还未收到定单目的地已经接受的确认。</li> <li>• <b>PendingCancel</b> - 表示你已经发送了一个取消定单的请求，但还未收到来自定单目的地的取消确认。此时，你的定单并没有被确认已取消。在你的取消请求等待期间，你可能仍然会得到执行。 <b>PendingSubmit</b>和<b>PendingCancel</b>定单状态不是由系统发出的，当一个定单被取消时应该由API设计者发出。</li> <li>• <b>PreSubmitted</b> - 表示一个模拟定单类型已经被系统接受，且这个定单还未被选中。定单存在系统中直到设定的条件被满足。此时定单被发送到指定的定单目的地。</li> <li>• <b>Submitted</b> - 表示你的定单已经被定单目的地接受，正在处理中。</li> <li>• <b>Cancelled</b> - 表示你的定单已经被系统确认取消了。这也可能会是在目的地拒绝了你的定单时发生的。</li> <li>• <b>Filled</b> - 表示定单已经被全部执行。</li> <li>• <b>Inactive</b> - 表示定单已经被系统接受(模拟定单)，或被某个交易所接受(本地定单)，但目前由于系统、交易所或其它原因，定单处于非活跃状态。</li> </ul>
<p>filled</p>	<p>指定已被执行的股票数目。 有关部分执行的详细信息，请参阅<a href="#">部分执行的定单状态</a>。</p>
<p>remaining</p>	<p>说明还未获执行的股数。</p>
<p>avgFillPrice</p>	<p>已执行的股票平均价格这个参数仅在如果<b>filled</b>参数值是大于零的情况下才有效。否则，价格参数将为零。</p>
<p>permId</p>	<p>用于确认定单的代号。整个过程中保持不变。</p>
<p>parentId</p>	<p>母定单的定单代号，应用于括号和自动跟踪止损定单。</p>
<p>lastFillPrice</p>	<p>执行股票的最后价格。仅在<b>filled</b>(执行的)参数值大于零是才有效。否则，价格参数将为零。</p>
<p>clientId</p>	<p>下达定单客户的代号。注意，应用程序定单使用一个固定的<b>clientId</b>和<b>orderId</b>为0的来和API定单相区别。</p>

### openOrderEx()

启动这个方法来输送开放定单。

**Sub openOrderEx(ByVal orderId As Integer, ByVal contract As TWSLib.IContract, ByVal order As TWSLib.IOrder, ByVal orderState As TWSLib.IOrderState)**

参数	描述
orderId	由TWS分派的定单代号。用于取消或更新定单。
contract	描述合约的合约级别属性。
order	定义定单详细的定单级别属性。
orderState	orderState属性包括交易前和交易后数据的保证金和佣金区域。

### nextValidId()

这个行动的启动是在成功与TWS连接后。

**Sub nextValidId(ByVal id As Integer)**

参数	描述
id	与TWS联通后接收到的下一个可用的定单代号。所有后续的定单的代号将根据这个代号增加1。

### permId()

这个行动永远是在定单状态行动之后收到。它将对特定的定单代号给与permId。permId将在所有过程中保持不变。

**Sub permId(ByVal id As Integer, ByVal permId As Integer)**

参数	描述
id	与TWS联通后接收到的下一个可用的定单代号。所有后续的定单的代号将根据这个代号增加1。
permId	这个代号将在所有过程中保持不变。

### updateAccountValue()

这个行动更新一个单一账户值。

**Sub updateAccountValue(ByVal key As String, ByVal value As String, ByVal currency As String, ByVal accountName As String)**

参数	描述
----	----

<p>key( 标签)</p>	<p>一个表示一种账户值的字符串。以下为TWS发送的一些标签.</p> <ul style="list-style-type: none"> <li>• Account Type( 账户类型)</li> <li>• Account Code( 账户代码)</li> <li>• Available Funds( 可用资金)</li> <li>• Buying Power( 购买力)</li> <li>• CashBalance - Account cash balance( 账户现金余额)</li> <li>• Currency - Currency string( 货币字符串)</li> <li>• DayTradesRemaining - Number of day trades left( 所剩交易天数)</li> <li>• EquityWithLoanValue - Equity with Loan Value( 含借贷值股权)</li> <li>• Excess Liquidity( 剩余流动性)</li> <li>• Full Available Funds( 全部可用资金)</li> <li>• Full Excess Liquidity( 全部剩余流动性)</li> <li>• Full Init Margin Req( 全部初始保证金要求)</li> <li>• Full Maint Margin Req( 全部维持保证金要求)</li> <li>• Future Option Value( 期货期权值)</li> <li>• Futures PNL( 期货PNL)</li> <li>• Gross Position Value( 头寸值总计)</li> <li>• InitMarginReq - Current initial margin requirement( 当前初始保证金要求)</li> <li>• Leverage( 杠杆)</li> <li>• Look Ahead Available Funds( 预计可用资金)</li> <li>• Look Ahead Next Change( 预计下一个改变)</li> <li>• Look Ahead Excess Liquidity( 预计剩余流动性)</li> <li>• Look Ahead Margin Req( 预计保证金要求)</li> <li>• Look Ahead Maint Margin Req( 预计维持保证金要求)</li> <li>• LongOptionValue - Long option value( 做多头寸值)</li> <li>• MaintMarginReq - Current maintenance margin( 当前维持保证金)</li> <li>• NetLiquidation - Net liquidation value( 净清算值)</li> <li>• OptionMarketValue - Option market value( 期权市值)</li> <li>• Realized PNL( 实现的PNL)</li> <li>• Settled Cash( 结算的现金)</li> <li>• ShortOptionValue - Short option value( 做空期权值)</li> <li>• StockMarketValue - Stock market value( 股票市值)</li> </ul>
-----------------	--

	<ul style="list-style-type: none"> <li>• Total Cash Balance( 总现金余额)</li> <li>• Total Cash Value( 总现金值)</li> <li>• UnalteredInitMarginReq - Overnight initial margin requirement( 隔夜初始保证金要求)</li> <li>• UnalteredMaintMarginReq - Overnight maintenance margin requirement( 隔夜维持保证金要求)</li> <li>• Unrealized PNL( 未实现的PNL)</li> </ul>
value( 值)	与标签相关的值。
currency( 货币)	定义货币的值，如果值是个金额。
account( 账户)	应用于说明账户的信息。对金融顾问子账户信息非常要有用。

**updatePortfolioEx()**

这个回调是响应[reqAccountUpdates\(\)](#)方法。

**Sub updatePortfolioEx(ByVal contract As TWSLib.IContract, ByVal position As Integer, ByVal marketPrice As Double, ByVal marketValue As Double, ByVal averageCost As Double, ByVal unrealizedPNL As Double, ByVal realizedPNL As Double, ByVal accountName As String)**

参数	描述
contract	这个目标包括交易合约的描述。合约的交易所区域没有为投资组合更新设置。
position	这个整数表示合约的头寸。如果头寸为0，代表头寸已被清空。
marketPrice	交易产品的单位价格。
marketValue	交易产品的总市值。
averageCost	每股平均成本的计算是用你头寸的数量除以你的成本( 执行价格+佣金) 。
unrealizedPNL	你的未平仓头寸的当前市值和平均成本或平均成本的值的差。
realizedPNL	显示你平仓头寸的利润，为你的买进执行成本( 建仓的执行价格+佣金) 和卖出执行成本( 平仓的执行价格+佣金) 的差。
accountName	信息所指的账户名称。对金融顾问子账户信息非常有用。

**updateAccountTime()**

这个行动在发送时间为账户值和投资组合市价被计算出后。

**Sub updateAccountTime(ByVal timeStamp As String)**

参数	描述
timeStamp	表示账户信息的最后更新时间。

**updateNewsBulletin()**

这个行动被客户订阅的每个新的公告触发( 例如，通过启动reqNewsBulletins()方法) 。

**Sub updateNewsBulletin(ByVal msgId As Short, ByVal msgType As Short, ByVal message As String, ByVal origExchange As String)**

参数	描述
msgId	公告代号，每个新公告的代号将增加。

msgType	指定公告类型。有效的值包括： <ul style="list-style-type: none"> <li>• 1 = 普通IB新闻公告</li> <li>• 2 = 不再提供交易的交易所。</li> <li>• 3 = 可用于交易的交易所。</li> </ul>
message	公告信息文字。
origExchange	发出信息的交易所。

### contractDetailsEx()

这个行动的启动仅在[reqContractDetailsEx\(\)](#)方法被启动时发生。

**Sub contractDetailsEx(ByVal reqId As Integer, ByVal contractDetails As TWSLib.IContractDetails**

参数	描述
reqId	数据请求的代号。如果处理几个请求时，确保结果和请求相匹配。
contractDetails	这个目标包括一个查看合约的完整描述。

### contractDetailsEnd()

这个方法在请求的所有合约详细收到后启动。这将帮助定义期权链的截止。

**Sub contractDetailsEnd(ByVal reqId As Integer)**

参数	描述
reqID	数据请求的代号。

### bondContractDetails()

从TWS版本921开始，某些债券合约数据受到限制和不能通过API使用。所有债券合约数据仍然通过交易平台可用，仅以下债券合约数据可以通过API使用：

- 合约代号
- 最小跳动
- CUSIP(如果你订阅了CUSIP服务)
- 评级(如果你已经订阅了评级服务)

**Sub bondContractDetails(ByVal symbol As String, ByVal secType As String, ByVal cusip As String, ByVal coupon As Double, ByVal maturity As String, ByVal issueDate As String, ByVal ratings As String, ByVal bondType As String, ByVal couponType As String, ByVal convertible As Integer, ByVal callable As Integer, ByVal putable As Integer, ByVal descAppend As String, ByVal exchange As String, ByVal curency As String, ByVal marketName As String, ByVal tradingClass As String, ByVal conId As Integer, ByVal minTick As Double, ByVal orderTypes As**



**String, ByVal validExchanges As String, ByVal nextOptionDate As String, ByVal nextOptionType As String, ByVal nextOptionPartial As Integer, ByVal notes As String)**

参数	描述
symbol	债券代码。
secType	债券
cusip	9位代码的债券CUSIP，或12位SEDOL。
coupon	利率—用于计算你将在一年过程中收到的利息。
maturity	债务发行人必须偿还债券面值的日期。
issueDate	债券发行的日期。
ratings	发行人的信用评级。信用评级越高表明投资风险越低。债券评级是由穆迪( Moody's) 和标普( S&P) 提供的。
bondType	债券类型，比如“Corp”为公司债券。
couponType	利息类型，比如“固定的”。
convertible	值为：真 或 假。如果为真，在一定条件下债券可以被转换为股票。
callable	值为：真 或 假。如果是真，在一定条件下，发行人可以召回债券。
putable	值为：真 或 假。如果是真，在一定条件下债券可以回售给发行人。
descAppend	包含关于债券详细描述字符串。
exchange	债券交易的交易所。
curency	债券交易的货币。
marketName	合约的市场名称。
tradingClass	合约的交易级别名称。
conId	债券的IB合约代码。
minTick	债券的最小价格增量。
orderTypes	用于这个债券的定单类型。
validExchanges	这个债券交易的交易所的逗号分隔的字符串。
nextOptionDate	下一个期权日。仅用于带有期权的债券。
nextOptionType	下一个期权类型。仅用于带有期权的债券。

nextOptionPartial	下一个部分期权。仅用于具有期权的债券(下一个期权是全或部分的?)。
notes	债券票据, 为IB的数据库中的债券填充。

**execDetailsEx()**

这个行动的启动是当[reqExecutionsEx\(\)](#)方法被调用时, 或当一个定单被执行时。

**Sub execDetailsEx(ByVal reqId As Integer, ByVal contract As TWSLib.IContract, ByVal execution As TWSLib.IExecution)**

参数	描述
orderId	之前在启动 <a href="#">placeOrderEx()</a> 中定义的定单代号。
contract	这个目标包括执行合约的完整描述。
execution	这个结构包括额外的定单执行信息。

**execDetailsEnd()**

这个方法启动是当所有执行都根据[reqExecutionsEx\(\)](#)发送给客户时。

**Sub execDetailsEnd(ByVal reqId As Integer)**

参数	描述
reqID	数据请求的代码。

**commissionReport()**

**Sub commissionReport(ByVal commissionReport As TWSLib.ICommissionReport)**

参数	描述
commissionReport	包括佣金详细的结构。

**updateMktDepth()**

这个功能的启动是当市场深度发生变化时。

**Sub updateMktDepth(ByVal id As Integer, ByVal position As Integer, ByVal operation As Integer, ByVal side As Integer, ByVal price As Double, ByVal size As Integer)**

参数	描述
id	之前启动 <a href="#">reqMktDepth()</a> 中指定的定单代号。
position	指定这个市场深度输入行的代号。

operation	指定这个定单将如何应用于市场深度。有效值包括： <ul style="list-style-type: none"> <li>• 0 = 插入( 在由'头寸'确定的行内插入这个新定单) 。</li> <li>• 1 = 更新( 在由'头寸'指定的行中更新现有定单) 。</li> <li>• 2 = 删除( 在由'头寸'指定的行内删除现有定单) 。</li> </ul>
side	定单属于的在册的边。有效值为： <ul style="list-style-type: none"> <li>• 0 = 卖价</li> <li>• 1 = 买价</li> </ul>
price	定单价格。
size	定单尺寸。

### updateMktDepthL2()

这个功能的启动是当二级市场深度改变时。

**Sub updateMktDepthL2(ByVal id As Integer, ByVal position As Integer, ByVal marketMaker As String, ByVal operation As Integer, ByVal side As Integer, ByVal price As Double, ByVal size As Integer)**

参数	描述
id	之前启动reqMktDepth()中指定的定单代号。
position	指定这个市场深度输入行的代号。
marketMaker	指定持有该定单的交易所。
operation	指定该定单应该被应用的市场深度的行。有效值包括： <ul style="list-style-type: none"> <li>• 0 = 插入( 在由'头寸'确定的行内插入这个新定单) 。</li> <li>• 1 = 更新( 在由'头寸'指定的行中更新现有定单) 。</li> <li>• 2 = 删除( 在由'头寸'指定的行内删除现有定单) 。</li> </ul>
side	指定该定单属于的在册边。有效值是： <ul style="list-style-type: none"> <li>• 0 = 卖价</li> <li>• 1 = 买价</li> </ul>
price	定单价格。
size	定单尺寸。

### managedAccounts()

这个行动代码的启动是当成功与金融顾问账户连接时。也是当reqManagedAccts()方法被启动时。

#### Sub managedAccounts(ByVal accountsList As String)

参数	描述
accountsList	FA管理账户的逗号分隔的列表。

### receiveFA()

这个行动接收之前请求的来自TWS的FA配置信息。

#### Sub receiveFA(ByVal faDataType As Integer, ByVal cxml As String)

参数	描述
faDataType	指定从TWS接收到的金融顾问配置数据的类型。有效值包括： <ul style="list-style-type: none"> <li>• 1 = 组</li> <li>• 2 = 概况</li> <li>• 3 = 账户别名</li> </ul>
cxml	包含之前请求的FA配置信息的XML字符串。

### accountSummary()

回复[reqAccountSummary\(\)](#)从TWS账户窗口概况标签返回的数据。

#### Sub accountSummary(ByVal messageType As Integer, ByVal version As Integer, ByVal requestId As Integer, ByVal account As String, tag As String, value As String, currency As String)

参数	类型	描述
messageType	整数	设为 62。
version	整数	设为 1。
requestId	整数	数据请求的代号。
account	字符串	账户代号。

参数	类型	描述
tag	字符串	<p>数据请求的标签。 可用的标签包括：</p> <ul style="list-style-type: none"> <li>• <i>AccountType</i></li> <li>• <i>TotalCashValue</i> — 包括期货 pnl 的总现金</li> <li>• <i>SettledCash</i> — 对每个现金账户，这个与 <i>TotalCashValue</i> 相同</li> <li>• <i>AccruedCash</i> — 净累计利息</li> <li>• <i>BuyingPower</i> — 账户可以买价的保证金美国股票的最大数额。</li> <li>• <i>EquityWithLoanValue</i> — 现金 + 股票 + 债券 + 共同基金</li> <li>• <i>PreviousEquityWithLoanValue</i></li> <li>• <i>GrossPositionValue</i> — 所有股票和股票期权头寸的绝对值总额。</li> <li>• <i>RegTEquity</i></li> <li>• <i>RegTMargin</i></li> <li>• <i>SMA</i> — 特殊备忘录账户</li> <li>• <i>InitMarginReq</i></li> <li>• <i>MaintMarginReq</i></li> <li>• <i>AvailableFunds</i></li> <li>• <i>ExcessLiquidity</i></li> <li>• <i>Cushion</i> — 以净清算值表示的剩余流动性</li> <li>• <i>FullInitMarginReq</i></li> <li>• <i>FullMaintMarginReq</i></li> <li>• <i>FullAvailableFunds</i></li> <li>• <i>FullExcessLiquidity</i></li> <li>• <i>LookAheadNextChange</i> — 预期值发生的时间</li> <li>• <i>LookAheadInitMarginReq</i></li> <li>• <i>LookAheadMaintMarginReq</i></li> <li>• <i>LookAheadAvailableFunds</i></li> <li>• <i>LookAheadExcessLiquidity</i></li> <li>• <i>HighestSeverity</i> — 衡量账户如何接近被清算</li> <li>• <i>DayTradesRemaining</i> — 在日交易模式被显示前用户可以进行的开仓/平仓交易。 值为“-1”表示用户可以进行无限制的日交易。</li> <li>• <i>Leverage</i> — <math>GrossPositionValue / NetLiquidation</math></li> </ul>

参数	类型	描述
value	字符串	标签的值。
currency	字符串	标签的货币。

### accountSummaryEnd

这个方法启动是当一个请求的所有账户的概括数据被收到时。

#### Sub accountSummaryEnd(ByVal reqId As Integer)

参数	类型	描述
reqId	整数	数据请求的代号。

### position()

这个行动返回[reqPositions\(\)](#)方法的所有账户的实时头寸。

#### Sub position(ByVal messageId As Integer, ByVal version As Integer, ByVal account as String, ByVal conid As Integer, ByVal underlying As String, ByVal securityType As String, ByVal expiry As String, ByVal strike As String, ByVal right As String, ByVal multiplier As String, ByVal exchange As String, ByVal currency As String, By Val ibLocalSymbol As String, ByVal position As double)

参数	类型	描述
messageId	整数	设为 62。
version	整数	设为 1。
account	字符串	账户。
conid	整数	独特的合约代码。
underlying	字符串	底层资产的代码。
securityType	字符串	证券类型。
expiry	字符串	过期日。
strike	字符串	行使价。
right	字符串	看跌或看涨。
multiplier	字符串	乘数。
exchange	字符串	交易所。
currency	字符串	货币。
ibLocalSymbol	字符串	本地代码。
position	double	头寸。

### positionEnd()

这个启动是当一个给定请求的所有头寸数据全部收到和position()数据的结束标志的功能。

#### Sub positionEnd(ByVal reqId As Integer)

参数	类型	描述
reqId	整数	数据请求的代号。

### historicalData()

这个行动接收来自TWS的请求的历史数据。

#### Sub historicalData(ByVal reqId As Integer, ByVal date As String, ByVal open As Double, ByVal high As Double, ByVal low As Double, ByVal close As Double, ByVal volume As Integer, ByVal barCount As Integer, ByVal WAP As Double, ByVal hasGaps As Integer)

参数	描述
reqId	柱所代表的请求的代码代号。
date	柱开始的日期时间标记。格式由reqHistoricalData() formatDate参数确定。
open	柱的开始价格。
high	柱所覆盖的时间内的高价。
low	柱所覆盖的时间内的低价。
close	柱的结束价。
volume	柱所覆盖时间内的交易量。
WAP	柱所覆盖时间内的加权平均价。
hasGaps	指定数据中是否有间断。

### scannerParameters()

#### Sub scannerParameters(ByVal xml As String)

参数	描述
xml	描述扫描仪参数可以使用有效参数的XML文件。

### scannerDataEx()

这个行动接收请求的市场扫描仪数据结果。

#### Sub scannerDataEx(ByVal reqId As Integer, ByVal rank As Integer, ByVal contractDetails As TWSLib.IContractDetails, ByVal distance As String, ByVal benchmark As String, ByVal projection As String, ByVal legsStr As String)

参数	描述
reqId	行所对应的请求代号。
rank	这个柱在结果中的排列。
contractDetails	这个目标包括合约的完整描述。
distance	根据询问而变化。
benchmark	根据询问而变化。
projection	根据询问而变化。
legsStr	当扫描的结果是EFP时，对组合边的描述。

### scannerDataEnd()

这个功能的启动是当收到快照和一次扫描结束的标志时。

#### Sub scannerDataEnd(ByVal reqId As Integer)

参数	描述
reqId	由这个参数结束的市场数据快照请求的代号。

### realtimeBar()

这个方法接收实时柱数据结果。

#### Sub realtimeBar(ByVal tickerId As Integer, ByVal time As Integer, ByVal open As Double, ByVal high As Double, ByVal low As Double, ByVal close As Double, ByVal volume As Integer, ByVal WAP As Double, ByVal Count As Integer)

参数	描述
reqId	该柱所对应的请求代码代号。
time	柱开始的日期-时间标志。格式由reqHistoricalData() formatDate参数决定。
open	柱的开始价。
high	柱所覆盖的时间内的高价。
low	柱所覆盖的时间内的低价。
close	柱的结束价。
volume	柱所覆盖时间内的交易量。
wap	柱所覆盖时间内的加权平均价。
count	当交易历史数据返回时，代表柱所覆盖的区间内发生的交易次数。



### fundamentalData()

启动这个方法接收路透社全球基本面市场数据。在你能够接收这个数据之前，你必须先通过账户管理订阅路透社基本面。

#### Sub fundamentalData(ByVal reqId As Integer, ByVal data As String)

参数	描述
reqId	数据请求的代号。
data	三个XML报告之一： <ul style="list-style-type: none"> <li>• 估计(estimate)</li> <li>• 财务报表(finstat)</li> <li>• 概况(snapshot)</li> </ul>

## ActiveX COM对象

下面表格定义了以下对象的特性：

- [IExecution](#)
- [IExecutionFilter](#)
- [ICommissionReport](#)
- [IContract](#)
- [IContractDetails](#)
- [IComboLeg](#)
- [IComboLegList](#)
- [IOrder](#)
- [IOrderState](#)
- [IScannerSubscription](#)
- [ITagValueList](#)
- [ITagValue](#)
- [IUnderComp](#)

这一章中，你必须使用生产“创建”方法来创建 COM对象。一旦一个COM对象通过一个生产方法被创建，该COM对象即与一个相应的TWS COM对象(一个实际COM对象)相连。不要试图将一个COM对象转给另一个实例TWS COM对象。

### IExecution

属性	描述
acctNumber() As String	客户账户号码。
avgPrice() As Double	平均价格。用于一般交易、组合交易和组合边。包括佣金。
clientId() As Integer	下达订单客户的代号。  <b>Note:</b> TWS订单的客户代号是固定的，为“0”。
cumQty() As Integer	累积数量。用于一般交易、组合交易和组合边。
exchange() As String	执行交易的交易所。
execId() As String	独特的订单执行代号。
liquidation() As Integer	在需要清算时，指定最后被清算的头寸。

orderId() As Integer	定单代号。 <b>Note:</b> TWS定单的固定定单代号为“0”。
permId() As Integer	用于确认定单的TWS代号，整个过程保持不变。
price() As Double	不包括佣金的定单执行价格。
shares() As Integer	执行的股数。
side() As String	说明交易是卖出或买进。有效值是： <ul style="list-style-type: none"> <li>• BOT ( 买进)</li> <li>• SLD ( 卖出)</li> </ul>
time() As String	定单执行时间。
evRule() As String	包含经济价值规则名称和相应的可选参数。两个值应该用冒号隔开。例如， aussieBond:YearsToExpiration=3。当没有提供可选参数时，第一个值后将使用一个冒号。
evMultiplier As Double	告诉你如果价格改变1，合约的市值改变是多少。不能用乘数乘以价格来获得市值。

**IExecutionFilter**

属性	描述
acctCode() As String	根据账户编码过滤reqExecutionsEx()方法的结果。注：这只是相对于金融顾问( FA) 账户。
clientId() As Integer	根据clientId过滤reqExecutionsEx()方法的结果。
exchange() As String	根据定单交易所过滤reqExecutionsEx()方法的结果。
secType() String	根据定单证券类型过滤reqExecutionsEx()方法的结果。 <b>Note:</b> 参阅有效证券类型列表的合约目标。
side() As String	根据定单行动过滤reqExecutionsEx()方法的结果。 <b>Note:</b> 参阅有效定单行动列表的定单目标。
symbol() As String	根据定单代码过滤reqExecutionsEx()方法的结果。
time() As String	根据指定时间后接收到的执行报告过滤reqExecutionsEx()方法结果。timeFilter的格式是“yyyymmdd-hh:mm:ss”

### ICommissionReport

属性	描述
commission() As Double	佣金额。
currency() As String	货币。
execId() As String	独特的定单执行代号。
realizedPNL() As Double	实现的利润和损失数额。
yield() As Double	收益率。
yieldRedemptionDate() As Double	使用YYYYMMDD格式。

### IContract

属性	描述
comboLegs() As Object	用于记录合约边定义的动态记忆结构。
comboLegsDescrip() As String	描述组合边。
conId() As Integer	独特的合约代号。
currency() As String	指定货币。可能需要对这个区域给出不确定的结果，例如，当使用SMART(智能传递)做为交易所和请求IBM时(IBM可以以英镑或美元交易)。考虑到这种不确定的存在，建议最好指定货币。
exchange() As String	定单目的地，比如Smart(智能)。
expiry() As String	过期日。使用格式YYYYMM。
includeExpired() As Integer	如果设置为真，可以对有关的过期合约进行合约详细请求和历史数据请求。 注：对过期合约的历史数据请求限制在合约寿命期限内的最后一年，并初始只支持过期的期货合约。
localSymbol() As String	这是底层资产的本地交易所代码。
multiplier() As String	允许你指定期货或期权合约乘数。这个只有当存在多种可能的情况下需要。
primaryExch() As String	指定合约的交易所列表(不需要列出Smart)。
right() As String	指定看跌或看涨。有效值是：P、PUT、C、CALL。

secId as String	独特的secIdType标识。
secIdType As String	<p>证券标识符，当寻求合约详细或下达定单时。支持的标识符包括：</p> <ul style="list-style-type: none"> <li>• ISIN (例如： Apple: US0378331005)</li> <li>• CUSIP (例如： Apple: 037833100)</li> <li>• SEDOL (包括6-AN + 校验位。 例如： BAE: 0263494)</li> <li>• RIC (包含交易所—独立RIC根和交易所确认后缀。 Exa</li> </ul>
secType() As String	<p>这是证券类型。有效值是：</p> <ul style="list-style-type: none"> <li>• STK</li> <li>• OPT</li> <li>• FUT</li> <li>• IND</li> <li>• FOP</li> <li>• CASH</li> <li>• BAG</li> </ul>
strike() As Double	行使价。
symbol() As String	这是底层资产代码。

### IContractDetails

属性	描述
category() As String	底层证券的行业分类。 例如， InvestmentSvc。
contractMonth() As String	合约月份。 通常为期货合约的底层证券合约月份。
industry() As String	底层证券/产品的行业分类。 例如， Financial( 金融) 。
liquidHours() As String	产品的流动交易时段。 例如， 20090507:0930-1600;20090508:CLOSED。
longName() As String	资产的描述名称。
marketName() String	该合约的市场名称。
minTick() As Double	最小价格跳动。

orderTypes() As String	该合约的有效定单类型列表。
priceMagnifier() Integer	允许执行和行使价的报告始终与市场数据、历史数据和定单价格相一致，例如， Z on LIFFE的报告是以指数点，而不是GBP。
ratings() As String	确定发行人的信用评级。比较高的信用评级表示比较低的投资风险。债券的评级由穆迪和标普给出。
secIdList() As Object	客户可以查看的合约标识符列表( CUSIP、ISIN。等)
subcategory() As String	底层证券的子行业类别。例如，经纪商。
summary() As Object	合约概括。
timeZoneId() As String	产品交易时间的时区代号。例如， EST( 东部标准时间) 。
tradingClass() As String	该合约的交易级别名称。
tradingHours() As String	产品的交易时间。例如， 20090507:0700-1830,1830-2330;20090508:CLOSED。
underConId() As String	底层合约代号。
validExchanges() As String	交易该合约的交易所列表。
evRule() As String	包含经济价值规则名称和有关的可选参数。两个值应该用冒号隔开。例如， aussieBond:YearsToExpiration=3。当可选参数不存在时，第一个值之后使用冒号。
evMultiplier As Double	告诉你如果价格改变1，合约的市值改变是多少。不能用乘数乘以价格来获得市值。
<b>债券值</b>	
bondType() As String	证券类型，比如“CORP.”( 公司)
callable() As Integer	值为真或假。如果为真，在一定条件下，债券可以被发行人召回。
convertible() As Integer	值为真或假。如果为真，在一定条件下，债券可以被转换为股票。
coupon() As Double	用于计算你在一年中将收到的利息的利率。
couponType() As String	债券利率类型，比如“FIXED”( 固定的) 。
cusip() As String	9位字符的债券CUSIP或12位字符的SEDOL。
descAppend() As String	包含关于债券详细描述信息的描述字符串。

issueDate() As String	债券的发行日期。
maturity() As String	发行人必须偿还债券面值的日期。
nextOptionDate)_ As String	应用于含有期权的债券。
nextOptionPartial() As Integer	应用于含有期权的债券。
nextOptionType() As String	应用于含有期权的债券。
notes() As String	如果将债券包括在IB数据库中
putable() As Integer	值为真或假。 如果为真，在一定条件下，债券可以被回售给发行人。

### IComboLeg

属性	描述
action() As String	你创建的边的一侧( 买或卖) 。
conId() As Integer	指定证券的独特合约标识符。
exchange() As String	将完整的组合定单发送到的交易所。
openClose() As Integer	指定定单是开放或关闭定单。有效值是： <ul style="list-style-type: none"> <li>• 0 - 与母定单相同。 这是零售客户的唯一选择。</li> <li>• 1 - 开放。 这个值只对机构客户有效。</li> <li>• 2 - 关闭。 这个值只对机构客户有效。</li> <li>• 不知道 - (3)</li> </ul>
ratio() As Integer	为你创建的边选择合约的相应数字。 来帮助确定一个特定组合定单的比率，请参考用户指南的互动分析部分。
<b>用于卖空股票边</b>	
designatedLocation() As String	如果shortSaleSlot = 2，则必须指定designatedLocation。 否则留为空白或定单将被拒绝。
shortSaleSlot() Integer	仅用于机构客户。 <ul style="list-style-type: none"> <li>• 0 - inapplicable ( 例如 零售客户或非卖空边)</li> <li>• 1 - 结算经纪商</li> <li>• 2 - 第三方。 如果使用这个值，你必须输入一个指定的地点。</li> </ul>

### IComboLegList

属性	描述
Add() As Object	添加组合边到组合边列表中。
Count() As Integer	边数目。
Item(Integer) As Object	通过指数获得边。

### IOrder

属性	描述
<b>订单标识符</b>	
clientId() As Integer	下达订单的客户代号。
orderId() As Integer	订单代号。
permId() As Integer	用于确认订单的TWS代号，整个TWS过程中保持不变。
<b>主订单区域</b>	
action() As String	确认那一侧。有效值是:买、卖、卖空
auxPrice() As Double	这是止损限价单的止损价格，和相对订单的抵消额。所有其它情况下为零。
lmtPrice() As Double	用于限价单、止损限价单和相对订单的限价价格。所有其它情况下为零。没有限价价格的相对订单也指定为零。
orderType() As String	确定订单类型。  关于支持的订单类型的更多信息，请参阅 <a href="#">支持的订单类型</a> 。
totalQuantity() As Integer	订单数量。
<b>扩展的订单区域</b>	
allOrNone() As Integer	0 = no, 1 = yes
blockOrder() As Integer	指定订单是ISE批量订单。
displaySize() As Integer	公开披露订单尺寸，在下达冰山( Iceberg) 订单时使用。
goodAfterTime() As String	交易“过时后有效”的格式 “YYYYMMDD hh:mm:ss (时区为可选)” 不适用时使用空白字符串。



goodTillDate() As String	使用这个字符串你必须输入GTD( 到时有有效) 做为有效时间。交易“到时有有效”格式“YYYYMMDD hh:mm:ss (时区为可选)” 不适用时使用空白字符串。
hidden() As Integer	指定当查看市场深度是，订单将为不可视。这个选择仅用于传递到ISLAND交易所的订单。
minQty() As Integer	指定最低数量订单类型。
ocaGroup() As String	确定一个OCA( 一取消全) 组。
ocaType() As Integer	<p>当一个订单或部分订单被执行时，指示如何处理OCA组中剩余的订单。有效值包括：</p> <ul style="list-style-type: none"> <li>• 1 = 取消批量和剩余的订单</li> <li>• 2 = 保持批量并对剩余订单按比率减少尺寸</li> <li>• 3 = 按比率减少剩余订单的尺寸，取消批量。</li> </ul> <p>如果你使用值”保持批量“将保护你的订单不会被超额执行。意味着每次仅发送组中的一个订单，消除了超额执行的可能。</p>
orderRef() As String	订单参考。意在仅用于机构客户，尽管所有客户均就可以使用它来确认发送订单的API客户( 当存在多个API客户时) 。
outsideRth() As Integer	指定订单是否可以在正常交易时段以外触发或执行。
overridePercentageConstraints() As Integer	<p>在TWS预设页面上定义了预防约束，以帮助保证你的订单值的价格和尺寸是合理的。从API发送的订单也受这些安全约束的限制，如果违反了任何限制，订单也可能被拒绝。取消验证，设定这个参数的值为真。</p> <p>有效值包括：</p> <ul style="list-style-type: none"> <li>• 0 = 假</li> <li>• 1 = 真</li> </ul>
parentId() As Integer	母订单的订单代号，用于括号单和自动跟踪止损单。
percentOffset() As Double	相对订单的百分比抵消额。

rule80A() As String	<p>值包括：</p> <ul style="list-style-type: none"> <li>• Individual = 'I'</li> <li>• Agency = 'A',</li> <li>• AgentOtherMember = 'W'</li> <li>• IndividualPTIA = 'J'</li> <li>• AgencyPTIA = 'U'</li> <li>• AgentOtherMemberPTIA = 'M'</li> <li>• IndividualPT = 'K'</li> <li>• AgencyPT = 'Y'</li> <li>• AgentOtherMemberPT = 'N'</li> </ul>
sweepToFill() As Integer	指定是否定单是一扫光定单。
tif() As String	有效时间。有效值是：DAY, GTC, IOC, GTD.
transmit() As Integer	指定是否定单将由TWS发送。
triggerMethod() As Integer	<p>指定如何模拟止损、止损限价和跟踪止损单被触发。有效值包括：</p> <ul style="list-style-type: none"> <li>• 0 - 默认值。将对场外交易股票和美国期权的定单使用”双买价/卖价“功能。所有其它定单将使用”最后“功能。</li> <li>• 1 - 使用”双买价/卖价“功能，止损单的触发基于两个连续的买价或卖价。</li> <li>• 2 - ”最后“功能，止损单的触发基于最后价格。</li> <li>• 3 双最后功能。</li> <li>• 4 买价/卖价功能。</li> <li>• 7 最后或买价/卖价功能。</li> <li>• 8 中点功能。</li> </ul>
trailStopPrice() As Double	仅用于TRAILLIMIT定单

trailingPercent() As Double	<p>指定跟踪止损单的跟踪额为一个百分比。当使用trailingPercent区域时遵循下列准则：</p> <ul style="list-style-type: none"> <li>• 这个区域与现有的跟踪额是相互排斥的。也就是，API客户可以发送其中之一，而不能两个都发。</li> <li>• 这个区域是在止损价( 屏障价格) 之后使用的：                     <ul style="list-style-type: none"> <li>deltaNeutralAuxPrice</li> <li>stopPrice</li> <li>trailingPercent</li> </ul>                     分段定单属性                 </li> <li>• 如果API客户的版本是 &gt;= 56，该区域也将在openOrder信息中发给API。该区域在stopPrice区域之后发送：                     <ul style="list-style-type: none"> <li>stopPrice</li> <li>trailingPct</li> <li>basisPoint</li> </ul> </li> </ul>
<b>金融顾问区域</b>	
faGroup() As String	金融顾问组的交易将被分配给 -- 使用一个空字符串，如果不适用时。
faMethod() As String	金融顾问分配功能的交易将被分配 -- 使用一个空字符串，如果不适用时。
faPercentage() As String	金融顾问百分比的交易分配 -- 使用一个空字符串，如果不适用时。
faProfile() As String	金融顾问分配概况的交易将被分配 -- 使用一个空字符串，如果不适用时。
<b>仅机构( 非清算的)</b>	
designatedLocation() As String	仅在 shortSaleSlot = 2 时使用。
openClose() As String	仅用于机构客户。有效值是O、C。
origin() As Integer	原始定单。仅用于机构客户。有效值是0 = 客户，1 = 公司
shortSaleSlot() As Integer	有效值是1或2。
<b>仅用于SMART传递</b>	
discretionaryAmt() As Double	全权委托定单允许使用的偏离限价的数额。
eTradeOnly() As Integer	使用电子报价交易。 0 = no, 1 = yes
firmQuoteOnly() As Integer	使用公司报价交易。 0 = no, 1 = yes

nbboPriceCap() As Double	偏离NBBP的最大智能传递订单。
optOutSmartRouting() As Integer	从默认的SmartRouting中将直接传递到ASX的订单排除。除非明确设置为真，这个属性的默认为假。当设置为假时，订单将不使用SmartRouting，而直接传递给ASX。当设置为真时，订单将使用SmartRouting直接传递到ASX。
notHeld() As Integer	仅用于IBDARK订单。传递到IBDARK的订单的标签是“post only(只发布)”并持有在IB的订单册中，来自其它IB客户的SmartRouting传递订单可以对它们进行交易。
<b>仅BOX或VOL订单</b>	
auctionStrategy() As Integer	包括的值： <ul style="list-style-type: none"> <li>• 匹配 = 1</li> <li>• 改进 = 2</li> <li>• 公开 = 3</li> </ul> 仅在BOX交易的订单。
<b>仅BOX交易所的订单</b>	
delta() As Double	股票delta。仅BOX交易的订单。
startingPrice() As Double	竞价起始价格。仅BOX交易的订单。
stockRefPrice() As Double	股票参考价格。参考价格被用于VOL订单来计算发送给交易所(不论是否选择了连续更新)的限价价格和价格范围监控。
<b>仅挂钩股票和VOL订单</b>	
stockRangeLower() As Double	可接受的底层股票价格范围的低端值。对具有动态管理的BOX和VOL订单的价格改善期权订单。
stockRangeUpper() As Double	可接受的底层股票价格范围的高端值。对具有动态管理的BOX和VOL订单的价格改善期权订单。
<b>仅波动率订单</b>	
continuousUpdate() As Integer	仅VOL(波动率)订单。指定在底层价格变动时是否TWS将自动更新订单的限价。
deltaNeutralOrderType() As String	仅VOL(波动率)订单。输入一个订单类型指示TWS对全额或部分执行的VOL订单发送一个delta中性交易。对将要发送的非对冲delta订单，指定为NONE(无)。

deltaNeutralAuxPrice() As Integer	仅VOL(波动率)定单。使用这个区域输入一个值，如果这个值在deltaNeutralOrderType区域中是需要辅助价格的定单类型，比如REL定单。
referencePriceType() As Integer	仅VOL(波动率)定单。指定你希望TWS任何计算期权的限价价格，和股票范围价格监控。 有效值包括： <ul style="list-style-type: none"> <li>• 1 = 平均NBBO</li> <li>• 2 = 根据行动和行使权，为NBB或NBO。</li> </ul>
volatility() As Double	期权价格波动率，通过TWS的期权分析计算得出的。这个值的表示为百分比，并被用于计算发送给交易所的限价价格。
volatilityType() As Integer	包括的值： <ul style="list-style-type: none"> <li>• 1 = 日波动率</li> <li>• 2 = 年波动率</li> </ul>
deltaNeutralOpenClose() As String	指定是否定单是开放或关闭定单，并被用于当对冲涉及CFD和定单由其它第三结算时。
deltaNeutralShortSale () As Integer	用于当对冲涉及股票和表明是否为沽空。
deltaNeutralShortSaleSlot() As Integer	值为1(清算经纪商持有股票)或2(由第三方提交)。如果你使用2，你必须指定一个deltaNeutralDesignatedLocation。
deltaNeutralDesignatedLocation() As String	仅用于当deltaNeutralShortSaleSlot = 2。
<b>仅组合定单</b>	
basisPoints() As Double	仅EFP定单
basisPointsType() As Integer	仅EFP定单
<b>仅分段定单</b>	
scaleAutoReset() As Integer	用于扩展分段定单。
scaleInitFillQty() As Integer	用于扩展分段定单。
scaleInitLevelSize() As Integer	用于分段定单：定义第一个尺寸，或初始尺寸，定单分组。

scaleInitPosition() As Integer	用于扩展分段定单。
scalePriceIncrement() As Double	用于分段定单：定义分段组之间的价格增量。这个区域是要求的。
scalePriceAdjustInterval() As Integer	用于扩展分段定单。
scalePriceAdjustValue() As Double	用于扩展分段定单。
scaleProfitOffset() As Double	用于扩展分段定单。
scaleRandomPercent() As Integer	用于扩展分段定单。
scaleSubsLevelSize() As Integer	用于分段定单：定义后续分段定单组的尺寸。与scaleInitLevelSize()一起使用。
<b>仅对冲定单</b>	
hedgeParam() As String	Beta = x 用于Beta对冲定单，比率 = y 用于双对冲定单
hedgeType() As String	用于对冲定单。可能的值包括： <ul style="list-style-type: none"> <li>• D = Delta</li> <li>• B = Beta</li> <li>• F = FX</li> <li>• P = Pair</li> </ul>
<b>清算信息</b>	
account() As String	账户。仅用于机构客户。
clearingAccount() As String	对IBExecution客户：指定定单的真实受益人。为向交易所报告，这个区域对FUT/FOP定单是要求的。
clearingIntent() As String	对IBExecution客户：有效值是：IB, Away, 和PTA 交易后分配) 。
settlingFirm() As String	仅机构。
<b>仅算法定单</b>	
algoStrategy() As String	有关API算法定单的信息，请参阅 <a href="#">IBAlgo 参数</a> 。
algoParams() As Object	支持IBAlgo参数。

<b>假设</b>	
whatIf() As Integer	用于请求交易前佣金和保证金信息。 如果设置为真，对 openOrder() callback 的保证金和佣金数据将通过 OrderState() 目标反馈接收。
<b>智能组合传递</b>	
smartComboRoutingParams() As Object	支持 <a href="#">智能组合传递</a> 。
<b>定单组合边</b>	
OrderComboLegs() As Object	持有组合定单中的所有边的属性。

### OrderComboLeg

属性	描述
double price	定单特定边价格。

### IOrderState

属性	描述
commission() As Double	显示定单的佣金数额。
commissionCurrency() As String	显示佣金值的货币。
equityWithLoan() As String	显示定单将对你的含借贷值股权的影响。
initMargin() As String	显示定单将对你的初始保证金的影响。
maintMargin() As String	显示定单将对你的维持保证金的影响。
maxCommission() As Double	配合 minCommission 区域一起使用，定义实际定单佣金可能范围的最高值。
minCommission() As Double	配合 maxCommission 区域一起使用，定义定义实际定单佣金可能范围的最低值。
status() As String	显示定单状态。
warningText() As String	必要时显示一个警告信息。

### IScannerSubscription

属性	描述
averageOptionVolumeAbove () As Integer	可以留为空白。
couponRateAbove() As String	过滤掉利率低于该值的合约。可以留为空白。
couponRateBelow() As String	过滤掉利率高于该值的合约。可以留为空白。
excludeConvertible() As Integer	过滤掉可转换债券。可以留为空白。
instrument() As String	定义扫描产品类型。
locations() As String	地点。
marketCapAbove() As Double	过滤掉市值低于该值的合约。可以留为空白。
marketCapBelow() As Double	过滤掉市值高于该值的合约。可以留为空白。
maturityDateAbove() As String	过滤掉到期日早于该值的合约。可以留为空白。
maturityDateBelow() As String	过滤掉到期日迟于该值的合约。可以留为空白。
moodyRatingAbove() As String	过滤掉穆迪评级低于该值的合约。可以留为空白。
moodyRatingBelow() As String	过滤掉穆迪评级高于该值的合约。可以留为空白。
numberOfRows() As Integer	定义查询返回的数据的行数。
priceAbove() As Double	过滤掉价格低于该值的合约。可以留为空白。
priceBelow() As Double	过滤掉价格高于该值的合约。可以留为空白。
scanCode() As String	可以留为空白。
scannerSettingPairs() As String	可以留为空白。例如，将"年，真"用于“最高期权隐含波动率百分比涨幅”扫描将可能返回年度波动率。
spRatingAbove() As String	过滤掉标普评级低于该值的合约。可以留为空白。
spRatingBelow() As String	过滤掉标普评级高于该值的合约。可以留为空白。



stockTypeFilter() As String	有效值包括： <ul style="list-style-type: none"> <li>• CORP = 公司</li> <li>• ADR = 美国存托凭证</li> <li>• ETF = 交易所交易基金</li> <li>• REIT = 房地产投资信托</li> <li>• CEF = 封闭式基金</li> </ul>
volumeAbove() As Integer	过滤掉交易量低于该值的合约。可以留为空白。

### ITagValueList

属性	描述
Count() As Integer	标签值对的数目( IBAalgo参数) 。
Item(Integer) As Object	一个标签值对( IBAalgo参数) 。有关详细信息，请参阅 <a href="#">IBAlgo参数</a> 。

### ITagValue

属性	描述
tag() As String	IBAlgo定单参数。更多信息，请参阅 <a href="#">IBAlgo参数</a> 。
value() As String	IBAlgo参数值。

### IUnderComp

属性	描述
conId() As Integer	识别证券的独特合约标识符。用于Delta—中性组合合约。
delta() As Double	底层股票或期货delta。用于Delta—中性组合合约。
price() As Double	底层证券价格。用于Delta—中性组合合约。

## ActiveX属性

下面表格定义了你在使用ActiveX与服务器连接时可以使用的属性。

属性	描述
String TwsConnectionTime	连接时间。
long serverVersion	服务器版本。

## 下达组合定单

组合定单是一种特殊类型的定单，由多个不同的边组成，但按一个单个交易执行。比如使用BAG证券类型(在合约目标中定义的)提交组合定单，如日历差价、转换和跨式期权等。使用API实施一个成功的API组合定单的关键是了解如何使用交易平台下达相同的定单。如果你熟悉在TWS中下达组合定单，则将非常容易使用API下达相同的定单，因为API仅是模仿TWS的操作。

### 举例

在这个例子中，一个客户对GOOG的日历差价下达一个买单。买进一个日历差价意味着：

**边 1: 卖 1 GOOG OPT SEP 18 '09 150.0 CALL (100)**

**边 2: 买 1 GOOG OPT JAN 21 '11 150.0 CALL (100)**

以下是使用API下达组合定单要求的步骤概括：

- 获取每个边的合约代号( conId) 。通过启动reqContractDetailsEx()方法获取这个号码。
- 填充相关的区域，包括IComboLeg COM目标的每个边。
- 使用IContract和IOrder COM目标实施placeOrderEx() 方法。

### 下达这个组合定单

1. 获取每个边定义的合约代码：

```
'第一边
Dim con1 As TWSLib.IContract
con1 = Tws1.createContract

con1.symbol = "GOOG"
con1.secType = "OPT"
con1.expiry = "200909"
con1.strike = 150.0
con1.right = "C"
con1.multiplier = "100"
con1.exchange = "SMART"
con1.currency = "USD"

Tws1.reqContractDetailsEx(1, con1)

'第二边
Dim con2 As TWSLib.IContract
con2 = Tws1.createContract

con2.symbol = "GOOG"
con2.secType = "OPT"
con2.expiry = "201101"
con2.strike = 150.0
con2.right = "C"
con2.multiplier = "100"
con2.exchange = "SMART"
con2.currency = "USD"
```

```
Tws1.reqContractDetailsEx(2, con2)

'所有的conId数字由ContractDetail()提交

Private Sub Tws1_contractDetailsEx(ByVal sender As Object, ByVal e As
AxTWSLib._DTwsEvents_contractDetailsExEvent) Handles Tws1.contractDetailsEx

Dim contractDetails As TWSLib.IContractDetails
contractDetails = e.contractDetails

Dim contract As TWSLib.IContract
contract = contractDetails.summary

'reqId = 1 是相应的第一个请求或第一个边
'reqId = 2 是相应的第二个请求或第二个边

如果 e.reqId = 1 则
边1 = contract.conId '获取第一边的conId
结束, 如果

如果e.reqId = 2 则
边2 = contract.conId '获取第二边的conId
结束, 如果

结束提交
```

- 一旦程序得到了每个边的conId, 将其包括在ComboLeg 目标中:

```
TWSLib.IComboLegList
addAllLegs = Tws1.createComboLegList

'第一组合边
Dim Leg1 As TWSLib.IComboLeg
Leg1 = addAllLegs.Add()

Leg1.conId = leg1_conId
Leg1.ratio = 1
Leg1.action = "SELL"
Leg1.exchange = "SMART"
Leg1.openClose = 0
Leg1.shortSaleslot = 0
Leg1.designatedLocation = ""

' 第二组合边
Dim Leg2 As TWSLib.IComboLeg
Leg2 = addAllLegs.Add()

Leg1.conId = leg2_conId
Leg1.ratio = 1
Leg1.action = "BUY"
Leg1.exchange = "SMART"
Leg1.openClose = 0
Leg1.shortSaleslot = 0
```

```
Leg1.designatedLocation = ""
```

3. 使用相应的合约和定单目标启动placeOrder()方法:

```
Dim contract As TWSLib.IContract
contract = Tws1.createContract

contract.symbol = "USD"
contract.secType = "BAG"
contract.exchange = "SMART"
contract.currency = "USD"
contract.comboLegs = addAllLegs

Dim order As TWSLib.IOrder
order = Tws1.createOrder

order.action = "BUY"
order.totalQuantity = 1
order.orderType = "MKT"

Tws1.placeOrderEx(OrderId, contract, order)
```



## C++

这一章介绍C++ API，包括以下内容：

- [使用TwsSocketClient.dll连接TWS](#)
- [使用C++ TestSocketClient样本程序](#)
- [Class EClientSocket功能](#)
- [Class EWrapper功能](#)
- [SocketClient属性](#)
- [下达组合定单](#)

## 使用TwsSocketClient.dll连接TWS

### 使用TwsSocketClient.dll连接TWS

1. 使用MS Visual Studio( 版本5.0或以上) 创建一个Window应用程序。
2. 在你的API安装文件夹中添加完整路径\SocketClient\include目录到你的项目的包括路径。 每个进入TwsSocketClient文库主文件的单个项目均需要完成这个步骤。
3. 在你的API安装文件夹中添加完整路径\SocketClient\lib\TwsSocketclient.lib目录到你的项目的文库路径。 每个进入TwsSocketClient文库的单个项目均需要完成这个步骤。
4. 在任何 Visual C++源代码中包括EWrapper.h和EClientSocket.h 进入其功能和数据结构。
5. EWrapper class的子类。
6. 取代下列功能：

Ewrapper功能	描述
tickPrice()	处理市场数据。
tickSize()	
tickOptionComputation()	
tickGeneric()	
tickString()	
tickEFP()	
orderStatus()	接收订单状态。
openOrder()	接收开放订单。
error()	接收出错信息。
connectionClosed()	在TWS终止连接时发通知。
updateAccountValue()	接收当前账户值。
updateAccountTime()	接收最后更新的账户信息。
updatePortfolio()	接收当前投资组合信息。



nextValidId()	接收连接时的下一个有效代码。
contractDetails()	接收合约信息。
contractDetailsEnd()	确认给出合约细节要求的终结。
bondContractDetails()	接收债券合约信息。
execDetails()	接收执行报告信息。
updateMktDepth()	接收市场深度信息。
updateMktDepthL2()	接收第二级市场深度信息。
updateNewsBulletin()	接收IB新闻公告。
managedAccounts()	接收金融顾问(FA)管理的账户列表。
receiveFA()	接收FA配置信息。
historicalData()	接收历史数据结果。
scannerParameters()	接收描述扫描仪订阅有效参数的XML文件。
scannerData()	接收市场扫描仪结果。
realTimeBar()	接收实时柱。
currentTime()	接收服务器的当前系统时间。
fundamentalData()	接收路透社全球基本面市场数据。

7. EClientSocket class实例。

8. 启动下列功能：

- Import com.ib.client.\* 到你的源代码文件。
- 实施EWrapper interface。这个class将从插口接收信息。
- 取代下列功能：

Ewrapper功能	描述
------------	----

tickPrice()	处理市场数据。
tickSize()	
tickOptionComputation()	
tickGeneric()	
tickString()	
tickEFP()	
orderStatus()	接收订单状态。
openOrder()	接收开放订单。
error()	接收出错信息。
connectionClosed()	在TWS终止连接时发通知。
updateAccountValue()	接收当前账户值。
updateAccountTime()	接收最后更新的账户信息。
updatePortfolio()	接收当前投资组合信息。
nextValidId()	接收连接时的下一个有效代码。
contractDetails()	接收合约信息。
contractDetailsEnd()	确认给出合约细节请求的终结。
bondContractDetails()	接收债券合约信息。
execDetails()	接收执行报告信息。
updateMktDepth()	接收市场深度信息。
updateMktDepthL2()	接收第二级市场深度信息。
updateNewsBulletin()	接收IB新闻公告。
managedAccounts()	接收金融顾问(FA)管理的账户列表。

receiveFA()	接收FA配置信息。
historicalData()	接收历史数据结果。
scannerParameters()	接收描述扫描仪订阅有效参数的XML文件。
scannerData()	接收市场扫描仪结果。
realTimeBar()	接收实时柱。
currentTime()	接收服务器的当前系统时间。
fundamentalData()	接收路透社全球基本面市场数据。

- EClientSocket class实例。这个目标将被用于发送信息给TWS。
- 启动下列功能：

EClientSocket功能	描述
eConnect()	连接TWS。
eDisconnect()	断开连接TWS。
reqMktData()	请求市场数据。
cancelMktData()	取消市场数据。
reqMktDepth()	请求市场深度。
cancelMktDepth()	取消市场深度。
reqContractDetails()	请求合约详细。
placeOrder()	下达订单。
cancelOrder()	取消订单。
reqAccountUpdates()	请求账户值、投资组合和账户更新时间信息。
reqExecutions()	请求日执行报告列表。
reqOpenOrders()	请求发出请求客户的当前客户订单列表和客户TWS相关的开放订单。相关信息仅当请求客户的客户代码是0时才会发生。
reqAllOpenOrders()	请求所有开放订单的列表。

reqAutoOpenOrders()	自动将一个新的TWS与客户相连。相关信息仅当请求客户的客户代码是0时才会发生。
reqNewsBulletin()	请求IB新闻公告。
cancelNewsBulletins()	取消IB新闻公告。
setServerLogLevel()	设定API请求级别和处理记录。
reqManagedAccts()	请求金融顾问(FA)管理的账户代号列表。
requestFA()	请求TWS的FA配置信息。
replaceFA()	修改API的FA配置信息。
reqScannerParameters()	请求描述扫描仪订阅有效参数的XML文件。
reqScannerSubscription()	请求市场扫描结果。
cancelScannerSubscription()	取消扫描仪订阅。
reqHistoricalData()	请求历史数据。
cancelHistoricalData()	取消历史数据。
reqRealTimeBars()	请求实时柱。
cancelRealTimeBars()	取消实时柱。
exerciseOptions()	行使期权。
reqCurrentTime()	期权当前服务器时间。
serverVersion()	返回API应用程序连接的TWS版本实例。
TwsConnectionTime()	返回API应用程序和TWS连接发生的时间。
reqFundamentalData()	请求路透社全球基本面数据。在你能够接收该数据之前，你必须通过账户管理设置路透社基本面的订阅。
cancelFundamentalData()	取消路透社全球基本面数据。

要运行程序，必须保证TwsSocketClient.dll是在与你的可执行的相同的目录中，或路径中。按默认状态，TwsSocketClient.dll文件将安装在你计算机的C:\Windows\system or C:\WINNT\system32目录中。

## 使用C++ TestSocketClient样本程序

你可以使用TWSSocketClient.dll组件通过微软的C++应用程序由TWS或IB网关进入IB交易系统。在你使用SocketClient组件连接TWS之前，你必须：

- 安装插口客户组件和样本程序
- 如果你在使用TWS，请配置TWS来支持API组件
- 在您的计算机中安装Microsoft Visual Studio(Visual C++ 5.0 或更高版本) 。

TestSocketClient程序是一个样本程序，指示你如何从基于Microsoft Windows C++应用程序使用插口与TWS连接。

### 运行预先创建的样本程序

我们在API软件中已经包括了一个完整的C++客户。运行这个样本程序，进入你的TWS API安装文件夹，打开\TestSocketClient\Release文件夹。运行文件client2.exe。

### 从Microsoft Visual Studio 2008运行TestSocketClient程序

#### 创建项目文件和汇编 TestSocketClient 应用程序( client2.exe) :

1. 下载和安装最新版本的API软件。
2. 创建所有项目相关文件将被保存的文件夹。例如， C:\SampleSocketClient。
3. 复制文件夹 TestSocketClient， Shared和SocketClient到你在第二步创建的文件夹中。( 例如， C:\SampleSocketClient。)
4. 在Visual Studio中，从文件菜单中选择**新> 现有编码中的项目**。
5. 选择**Visual C++**做为项目类型。
6. 在现有编码文件对话框的创建新项目：
  - a. 为项目文件地址选择你在第二步中创建的文件夹。( 例如， C:\SampleSocketClient。)
  - b. 为项目名称输入 **SampleSocketClient**。
  - c. 点击**完成**。
7. 在方案管理器( Solution Explorer) 中右键点击项目并选择**Properties**。在属性页面对话框中：
  - a. 在对话框的左侧，选择**Configuration Parameters( 配置参数) > C/C++ > General( 一般)**。在对话框右侧的其它包括目录区域，输入：
 

```
./Shared;./SocketClient/src
```
  - b. 在对话框的左侧，选择**Configuration Parameters ( 配置参数) > C/C++ > Code Generation( 编码生成)**。在对话框右侧的运行文库区域，选择**Multi-threaded (/MT)**。
8. 点击**OK**保存你的更改到项目属性中。
9. 创建项目。

## Class EClientSocket功能

下列表中定义了在与TWS连接时可以使用class EClientSocket功能。功能包括：

<p><b>连接和服务端</b></p> <p><a href="#">EClientSocket()</a>  <a href="#">eConnect()</a>  <a href="#">eDisconnect()</a>  <a href="#">isConnected()</a>  <a href="#">reqCurrentTime()</a>  <a href="#">serverVersion()</a>  <a href="#">TwsConnectionTime()</a>  <a href="#">setLogLevel()</a>  <a href="#">checkMessages()</a></p> <p><b>市场数据</b></p> <p><a href="#">reqMktData()</a>  <a href="#">cancelMktData()</a>  <a href="#">calculateImpliedVolatility()</a>  <a href="#">cancelCalculateImpliedVolatility()</a>  <a href="#">calculateOptionPrice()</a>  <a href="#">cancelCalculateOptionPrice()</a>  <a href="#">reqMarketDataType()</a></p> <p><b>定单</b></p> <p><a href="#">placeOrder()</a>  <a href="#">cancelOrder()</a>  <a href="#">reqOpenOrders()</a>  <a href="#">reqAllOpenOrders()</a>  <a href="#">reqAutoOpenOrders()</a>  <a href="#">reqIDs()</a>  <a href="#">exerciseOptions()</a></p> <p><b>账户</b></p> <p><a href="#">reqAccountUpdates()</a></p> <p><b>执行</b></p> <p><a href="#">reqExecutions()</a></p>	<p><b>合约详细</b></p> <p><a href="#">reqContractDetails()</a></p> <p><b>市场深度</b></p> <p><a href="#">reqMktDepth()</a>  <a href="#">cancelMktDepth()</a></p> <p><b>新闻公告</b></p> <p><a href="#">reqNewsBulletins()</a>  <a href="#">cancelNewsBulletins()</a></p> <p><b>金融顾问</b></p> <p><a href="#">reqManagedAccts()</a>  <a href="#">requestFA()</a>  <a href="#">replaceFA()</a>  <a href="#">reqAccountSummary()</a>  <a href="#">cancelAccountSummary()</a>  <a href="#">reqPositions()</a>  <a href="#">cancelPositions()</a></p> <p><b>历史数据</b></p> <p><a href="#">reqHistoricalData()</a>  <a href="#">cancelHistoricalData()</a></p> <p><b>市场扫描仪</b></p> <p><a href="#">reqScannerParameters()</a>  <a href="#">reqScannerSubscription()</a>  <a href="#">cancelScannerSubscription()</a></p> <p><b>实时柱</b></p> <p><a href="#">reqRealTimeBars()</a>  <a href="#">cancelRealTimeBars()</a></p> <p><b>基本面数据</b></p> <p><a href="#">reqFundamentalData()</a>  <a href="#">cancelFundamentalData()</a></p>
---	---

### **EClientSocket()**

这是一个构造函数。

#### **EClientSocket( EWrapper \*ptr)**

参数	描述
ptr	指向来自EWrapper基本class对象的指针。

### **eConnect()**

必须最先启动这个功能。没有成功连接的反馈，但如果再继续进行连接的话将返回“已经连接”的信息。

#### **bool eConnect( const char \*host, UINT port, int clientId=0)**

参数	描述
host	TWS运行的主机名或IP地址。保持空白与本地主机连接。
port	必须和TWS中Configure>API>Socket Port区域指定的端口匹配。
clientId	用于确认这个客户连接的号码。所有来自这个客户下达的/修改的订单将与这个客户代码相连。 注：每个客户必须使用独特的clientId相连。

### **eDisconnect()**

启动这个功能终止与TWS的连接。启动这个功能并不取消已经发送的订单。

#### **void eDisconnect()**

参数	描述
ptr	指向来自EWrapper基本class对象的指针。

### **isConnected()**

启动这个功能查看是否和TWS有连接。

#### **void isConnected()**

### **reqCurrentTime()**

反馈服务器的当前系统时间。

#### **void reqCurrentTime()**

### **serverVersion()**

反馈API应用程序连接的TWS实例的版本。

**serverVersion()**

**setLogLevel()**

默认的详细级别是ERROR( 出错) 。有关详细信息，请查看[API记录](#)。

**void setLogLevel(int logLevel)**

参数	描述
logLevel	指定当处理API请求时服务器( TWS) 使用的记录输入详细的级别。有效值包括： <ul style="list-style-type: none"> <li>• 1 = 系统</li> <li>• 2 = 出错</li> <li>• 3 = 警告</li> <li>• 4 = 信息</li> <li>• 5 = 详细</li> </ul>

**TwConnectionTime()**

反馈API应用程序和TWS连接的时间。

**TwConnectionTime()**

**checkMessages()**

应该频繁启动这个功能( 每一秒钟) 来检查从TWS接收的信息。

**void checkMessages()**

**reqMktData()**

启动这个功能请求市场数据。 市场数据将通过tickPrice和tickSize事件返回。

**void reqMktData(TickerID id, const Contract &contract, CString genericTicklist, bool snapshot)**

参数	描述
id	代码代号。必须为一个特殊值。当返回市场数据时，将由这个标签确认。也用于当取消市场数据时。
contract	这个结构包括被请求市场数据的合约的描述。
genericTicklist	一个有逗号分隔的通用跳动类型列表。跳动类型可以在 <a href="#">通用跳动类型</a> 页面中找到。
snapshot	选取该项将返回一个市场数据的截图和取消市场数据的订阅。如果你使用snapshot( 截图) ，不要输入任何genericTicklist值。



### cancelMktData()

启动这个功能后，特定代号的的市场数据将停止输送。

**void cancelMktData(TickerID id)**

参数	描述
id	启动 reqMktData()中指定的代号。

### calculateImpliedVolatility()

调用这个功能来计算提供的期权价格和底层价格的波动率。

**void calculateImpliedVolatility(TickerID reqId, Contract &contract, double optionPrice, double underPrice)**

参数	描述
reqId	代码代号。
contract	合约描述。
optionPrice	期权价格。
underPrice	底层价格。

### cancelCalculateImpliedVolatility()

调用这个功能来取消请求对提供的期权价格和底层价格的计算。

**calculateImpliedVolatility(TickerId reqId)**

参数	描述
reqId	代码代号。

### calculateOptionPrice()

调用这个功能来计算某个提供的波动率和底层价格的期权价格和希腊值。

**void calculateOptionPrice(TickerId reqId, const Contract &contract, double volatility, double underPrice)**

参数	描述
reqId	代码代号。
contract	合约描述。
volatility	波动率。
underPrice	底层价格。

### cancelCalculateOptionPrice()

调用这个功能来取消对提供的波动率和底层价格计算期权价格和希腊值的请求。

#### cancelCalculateOptionPrice(TickerId reqId)

参数	描述
reqId	代码代号。

### reqMarketDataType()

API可以从交易平台接收冻结的市场数据。冻结市场数据是我们系统中记录的最后的数据。在正常交易时段中，API接收实时市场数据。如果你使用这个功能，你是在指示TWS在收盘后自动转换到冻结市场数据。然后，在下一个交易日开始前，市场数据将自动转换回实时市场数据。

#### reqMarketDataType(int marketDataType)

参数	描述
marketDataType	1是实时市场数据流或2是冻结市场数据。

### placeOrder()

调用这个功能用来下达定单。定单状态将由orderStatus结果返回。

#### void placeOrder( OrderID id, const Contract &contract, const Order &order)

参数	描述
id	定单代号。你必须指定一个特殊的值。当定单状态返回时，将通过这个标签确认。这个标签也被用于取消定单时。
contract	这个结构包括交易合约的描述。
order	这个结构包括定单的详细信息。注：每个客户必须和一个独特的clientId相连。

### cancelOrder()

调用这个功能取消定单。

#### void cancelOrder(OrderID id)

参数	描述
id	之前在调用placeOrder()中指定的定单代号。

### reqOpenOrders()

调用这个功能请求由这个客户下达的开放订单。每个开放订单将通过EWrapper中的openOrder()和orderStatus()功能反馈回来。

#### void reqOpenOrders()

**Note:** 具有clientId为0的客户也将收到TWS拥有的开放订单。这些订单将与客户相连并将生成一个新的orderId。这个相连将保持在多个API和TWS作业中。

### reqAllOpenOrders()

调用这个功能请求包括所有客户以及TWS下达的开放订单。每个开放订单将通过EWrapper中的openOrder()和orderStatus()功能反馈回来。

#### void reqAllOpenOrders()

**Note:** 在返回的订单和请求客户之间没有相连。

### reqAutoOpenOrders()

调用这个功能请求新近创建的TWS订单和客户隐式相连。当一个新TWS订单被创建时，订单将与客户相连，并通过EWrapper的openOrder()和orderStatus()功能反馈。

#### reqAutoOpenOrders (bool bAutoBind)

**Note:** 这个请求只能由clientId为0的客户提出。

参数	描述
bAutoBind	如果设置为TRUE(真)，新近创建的TWS订单将与客户产生隐式相连。如果设置为FALSE(假)，将没有相连。

### reqIDs()

调用这个功能请求TWS在下达订单时可用的下一个代号。调用功能后，nextValidId()事件将被触发，并将返回下一个代号。这个代号将反映任何发生的自动绑定(生成新的代号和下一个有效代号其中的增量)。

#### void reqIds(int numIds)

参数	描述
numIds	你希望保留的代号数目。

### exerciseOptions()

void exerciseOptions(long id, const Contract &contract, int exerciseAction, int exerciseQuantity, const CString &account, int override)

参数	描述
id	代码代号。必须为唯一的值。

contract	这个结构包括请求的市场深度数据的合约描述。
exerciseAction	指定是否你希望期权失效或被行使。 值为 1 = 行使， 2 = 失效。
exerciseQuantity	你希望行使的数量。
override	指定是否你的设置将取代系统的自然行动。 例如， 如果你的行动是“行使”， 且期权不到价， 按自然行动， 期权将不会被行使。 如果你的取代设置为“是”， 则自然行动将被取代， 未到价期权将被行使。 值是: 0 = no, 1 = yes。

### reqAccountUpdates()

调用这个功能开始接收账户值、投资组合、和最后更新时间的信息。

**void reqAccountUpdates(bool subscribe, const CString& acctCode)**

参数	描述
subscribe	如果设为 TRUE( 真) ， 客户将开始接收账户和投资组合更新。 如果设为 FALSE( 假) ， 客户将停止接收这个信息。
acctCode	接收账户和投资组合更新的账户代码。

### reqExecutions()

当调用这个方法时， 满足过滤条件的过去24小数的执行报告将通过execDetails()方法下载给客户。 查看24小时以外的执行报告， 打开TWS中的交易记录， 当小数交易记录时， 请求再次从API下载执行报告。

**void reqExecutions(int reqID, const ExecutionFilter& filter)**

参数	描述
reqId	数据请求的代号。 如果同时处理几个请求时， 保证结果与请求匹配。
filter	这个对象包括用于描述确定返回的执行报告的过滤条件的属性。

### reqContractDetails()

调用这个功能下载特定底层证券的所有详细信息。 合约详细假通过EWrapper中的contractDetails()方法接收。

**void reqContractDetails (const Contract &contract)**

参数	描述
reqId	数据请求的代号。 如果同时处理几个请求时， 保证结果与请求匹配。

Contract	被查看合约的概括描述。
----------	-------------

### reqMktDepth()

调用这个功能请求特定合约的市场深度。市场深度将通过updateMktDepth()和updateMktDepthL2()事件反馈。

**void reqMktDepth (TickerID id, const Contract &contract, long numRows)**

参数	描述
id	代码代号。必须是唯一的值。当返回市场深度数据时，将由这个标签确定。这个也用于取消市场深度时。
contract	这个结构包括请求的市场深度数据的合约的描述。
numRows	- 指定显示的市场深度行的数目。

### cancelMktDepth()

调用这个方法之后，特定代码的市场深度数据将停止输送。

**void cancelMktDepth (TickerID id)**

参数	描述
id	调用reqMktDepth()中指定的代号。

### reqNewsBulletins()

调用这个功能开始接收新闻公告。每个公告将通过updatedNewsBulletin()事件返回。

**void reqNewsBulletins(bool allMsgs)**

参数	描述
allMsgs	如果设为TRUE(真)，将返回当天所有现有的公告和任何新的公告。如果设为FALSE(假)，假仅返回新的公告。

### cancelNewsBulletins()

调用这个功能停止接收新闻公告。

**void cancelNewsBulletins()**

### reqManagedAccts()

调用这个功能请求管理的账户列表。列表将通过EWrapper中的managedAccounts()功能反馈。

**Note:** 该请求只能在和金融顾问管理的账户连接时才可以发出。

**void reqManagedAccts()**

### requestFA()

调用这个功能请求TWS的金融顾问配置信息。数据通过"receiveFA" ActiveX事件以XML字符串形式反馈。

#### requestFA(long faDataType)

参数	描述
faDataType	指定请求的金融顾问配置数据的类型。有效值包括： <ul style="list-style-type: none"> <li>• 1 = 组</li> <li>• 2 = 概括</li> <li>• 3 = 账户别名</li> </ul>

### replaceFA()

调用该功能从API修改金融顾问配置信息。注意，也可以在TWS中手动修改。

#### replaceFA(long faDataType, string XML)

参数	描述
faDataType	指定通过API修改的金融顾问配置数据类型。有效值包括： <ul style="list-style-type: none"> <li>• 1 = 组</li> <li>• 2 = 概括</li> <li>• 3 = 账户别名</li> </ul>
XML	XML字符串包含新的FA配置信息。

### reqAccountSummary()

调用该功能请求并保持TWS账户窗口上概括标签中的数据为最新的。数据通过[accountSummary\(\)](#)反馈。

**Note:** 该请求只有在连接FA管理的账户时开可以进行。

#### void reqAccountSummary(int messageType, int version, int reqId, String groupName, String tags)

参数	类型	描述
messageType	整数	设置为62。
version	整数	设置为1。
reqId	整数	

参数	类型	描述
groupName	字符串	设置为 <i>所有</i> 返回所有账户的账户概括数据，或设置为一个已经在TWS全局配置中创建的特定的顾问账户组名称。

参数	类型	描述
tags	字符串	<p>一个由逗号隔开的账户标签列表。 可用的标签包括：</p> <ul style="list-style-type: none"> <li>• <i>AccountType</i></li> <li>• <i>NetLiquidation</i>,</li> <li>• <i>TotalCashValue</i> — 包括期货pnl的全部现金</li> <li>• <i>SettledCash</i> — 对现金账户，与<i>TotalCashValue</i>相同。</li> <li>• <i>AccruedCash</i> — 净累计利息</li> <li>• <i>BuyingPower</i> — 账户可以使用保证金购买美国股票的最大额。</li> <li>• <i>EquityWithLoanValue</i> — 现金 + 股票 + 债券 + 共同基金</li> <li>• <i>PreviousEquityWithLoanValue</i>,</li> <li>• <i>GrossPositionValue</i> — 所有股票和股票期权头寸的绝对值总额。</li> <li>• <i>RegTEquity</i>,</li> <li>• <i>RegTMargin</i>,</li> <li>• <i>SMA</i> — 特殊备忘录账户</li> <li>• <i>InitMarginReq</i>,</li> <li>• <i>MaintMarginReq</i>,</li> <li>• <i>AvailableFunds</i>,</li> <li>• <i>ExcessLiquidity</i>,</li> <li>• <i>Cushion</i> — 做为净清算值百分比的剩余流动性</li> <li>• <i>FullInitMarginReq</i>,</li> <li>• <i>FullMaintMarginReq</i>,</li> <li>• <i>FullAvailableFunds</i>,</li> <li>• <i>FullExcessLiquidity</i>,</li> <li>• <i>LookAheadNextChange</i> — 当预期值生效的时间</li> <li>• <i>LookAheadInitMarginReq</i>,</li> <li>• <i>LookAheadMaintMarginReq</i>,</li> <li>• <i>LookAheadAvailableFunds</i>,</li> <li>• <i>LookAheadExcessLiquidity</i>,</li> <li>• <i>HighestSeverity</i> — 衡量账户是如何接近被清算</li> <li>• <i>DayTradesRemaining</i> — 在日交易模式被显现之前，用户可以交易的开仓/平仓的数目。值为 "-1"表示用户可以进行无限制的日交易。</li> </ul>



参数	类型	描述
		<ul style="list-style-type: none"> <li><i>Leverage</i> — GrossPositionValue / NetLiquidation</li> </ul>

### cancelAccountSummary()

取消账户窗口概括标签数据的请求。

**void cancelAccountSummary(int messageId, int version, int reqId)**

参数	类型	描述
messageId	整数	设置为 63。
version	整数	设置为 1。
reqId	整数	被取消的数据请求代号。

### reqPositions()

请求所有账户的实时头寸数据。

**Note:** 该请求只有在连接FA管理的账户时才可以进行。

**void reqPositions(int messageId, int version)**

参数	类型	描述
messageId	整数	设置为 62。
version	整数	设置为 1。

### cancelPositions()

取消实时头寸更新。

**void cancelPositions(int messageId, int version)**

参数	类型	描述
messageId	整数	设为 64。
version	整数	设为 1。

### reqHistoricalData()

**void reqHistoricalData (TickerID id, const Contract &contract, String endDateTime, String durationStr, long barSizeSetting String whatToShow, int useRTH, int formatDate)**

参数	描述
id	请求的代号。必须为特殊的值。当返回市场数据时，将由这个标签确认。也被用于取消市场数据时。

contract	这个对象包含请求市场数据合约的描述。
endDateTime	定义过去6个月期间的任何请求截止日期和时间点。有效值包括过去6个月内任何日期/时间，格式为：yyymmdd HH:mm:ss ttt 这里"ttt"是可选的时区。
durationStr	设置请求区间上至一个星期，使用的时间单位为秒、天和星期。有效值包括任何整数及空格和S(秒)，D(天)或W(星期)。如果没有指定单位，将使用秒为单位。
barSizeSetting	指定将要返回的柱的尺寸(在IB/TWS限制范围内)。有效值包括： <ul style="list-style-type: none"> <li>• 1 秒</li> <li>• 5 秒</li> <li>• 15 秒</li> <li>• 30 秒</li> <li>• 1 分钟</li> <li>• 2 分钟</li> <li>• 3 分钟</li> <li>• 5 分钟</li> <li>• 15 分钟</li> <li>• 30 分钟</li> <li>• 1 小时</li> <li>• 1 天</li> </ul>
whatToShow	决定被提取数据的性质。有效值包括： <ul style="list-style-type: none"> <li>• TRADES</li> <li>• MIDPOINT</li> <li>• BID</li> <li>• ASK</li> <li>• BID_ASK</li> <li>• HISTORICAL_VOLATILITY</li> <li>• OPTION_IMPLIED_VOLATILITY</li> </ul>

useRTH	<p>决定是否返回请求时间范围内的所有可用数据，或仅属于正常交易时段内的数据。有效值包括：</p> <ul style="list-style-type: none"> <li>• 0 - 返回请求的包括正常交易时段以外的所有数据。</li> <li>• 1 - 仅返回属于正常交易时段内的数据，尽管请求的时间段指示部分或全部在RTH( 正常交易时段) 以外。</li> </ul>
formatDate	<p>决定应用于返回柱的日期格式。有效值包括：</p> <ul style="list-style-type: none"> <li>• 1 - 应用于返回柱的日期格式： yyymmdd {space} {space} hh:mm:dd</li> <li>• 2 - 指定自 1/1/1970 GMT的秒数的长的整数做为返回的日期。</li> </ul>

有关历史数据的请求限制信息，请参阅[历史数据限制](#)。

### cancelHistoricalData()

使用在网路连接切断时，或请求的结果被延迟和应用程序不再需要接收数据时。

**void cancelHistoricalData (int tickerId)**

参数	描述
tickerId	代码代号。必须是唯一的值。

### reqScannerParameters()

请求一个描述所有可能的扫描仪查询的XML字符串。

**void reqScannerParameters()**

### reqScannerSubscription()

**void reqScannerSubscription(int tickerId, const ScannerSubscription&subscription)**

参数	描述
tickerId	代码代号。必须是唯一的值。
ScannerSubscription	这个结构包括用于过滤结果的可能参数。

### cancelScannerSubscription()

**void cancelScannerSubscription(int tickerId)**

参数	描述
tickerId	代码代号。必须是唯一的值。

### reqRealTimeBars()

调用 reqRealTimeBars() 功能开始通过 realtimeBar() EWrapper 功能接收实时时间柱结果。

**void reqRealTimeBars(int tickerId, Contract contract, int barSize, String whatToShow, boolean useRTH)**

参数	描述
tickerId	请求的代号。必须是唯一的值。当收到数据时，将用这个代号确认。也被用于取消请求。
contract	该对象包括被请求的实时柱的合约的描述。
barSize	目前仅支持5秒柱，如果使用任何其它值，将呈现例外。
whatToShow	决定被提取数据的性质。有效值包括： <ul style="list-style-type: none"> <li>• TRADES</li> <li>• BID</li> <li>• ASK</li> <li>• MIDPOINT</li> </ul>
useRTH	仅正常交易时段。有效值包括： <ul style="list-style-type: none"> <li>• 0 = 返回请求的时间区间内的所有可用数据，包括请求的市场处于正常交易时段以外的时间。</li> <li>• 1 = 仅返回请求产品正常交易时段内的数据，尽管可能时间部分或全部处于正常交易时段以外。</li> </ul>

### cancelRealTimeBars()

调用 cancelRealTimeBars() 功能停止接收实时柱结果。

**void cancelRealTimeBars (int tickerId)**

参数	描述
tickerId	调用 reqRealTimeBars() 中指定的代号。

### reqFundamentalData()

调用这个功能接收路透社全球基本面数据。必须在接收该数据前在账户管理中完成路透社基本面的订阅。

**void reqFundamentalData(int reqId, const Contract &contract, String reportType)**

参数	描述
reqId	数据请求的代号。如果存在几个请求时，保证结果与请求相匹配。
contract	这个结构包括被请求的路透社基本面数据合约的描述。
reportType	确认以下的报告类型之一： <ul style="list-style-type: none"> <li>• 估计</li> <li>• 财务报告</li> <li>• 概括</li> </ul>

**cancelFundamentalData()**

调用该功能停止接收路透社全球基本面数据。

**void cancelFundamentalData(int reqId)**

参数	描述
reqId	数据请求的代号。

## Class EWrapper功能

下列表格定义在连接TWS时你可以使用的class EWrapper功能。 这些功能从TWS接收事件。列出的功能包括：

<p><b>连接和服务端</b></p> <p><a href="#"><u>winError()</u></a>  <a href="#"><u>error()</u></a>  <a href="#"><u>connectionClosed()</u></a>  <a href="#"><u>currentTime()</u></a></p> <p><b>市场数据</b></p> <p><a href="#"><u>tickPrice()</u></a>  <a href="#"><u>tickSize()</u></a>  <a href="#"><u>tickOptionComputation()</u></a>  <a href="#"><u>tickGeneric()</u></a>  <a href="#"><u>tickString()</u></a>  <a href="#"><u>tickEFP()</u></a>  <a href="#"><u>tickSnapshotEnd()</u></a>  <a href="#"><u>marketDataType()</u></a></p> <p><b>定单</b></p> <p><a href="#"><u>orderStatus()</u></a>  <a href="#"><u>openOrder()</u></a>  <a href="#"><u>nextValidId()</u></a></p> <p><b>账户和投资组合</b></p> <p><a href="#"><u>updateAccountValue()</u></a>  <a href="#"><u>updatePortfolio()</u></a>  <a href="#"><u>updateAccountTime()</u></a></p> <p><b>新闻公告</b></p> <p><a href="#"><u>updateNewsBulletin()</u></a></p>	<p><b>合约详细</b></p> <p><a href="#"><u>contractDetails()</u></a>  <a href="#"><u>contractDetailsEnd()</u></a>  <a href="#"><u>bondContractDetails()</u></a></p> <p><b>执行</b></p> <p><a href="#"><u>execDetails()</u></a>  <a href="#"><u>execDetailsEnd()</u></a>  <a href="#"><u>commissionReport()</u></a></p> <p><b>市场深度</b></p> <p><a href="#"><u>updateMktDepth()</u></a>  <a href="#"><u>updateMktDepthL2()</u></a></p> <p><b>金融顾问</b></p> <p><a href="#"><u>managedAccounts()</u></a>  <a href="#"><u>receiveFA()</u></a>  <a href="#"><u>accountSummary()</u></a>  <a href="#"><u>accountSummaryEnd()</u></a>  <a href="#"><u>position()</u></a>  <a href="#"><u>positionEnd()</u></a></p> <p><b>历史数据</b></p> <p><a href="#"><u>historicalData()</u></a></p> <p><b>市场扫描仪</b></p> <p><a href="#"><u>scannerParameters()</u></a>  <a href="#"><u>scannerData()</u></a>  <a href="#"><u>scannerDataEnd()</u></a></p> <p><b>实时柱</b></p> <p><a href="#"><u>realtimeBar()</u></a></p> <p><b>基本面数据</b></p> <p><a href="#"><u>fundamentalData()</u></a></p>
--	---

### winError()

这个事件在客户一侧出错时被调用。

**virtual void winError(const CString &str, int lastError)**

参数	描述
str	这是出错信息文字。
lastError	由GetLastError()返回的错误编码。

### error()

这个事件是当出现与TWS有通讯错误时，或当TWS希望给客户发送一个信息时被调用。

**virtual void error(const int id, const int errorCode, const CString errorString)**

参数	描述
id	这是orderId或出现错误的请求的tickerId。
errorCode	错误编码在 <a href="#">API信息编码</a> 题目中介绍。
errorString	这是错误信息的文字描述，也在错误编码标题中介绍。

### connectionClosed()

当TWS关闭了ActiveX控制的插口连接或当TWS关闭时调用这个功能。

**virtual void connectionClosed()**

### currentTime()

这个功能接收服务器一侧的当前系统时间。

**virtual void currentTime(long time)**

参数	描述
time	服务器一侧的当前系统时间。

### tickPrice()

当市场数据改变时调用这个功能。价格立即更新，没有延迟。

**virtual void tickPrice(TickerID id, TickType field, double price, int canAutoExecute)**

参数	描述
id	之前调用reqMktData()中指定的代码代号。

tickType	指定的价格类型。可能的值包括： <ul style="list-style-type: none"> <li>• 1 = 买价</li> <li>• 2 = 卖价</li> <li>• 4 = 最后价</li> <li>• 6 = 最高价</li> <li>• 7 = 最低价</li> <li>• 9 = 收盘价</li> </ul>
price	- 根据tickType的值，可以是买价、卖价、最后价、日内最高价、日内最低价或前一天收盘价。
canAutoExecute	指定是否价格跳动可被用于自动执行。可能的值包括： <ul style="list-style-type: none"> <li>• 0 = 不可用于自动执行</li> <li>• 1 = 可用于自动执行</li> </ul>

### tickSize()

当市场数据改变时调用这个功能。尺寸立即更新，没有延迟。

**virtual void tickSize(TickerID id, TickType field, int size)**

参数	描述
id	之前调用reqMktData()中指定的代码代号。
tickType	指定的尺寸类型。可能的值包括： <ul style="list-style-type: none"> <li>• 0 = 买价尺寸</li> <li>• 3 = 卖价尺寸</li> <li>• 5 = 最后价尺寸</li> <li>• 8 = 交易量</li> </ul>
size	根据tickType值，可以是买价尺寸、卖价尺寸、最后价尺寸或交易量。

### tickOptionComputation()

当期权所在市场或期权的底层代码变得时调用这个功能。TWS的期权模型波动率、价格和deltas，以及期权底层证券将收到的股息的当前值。

**virtual void tickOptionComputation(TickerID tickerId, TickType tickType, double impliedVol, double delta, double modelPrice, double pvDividend, double gamma, double vega, double theta, double undPrice)**

参数	描述
----	----



id	之前调用 reqMktData() 中指定的代码代号。
tickType	指定跳动类型。可能的值包括： <ul style="list-style-type: none"> <li>• 10 = 买价</li> <li>• 11 = 卖价</li> <li>• 12 = 最后价</li> </ul>
impliedVol	使用特定的 ticktype 值，由 TWS 期权模型计算的隐含波动率。
delta	期权 delta 值。
optPrice	期权价格。
pvDividend	期权底层产品预期股息的当前值。
gamma	期权 gamma 值。
vega	期权 vega 值。
theta	期权 theta 值。
undPrice	底层证券的价格。

### tickGeneric()

当市场数据变化时调用这个功能。值被立即更新，没有延迟。

**virtual void tickGeneric(TickerId tickerId, TickType tickType, double value)**

参数	描述
tickerId	之前调用 reqMktData() 中指定的代码代号。
tickType	指定价格类型。 将区域值传递到 TickType.getField(int tickType) 接收区域描述。例如，区域值 46 将连接到 shortable(卖空)，等等。
value	指定区域的值。

### tickString()

当市场数据变化时调用这个功能。值被立即更新，没有延迟。

**virtual void tickString(TickerId tickerId, TickType tickType, const CString& value)**

参数	描述
tickerId	之前调用 reqMktData() 中指定的代码代号。

field	指定价格类型。 将区域值传递到TickType.getField(int tickType)接收区域描述。 例如，区域值45将与lastTimestamp相连，等等。
value	指定区域的值。

### tickEFP()

当市场数据变化时调用这个功能。 值被立即更新，没有延迟。

**virtual void tickEFP(TickerId tickerId, TickType tickType, double basisPoints, const CString& formattedBasisPoints, double totalDividends, int holdDays, const CString& futureExpiry, double dividendImpact, double dividendsToExpiry)**

参数	描述
tickerId	之前调用reqMktData()中指定的代码代号。
field	指定价格类型。 将区域值传递到TickType.getField(int tickType)接收区域描述。 例如，区域值38将连接到bidEFP，等等。
basisPoints	代表可以直接和经纪商利率比较的融资利率的年基点。
formattedBasisPoints	以百分比形式表达的格式化字符串的年基点。
impliedFuture	隐含期货价格。
holdDays	EFP过期前的持有天数。
futureExpiry	单一股票期货的过期日。
dividendImpact	股息对年基点利率的影响。
dividendsToExpiry	单一股票期货过期前的预计股息。

### tickSnapshotEnd()

当截图市场数据订阅被完全处理后，且没有任何需要等待处理时调用该功能。这也适用于超时的情况。

**virtual void tickSnapshotEnd(int reqId)**

参数	描述
reqID	数据请求的代号。

### marketDataType()

TWS给API发送一个marketDataType(type)回调，类型为冻结或实时，宣布市场数据已经在冻结和实时之间转换了。这个通知仅当市场数据在实时和冻结之间转换时发生。marketDataType( )回调接受reqId参数，并按每个订阅发送，因为不同的合约通常可以按不同的时间来交易。

#### virtual void marketDataType(TickerId reqId, int marketDataType)

参数	描述
reqId	数据请求的代号。
marketDataType	1为实时数据流市场数据，或2为冻结市场数据。

### orderStatus()

每当订单状态变化时调用这个事件。如果客户有任何开放订单，再次与TWS接通时也启动这个事件。

#### virtual void orderStatus(OrderId id, const CString &status, int filled, int remaining, double avgFillPrice, int permId, int parentId, double lastFillPrice, int clientId, const CString& whyHeld)

**Note:** 有可能orderStatus()返回重复信息。关键是你要对信息进行过滤。

参数	描述
id	之前调用placeOrder()中指定的订单代号。

<p>status</p>	<p>定单状态。可能的值包括：</p> <ul style="list-style-type: none"> <li>• <b>PendingSubmit</b> - 表明你已经发送了定单，但还未接收到定单已被定单目的地接受的确认信息。<b>注：</b>该定单状态不是由TWS发出的，应该在定单发送时由API开发人设置。</li> <li>• <b>PendingCancel</b> - 表明你已经发出了取消定单的请求，但还未收到来自定单目的地的取消确认。此时，你的定单并没有被确认取消。在等待取消请求期间，你将仍有可能得到执行。<b>注：</b>该定单状态不是由TWS发出的，应该在定单取消时由API开发人设置。</li> <li>• <b>PreSubmitted</b> - 表明模拟定单类型已经被IB系统接受了，但该定单还未被选中。定单将停留在IB系统中直到选择条件被满足。之后定单将被发送到指定的定单目的地。</li> <li>• <b>Submitted</b> - 表示你的定单已经被定单目的地接收，正在处理中。</li> <li>• <b>Cancelled</b> - 表示IB系统已经确认了你的定单余额取消。这也可能在IB或目的地拒绝了你的定单时不期发生。</li> <li>• <b>Filled</b> - 表明定单已经被全部执行。</li> <li>• <b>Inactive</b> - 表明定单已经被系统(模拟定单)或交易所(本地定单)接收。但当前由于系统、交易所或其它原因，定单处于非活跃状态。</li> </ul>
---------------	---

filled	指明已被执行的股数。 关于部分执行的详细信息，请参阅 <a href="#">部分执行订单状态</a> 。
remaining	标明仍处于未执行的股数。
avgFillPrice	已经执行股票的平均价格。这个参数仅在 <b>filled</b> 参数值大于零时才有效。否则，价格参数将为零。
permId	用于确认订单的TWS代号。整个TWS运行过程保持不变。
parentId	母订单的订单代号，用于括号和自动跟踪止损订单。
lastFilledPrice	股票执行的最后价格。这个参数仅在 <b>filled</b> 参数值大于零时才有效。否则，价格参数将为零。
clientId	下达订单的客户代号(或TWS)。注意，TWS订单具有固定的clientId和与API订单区分的0的orderId。
whyHeld	这个区域被用来确认当TWS为卖空寻求股票时保持的订单。用于表示这个的值是'locate'(寻找)。

### openOrder()

调用该功能提供开放订单。

**virtual void openOrder(OrderId orderId, const Contract &contract, const Order &order, const OrderState &orderState)**

更多信息，请查看[扩展订单属性](#)。

参数	描述
orderId	由TWS分配的订单代号。用于取消或更新订单。
contract	描述合约的合约级别属性。
order	描述开放订单详细的订单级别。
orderState	orderState class包括用于交易前和交易后保证金以及佣金数据的属性。

### nextValidId()

成功与TWS连接后调用这个功能。

**virtual void nextValidId(OrderID orderId)**

参数	描述
----	----

orderId	连接TWS时收到的下一个可用的定单代号。所有后续定单代号均 根据这个代号增量为1。
---------	---

### updateAccountValue()

这个功能仅在EClientSocket对象的ReqAccountUpdates已经被调用时调用。

**virtual void updateAccountValue(const CString& key, const CString& value, const CString& currency, const CString& accountName)**

参数	描述
key	表明一种账户值的字符串。 以下为由TWS发出的keys。 <ul style="list-style-type: none"> <li>• CashBalance - 账户现金余额</li> <li>• Currency - 货币字符串</li> <li>• DayTradesRemaining - 剩余日交易次</li> <li>• EquityWithLoanValue - 含贷款价值权益</li> <li>• InitMarginReq - 当前初始保证金要求</li> <li>• LongOptionValue - 做多期权头寸</li> <li>• MaintMarginReq - 当前维持保证金</li> <li>• NetLiquidation - 净清算值</li> <li>• OptionMarketValue - 期权市值</li> <li>• ShortOptionValue - 做空期权值</li> <li>• StockMarketValue - 股票市值</li> <li>• UnalteredInitMarginReq - 隔夜初始保证金要求</li> <li>• UnalteredMaintMarginReq - 隔夜维持保证金要求</li> </ul>
value	和key有关的值。
currency	在值为货币类型时定义货币类型。
account	指明信息应用的账户。 对金融顾问子账户信息非常有用。

### updatePortfolio()

这个功能仅在EClientSocket对象的ReqAccountUpdates已经被调用时调用。

**virtual void updatePortfolio(const Contract& contract, int position, double marketPrice, double marketValue, double averageCost, double unrealizedPNL, double realizedPNL, const CString& accountName)**

参数	描述
----	----

contract	这个结构包括交易合约的描述。合约的交易所区域没有为投资组合更新设置。
position	该整数表明合约的头寸。如果头寸为0，表示头寸刚被清零。
marketPrice	产品的单位价格。
marketValue	产品的总市值。
averageCost	平均每股成本的计算是用你的头寸数量除以你的成本( 执行价格+佣金) 。
unrealizedPNL	当前你未平仓头寸的市值和平均成本或平均成本值的差。
realizedPNL	显示你的平仓头寸的利润，即你的建仓执行成本( 建仓执行价格+佣金) 和平仓执行成本( 平仓执行价格+佣金) 的差。
accountName	表明信息应用的账户。对金融顾问子账户信息非常有用。

### updateAccountTime()

这个功能仅在EClientSocket对象的ReqAccountUpdates已经被调用时调用。

#### virtual void updateAccountTime(const CString& timeStamp)

参数	描述
timeStamp	表明账户信息的最后更新时间。

### updateNewsBulletin()

这个事件由客户订阅的每个新公告启动( 例如 调用reqNewsBulletins()功能。)

#### virtual void updateNewsBulletin(int msgId, int msgType, const CString& message, const CString& origExchange

参数	描述
msgId	公告代号，每个新公告具有增加。
msgType	指明公告类型。有效值包括： <ul style="list-style-type: none"> <li>• 1 = 普通新闻公告</li> <li>• ·2 = 交易所不在用于交易</li> <li>• ·3 = 交易所可用于交易</li> </ul>
message	公告的信息文字。
origExchange	发出信息发的交易所。

### contractDetails()

这个功能仅在EClientSocket对象的reqContractDetails功能已经被调用时调用。

#### virtual void contractDetails(const ContractDetails &contractDetails)

参数	描述
reqId	数据请求代号。 保证在处理多个请求时， 结果与请求匹配。
contractDetails	该结果包括被查询合约的完整描述。

### contractDetailsEnd()

该功能在某个请求的所有合约信息全部收到后调用。用于帮助确认一个期权链的截止。

#### void contractDetailsEnd(int reqId)

参数	描述
reqID	数据请求的代号。

### bondContractDetails()

这个功能仅在EClientSocket对象的reqContractDetails功能已经为债券调用时调用。

#### virtual void bondContractDetails(const contractDetails&contractDetails)

参数	描述
reqId	数据请求的代号。
contractDetails	该结构包括交易合约的描述。 合约的交易所区域没有为投资组合更新设置。

### execDetails()

该事件被启动当reqExecutions()功能被调用时， 或当一个定单被执行时。

#### virtual void execDetails( OrderId orderId, const Contract& contract, const Execution& execution, long liquidation)

参数	描述
id	之前在调用placeOrder()中指定的定单代号。
contract	该结构包括已执行合约的完整描述。
execution	该结构包括附加的详细定单执行信息。



### execDetailsEnd()

该功能在根据reqExecutions()向客户发送了所有的执行后调用。

**virtual void execDetailsEnd(int reqId)**

参数	描述
reqID	数据请求的代号。

### commissionReport

**virtual void commissionReport(const CommissionReport &commissionReport)**

参数	描述
commissionReport	包括佣金详细的结构。

### UpdateMktDepth()

当市场深度变化时调用该功能。

**virtual void updateMktDepth(TickerId id, int position, int operation, int side, double price, int size)**

参数	描述
id	之前调用reqMktDepth()中指定的定单代号。
position	指定该市场深度输入的行代号。
operation	指明该定单如何应用于市场深度。有效值包括： <ul style="list-style-type: none"> <li>• 0 = 插入( 将这个新定单插入到标签为'头寸'的行) 。</li> <li>• 1 = 更新( 更新标签为'头寸'行内的现有定单) 。</li> <li>• 2 = 删除( 删除标签为'头寸'行内的现有定单)</li> </ul>
side	确定该定单属于的定单册的边。有效值包括： <ul style="list-style-type: none"> <li>• 0 = 卖价</li> <li>• 1 = 买价</li> </ul>
price	定单价格。
size	定单尺寸。

### updateMktDepthL2()

当二级市场深度改变时调用这个功能。

**virtual void updateMktDepthL2(TickerId id, int position, CString marketMaker, int operation, int side, double price, int size)**

参数	描述
id	之前调用 reqMktDepth() 中指定的订单代号。
position	指定该市场深度输入的行代号。
marketMaker	指定持有该订单的交易所。
operation	指明该订单如何应用于市场深度。有效值包括： <ul style="list-style-type: none"> <li>• 0 = 插入 ( 将这个新订单插入到标签为'头寸'的行) 。</li> <li>• 1 = 更新 ( 更新标签为'头寸'行内的现有订单) 。</li> <li>• 2 = 删除 ( 删除标签为'头寸'行内的现有订单)</li> </ul>
side	确定该订单属于的订单册的边。有效值包括： <ul style="list-style-type: none"> <li>• 0 = 卖价</li> <li>• 1 = 买价</li> </ul>
price	订单价格。
size	订单尺寸。

**managedAccounts()**

当成功连接金融顾问账户后调用这个功能。也在调用 reqManagedAccts() 功能时调用该功能。

**virtual void managedAccounts(const CString& accountsList)**

参数	描述
accountsList	有逗号分隔的FA管理的账户列表。

**receiveFA()**

这个事件接收之前请求的TWS的FA配置信息。

**virtual receiveFA(long faDataType, string XML)**

参数	描述
----	----

faDataType	指定从TWS接收的金融顾问配置数据类型。有效值包括： <ul style="list-style-type: none"> <li>• 1 = 组</li> <li>• 2 = 概括</li> <li>• 3 = 账户别名</li> </ul>
XML	XML字符串包括之前请求的FA配置信息。

**accountSummary()**

返回由[reqAccountSummary\(\)](#)功能生成的自TWS账户窗口概括标签的数据。

**virtual void accountSummary(int messageType, int version, int requestId, String account, String tag, String value, String currency)**

参数	类型	描述
messageType	整数	设为62。
version	整数	设为1。
requestId	整数	数据请求的代号。
account	字符串	账户代号。

参数	类型	描述
tag	字符串	<p>来自数据请求的标签。 可用的标签包括：</p> <ul style="list-style-type: none"> <li>• <i>AccountType</i></li> <li>• <i>TotalCashValue</i> — 包括期货 pnl 的总现金</li> <li>• <i>SettledCash</i> — 对现金账户，与 <i>TotalCashValue</i> 相同</li> <li>• <i>AccruedCash</i> — 净累计利息</li> <li>• <i>BuyingPower</i> — 账户可以购买的适用于保证金的美国股票的最大额</li> <li>• <i>EquityWithLoanValue</i> — 现金 + 股票 + 债券 + 共同基金</li> <li>• <i>PreviousEquityWithLoanValue</i></li> <li>• <i>GrossPositionValue</i> — 所有股票和股票期权头寸的绝对值总合</li> <li>• <i>RegTEquity</i></li> <li>• <i>RegTMargin</i></li> <li>• <i>SMA</i> — 特殊备忘录账户</li> <li>• <i>InitMarginReq</i></li> <li>• <i>MaintMarginReq</i></li> <li>• <i>AvailableFunds</i></li> <li>• <i>ExcessLiquidity</i></li> <li>• <i>Cushion</i> — 净清算值百分比的剩余流动性</li> <li>• <i>FullInitMarginReq</i></li> <li>• <i>FullMaintMarginReq</i></li> <li>• <i>FullAvailableFunds</i></li> <li>• <i>FullExcessLiquidity</i></li> <li>• <i>LookAheadNextChange</i> — 预期值生效的时间</li> <li>• <i>LookAheadInitMarginReq</i></li> <li>• <i>LookAheadMaintMarginReq</i></li> <li>• <i>LookAheadAvailableFunds</i></li> <li>• <i>LookAheadExcessLiquidity</i></li> <li>• <i>HighestSeverity</i> — 衡量账户接近被清算的程度</li> <li>• <i>DayTradesRemaining</i> — 在日交易模式被显露之前，用户可以进行开仓/平仓的数目。值为 "-1" 表示用户可以进行的日交易没有限制。</li> </ul>

参数	类型	描述
		<ul style="list-style-type: none"> <li><i>Leverage</i> — GrossPositionValue / NetLiquidation</li> </ul>
value	字符串	标签的值。
currency	字符串	标签的货币。

### accountSummaryEnd

当某个请求的所有账户概括数据被收到时调用该方法。

**virtual void accountSummaryEnd(int reqId)**

参数	类型	描述
reqId	整数	数据请求的代号。

### position()

这个事件返回相应于 [reqPositions\(\)](#) 方法的所有账户的实时头寸。

**virtual void position(int messageId, int version, String account, int conid, String underlying, String securityType, String expiry, String strike, String right, String multiplier, String exchange, String currency, String ibLocalSymbol, double position)**

参数	类型	描述
messageId	整数	设为 62。
version	整数	设为 1。
account	字符串	账户。
conid	整数	独特的合约确认代号。
underlying	字符串	底层资产的代码。
securityType	字符串	证券类型。
expiry	字符串	过期日。
strike	字符串	执行价。
right	字符串	看跌或看涨。
multiplier	字符串	乘数。
exchange	字符串	交易所。
currency	字符串	货币。
ibLocalSymbol	字符串	本地代码。
position	双	头寸。

### positionEnd()

当收到一个给定请求的所有头寸数据后合调用该功能，以及做为 position()数据截止的标记功能。

#### virtual void positionEnd(int reqId)

参数	类型	描述
reqId	整数	数据请求的代号。

### historicalData()

该功能接收请求的历史数据结果。

#### virtual void historicalData(TickerId reqId, const CString& date, double open, double high, double low, double close, int volume, double WAP, int hasGaps)

参数	描述
reqId	柱相对应的请求代码代号。
date	柱的日期时间开始标记。其格式由reqHistoricalData() formatDate 参数决定。
open	柱开始价格。
high	柱覆盖时间的最高价格。
low	柱覆盖时间的最低价格。
close	柱截止价格。
volume	柱覆盖时间的交易量。
WAP	柱覆盖时间的加权平均价格。
count	当返回TRADES(交易)历史数据时，代表柱覆盖时间区域内发生的交易次数。
hasGaps	报告是否数据存在间断。

### scannerParameters()

该功能接收描述扫描仪订阅可以拥有的有效参数的XML文件。

#### virtual void scannerParameters(String xml)

参数	描述
xml	一个描述请求的有效参数的XML文件。

### scannerData()

该功能接收请求的市场扫描仪数据结果。

**virtual void scannerData(int reqId, int rank, const ContractDetails &contractDetails, String distance, String benchmark, String projection)**

参数	描述
reqId	该行代表的请求的代码代号。
rank	相对该柱结果的排序。
contractDetails	该对象包括合约的完整描述。
distance	根据查询而变化。
benchmark	根据查询而变化。
projection	根据查询而变化。
legsStr	描述扫描反馈的EFP的组合边。

**scannerDataEnd()**

当收到截图时调用该功能并标记一个扫描的结束。

**virtual void scannerDataEnd(int reqId)**

参数	描述
reqId	由这个参数关闭市场扫描仪请求代号。

**realtimeBar()**

该功能接收实时柱数据结果。

**virtual void realtimeBar(TickerId reqId, long time, double open, double high, double low, double close, long volume, double wap, int count)**

参数	描述
reqId	该柱代表的请求代码代号。
time	柱的日期-时间开始标记。格式由reqHistoricalData() formatDate 参数确定。
open	柱开始价格。
high	柱覆盖时间的最高价格。
low	柱覆盖时间的最低价格。
close	柱截止价格。
volume	柱覆盖时间的交易量。
wap	柱覆盖时间的加权平均价格。

count	当返回TRADES( 交易) 历史数据时，代表柱覆盖时间区域内发生的交易次数。
-------	---

**fundamentalData()**

调用该功能接收路透社全球基本面市场数据。接收该数据前必须通过账户管理订阅路透社基本面。

**virtual void fundamentalData(int reqId, string data)**

参数	描述
reqId	数据请求的代号。
data	下列三种XML报告之一： <ul style="list-style-type: none"> <li>• 估计 (estimates)</li> <li>• 财务报告 (finstat)</li> <li>• 概括 (snapshot)</li> </ul>



## SocketClient资源文件

下面列出的是用于执行、合约和定单classes，以及与其直接相关的classes的定义资源文件。

- [Execution](#)
- [ExecutionFilter](#)
- [Contract](#)
- [ContractDetails](#)
- [ComboLeg](#)
- [Order](#)
- [OrderState](#)
- [ScannerSubscription](#)
- [UnderComp](#)
- [CommissionReport](#)

### Execution

属性	描述
IBString execId	独特的定单执行代号。
IBString time	定单执行时间。
IBString acctNumber	客户账户号码。
IBString exchange	执行定单的交易所。
IBString side	指定执行的交易是卖出还是买进。有效值包括： <ul style="list-style-type: none"> <li>• BOT( 买进)</li> <li>• SLD( 卖出)</li> </ul>
int shares	执行的股数。
double price	不包括佣金的定单执行价格。
int permId	用于确认定单的TWS代号，整个TWS运行过程保持不变。
long clientId	下达定单客户的代号。 <b>Note:</b> TWS定单的客户代号固定为0。
long orderId	定单代号。 <b>Note:</b> TWS定单的定单代号固定为0。
int liquidation	确定需要时最后清算的头寸。

int cumQty	累积数量。用于正常交易、组合交易和组合边。
double avgPrice	平均价格。用于正常交易、组合交易和组合边。包括佣金。
IBString evRule	包括经济价值规则名称和相应的可选参数。两个值应该用冒号分开。例如， aussieBond:YearsToExpiration=3。 当可选参数不存在时， 第一个值之后将使用冒号。
double evMultiplier	告诉你如果价格变化为1时， 大约一个合约的市值改变为多少。 不可以通过用近似的乘数乘以价格得出市值。

### ExecutionFilter

属性	描述
IBString acctCode	根据账户代码过滤reqExecutions()功能的结果。 注： 仅与FA管理的账户相关。
IBString exchange	根据定单交易所过滤reqExecutions()功能的结果。
IBString secType	根据定单证券类型过滤reqExecutions()功能结果。 注： 参考有效证券类型列表的合约结构。
IBString side	根据定单活动过滤reqExecutions()功能结果。 注： 参考有效定单活动列表的定单结构。
IBString symbol	根据定单代码过滤reqExecutions()功能结果。
IBString time	根据在指定时间后接收到的执行报告过滤reqExecutions()功能结果。 timeFilter的格式是“yyyymmdd-hh:mm:ss”
long clientId	根据 clientId过滤 reqExecutions()功能结果。

### Contract

属性	描述
vector<ComboLeg> comboLegs	用于存储合约边定义的动态记忆结构。
IBString comboLegsDescrip	描述组合边。
int conId	独特合约标识符。

IBString currency	指定货币。不明确的可能需要指定这个区域，例如，当SMART( 智能) 为交易所和IBM被请求时( IBM的交易可以是英镑或美元)。考虑到这种不明确的存在，最好是指定货币。
IBString exchange	定单目的地，比如Smart( 智能) 。
IBString expiry	过期日。使用格式YYYYMM。
bool includeExpired	如果设置为真，以对过期的合约进行合约详细请求和历史数据请求。  <b>Note:</b> 对过期合约的历史数据请求限制在合约寿命的最后一年，并且开始时仅支持过期的期货合约。
IBString localSymbol	这是底层资产的本地交易所代码。
IBString multiplier	允许你指定一个期货或期权乘数。这只是当多个可能存在的情况下才需要。
IBString primaryExchange	澄清智能传递合约的任何模糊不清，包括目的地的主交易所，以及智能目的地。
IBString right	指定看跌或看涨。有效值包括：P、PUT、C、CALL。
IBString secId	secIdType的独特标识符。
IBString secIdType	当查询合约信息或下达定单时的证券标识符。支持的标识符包括： <ul style="list-style-type: none"> <li>• ISIN 例如：Apple: US0378331005)</li> <li>• CUSIP ( 例如：Apple: 037833100)</li> <li>• SEDOL ( 包括6-AN + 校验位。例如：BAE: 0263494)</li> <li>• RIC ( 确认交易所的包括独立于交易所的RIC根和前缀)。例如：AAPL.O 是NASDAQ上的Apple股票。)</li> </ul>

IBString secType	这是证券类型。有效值包括： <ul style="list-style-type: none"> <li>• STK</li> <li>• OPT</li> <li>• FUT</li> <li>• IND</li> <li>• FOP</li> <li>• CASH</li> <li>• BAG</li> </ul>
double strike	执行价。
IBString symbol	底层资产的代码。

### ContractDetails

属性	描述
bool callable	用于债券。值为真或假。如果是真，在一定条件下，债券发行人可以赎回债券。
IBString category	底层证券的行业类别。例如，InvestmentSvc。
IBString contractMonth	合约月。期货合约的底层证券的合约月。
bool convertible	用于债券。值为真或假。如果为真，在一定条件下，债券可以被转换为股票。
double coupon	用于债券。用于计算你将在一年期间收到的利息支付的利率。
IBString industry	底层/产品的行业分类。例如，Financial( 金融) 。
IBString liquidHours	产品的流通交易时间。例如，20090507:0930-1600;20090508:CLOSED。
IBString longName	资产的描述名称。
IBString marketName	该合约的市场名称。
double minTick	最低价格跳动。
Bool nextOptionPartial	对债券，如果债券嵌入了期权则相关，例如，下一个是全期权或部分期权？
IBString orderTypes	该合约的有效定单类型列表。
long priceMagnifier	允许执行和行使价的报告始终和市场数据、历史数据和定单价格一致，例如LIFFE市场Z的报告是指数点而非英镑。

bool putable	用于债券。值为真或假。如果为真，一定条件下，债券可以回售给发行人。
TagValueListSPtr secIdList()	允许客户查看的合约标识符列表( CUSIP, ISIN, 等。)
IBString subcategory	底层资产的子行业类别。例如， Brokerage( 经纪商) 。
Contract summary	合约结构。
IBString tradingClass	该合约的交易级别名称。
IBString tradingHours	产品的交易时间。例如， 20090507:0700-1830,1830-2330;20090508:CLOSED。
IBString timeZoneId	产品交易时间的时区代号。例如， EST( 东部标准时间) 。
IBString underConId	底层合约代号。
IBString evRule	包括经济价值规则名称和相关的可选参数。两个值应该用冒号隔开。例如， aussieBond:YearsToExpiration=3。当可选参数不存在时，第一个值之后将跟随一个冒号。
double evMultiplier	指出在价格改变为1时，合约市值的大概变化是多少。不可以使用乘数乘以价格来得出市值。
<b>债券值</b>	
IBString bondType	用于债券。债券类型，比如“CORP。”
IBString couponType	用于债券。债券利率类型，比如“FIXED。”( 固定的)
IBString cusip	用于债券。九位字符债券CUSIP或12位的SEDOL。
IBString descAppend	用于债券。包括债券进一步描述信息的描述字符串。
IBString issueDate	用于债券。债券发行的日期。
IBString maturity	用于债券。债券发行人必须偿还债券面值的日期。
IBString nextOptionDate	对债券，如果债券包含嵌入期权，则相关。
IBString nextOptionType	对债券，如果债券包含嵌入期权，则相关。
IBString notes	对债券，如果在IB数据库中包括了债券。
IBString ratings	用于债券。确定债券发行人的信用评级。较高的信用评级表明较低的投资风险。债券评级分别来自Moody's和S&P。
IBString validExchanges	交易合约的交易所列表。

### ComboLeg

属性	描述
IBString action	你创建边的一侧( 买或卖) 。
long conId	代表证券的特殊合约标识符。
IBString designatedLocation	如果 shortSaleSlot = 2, 必须指定 designatedLocation。 否则留为空白或订单将被拒绝。
IBString exchange	完整组合订单将被传递的交易所。
long openClose	指定是否订单是开放或关闭订单。有效值包括： <ul style="list-style-type: none"> <li>• 相同 - (0) 和母证券相同。这是零售客户的唯一选择。</li> <li>• 开放 - (1)仅适用于机构客户。</li> <li>• 关闭 - (2)仅适用于机构客户。</li> <li>• 未知</li> </ul>
long ratio	为你创建的边选择相应的若干合约。帮助确定特定组合订单的比率，请参考用户指南中的投资组合分析部分。
int shortSaleSlot	仅适用于机构客户。 <ul style="list-style-type: none"> <li>• 0 - 不适用( 例如 零售客户或非卖空边)</li> <li>• 1 - 清算经纪商</li> <li>• 2 - 第三方。如果使用这个值，你必须输入一个指定的地点。</li> </ul>

### Order

属性	描述
<b>订单标识符</b>	
long clientId	下达订单的客户代号。
long orderId	订单代号。
long permId	用于确认定单的TWS代号，整个TWS运行过程保持不变。
<b>主订单区域</b>	
IBString action	确认边。有效值是：BUY, SELL, SSHORT

double auxPrice	是止损限价单的止损价格，和相对定单的抵消额。所有其它情形下，设定为零。
double lmtPrice	限价价格，用于限价单、止损限价单和相对定单。所有其它情形下为零。对没有限价价格的相对定单，也为零。
IBString orderType	确认定单类型。  关于支持的定单类型信息，请参阅 <a href="#">支持的定单类型</a> 。
long totalQuantity	定单数量。
<b>扩展定单区域</b>	
bool allOrNone	0 = no, 1 = yes
bool blockOrder	如果设为真，指定定单是ISE Block定单。
int displaySize	公开披露定单尺寸，用于下达Iceberg定单时。
IBString goodAfterTime	交易“过时后有效，”的格式 “YYYYMMDD hh:mm:ss ( 可选的时区) “(optional time zone)” 不适用时使用空字符串。
IBString goodTillDate	使用这个字符串你必须输入GTD做为有效时间。交易“到日期前有效”格式“YYYYMMDD hh:mm:ss( 可选时区) ” 不适用时使用空字符串。
boolean hidden	如果设为真，当查看市场深度时，定单将为不可视。这个选择仅用于发送到ISLAND交易所的定单。
int minQty	指定最低数量定单类型。
IBString ocaGroup	指定OCA ( 一取消全) 组。
int ocaType	当一个定单或部分定单执行后，指示如何处理OCA组中剩余的定单。有效值包括： <ul style="list-style-type: none"> <li>• 1 = 取消批量中的所有剩余定单</li> <li>• 2 = 随批量按比例减少剩余定单尺寸</li> <li>• 3 = 不随批量按比例减少定单尺寸</li> </ul> <p>如果你使用值“随批量”则使你的定单具有超额执行的保护。意味着每次只传递组中的一个定单以避免发生超执行的可能性。</p>

IBString orderRef	定单参考。仅计划用于机构客户，尽管当多个API客户运行时，所有的客户可以使用它来确认发送定单的API客户。
boolean outsideRth()	如果设为真，允许定单在正常交易时段外触发或执行。
bool overridePercentageConstraints	TWS预设页面中定义的预防约束，帮助确保你的价格和尺寸定单值是合理的。从API发出的定单也受这些安全约束的验证，并在违反约束时拒绝定单。如想取消验证，设置该参数的值为真。 有效值包括： <ul style="list-style-type: none"> <li>• 0 = 假</li> <li>• 1 = 真</li> </ul>
long parentId	母定单的定单代号，用于括号和自动跟踪止损定单。
double percentOffset	相对定单的百分比抵消额。
IBString rule80A	包括的值： <ul style="list-style-type: none"> <li>• Individual = 'I'</li> <li>• Agency = 'A',</li> <li>• AgentOtherMember = 'W'</li> <li>• IndividualPTIA = 'J'</li> <li>• AgencyPTIA = 'U'</li> <li>• AgentOtherMemberPTIA = 'M'</li> <li>• IndividualPT = 'K'</li> <li>• AgencyPT = 'Y'</li> <li>• AgentOtherMemberPT = 'N'</li> </ul>
IBString tif	有效时间。有效值包括：DAY, GTC, IOC, GTD。
bool sweepToFill	如果设为真，指定定单为一扫光定单。



<p>double trailingPercent</p>	<p>指定跟踪止损订单的跟踪额为一个百分比。使用trailingPercent区域时要遵循下列原则：</p> <ul style="list-style-type: none"> <li>• 这个区域与现有的跟踪额是相互排斥的。也就是，API客户可以发送其中一个或另一个，但非两者。</li> <li>• 这个区域是在止损价格( 保护价格) 之后作用的： deltaNeutralAuxPrice stopPrice trailingPercent 分段订单属性</li> <li>• 该区域也将通过openOrder信息发API，如果API客户版本是&gt;= 56。并且是在stopPrice区域之后发送的： stopPrice trailingPct basisPoint</li> </ul>
<p>double trailStopPrice</p>	<p>仅用于TRAILLIMIT( 跟踪限价单) 订单</p>
<p>bool transmit</p>	<p>指定是否订单将由TWS发送。如果设为假，订单将在TWS中创建，但不会被发送。</p>
<p>int triggerfunction</p>	<p>指定如何触发模拟止损、止损限价和跟踪止损订单。有效值为：</p> <ul style="list-style-type: none"> <li>• 0 - 默认值。“双买价/卖价”功能将被用于OTC( 场外) 股票和美国期权的订单。所有其它订单将使用“最后价”功能。</li> <li>• 1 - 当止损订单的触发是基于两个连续买价或卖价时使用“双买价/卖价”功能。</li> <li>• 2 - “最后价”功能，止损订单的触发时基于最后价格。</li> <li>• 3 双最后价功能</li> <li>• 4 买价/卖价功能。</li> <li>• 7 最后价或买价/卖价功能。</li> <li>• 8 中点功能。</li> </ul>
<p><b>金融顾问区域</b></p>	
<p>IBString faGroup</p>	<p>金融顾问组的交易将被分配到 -- 如果不适用，使用一个空字符串。</p>
<p>IBString faMethod</p>	<p>金融顾问分配功能的交易将被分配 -- 如果不适用，使用一个空字符串。</p>

IBString faPercentage	金融顾问百分比关系到交易的分配 -- 如果不适用，使用一个空字符串。
IBString faProfile	金融顾问分配方式的交易将被分配到 -- 如果不适用，使用一个空字符串。
<b>仅机构(非结算的)</b>	
IBString designatedLocation	仅当shortSaleSlot = 2时使用。
IBString openClose	仅用于机构客户。有效值为O, C。
int origin	订单源。仅用于机构客户。有效值是0 = 客户, 1 = 公司
int shortSaleSlot	有效值是1 或 2。
<b>仅智能传递</b>	
double discretionaryAmt	全权委托单允许的偏离限价价格的数额。
bool eTradeOnly	使用电子报价交易。 0 = no, 1 = yes
bool firmQuoteOnly	使用公司报价交易。 0 = no, 1 = yes
double nbboPriceCap	偏离NBBO的最大智能定单距离。
bool optOutSmartRouting	选择不使用默认的SmartRouting(智能传递) 将定单直接传递到ASX。这个属性的默认是假的，除非明确设置为真。当设为假时，直接传递到ASX的定单将不使用SmartRouting。当设为真时，直接传递到ASX的定单将使用SmartRouting。
<b>仅BOX或VOL定单</b>	
int auctionStrategy	包括的值： <ul style="list-style-type: none"><li>• 匹配 = 1</li><li>• 改进 = 2</li><li>• 透明 = 3</li></ul> 仅BOX定单。
<b>仅BOX交易所定单</b>	
double delta	股票delta。仅BOX定单。
double startingPrice	竞价开始价格。仅BOX定单。

double stockRefPrice	股票参考价格。参考价格被用于VOL订单来计算发给交易所的限价价格(不论是否选择了连续更新)和价格范围监控。
仅挂钩股票和波动率订单。	
double stockRangeLower	可接受的底层股票价格范围的低端值。用于BOX和VOL订单的价格改善期权订单的动态管理。
double stockRangeUpper	可接受的底层股票价格范围的高端值。用于BOX和VOL订单的价格改善期权订单的动态管理。
仅波动率订单	
bool continuousUpdate	仅VOL订单。指定是否在底层价格变化时TWS将自动更新订单的限价价格。
int deltaNeutralAuxPrice	仅VOL订单。使用这个区域输入值, 如果deltaNeutralOrderType区域中的值是订单类型, 需要辅助价格, 比如相对订单。
IBString deltaNeutralOrderType	仅VOL订单。输入订单类型指示TWS对全部或部分执行的VOL订单发送一个delta中性订单。对不需要发送对冲delta订单的情况, 指定为NONE(无)。
int referencePriceType	仅VOL订单。指定是否你希望TWS为期权计算限价价格, 和股票范围价格监控。 有效值包括: <ul style="list-style-type: none"> <li>• 1 = 平均NBBO</li> <li>• 2 = 根据活动和权力, 为NBB或NBO。</li> </ul>
double volatility	波动率的期权价格, 由TWS的期权分析计算。这个值的表示是一个百分比并用于计算发给交易所的限价价格。
int volatilityType	包括的值: <ul style="list-style-type: none"> <li>• 1 = 日波动率</li> <li>• 2 = 年波动率</li> </ul>
IBString deltaNeutralOpenClose	指定是否订单为开放或关闭订单, 并在对冲涉及到CFD和订单交与第三方结算时使用。
bool deltaNeutralShortSale	当对冲涉及到股票时使用并表示是否为卖空出售。

int deltaNeutralShortSaleSlot	值为1( 结算经纪商持有股票) 或2( 由第三方交付) 。如果你使用2, 则你必须指定一个deltaNeutralDesignatedLocation。
IBString deltaNeutralDesignatedLocation	仅当deltaNeutralShortSaleSlot = 2时使用。
<b>仅组合定单</b>	
double basisPoints	仅EFP定单
int basisPointsType	仅EFP定单
<b>仅分段定单</b>	
bool scaleAutoReset()	用于扩展分段定单。
int scaleInitFillQty()	用于扩展分段定单。
int scaleInitLevelSize	对分段定单: 定义第一个尺寸, 或初始尺寸, 定单组。
int scaleInitPosition()	用于扩展分段定单。
int scalePriceAdjustInterval()	用于扩展分段定单。
double scalePriceAdjustValue()	用于扩展分段定单。
double scalePriceIncrement	对分段定单: 定义分段组之间的价格增量。这个区域是要求的。
double scaleProfitOffset()	用于扩展分段定单。
bool scaleRandomPercent()	用于扩展分段定单。
int scaleSubsLevelSize	对分段定单: 定义后续分段定单组的定单尺寸。与scaleInitLevelSize()配合一起使用。
<b>仅对冲定单</b>	
IBString hedgeParam	Beta = x 是Beta对冲定单, 比率 = y是成双对冲定单

IBString hedgeType	用于对冲定单。可能的值为： <ul style="list-style-type: none"> <li>• D = Delta</li> <li>• B = Beta</li> <li>• F = FX</li> <li>• P = Pair</li> </ul>
<b>结算信息</b>	
IBString account	账户。仅用于机构客户。
IBString clearingAccount	用于IBExecution客户：指定定单的真实受益人。该值对报告到交易所的FUT/FOP的定单是要求的。
IBString clearingIntent	对IBExecution客户：有效值是：IB, Away, 和PTA (后交易分配)。
IBString settlingFirm	仅机构。
<b>仅算法定单</b>	
IBString algoStrategy	有关API算法定单的信息，请参考 <a href="#">IBAlgo 参数</a> 。
Vector<TagValue> algoParams	支持IBAlgo参数。
<b>假设</b>	
bool whatIf	用于请求交易前佣金和保证金信息。 如果设为真，保证金和佣金数据是通过openOrder()回调的OrderState()对象接收的。
<b>定单组合边</b>	
Vector<TagValue> OrderComboLegs	拥有一个组合定单中所有边的属性。
<b>不拥有</b>	
bool m_notHeld	仅用于IBDARK定单。传递到IBDARK的定单的标签为“仅交易后”并保持在IB的定单册中，定单册中来自其它IB客户的SmartRouted(智能传递)的定单可以对它们进行交易。

### OrderState

属性	描述
----	----

double commission	显示定单的佣金额。
IBString commissionCurrency	显示佣金值的货币。
IBString equityWithLoan	显示定单可能对你的含借贷价值股权的影响。
IBString initMargin	显示定单可能对你的初始保证金的影响。
IBString maintMargin	显示定单可能对你的维持保证金的影响。
double maxCommission	与 minCommission 区域配合使用，这将定义实际定单佣金将落入的可能范围的最高端。
double minCommission	与 maxCommission 区域配合使用，这将定义实际定单佣金将落入的可能范围的最低端。
IBString status	显示定单状态。
IBString warningText	必要时，显示一个警告信息。

### ScannerSubscription

属性	描述
double abovePrice	过滤掉价格低于这个值的合约。可以是空白。
int aboveVolume	过滤掉交易量低于这个值的合约。可以是空白。
int averageOptionVolumeAbove	可以留为空白。
double belowPrice	过滤掉价格高于这个值的合约。可以是空白。
double couponRateAbove	过滤掉利率低于这个值的合约。可以是空白。
double couponRateBelow	过滤掉利率高于这个值的合约。可以是空白。
IBString excludeConvertible	过滤掉可转换债券。可以留为空白。
IBString instrument	定义扫描的产品类型。
IBString locationCode	地点。
double marketCapAbove	过滤掉市值低于这个值的合约。可以是空白。
double marketCapBelow	过滤掉市值高于这个值的合约。可以是空白。

IBString maturityDateAbove	过滤掉到期日期早于这个值的合约。可以是空白。
IBString maturityDateBelow	过滤掉到期日期迟于这个值的合约。可以是空白。
IBString moodyRatingAbove	过滤掉穆迪评级低于这个值的合约。可以是空白。
IBString moodyRatingBelow	过滤掉穆迪评级高于这个值的合约。可以是空白。
int numberOfRows	定义查询返回的数据行数。
IBString scanCode	可以为空白。
IBString scannerSettingPairs	可以留为空白。例如，用于“获取最高期权隐含波动率”扫描的一对“年度，真”将返回年度波动率。
IBString spRatingAbove	过滤掉标普评级低于这个值的合约。可以是空白。
IBString spRatingBelow	过滤掉标普评级高于这个值的合约。可以是空白。
IBString stockTypeFilter	有效值是： <ul style="list-style-type: none"> <li>• CORP = 公司</li> <li>• ADR = 美国存托凭证</li> <li>• ETF = 交易所交易基金</li> <li>• REIT = 房地产投资信托</li> <li>• CEF = 定额基金</li> </ul>

### UnderComp

属性	描述
int conId	确定证券的独特合约代号。用于Delta-中性组合合约。
double delta	底层股票或期货 delta。用于Delta-中性组合合约。
double price	底层证券价格。用于Delta-中性组合合约。

### CommissionReport

属性	描述
double commission	佣金额。
IBString currency()	货币。

IBString execId()	独特定单执行代号。
double realizedPNL()	实现的利润和损失数额。
double yield()	收益。
int yieldRedemptionDate()	接受YYYYMMDD格式。



## 下达一个组合定单

组合定单是一种特殊的定单类型，由多个不同的边组成，但以单个交易执行。提交组合定单，比如使用BAG证券类型(在Contract对象中定义)的日历差价，转换和跨越定单。使用API成功的实施API组合定单的关键是了解如何使用交易平台下达相同定单。如果你熟悉在TWS中下达组合定单，就会容易使用API下达相同定单，因为API仅模仿TWS的运用。

### 举例

举例中的客户对CLK9期货合约下达一个买单和CLM9期货合约的卖单。在这个过程中，客户必须调用reqContractDetails()获取CLK9和CLM9的conId。

#### 边 1: 买 1 CLK9期货合约

#### 边 2: 卖 1 CLM9期货合约

以下为使用API下达组合定单要求的步骤：

- 获取每一边的合约代号( conId) 。通过调用reqContractDetails()方法获取这个数字。
- 通过填充相应的区域包含ComboLeg对象的每一边。
- 使用合约和定单插口客户资源文件实施placeOrder()方法。

### 下达这个组合定单

1. 获取两边定义的合约代号。请求1分配给CLK9和请求2分配给CLM9。

```
con1.localSymbol = "CLK9";
con1.secType = "FUT";
con1.exchange = "NYMEX";
con1.currency = "USD";

m_client->reqContractDetails(1, con1->getContract()); // request 1

con2.m_localSymbol = "CLM9";
con2.m_secType = "FUT";
con2.m_exchange = "NYMEX";
con2.m_currency = "USD";

m_client->reqContractDetails(2, con2->getContract()); // request 2
```

conId值通过以下活动提供。如果reqId等于1，则conid是CLK9合约代号。如果reqId等于2，则conId是CLM9。

```
::contractDetails( int reqId, const ContractDetails &contractDetails)
{

// to obtain conId for CLK9
if (reqId == 1)
...
// to obtain conid for CLM9
if (reqId == 2)
...
}
```

```
}
```

2. 为组合定单和组合分配所有相关的值:

```
leg1.conId = Leg1_conId;
leg1.ratio = 1;
leg1.action = "BUY";
leg1.exchange = "NYMEX";
leg1.openClose = 0;
leg1.shortSaleSlot = 0;
leg1.designatedLocation = "";

leg2.conId = Leg2_conId;
leg2.ratio = 1;
leg2.action = "SELL";
leg2.exchange = "NYMEX";
leg2.openClose = 0;
leg2.shortSaleSlot = 0;
leg2.designatedLocation = "";
```

3. 对相应的合约和定单对象调用placeOrder()方法。如下所示，这包括合约对象中的addAllLegs声明。

```
contract.symbol = "USD"; // arbitrary value only combo orders
contract.secType = "BAG"; // BAG is the security type for COMBO order
contract.exchange = "NYMEX";
contract.currency = "USD";
contract.comboLegs = addAllLegs; //including combo order in contract object

order.m_action = "BUY";
order.m_totalQuantity = 1;
order.m_orderType = "MKT";

m_client->placeOrder(Orderid, contract->getContract(), order->getOrder());
```

## Java

这一章介绍Java API，包括下列题目：

- [使用Java API连接TWS](#)
- [运行Java测试客户样本程序](#)
- [和Eclipse一起运行Java测试客户程序](#)
- [Java测试客户概括](#)
- [Java API概括](#)
- [Java EClientSocket方法](#)
- [Java EWrapper方法](#)
- [Java SocketClient资源](#)
- [下达组合定单](#)
- [Java编码举例：合约参数](#)

## 使用Java API连接TWS

### 使用Java API连接TWS

1. 将com.ib.client.\* 输入到您的源代码文件。
2. 实施EWrapper接口。 这个class将从插口接收信息。
3. 取代下列方法：

Ewrapper方法	描述
tickPrice()	处理市场数据。
tickSize()	
tickOptionComputation()	
tickGeneric()	
tickString()	
tickEFP()	
orderStatus()	
openOrder()	接收开放订单。
error()	接收出错信息。
connectionClosed()	在TWS终止连接时发出通知。
updateAccountValue()	接收当前账户值。
updateAccountTime()	接收最后一次账户更新信息。
updatePortfolio()	接收当前投资组合信息。
nextValidId()	接收连接时的下一个有效订单代号。
contractDetails()	接收合约信息。
contractDetailsEnd()	确认一个给定合约细节请求的结束。

bondContractDetails()	接收债券合约信息。
execDetails()	接收执行报告信息。
updateMktDepth()	接收市场深度信息。
updateMktDepthL2()	接收二级市场深度信息。
updateNewsBulletin()	接收IB新闻公告。
managedAccounts()	接收金融顾问(FA)管理的账户列表。
receiveFA()	接收FA配置信息。
historicalData()	接收历史数据结果。
scannerParameters()	接收描述扫描仪订阅的有效参数的XML文件。
scannerData()	接收市场扫描仪结果。
realTimeBar()	接收实时柱。
currentTime()	接收服务器的当前系统时间。
fundamentalData()	接收路透社全球基本面市场数据。

4. 实例EClientSocket class。该对象将被用于发送信息到TWS。

5. 调用下列方法：

EClientSocket方法	描述
eConnect()	连接TWS。
eDisconnect()	断开TWS。
reqMktData()	请求市场数据。
cancelMktData()	取消市场数据。
reqMktDepth()	请求市场深度。
cancelMktDepth()	取消市场深度。
reqContractDetails()	请求合约详细。
placeOrder()	下达订单。
cancelOrder()	取消订单。
reqAccountUpdates()	请求账户值、投资组合、和账户更新时间信息。

reqExecutions()	请求日执行报告列表。
reqOpenOrders()	请求关于请求客户的当前开放定单列表和将TWS开发定单与客户相连。相连仅在请求客户的客户代号为0时发生。
reqAllOpenOrders()	请求所有开放定单的列表。
reqAutoOpenOrders()	自动将新的TWS运作与客户相连。相连仅在请求客户的客户代号为0时发生。
reqNewsBulletin()	请求IB新闻公告。
cancelNewsBulletins()	取消IB新闻公告。
setServerLogLevel()	设置API请求和处理记录的级别。
reqManagedAccts()	请求金融顾问(FA)管理的账户编码列表。
requestFA()	请求TWS的FA配置信息。
replaceFA()	修改API的FA配置信息。
reqScannerParameters()	请求描述扫描仪订阅有效参数的XML文件。
reqScannerSubscription()	请求市场扫描仪结果。
cancelScannerSubscription()	取消扫描仪订阅。
reqHistoricalData()	请求历史数据。
cancelHistoricalData()	取消历史数据。
reqRealTimeBars()	请求实时柱。
cancelRealTimeBars()	取消实时柱。
exerciseOptions()	行使期权。
reqCurrentTime()	请求当前服务器时间。
serverVersion()	反馈API应用程序连接的TWS实例的版本。
TwscConnectionTime()	反馈API应用程序连接到TWS的时间。

<code>reqFundamentalData()</code>	请求路透社全球基本面数据。必须通过账户管理订阅路透社基本面才可以接收数据。
<code>cancelFundamentalData()</code>	取消路透社全球基本面数据。

## 运行Java测试客户端样本程序

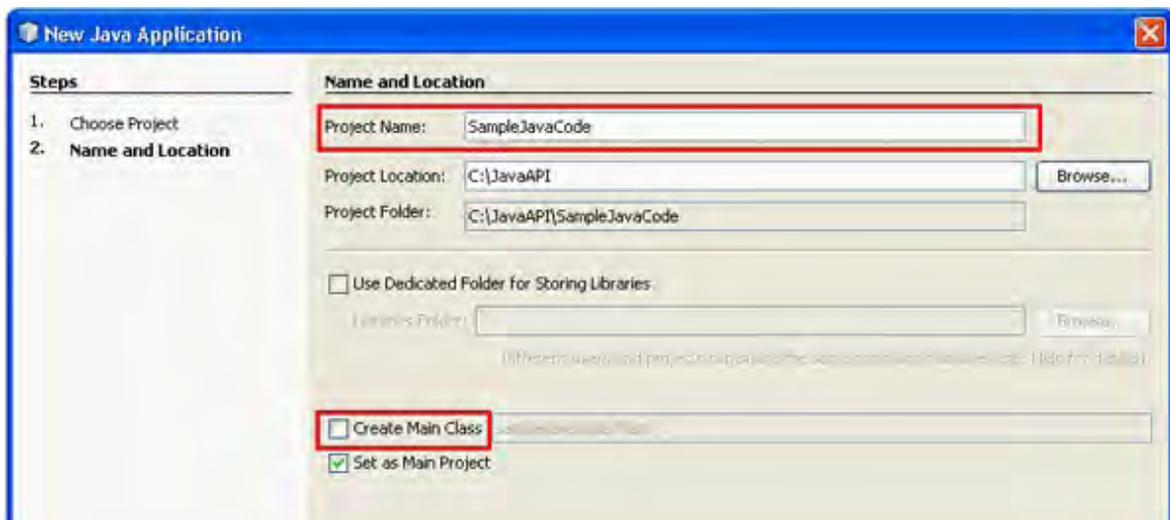
你可以通过TWS进入IB交易系统或使用插口客户端组件通过Java应用程序进入IB网关。在你连接TWS之前，你必须：

### 在Windows上运行Java测试客户端样本程序

1. 从Windows Explorer，进入Its:\Java 文件夹。
2. 运行文件 *run.bat*。

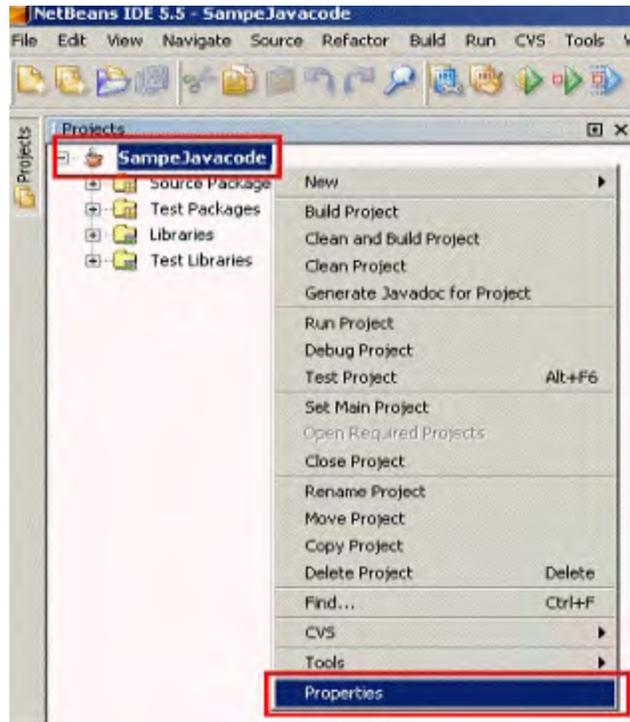
### 在NetBeans中从一个新的项目运行Java测试客户端样本程序

1. 打开NetBeans并点击 *New Project* ( 新项目 ) 开始向导。
2. 在项目区域，选择 *Java Application* ( Java应用程序 ) 并点击 **Next** ( 继续 ) 。
3. 在 **New Java Application** ( 新Java应用程序 ) 窗口中命名你的项目、选择一个地点、并取消选取 *Create Main Class* ( 创建主要Class ) 选取框。
4. 点击 **Finish** ( 完成 ) 。

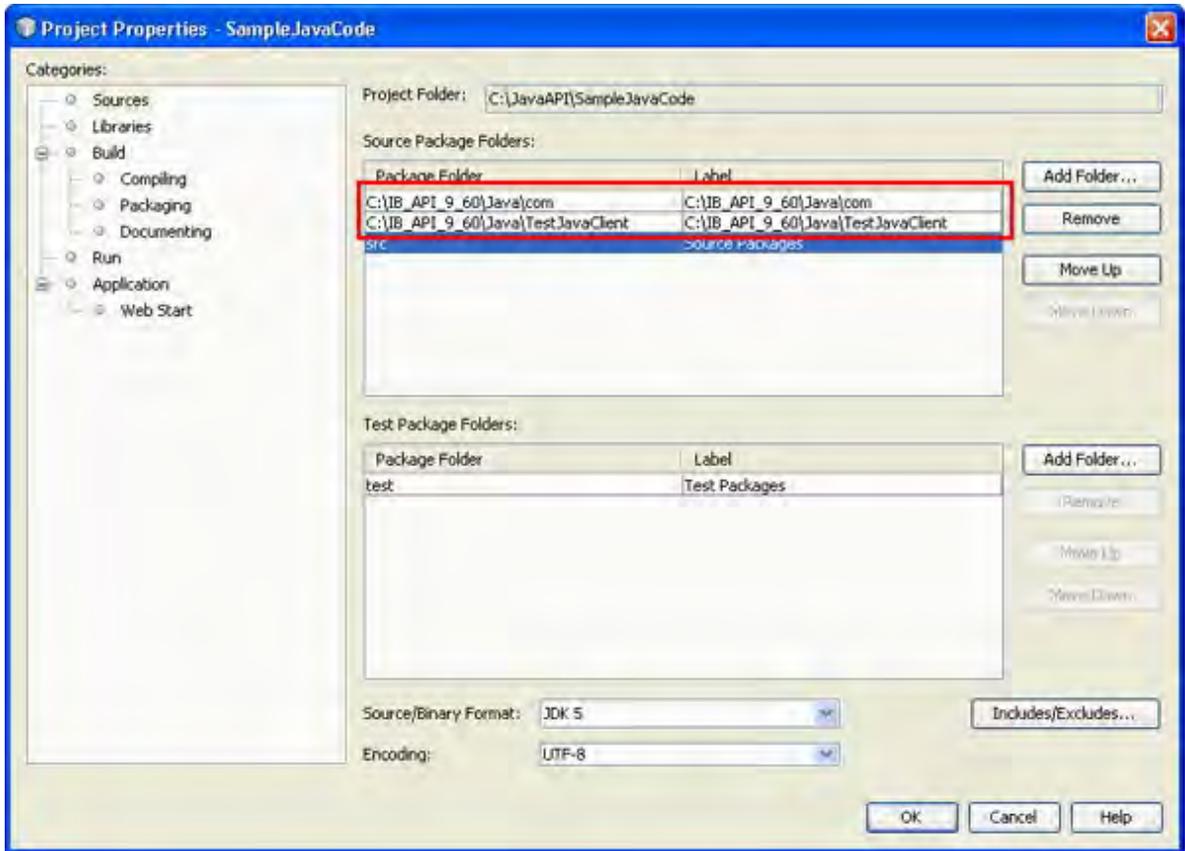




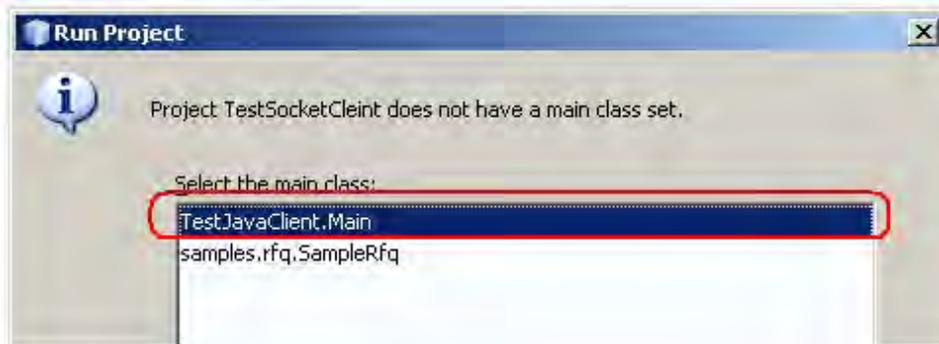
5. 设置Java来使用API，右键点击你刚创建的项目，然后选择 **Properties(资源)**。



6. 从来源分类点击 *Add Folder(添加文件夹)*。
7. 导航进入安装TWS API的文件夹。你希望添加的文件夹的名称是 *Java\com*和 *Java\TestJavaClient*。点击 **OK**。



- 按下 **F6** 运行样本java项目。当信息说“Project Samplejavacode does not have main class set(项目 Samplejavacode还没有设置主 class)” select *TestJavaclient.Main*， 点击 **OK**。



Java测试客户端的样本程序窗口如下图。



## 运行Java测试客户程序和Eclipse

这一部分描述如何运行Java测试客户程序和Eclipse IDE。 下列步骤假设你已经下载并安装了TWS API软件。

**Note:** 这个过程假设你使用Eclipse Helios 3.6.2, 但其它版本Eclipse也应该工作。

### 运行Java测试客户程序和Eclipse

1. 从<http://www.eclipse.org/downloads/>下载Eclipse IDE。
  - 根据你的操作系统下载Windows 32位或Windows 64位版本。
  - 下载过程中将建议最适合你的自动下载。 点击链接并在您的计算机中保存zip文件。
2. Unzip下载的Eclipse文件。
3. 运行文件 **eclipse.exe**( 将被unzipped到eclipse目录中) 启动Eclipse IDE。
4. 开始一个新项目：
  - a. 在Eclipse中开始一个新项目，选择 **File > New > Java Project**。
  - b. 输入项目名称。此例中，命名项目为 *My API Program*。
  - c. 可选择改变各种设置，比如JRE和工作区位置。
  - d. 点击 **Finish( 完成)** 。
5. 导入TWS API源文件：
  - a. 在左侧的Package Explorer面板中扩展你刚创建的项目。
  - b. 右键点击**src**文件夹，然后选择 **New > Package**。
  - c. 输入 *com.ib.client* 做为 package com, 然后点击 **Finish**。
  - d. 在Package Explorer面板中右键点击 **packagecom.ib.client**, 然后选择 **Import**。
  - e. 选择 **General > File System**, 然后点击 **Next**。
  - f. 点击 **Browse...**, 然后找出安装API的文件夹地点( 通常为Java\com\ib\client) 。选择 **client** 文件夹( 例如, C:\Jts\Java\com\ib\client\), 然后点击 **OK**。
  - g. 复选 **client** 文件夹, 然后点击 **Finish**。
8. 输入Java样本测试客户文件：
  - a. 在左侧的Package Explorer面板中扩展你刚创建的项目。
  - b. 右键点击**src**文件夹，然后选择 **New > Package**。
  - c. 输入 *TestJavaClient* 做为 package com, 然后点击 **Finish**。
  - d. 在Package Explorer面板中右键点击 **packageTestJavaClient**, 然后选择 **Import**。

- e. 选择 **General > File System**，然后点击 **Next**。
  - f. 点击 **Browse...**，然后找到安装Java测试客户的文件夹(通常为Java\TestJavaClient)。选择 **TestJavaClient** 文件夹(例如，C:\Jts\Java\TestJavaClient)，然后点击 **OK**。
  - g. 复选 **client**文件夹，然后点击 **Finish**。
8. 运行样本测试客户：
- a. 右键点击TestJavaClient package并选择 **Run As... > Java Application**。
  - b. 下面是你应该看到的内容，此时你可以创建你自己的客户化项目：



## Java测试客户概括

这一部分介绍用于Java测试客户端的package，classes和方法，TWS API包括样本Java程序。

**Note:** 用于Java测试客户端的classes和方法是为样本程序设计的，不属于TWS Java API。

### Package

Java测试括号弹包括package TestJavaClient。这个package包括所有Java测试括号弹样本程序要求的classes和方法。

### TestJavaClient Classes

TestJavaClient包括下列classes:

Class	描述
AccountDlg	定义账户/投资组合对话框，是Java测试客户端样本程序用来显示一个客户账户和投资组合信息。
AcctUpdatesDlg	定义账户更新对话框，是Java测试客户端样本程序用来让FA客户订阅账户更新。
ComboLegDlg	定义组合边对话框，是Java测试客户端样本程序用来让客户添加和取消组合边。
ConnectDlg	定义连接对话框，是Java测试客户端样本程序用来让客户连接TWS。
ExecFilterDlg	定义执行报告过滤器对话框，是Java测试客户端样本程序用来让客户为执行报告输入过滤条件。
ExtOrdDlg	定义扩展定单信息对话框，是Java测试客户端样本程序用来让客户为扩展定单属性输入值。
FAAllocationInfoDlg	定义FA分配信息对话框，是Java测试客户端样本程序用来让金融顾问客户输入关于分配方式和账户组的信息。
FinancialAdvisorDlg	定义金融顾问对话框，是金融顾问客户在Java是Java测试客户端样本程序中使用。
LogConfigDlg	定义记录配置对话框，是Java测试客户端样本程序用来让客户指定输入到记录文件中的记录级别。

MktDepthDlg	定义市场深度对话框，是Java测试客户端样本程序用来让客户查看特定合约的市场深度。
NewsBulletinDlg	定义IB新闻公告订阅对话框，是Java测试客户端样本程序用来让客户订阅和取消订阅新闻公告。
OrderDlg	定义样本对话框，是Java测试客户端样本程序用来让客户为定单和请求合约数据、市场深度、市场数据、期权行使、和历史数据查询输入合约和定单信息。
SampleFrame	定义Java测试客户端样本程序主窗口。主窗口上的所有文字面板和按钮均通过这个class定义。
ScannerDlg	定义市场扫描仪对话框(也称之为测试客户端的样本对话)，是Java测试客户端样本程序用来让客户订阅和取消订阅市场扫描，以及请求市场扫描参数。

**Note:** 有关是Java测试客户端样本程序中Java编码的详细信息，请参阅[TWS Java API指南入门](#)。

### Java API概括

TWS Java API包括package **com.ib.client**，其包含以下classes:

Class	描述
<a href="#">EWrapper</a>	该接口负责从TWS接收信息。
<a href="#">ComboLeg</a>	该class包含用于描述组合边的属性。
<a href="#">Contract</a>	该class包括用于描述合约的属性。
<a href="#">ContractDetails</a>	该class包括用于描述合约详细的属性，包括债券详细。
<a href="#">EClientSocket</a>	该class负责向TWS发送信息。
<a href="#">Execution</a>	该class包含用于描述交易的属性。
<a href="#">ExecutionFilter</a>	该class包含用于描述执行过滤器条件的属性。
<a href="#">Order</a>	该class包含用于描述定单的属性。

<a href="#"><u>OrderState</u></a>	该class包含用于描述定单状态的属性。
<a href="#"><u>ScannerSubscription</u></a>	该class包含用于描述市场扫描元素的属性。
<a href="#"><u>TickType</u></a>	该class定义通用跳动类型和其跳动值。

这些classes的方法和属性将在本章剩下的部分中阐述。



## Java EClientSocket方法

这一节介绍当连接TWS时你将使用的class EClientSocket方法。方法列表包括：

<p><b>连接和服务端</b></p> <p><a href="#">EClientSocket()</a>  <a href="#">eConnect()</a>  <a href="#">eDisconnect()</a>  <a href="#">isConnected()</a>  <a href="#">setServerLogLevel()</a>  <a href="#">reqCurrentTime()</a>  <a href="#">serverVersion()</a>  <a href="#">TwsConnectionTime()</a></p> <p><b>市场数据</b></p> <p><a href="#">reqMktData()</a>  <a href="#">cancelMktData()</a>  <a href="#">calculateImpliedVolatility()</a>  <a href="#">cancelCalculateImpliedVolatility()</a>  <a href="#">calculateOptionPrice()</a>  <a href="#">cancelCalculateOptionPrice()</a>  <a href="#">reqMarketDataType()</a></p> <p><b>定单</b></p> <p><a href="#">placeOrder()</a>  <a href="#">cancelOrder()</a>  <a href="#">reqOpenOrders()</a>  <a href="#">reqAllOpenOrders()</a>  <a href="#">reqAutoOpenOrders()</a>  <a href="#">reqIDs()</a>  <a href="#">exerciseOptions()</a></p> <p><b>账户</b></p> <p><a href="#">reqAccountUpdates()</a></p> <p><b>执行</b></p> <p><a href="#">reqExecutions()</a></p>	<p><b>合约详细</b></p> <p><a href="#">reqContractDetails()</a></p> <p><b>市场深度</b></p> <p><a href="#">reqMktDepth()</a>  <a href="#">cancelMktDepth()</a></p> <p><b>新闻公告</b></p> <p><a href="#">reqNewsBulletins()</a>  <a href="#">cancelNewsBulletins()</a></p> <p><b>金融顾问</b></p> <p><a href="#">reqManagedAccts()</a>  <a href="#">requestFA()</a>  <a href="#">replaceFa()</a>  <a href="#">reqAccountSummary()</a>  <a href="#">cancelAccountSummary()</a>  <a href="#">reqPositions()</a>  <a href="#">cancelPositions()</a></p> <p><b>市场扫描仪</b></p> <p><a href="#">reqScannerParameters()</a>  <a href="#">reqScannerSubscription()</a>  <a href="#">cancelScannerSubscription()</a></p> <p><b>历史数据</b></p> <p><a href="#">reqHistoricalData()</a>  <a href="#">cancelHistoricalData()</a></p> <p><b>实时柱</b></p> <p><a href="#">reqRealTimeBars()</a>  <a href="#">cancelRealTimeBars()</a></p> <p><b>基本面数据</b></p> <p><a href="#">reqFundamentalData()</a>  <a href="#">cancelFundamentalData()</a></p>
--	---

### **EClientSocket()**

这是个构造方法。

### **EClientSocket(AnyWrapper anyWrapper)**

参数	描述
anyWrapper	对AnyWrapper基本接口衍生的对象的参考。注意，EWrapper是AnyWrapper的延伸。

### **eConnect()**

必须首先调用这个功能。没有成功连接的反馈，但接下来的连接试图将返回“已经连接”的信息。

### **void eConnect( String host, int port, int clientId)**

参数	描述
host	运行TWS的主机名或IP地址。保持空白连接本地主机。
port	必须匹配TWS中指定的端口Configure>API>Socket Port区域。
clientId	用于确认该客户连接的数字。从该客户下达的/修改的所有定单将与这个标识符相连。 <b>Note:</b> 注：每个客户必须使用独特的clientId连接。

### **eDisconnect()**

调用这个方法终止与TWS的连接。调用该方法并不取消已经发送的定单。

### **void eDisconnect()**

### **isConnected()**

调用该方法查看是否存在与TWS的连接。

### **void isConnected()**

### **setServerLogLevel()**

默认级别是出错。参考[API记录](#)页面了解详细信息。

### **void setServerLogLevel(int logLevel)**

参数	描述
----	----

logLevel	<p>当处理 API 请求时，指定服务器( TWS) 使用的输入详细的记录级别。有效值包括：</p> <ul style="list-style-type: none"> <li>• 1 = 系统</li> <li>• 2 = 错误</li> <li>• 3 = 警告</li> <li>• 4 = 信息</li> <li>• 5 = 详细</li> </ul>
----------	---

### reqCurrentTime()

通过 [currentTime\(\)](#) EWrapper 方法反馈服务器的当前系统时间。

void reqCurrentTime()

### serverVersion()

反馈 API 应用程序连接的 TWS 实例的版本。

void serverVersion()

### TwsConnectionTime()

反馈 API 与 TWS 开始连接的时间。

void TwsConnectionTime ()

### reqMktData()

调用该方法请求市场数据。将通过 [tickPrice\(\)](#)， [tickSize\(\)](#)， [tickOptionComputation\(\)](#)， [tickGeneric\(\)](#)， [tickString\(\)](#) 和 [tickEFP\(\)](#) 方法返回市场数据。

void reqMktData(int tickerId, Contract contract, String genericTicklist, boolean snapshot)

参数	描述
tickerId	代码代号。必须是唯一的值。当返回市场数据时，必须通过这个标签来确认。 也被用来取消市场数据。
<a href="#">contract</a>	该 class 包含用于描述合约的属性。
genericTicklist	一个由冒号分隔的通用跳动类型列表。 跳动类型信息可以通过 <a href="#">通用跳动类型</a> 页面查看。
snapshot	选取返回市场数据的单个快照和取消市场数据订阅。 如果使用快照，不要输入任何 genericTicklist 值。

### cancelMktData()

调用该方法后，将停止输送特定代号的的市场数据。

**void cancelMktData(int tickerId)**

参数	描述
tickerId	调用 reqMktData()中指定的代号。

**calculateImpliedVolatility()**

调用该方法计算提供的期权价格和底层价格的波动率。

**calculateImpliedVolatility(int reqId, Contract optionContract, double optionPrice, double underPrice)**

参数	描述
reqId	代码代号。
optionContract	描述合约。
optionPrice	期权价格。
underPrice	底层价格。

**cancelCalculateImpliedVolatility()**

调用这个功能取消计算提供的期权价格和底层价格波动率的请求。

**calculateImpliedVolatility(int reqId)**

参数	描述
reqId	代码代号。

**calculateOptionPrice()**

调用这个功能计算提供的波动率和底层价格的期权价格和希腊值。

**void calculateOptionPrice(int reqId, Contract contract, double volatility, double underPrice)**

参数	描述
conid	代码代号。
contract	描述合约。
volatility	波动率。
underPrice	底层价格。

**cancelCalculateOptionPrice()**

调用这个功能取消对提供的波动率和底层价格计算期权价格和希腊值的请求。

**cancelCalculateOptionPrice(int reqId)**

参数	描述
reqId	代码代号。

### reqMarketDataType()

API可以从交易平台接收冻结市场数据。冻结市场数据是我们系统中记录的最后数据。在正常交易时段中，API接收实时市场数据。如果你使用该功能，你是在指示TWS在收盘后自动切换到冻结市场数据。然后，在下一个开盘日之前，市场数据将自动转回到实时市场数据。

### reqMarketDataType(int type)

参数	描述
type	1为实时流市场数据或2为冻结市场数据。

### placeOrder()

void placeOrder( int id, Contract contract, Order order)

参数	描述
id	订单代号。你必须指定一个独特的值。当反馈订单状态时，将由这个标签确认。这个标签也被用于取消订单。
contract	该class包括用于描述合约的属性。
order	该结构包括订单详细。注：每个客户必须使用一个独特的clientId。

### cancelOrder()

调用这个方法取消订单。

void cancelOrder(int id)

参数	描述
id	之前调用placeOrder()中指定的订单代号

### reqOpenOrders()

调用该方法请求任何由这个API客户下达的开放订单。每个开放订单将通过EWrapperEach中的[openOrder\(\)](#)和[orderStatus\(\)](#)方法反馈。

**Note:** 具有clientId为“0”的客户也将会收到TWS拥有的开放订单。这些订单将和客户相连，并生成一个新的orderId。这个相连将在多个API和TWS运行过程中保持不变。

void reqOpenOrders()

### reqAllOpenOrders

调用该方法请求和一个TWS链接的所有客户下达的所有开放订单，和来自TWS的所有开放订单。注，你可以从一个单一TWS运行8个API客户。每个开放订单将通过EWrapper的[openOrder\(\)](#)和[orderStatus\(\)](#)方法反馈回来。

**Note:** 在返回的订单和请求的客户之间没有产生关联。

**void reqAllOpenOrders()**

### reqAutoOpenOrders()

调用这个方法请求将新近创建的TWS订单和客户隐含相连。当一个新的TWS订单被创建后，订单将被与客户相连并自动的通过EWrapper中的[openOrder\(\)](#)和[orderStatus\(\)](#)方法反馈。

**Note:** TWS订单只可以和clientId为0的客户相连。

**void reqAutoOpenOrders(boolean bAutoBind)**

参数	描述
bAutoBind	如果设为真( TRUE) ，新近创建的TWS订单将和客户隐形相连。如果设为假( FALSE) ，不生成相连。

### reqIDs()

调用该功能请求下一个有效的代号，可以在下达订单时使用。调用该功能后，将触发[nextValidId\(\)](#)事件，返回的代号将是下一个有效代号。该代号将反映发生的任何自动绑定(生成新的、和下一个有效代号的其中的增量)。

**Public synchronized Void reqIds (int numIds)**

参数	描述
numIds	设为1。

### exerciseOptions()

调用exerciseOptions()方法行使期权。

**Note:** SMART( 智能) 在exerciseOptions()调用中是不允许的交易所，在API启动行使或试图放弃时，TWS会对有问题的期权发出请求。

**void exerciseOptions(int tickerId, Contract contract, int exerciseAction, int exerciseQuantity, String account, int override)**

参数	描述
tickerId	行使请求的代号。
contract	该class包括用于描述合约的属性。

exerciseAction	指定是否行使特定的期权或让期权过期。有效值是： <ul style="list-style-type: none"> <li>• 1 = 行使</li> <li>• 2 = 放弃</li> </ul>
exerciseQuantity	被行使的合约数目。
account	用于机构定单。指定IB账户。
override	指定是否你的设置将取代系统的自然行动。例如，如果你的行动是“行使”，期权不是溢价期权，自然行动将不会行使期权。如果你将取代设置为“yes”，自然行动将被取代，且蚀价期权将被行使。值为： <ul style="list-style-type: none"> <li>• 0 = 不取代</li> <li>• 1 = 取代</li> </ul>

### reqAccountUpdates()

调用这个功能开始获得账户值、投资组合、和最后更新时间信息。账户数据将通过 [updateAccountTime\(\)](#), [updateAccountValue\(\)](#)和 [updatePortfolio\(\)](#) EWrapper methods 反馈。

**void reqAccountUpdates (boolean subscribe, String acctCode)**

参数	描述
subscribe	如果设为真，客户将开始接收账户和投资组合更新。如果设为假，客户将停止接收该信息。
acctCode	接收账户和投资组合更新的账户代码。

### reqExecutions()

当调用这个方法时，满足过滤条件的过去24小数的执行报告将通过 [execDetails\(\)](#)方法下载给客户。查看24小时以外的执行报告，打开TWS中的交易记录，当小数交易记录时，请求再次从API下载执行报告。

**void reqExecutions(ExecutionFilter filter)**

参数	描述
filter	用于确定返回具体执行报告的过滤器条件。

### reqContractDetails()

调用该方法下载特定合约的所有详细信息。合约详细将通过EWrapper的 [contractDetails\(\)](#) 方法接收。

**void reqContractDetails (int reqId, Contract contract)**

参数	描述
reqId	数据请求代号。如果存在几个请求时，保证结果与请求相匹配。
contract	该class包括用于描述合约的属性。

### reqMktDepth()

调用该方法请求特定合约的市场深度。市场深度将通过[updateMktDepth\(\)](#)和[updateMktDepthL2\(\)](#)方法返回。

#### void reqMktDepth(int tickerId, Contract contract, int numRows)

参数	描述
tickerId	代码代号。必须是唯一的值。当返回市场深度数据时，将由这个标签确认。也被用于取消市场深度时。
contract	这个class包括用于描述合约的属性。
numRows	指定返回的市场深度行的数目。

### cancelMktDepth()

调用这个方法之后，特定代号的市場深度数据将停止输送。

#### void cancelMktDepth(int id)

参数	描述
tickerId	调用reqMktDepth()中指定的代号。

### reqNewsBulletins()

调用这个方法开始接收新闻公告。每个公告将通过[updateNewsBulletin\(\)](#)方法反馈。

#### void reqNewsBulletins(boolean allMsgs)

参数	描述
allMsgs	如果设为真，返回当天的所有现有公告和任何新的公告。如果设为假，将仅返回新的公告。

### cancelNewsBulletins()

调用该方法停止接收新闻公告。

#### void cancelNewsBulletins()

### reqManagedAccts()

调用这个方法请求管理账户列表。该列表将通过EWrapper的[managedAccounts\(\)](#)方法返回。



**Note:** 这个请求只能在连接金融顾问( FA) 账户时才可以进行。

**void reqManagedAccts()**

**requestFA()**

调用这个方法请求TWS中的FA配置信息。数据通过[receiveFA\(\)](#)方法以XML字符串形式返回。

**void requestFA(long faDataType)**

参数	描述
faDataType	指定被请求的金融顾问配置数据类型。有效值包括： <ul style="list-style-type: none"> <li>• 1 = 组</li> <li>• 2 = 概括</li> <li>• 3 = 账户别名</li> </ul>

**replaceFA()**

调用该方法请求TWS中新的FA配置信息。数据通过"receiveFA"方法以XML字符串形式返回。

**void replaceFA(long faDataType, string xml)**

参数	描述
faDataType	指定被请求的金融顾问配置数据的类型。有效值包括： <ul style="list-style-type: none"> <li>• 1 = 组</li> <li>• 2 = 概括</li> <li>• 3 = 账户别名</li> </ul>
xml	包含新的FA配置信息的XML字符串。

**reqAccountSummary()**

调用该方法请求出现在TWS账户窗口概括标签上的数据，并保持更新。数据通过[accountSummary\(\)](#)返回。

**Note:** 请求只有当连接金融顾问( FA) 账户时才能进行。

**void reqAccountSummary(int messageType, int version, int reqId, String groupName, String tags)**

参数	类型	描述
messageType	整数	设为62。
version	整数	设为1。

参数	类型	描述
reqId	整数	
groupName	字符串	设为 <i>All</i> (全部) 返回所有账户的账户概括数据，或设为一个已经通过TWS全局配置创建的特定的顾问账户组名称。

参数	类型	描述
tags	字符串	<p>一个由冒号隔开的账户标签列表。 可用的标签包括：</p> <ul style="list-style-type: none"> <li>• <i>AccountType</i></li> <li>• <i>NetLiquidation</i>,</li> <li>• <i>TotalCashValue</i> — 包括期货pnl的总现金</li> <li>• <i>SettledCash</i> — 对现金账户，与<i>TotalCashValue</i>相同</li> <li>• <i>AccruedCash</i> — 净累计利息</li> <li>• <i>BuyingPower</i> — 账户可买进的可适用保证金的美国股票的最大额。</li> <li>• <i>EquityWithLoanValue</i> — 现金 + 债券 + 共同基金</li> <li>• <i>PreviousEquityWithLoanValue</i>,</li> <li>• <i>GrossPositionValue</i> — 所有股票和股票期权头寸的绝对值总合</li> <li>• <i>RegTEquity</i>,</li> <li>• <i>RegTMargin</i>,</li> <li>• <i>SMA</i> — 特殊备忘录账户</li> <li>• <i>InitMarginReq</i>,</li> <li>• <i>MaintMarginReq</i>,</li> <li>• <i>AvailableFunds</i>,</li> <li>• <i>ExcessLiquidity</i>,</li> <li>• <i>Cushion</i> — 净清算值百分比的剩余流动性</li> <li>• <i>FullInitMarginReq</i>,</li> <li>• <i>FullMaintMarginReq</i>,</li> <li>• <i>FullAvailableFunds</i>,</li> <li>• <i>FullExcessLiquidity</i>,</li> <li>• <i>LookAheadNextChange</i> — 预期值生效时间</li> <li>• <i>LookAheadInitMarginReq</i>,</li> <li>• <i>LookAheadMaintMarginReq</i>,</li> <li>• <i>LookAheadAvailableFunds</i>,</li> <li>• <i>LookAheadExcessLiquidity</i>,</li> <li>• <i>HighestSeverity</i> — 衡量账户如何接近被清算</li> <li>• <i>DayTradesRemaining</i> — 在日交易模式被显示出来之前，一个用户能够进行开仓/关仓交易的数目。 值为"-1"表示用户可以进行</li> </ul>

参数	类型	描述
		无限制的日交易。 • <i>Leverage</i> — GrossPositionValue / NetLiquidation

### cancelAccountSummary()

取消请求账户窗口概况标签数据。

**void cancelAccountSummary(int messageId, int version As Integer, int reqId As Integer)**

参数	类型	描述
messageId	整数	设为 63。
version	整数	设为 1。
reqId	整数	被取消数据请求的代号。

### reqPositions()

请求所有账户的实时头寸数据。

**Note:** 该请求只有当连接金融顾问( FA) 账户时才能够进行。

**void reqPositions(int messageId, int version)**

参数	类型	描述
messageId	整数	设为 62。
version	整数	设为 1。

### cancelPositions()

取消实时头寸更新。

**void cancelPositions(in messageId, int version)**

参数	类型	描述
messageId	整数	设为 64。
version	整数	设为 1。

### reqScannerParameters()

调用 reqScannerParameters()方法接收描述扫描仪订阅可以使用的有效参数的XML文件。

**void reqScannerParameters()**

### reqScannerSubscription()

调用 reqScannerSubscription()方法开始通过 [scannerData\(\)](#) EWrapper方法接收市场扫描仪结果。

**void reqScannerSubscription(int tickerId, ScannerSubscription subscription)**

参数	描述
tickerId	订阅代号。必须为唯一的值。当接收订阅数据时，将通过这个代号确认。该代号也被用于取消扫描仪。
subscription	包括过滤器的扫描仪订阅参数概括。

**cancelScannerSubscription()**

调用 cancelScannerSubscription()方法停止接收市场扫描仪结果。

**void cancelScannerSubscription(int tickerId)**

参数	描述
tickerId	调用 reqScannerSubscription()中指定的代号。

**reqHistoricalData()**

调用 reqHistoricalData()方法开始通过 [historicalData\(\)](#) EWrapper方法接收历史数据结果。

**void reqHistoricalData (int id, Contract contract, String endDateTime, String durationStr, String barSizeSetting, String whatToShow, int useRTH, int formatDate)**

参数	描述
tickerId	请求的代号。必须是唯一的值。当接收数据时，由这个代号确认。该代号也被用于取消历史数据请求。
contract	该class包括用于描述合约的属性。
endDateTime	使用 yyyyymmdd hh:mm:ss tmz格式，这里结尾空格后允许使用时区(可选择)。
durationStr	<p>这是请求覆盖的时间区域，并使用： &lt;integer( 整数) &gt; &lt;unit( 单位) &gt; 格式来指定，例如， 1 D，有效的单位包括：</p> <ul style="list-style-type: none"> <li>• " S( 秒)</li> <li>• " D( 天)</li> <li>• " W( 星期)</li> <li>• " M( 月)</li> <li>• " Y( 年)</li> </ul> <p>如果没有指定单位，将使用秒。还有目前“年”限制为一年。</p>

<p>barSizeSetting</p>	<p>指定将返回的柱的尺寸( 在IB/TWS限制范围内) 。有效柱尺寸值包括：</p> <ul style="list-style-type: none"> <li>• 1 秒</li> <li>• 5 秒</li> <li>• 15 秒</li> <li>• 30 秒</li> <li>• 1 分钟</li> <li>• 2 分钟</li> <li>• 3 分钟</li> <li>• 5 分钟</li> <li>• 15 分钟</li> <li>• 30 分钟</li> <li>• 1 小时</li> <li>• 1 天</li> </ul>
<p>whatToShow</p>	<p>决定被提取数据的性质。有效值包括：</p> <ul style="list-style-type: none"> <li>• TRADES</li> <li>• MIDPOINT</li> <li>• BID</li> <li>• ASK</li> <li>• BID_ASK</li> <li>• HISTORICAL_VOLATILITY</li> <li>• OPTION_IMPLIED_VOLATILITY</li> </ul>
<p>useRTH</p>	<p>确定是否返回请求时间范围内的所有可用数据，或仅返回处于正常交易时段范围内的数据。有效值包括：</p> <ul style="list-style-type: none"> <li>• 0 - 返回所有数据，即使请求的市场是处于其正常交易时段以外。</li> <li>• 1 - 仅返回正常交易时段以内的数据，即使请求的时间区间部分或全部属于RTH( 正常交易时段) 以外。</li> </ul>
<p>formatDate</p>	<p>确定应用于返回柱的日期格式。有效值包括：</p> <ul style="list-style-type: none"> <li>• 1 - 应用于返回柱的时间格式： yyymmdd {space} {space} hh:mm:dd</li> <li>• 2 - 返回的日期为一个长的整数，指定自 1/1/1970 GMT 的秒数。</li> </ul>

有关历史数据请求限制的详细信息，请参阅[历史数据限制](#)题目。

### cancelHistoricalData()

调用cancelHistoricalData()方法停止接收历史数据结果。

**void cancelHistoricalData (int tickerId)**

参数	描述
tickerId	调用reqHistoricalData()中指定的代号。

### reqRealTimeBars()

调用reqRealTimeBars()方法开始通过[realtimeBar\(\)](#) EWrapper方法接收实时柱结果。

**void reqRealTimeBars(int tickerId, Contract contract, int barSize, String whatToShow, boolean useRTH)**

参数	描述
tickerId	请求的代号。必须是唯一的值。当接收数据时，将使用这个代号确认。代号也将被用于取消历史数据请求。
contract	该class包括用于描述合约的属性。
barSize	目前仅支持5秒的柱，如果使用任何其它值，将出现不正常。
whatToShow	确定提取数据的性质。有效值包括： <ul style="list-style-type: none"> <li>• TRADES</li> <li>• BID</li> <li>• ASK</li> <li>• MIDPOINT</li> </ul>
useRTH	仅正常交易时段。有效值包括： <ul style="list-style-type: none"> <li>• 0 = 返回所有请求时间区域内的可用数据，包括请求的市场是处于其正常交易时段以外的数据。</li> <li>• 1 = 仅返回请求产品正常交易时段以内的数据，即使请求的时间区间部分或全部属于正常交易时段以外。</li> </ul>

### cancelRealTimeBars()

调用这个方法停止接收实时柱结果。

**void cancelRealTimeBars (int tickerId)**

参数	描述
tickerId	调用 reqRealTimeBars() 中指定的代号。

### reqFundamentalData()

调用这个方法接收路透社全球基本面数据。在你接收这个数据之前，你必须通过账户管理订阅路透社基本面。

#### void reqFundamentalData(int reqId, Contract contract, string reportType)

参数	描述
reqId	数据请求代号。保证在同时处理几个请求时，结果与请求相匹配。
contract	这个结构包括请求的路透社基本面数据的合约描述。
reportType	确认报告类型，为以下其中之一： <ul style="list-style-type: none"> <li>• 估计</li> <li>• 财务报告</li> <li>• 概况</li> </ul>

### cancelFundamentalData()

调用该方法停止接收路透社全球基本面数据。

#### void cancelFundamentalData(int reqId)

参数	描述
reqId	数据请求代号。



## Java EWrapper 方法

这一节介绍在连接TWS时你可以使用的的class EWrapper 方法。列出的方法包括：

<p><b>连接和服务端</b></p> <p><a href="#"><u>currentTime()</u></a>  <a href="#"><u>error()</u></a>  <a href="#"><u>connectionClosed()</u></a></p> <p><b>市场数据</b></p> <p><a href="#"><u>tickPrice()</u></a>  <a href="#"><u>tickSize()</u></a>  <a href="#"><u>tickOptionComputation()</u></a>  <a href="#"><u>tickGeneric()</u></a>  <a href="#"><u>tickString()</u></a>  <a href="#"><u>tickEFP()</u></a>  <a href="#"><u>tickSnapshotEnd()</u></a>  <a href="#"><u>marketDataType()</u></a></p> <p><b>定单</b></p> <p><a href="#"><u>orderStatus()</u></a>  <a href="#"><u>openOrder()</u></a>  <a href="#"><u>nextValidId()</u></a></p> <p><b>账户和投资组合</b></p> <p><a href="#"><u>updateAccountValue()</u></a>  <a href="#"><u>updatePortfolio()</u></a>  <a href="#"><u>updateAccountTime()</u></a></p> <p><b>合约详细</b></p> <p><a href="#"><u>contractDetails()</u></a>  <a href="#"><u>contractDetailsEnd()</u></a>  <a href="#"><u>bondContractDetails()</u></a></p>	<p><b>执行</b></p> <p><a href="#"><u>execDetails()</u></a>  <a href="#"><u>execDetailsEnd()</u></a>  <a href="#"><u>commissionReport()</u></a></p> <p><b>市场深度</b></p> <p><a href="#"><u>updateMktDepth()</u></a>  <a href="#"><u>updateMktDepthL2()</u></a></p> <p><b>新闻公告</b></p> <p><a href="#"><u>updateNewsBulletin()</u></a></p> <p><b>金融顾问</b></p> <p><a href="#"><u>managedAccounts()</u></a>  <a href="#"><u>receiveFA()</u></a>  <a href="#"><u>accountSummary()</u></a>  <a href="#"><u>accountSummaryEnd()</u></a>  <a href="#"><u>position()</u></a>  <a href="#"><u>positionEnd()</u></a></p> <p><b>历史数据</b></p> <p><a href="#"><u>historicalData()</u></a></p> <p><b>市场扫描仪</b></p> <p><a href="#"><u>scannerParameters()</u></a>  <a href="#"><u>scannerData()</u></a>  <a href="#"><u>scannerDataEnd()</u></a></p> <p><b>实时柱</b></p> <p><a href="#"><u>realtimeBar()</u></a></p> <p><b>基本面数据</b></p> <p><a href="#"><u>fundamentalData()</u></a></p>
---	---

### currentTime()

该方法接收服务器的当前系统时间。

**void currentTime(long time)**

参数	描述
time	服务器的当前系统时间

**error()**

当通讯出现错误或当TWS要给客户发送消息时该方法被调用。

**void error(int id, int errorCode, String errorString)**

参数	描述
id	这是发生错误请求的orderId或tickerId。
errorCode	有关错误编码，请参阅 <a href="#">错误编码</a> 。
errorString	错误的文字描述。

该方法在处理请求时发生意外情况时调用。

**void error(Exception e)**

参数	描述
e	发生的意外。

该方法在TWS希望给客户发送错误信息时调用。(V1).

**void error(String str)**

参数	描述
str	这是错误的文字描述

**connectionClosed()**

该方法在TWS关闭插口连接时，或当TWS被关闭时调用。

**void connectionClosed()**

**tickPrice()**

当市场数据改变时调用该方法。没有延迟的立即更新价格。

**void tickPrice(int tickerId, int field, double price, int canAutoExecute)**

参数	描述
----	----

tickerId	之前调用reqMktData()中指定的代码代号。
field	指定价格类型。将区域值转至TickType.getField(int tickType)接收区域描述。例如，区域值1将显示bidPrice，区域值2将显示askPrice，等。  <ul style="list-style-type: none"> <li>• 1 = 买价</li> <li>• 2 = 卖价</li> <li>• 4 = 最后价</li> <li>• 6 = 最高价</li> <li>• 7 = 最低价</li> <li>• 9 = 收盘价</li> </ul>
price	为特定区域指定的价格
canAutoExecute	指定是否价格跳动可用于自动执行。可能的值包括：  <ul style="list-style-type: none"> <li>• 0 = 不适用于自动执行</li> <li>• 1 = 可用于自动执行</li> </ul>

### tickSize()

当市场数据变动时调用该方法。尺寸立即更新，没有延迟。

**void tickSize(int tickerId, int field, int size)**

参数	描述
tickerId	之前调用reqMktData()中指定的代码代号。
field	指定价格类型。 将区域值转入TickType.getField(int tickType)获取区域描述。例如，区域值1将显示bidPrice，区域值2将显示askPrice，等。  <ul style="list-style-type: none"> <li>• 0 = 买价尺寸</li> <li>• 3 = 卖价尺寸</li> <li>• 5 = 最后价尺寸</li> <li>• 8 = 交易量</li> </ul>
size	指定特定区域的尺寸

### tickOptionComputation()

当期权市场或期权底层变化时调用该方法。TWS的期权模型波动率、价格、和deltas以及预期收到的期权底层证券股息的当前值。

**void tickOptionComputation(int tickerId, int field, double impliedVol, double delta, double optPrice, double pvDividend, double gamma, double vega, double theta, double undPrice)**

参数	描述
tickerId	之前调用 reqMktData() 中指定的代码代号。
field	指定期权计算类型。 将区域值植入 TickType.getField(int tickType) 获取区域描述。例如，区域值 13 将反映 modelOptComp，等。 <ul style="list-style-type: none"> <li>• 10 = 买价</li> <li>• 11 = 卖价</li> <li>• 12 = 最后价</li> </ul>
impliedVol	由 TWS 期权模型使用特定的 ticktype 值计算的隐含波动率。
delta	期权 delta 值。
optPrice	期权价格。
pvDividend	预期的期权底层证券股息的当前值。
gamma	期权 gamma 值。
vega	期权 vega 值。
theta	期权 theta 值。
undPrice	底层证券价格。

**tickGeneric()**

当市场数据改变时调用该方法。值被立即更新，没有延迟。

**void tickGeneric(int tickerId, int tickType, double value)**

参数	描述
tickerId	之前调用 reqMktData() 中指定的代码代号。
tickType	指定价格类型。 将该区域值移入 TickType.getField(int tickType) 中获取区域描述。例如，区域值 46 将连接 shortable，等。
value	特定区域的值。

**tickString()**

当市场数据改变时调用这个方法。值被立即更新，没有延迟。

**void tickString(int tickerId, int tickType, String value)**

参数	描述
tickerId	之前调用 reqMktData() 中指定的代码代号。
field	指定价格类型。 将该区域值移入 TickType.getField(int tickType) 中获取区域描述。例如，区域值 45 将连接到 lastTimestamp，等。
value	特定区域的值。

### tickEFP()

当市场数据改变时调用这个方法。值被立即更新，没有延迟。

**void tickEFP(int tickerId, int tickType, double basisPoints, String formattedBasisPoints, double impliedFuture, int holdDays, String futureExpiry, double dividendImpact, double dividendsToExpiry)**

参数	描述
tickerId	之前调用 reqMktData() 中指定的代码代号。
field	指定价格类型。 将该区域值移入 TickType.getField(int tickType) 中获取区域描述。例如，区域值 38 将给出 bidEFP，等。
basisPoints	年度基点，代表可以直接和经纪商利率比较的借贷利率。
formattedBasisPoints	以百分比形式描述的格式化字符串的年度基点
impliedFuture	隐含期货价格
holdDays	EFP 过期前的持有天数
futureExpiry	单一股票期货的过期日
dividendImpact	股息对年度基点利率的影响
dividendsToExpiry	单一股票期货过期前的预期股息

### tickSnapshotEnd()

当快照市场数据订阅已经被完成时调用该功能，并没有任何需要等待处理的事情。这个也包括超时的情况。

**void tickSnapshotEnd(int reqId)**

参数	描述
reqID	数据请求代号。

### marketDataType()

TWS发送marketDataType(type)反馈到API宣布市场数据已经在冻结和实时之间转换了，这里类型被设为冻结或实时。这个通知仅在市场数据在实时和冻结之间转换时才发送。marketDataType()反馈接受reqId参数并按每个订阅发送，因为不同的合约可以通常以不同时间表进行交易。

#### void marketDataType(int reqId, int type)

参数	描述
int reqId	数据请求代号
type	1为实时流市场数据或2为冻结市场数据。

### orderStatus()

该方法在定单状态改变时调用。在与TWS重新连接时，如果客户有任何开放定单也启动这个方法。

#### void orderStatus(int orderId, String status, int filled, int remaining, double avgFillPrice, int permId, int parentId, double lastFillPrice, int clientId, String whyHeld)

**Note:** 有可能orderStatus()返回重复的信息。应该根据情况对信息进行过滤。

参数	描述
id	之前调用placeOrder()中指定的定单代号。

status	<p>定单状态。可能的值包括：</p> <ul style="list-style-type: none"> <li>• <b>PendingSubmit</b> - 表示你已经传递了定单，但还未接到来自定单目的地接收的确认信息。注：这个定单状态不是由TWS发送的，应该由API设计人设置为当定单发送时发出。</li> <li>• <b>PendingCancel</b> - 表示你已经发送了取消定单的请求，但还未收到来自定单目的地的取消确认。此时，你的定单没有被确认取消。在你的取消请求等待期间。你的定单仍有可能得到执行。注：这个定单状态不是由TWS发送的，应该由API设计人设置为当定单发送时发出。</li> <li>• <b>PreSubmitted</b> - 表示模拟定单类型已经被IB系统接收，但还未被选中。定单被保持在IB系统中直到选择条件被满足。届时，定单将被发送到指定的定单目的地。</li> <li>• <b>Submitted</b> - 表示你的定单已经被定单目的地接受，并处于工作中。</li> <li>• <b>Cancelled</b> - 表示你定单的剩余部分已被IB系统确认取消了。这也会在IB或目的地拒绝你的定单时发生。</li> <li>• <b>Filled</b> - 表示定单已被全部执行。</li> <li>• <b>Inactive</b> - 表示定单已被系统接收(模拟定单)或交易所接收(本地定单)，但由于系统、交易所或其它原因，目前定单处于非工作状态。</li> </ul>
filled	<p>指定已被执行的股票数。</p> <p>有关部分执行的信息，请参考<a href="#">部分执行的定单状态</a>。</p>
remaining	<p>指定剩余部分的股票数。</p>
avgFillPrice	<p>执行股票的平均价格。该参数仅在<b>filled</b>参数值大于零时有效。否则，价格参数将为零。</p>
permId	<p>用于确认定单的TWS代号。整个TWS运行中保持不变。</p>
parentId	<p>母定单的定单代号，用于括号单和自动跟踪止损单。</p>
lastFilledPrice	<p>被执行股票的最后价格。该参数仅在<b>filled</b>参数值大于零时有效。否则，价格参数将为零。</p>

clientId	下达定单的客户代号( 或TWS) 。注意，TWS定单的clientId是固定的，其orderId为0以区别API定单。
whyHeld	这个区域被用于在TWS寻求卖空股票时确认持有的定单。用于指示这个的值是'locate'。

### openOrder()

调用这个方法提供开放定单。

**void openOrder(int orderId, Contract contract, Order order, OrderState orderState )**

参数	描述
orderId	由TWS分派的定单代号。用于取消或更新定单。
contract	描述合约的合约级别属性。
order	定义定单详细的定单级别属性。
orderState	orderState属性包括交易前和交易后数据的保证金和佣金区域。

### nextValidId()

成功连接TWS后调用这个方法。

**void nextValidId(int orderId)**

参数	描述
orderId	连接TWS时接收到的下一个可用的定单代号。所有后续定单将基于这个代号增量为一。

### updateAccountValue()

这个方法仅在已经调用EClientSocket对象的reqAccountUpdates()方法时调用。

**void updateAccountValue(String key, String value, String currency, String accountName)**

参数	描述
----	----



key	表示一种账户值类型的字符串。有很多可被发送的可用的标签，这里仅列出几个样本： <ul style="list-style-type: none"> <li>• CashBalance - 账户现金余额</li> <li>• DayTradesRemaining - 剩余交易日</li> <li>• EquityWithLoanValue - 含借贷值股权</li> <li>• InitMarginReq - 当前初始保证金要求</li> <li>• MaintMarginReq - 当前维持保证金</li> <li>• NetLiquidation - 净清算值</li> </ul>
value	与标签相关的值。
currency	在值为货币类型的情况下，定义货币类型。
account	说明信息应用的账户。适用于金融顾问子账户信息。

### updatePortfolio()

这个方法仅在已经调用EClientSocket对象的reqAccountUpdates()方法时调用。

**void updatePortfolio(Contract contract, int position, double marketPrice, double marketValue, double averageCost, double unrealizedPNL, double realizedPNL, String accountName)**

参数	描述
contract	该结构包括交易合同的描述。合约的交易所区域没有为投资组合更新而设置。
position	该整数表示合约头寸。如果头寸为0，表示头寸刚被平仓。
marketPrice	产品的单位价格。
marketValue	产品的总市值。
averageCost	每股的平均成本的计算是用你头寸的数量除以你的成本( 执行价格 + 佣金) 。
unrealizedPNL	你未平仓头寸的当前市值和平均成本或平均成本值的差。
realizedPNL	显示平仓头寸的利润，为你的建仓执行成本( 执行价格 + 建仓佣金) 和平仓执行成本( 执行价格 + 平仓头寸佣金) 的差。
accountName	信息应用于的账户名称。适用于金融顾问子账户信息。

### updateAccountTime()

这个方法仅在已经调用EClientSocket对象的reqAccountUpdates()方法时调用。

**void updateAccountTime(String timeStamp)**

参数	描述
timeStamp	表示账户信息的最后更新时间。

**contractDetails()**

该方法仅当EClientSocket对象的reqContractDetails方法已经被调用时调用。

**void contractDetails(int ReqId, ContractDetails contractDetails)**

参数	描述
reqID	数据请求的代号。在处理几个请求时保证结果与请求相匹配。
contractDetails	该结构包括查询合约的详细描述。

**contractDetailsEnd()**

该方法在收到给定请求的所有合约详细时调用。以此帮助定义期权链的截止。

**void contractDetailsEnd(int reqId)**

参数	描述
reqID	数据请求的代号。

**bondContractDetails()**

该方法仅当EClientSocket对象的reqContractDetails方法对债券调用后才调用。

**void bondContractDetails(int reqId, ContractDetails contractDetails)**

参数	描述
reqId	数据请求的代号。
contractDetails	该结构包括查询债券合约的详细描述。

**execDetails()**

该方法在[reqExecutions\(\)](#)方法被启动后，或当订单被执行后调用。

**void execDetails(int reqId, Contract contract, Execution execution)**

参数	描述
reqId	为之前调用reqExecution()中指定的reqID。

contract	该结构包括已执行合约的完整描述。  <b>Note:</b> 参考 <a href="#">Java SocketClient资源</a> 页面了解更多信息。
execution	该结构包括其它的订单执行详细。

### execDetailsEnd()

该方法在针对 reqExecutions() 的已将所有的执行发送给了客户后调用。

#### void execDetailsEnd(int reqId)

参数	描述
reqID	数据请求代号。

### commissionReport()

#### void commissionReport(CommissionReport commissionReport)

参数	描述
commissionReport	包括佣金详细的结构。

### updateMktDepth()

该方法在市场深度变化时调用。

#### void updateMktDepth(int tickerId, int position, int operation, int side, double price, int size)

参数	描述
tickerId	为之前调用 reqMktDepth() 中指定的代码代号。
position	指定该市场深度输入的行代号。
operation	确定订单如何应用于市场深度。有效值为： <ul style="list-style-type: none"> <li>• 0 = 插入 ( 将这个新订单插入到标识为 'position' 的行) 。</li> <li>• 1 = 更新 ( 更新行的标识为 'position' 内的现有订单) 。</li> <li>• 2 = 删除 ( 删除行的标识为 'position' 内的现有订单) 。</li> </ul>

side	确认该定单属于的在册边。有效值是： <ul style="list-style-type: none"> <li>• 0 = ask( 卖价)</li> <li>• 1 = bid ( 买价)</li> </ul>
price	定单价格。
size	定单尺寸。

### updateMktDepthL2()

当二级市场深度变化时调用该方法。

**void updateMktDepthL2(int tickerId, int position, String marketMaker, int operation, int side, double price, int size)**

参数	描述
tickerId	为之前调用reqMktDepth()中指定的代码代号
position	指定该市场深度输入的行代号。
marketMaker	指定持有该定单的交易所。
operation	指定该定单将如何被应用于市场深度。有效值是： <ul style="list-style-type: none"> <li>• 0 = 插入( 将这个新定单插入到标识为'position'的行) 。</li> <li>• 1 = 更新( 更新行的标识为'position'内的现有定单) 。</li> <li>• 2 = 删除( 删除行的标识为'position'内的现有定单) 。</li> </ul>
side	确认该定单属于的在册边。有效值是： <ul style="list-style-type: none"> <li>• 0 = ask( 卖价)</li> <li>• 1 = bid( 买价)</li> </ul>
price	定单价格。
size	定单尺寸。

### updateNewsBulletin()

该方法针对客户订阅的每个新公告启动( 例如， 通过调用reqNewsBulletins()方法) 。

**void updateNewsBulletin(int msgId, int msgType, String message, String origExchange)**

参数	描述
msgId	每个新的公告均递增的公告代号。

msgType	指定公告类型。有效值包括： <ul style="list-style-type: none"> <li>• 1 = 普通新公告</li> <li>• 2 = 交易所不在用于交易</li> <li>• 3 = 可用于交易的交易所</li> </ul>
message	公告的信息文字。
origExchange	发出信息的交易所。

### managedAccounts()

当成功与金融顾问账户连接时调用该方法。还在启动reqManagedAccts()方法时调用该功能。

#### void managedAccounts(String accountsList)

参数	描述
accountsList	由逗号隔开的FA管理的账户列表。

### receiveFA()

该方法接收之前请求的TWS的FA配置信息。

#### receiveFA(long faDataType, string xml)

参数	描述
faDataType	指定接收自TWS的金融顾问配置数据类型。有效值包括： <ul style="list-style-type: none"> <li>• 1 = 组</li> <li>• 2 = 概括</li> <li>• 3 = 账户别名</li> </ul>
xml	XML字符串包括之前请求的FA配置信息。

### accountSummary()

相应[reqAccountSummary\(\)](#)方法返回的TWS账户窗口概况标签的数据。

#### void accountSummary(int messageType, int version, int requestId, String account, String tag, String value, String currency)

参数	类型	描述
messageType	整数	设为62。

参数	类型	描述
version	整数	设为 1。
requestId	整数	数据请求的代号。
account	字符串	账户代号。

参数	类型	描述
tag	字符串	<p>数据请求标签。 可用的标签包括：</p> <ul style="list-style-type: none"> <li>• <i>AccountType</i></li> <li>• <i>TotalCashValue</i> — 包括期货pnl的总现金</li> <li>• <i>SettledCash</i> — 对现金账户，该标签和<i>TotalCashValue</i>相同</li> <li>• <i>AccruedCash</i> — 净累计利息</li> <li>• <i>BuyingPower</i> — 账户可以买进的应用保证金的美国股票的最大额</li> <li>• <i>EquityWithLoanValue</i> — 现金 + 股票 + 债券 + 共同基金</li> <li>• <i>PreviousEquityWithLoanValue</i></li> <li>• <i>GrossPositionValue</i> — 所有股票和股票期权头寸绝对值的总合</li> <li>• <i>RegTEquity</i></li> <li>• <i>RegTMargin</i></li> <li>• <i>SMA</i> — 特殊备忘账户</li> <li>• <i>InitMarginReq</i></li> <li>• <i>MaintMarginReq</i></li> <li>• <i>AvailableFunds</i></li> <li>• <i>ExcessLiquidity</i></li> <li>• <i>Cushion</i> — 净清算值百分比的剩余流动性</li> <li>• <i>FullInitMarginReq</i></li> <li>• <i>FullMaintMarginReq</i></li> <li>• <i>FullAvailableFunds</i></li> <li>• <i>FullExcessLiquidity</i></li> <li>• <i>LookAheadNextChange</i> — 预期值生效时间</li> <li>• <i>LookAheadInitMarginReq</i></li> <li>• <i>LookAheadMaintMarginReq</i></li> <li>• <i>LookAheadAvailableFunds</i></li> <li>• <i>LookAheadExcessLiquidity</i></li> <li>• <i>HighestSeverity</i> — 衡量账户接近被清算的程度</li> <li>• <i>DayTradesRemaining</i> — 日交易模式被检测出来之前用户能够进行开仓/平仓交易的数目。值为"-1"表示用户能够进行无限制的日交易。</li> </ul>

参数	类型	描述
		<ul style="list-style-type: none"> <li><i>Leverage</i> — GrossPositionValue / NetLiquidation</li> </ul>
value	字符串	标签值。
currency	字符串	标签货币。

### accountSummaryEnd

当给定请求的所有账户概括数据被收到后调用该方法。

**void accountSummaryEnd(int reqId)**

参数	类型	描述
reqId	整数	数据请求的代号。

### position()

该事件返回相应[reqPositions\(\)](#)方法的所有账户的实时头寸数据。

**void position(int messageId, int version, String account, int conid, String underlying, String securityType, String expiry, String strike, String right, String multiplier, String exchange, String currency, String ibLocalSymbol, double position)**

参数	类型	描述
messageId	整数	设为 62。
version	整数	设为 1。
account	字符串	账户。
conid	整数	独特的合约标识符。
underlying	字符串	底层资产代码。
securityType	字符串	证券类型。
expiry	字符串	过期日。
strike	字符串	行使价。
right	字符串	看跌或看涨。
multiplier	字符串	乘数。
exchange	字符串	交易所。
currency	字符串	货币。
ibLocalSymbol	字符串	本地代码。
position	双	头寸。



### positionEnd()

当收到给定请求的所有头寸数据后调用该功能，并做为position()数据的截止标志功能。

#### void positionEnd(int reqId)

参数	类型	描述
reqId	整数	数据请求的代号。

### historicalData()

该方法接收请求的历史数据结果。

#### void historicalData (int reqId, String date, double open, double high, double low, double close, int volume, int count, double WAP, boolean hasGaps)

参数	描述
reqId	柱相对应的请求的代码代号。
date	柱开始的日期时间标记。格式由reqHistoricalData() formatDate参数决定。
open	柱开始价格。
high	柱覆盖时间区间的最高价。
low	柱覆盖时间区间的最低价。
close	柱截止价。
volume	柱覆盖时间区间的交易量。
count	当交易历史数据返回时，代表柱覆盖时间区间内发生的交易次数。
WAP	柱覆盖时间区间的加权平均价。
hasGaps	是否数据中存在空白。

### scannerParameters()

该方法接收描述扫描仪订阅可以选择的有效参数的XML文件。

#### void scannerParameters(String xml)

参数	描述
xml	描述可用的扫描仪订阅参数的文件。

### scannerData()

该方法接收请求的市场扫描仪数据结果。

**void scannerData(int reqId, int rank, ContractDetails contractDetails, String distance, String benchmark, String projection, String legsStr)**

参数	描述
reqId	行所对应的请求代号。
rank	返回柱内的排列。
contractDetails	该结构包括被执行合约的完整描述。
distance	随查询而不同。
benchmark	随查询而不同。
projection	随查询而不同。
legsStr	当扫描返回EFP时描述组合边。

**scannerDataEnd()**

该方法在收到快照和标识一个扫描的结束时调用。

**void scannerDataEnd(int reqId)**

参数	描述
reqId	通过该参数结束市场数据快照请求的代号。

**realtimeBar()**

该方法接收实时柱数据结果。

**void realtimeBar(int reqId, long time, double open, double high, double low, double close, long volume, double wap, int count)**

参数	描述
reqId	柱所对应的请求代码代号。
time	柱开始的日期时间标记。格式由reqHistoricalData() formatDate 参数确定。
open	柱开始价格。
high	柱覆盖期间的最高价。
low	柱覆盖期间的最低价。
close	柱结束价。
volume	柱覆盖期间的交易量。
wap	柱覆盖期间的加权平均价。

count	当交易历史数据返回时，代表柱覆盖期间的发生的交易次数。
-------	-----------------------------

### fundamentalData()

调用该方法接收路透社全球基本面市场数据。在你能够接收数据之前，你必须通过账户管理设置订阅路透社基本面。

#### void fundamentalData(int reqId, String data)

参数	描述
reqId	数据请求代号。
data	三种XML报告之一： <ul style="list-style-type: none"> <li>• 估计 (estimates)</li> <li>• 财务报告 (finstat)</li> <li>• 概况 (snapshot)</li> </ul>

## Java SocketClient资源文件

下列题目定义了以下 classes 的属性：

- [Execution](#)
- [ExecutionFilter](#)
- [CommissionReport](#)
- [Contract](#)
- [ContractDetails](#)
- [ComboLeg](#)
- [Order](#)
- [OrderState](#)
- [ScannerSubscription](#)
- [UnderComp](#)

### Execution

属性	描述
String m_acctNumber	客户账户号码。
double m_avgPrice	平均价。用于正常交易、组合交易和组合边。包括佣金。
int m_clientId	下达定单的客户代号。 注：TWS定单的固定客户代号是“0”。
int m_cumQty	累积数量。用于正常交易、组合交易和组合边。
String m_exchange	执行定单的交易所。
String m_execId	独特的定单执行代号。
int m_liquidation	确认在必要时最后清算的头寸。
int m_orderId	定单代号。 <b>Note:</b> TWS定单的固定定单代号是“0”。
int m_permId	用于确认定单的TWS代号在整个TWS运行期间保持不变。
double m_price	定单执行价格，不包括佣金。
int m_shares	执行的股数。

String m_side	确定交易是卖出或买进。有效值是： <ul style="list-style-type: none"> <li>• BOT ( 买进)</li> <li>• SLD ( 卖出)</li> </ul>
String m_time	订单执行时间。
String m_evRule	包含经济值规则名称和相应的可选参数。两个值应该由冒号分开。例如， aussieBond:YearsToExpiration=3. 当不存在可选参数时，第一个值之后将用冒号。
double m_evMultiplier	提示你在价格改变是1的情况下，合约的市值改变大约是多少。不可以通过用大约的乘数乘以价格得出市值。

### ExecutionFilter

属性	描述
String m_acctCode	根据账户编码过滤reqExecutions()方法结果。注：这仅与金融顾问( FA) 账户相关。
int m_clientId	根据clientId过滤reqExecutions()方法结果。
String m_exchange	根据订单交易所过滤reqExecutions()方法的结果。
String m_secType	根据订单证券类型过滤reqExecutions()方法的结果。 注：关于有效证券类型列表，参考合约结构。
String m_side	根据订单行动过滤reqExecutions()方法的结果。 注：关于有效订单行动列表，参考订单级别。
String m_symbol	根据订单代码过滤reqExecutions()方法的结果。
String m_time	根据特定时间后接收到的执行报告过滤reqExecutions()方法的结果。 timeFilter的格式是 "yyyymmdd-hh:mm:ss"

### CommissionReport

属性	描述
double m_commission()	佣金额。
String m_currency()	货币。

String m_execId()	独特定单执行代号。
double m_realizedPNL()	实现的益损额。
double m_yield()	收益率。
int m_yieldRedemptionDate()	接受YYYYMMDD格式。

**Contract**

属性	描述
Vector m_comboLegs	用于储存合约边定义的动态记忆结构。
String m_comboLegsDescrip	描述组合边
int m_conId	独特的合约标识符。
String m_currency	指定货币。可能要求该区域为不确定性，例如，当SMART(智能)是交易所和请求的是IBM(IBM可用GBP或USD交易)。考虑待这种不确定性，最好是指定货币。
String m_exchange	定单目的地，比如Smart。
String m_expiry	过期日。使用格式YYYYMM。
boolean m_includeExpired	如果设为真，合约详细请求和历史数据查询可以针对已过期合约进行。 注：对过期合约的历史数据查询被限制在合约期的最后一年，并且最初仅支持过期的期货合约。
String m_localSymbol	这是底层资产的本地交易所代码。
String m_multiplier	允许你指定一个期货或期权合约乘数。仅在存在多个可能性时需要。
String m_primaryExch	确定合约的挂牌交易所(不包括SMART)。
String m_right	指定看跌或看涨。有效值是：P, PUT, C, CALL。
String m_secId	secIdType的独特标识符。

String m_secIdType	<p>证券标识符，当查询合约详细或下达定单时。支持的标识符包括：</p> <ul style="list-style-type: none"> <li>• SIN (Example: Apple: US0378331005)</li> <li>• CUSIP ( 例如： Apple: 037833100)</li> <li>• SEDOL ( 由6位字母数字代码+校验位。例如 BAE: 0263494)</li> <li>• RIC ( 由独立于交易所的RIC根和一个确认交易所的前缀组成。例如： AAPL.O 代表在NASDAQ上交易的Apple。)</li> </ul>
String m_secType	<p>这是证券类型。有效值是：</p> <ul style="list-style-type: none"> <li>• STK</li> <li>• OPT</li> <li>• FUT</li> <li>• IND</li> <li>• FOP</li> <li>• CASH</li> <li>• BAG</li> </ul>
double m_strike	行使价。
String m_symbol	底层资产的代码。

### ContractDetails

属性	描述
String m_category	底层证券的行业类别。例如， InvestmentSvc。
String m_contractMonth	合约月。通常为期货合约底层证券的合约月。
String m_industry	底层证券/产品的行业分类。例如， Financial。
String m_liquidHours	产品的流动交易时间。例如， 20090507:0930-1600;20090508:CLOSED.
String m_longName	资产的描述名称。
String m_marketName	合约的市场名称。
double m_minTick	最小价格跳动。
String m_orderTypes	该合约的有效定单类型列表。

String m_priceMagnifier	允许执行和行使价报告和市场数据、历史数据和定单价格保持一致，例如，在LIFFE交易的Z的报告是指数点，不是GBP。
Vector<TagValue> m_secIdList()	允许客户查看的合约标识符列表( CUSIP, ISIN, 等)
String m_subcategory	底层证券的子行业类别。例如，Brokerage。
Contract m_summary	合约概况。
String m_timeZoneId	产品交易时间的时区代号。例如，EST。
String m_tradingHours	产品交易时间。例如，20090507:0700-1830,1830-2330;20090508:CLOSED。
String m_tradingClass	该合约的交易级别名称。
String m_underConId	底层合约代号。
String m_validExchanges	合约挂牌交易的交易所列表。
String m_evRule	包含经济价值规则名称和相应的可选参数。两个值应该用冒号分开。例如，aussieBond:YearsToExpiration=3。当可选的出色不存在时，在第一个值后面使用一个冒号。
double m_evMultiplier	显示如果价格的改变为1的话，合约市值的改变将大约为多少。不能够使用大约乘数乘以价格获得市值。
<b>债券值</b>	
String m_bondType	用于债券。债券类型，比如"CORP。"
boolean m_callable	用于债券。值为真或假。如果是真，一定条件下，债券发行人可以回购债券。
boolean m_convertible	用于债券。值为真或假。如果是真，一定条件下，债券可以被转换成股票。
double m_coupon	用于债券。用于计算你将在一年内收到的利息支付额的利率。
String m_couponType	用于债券。债券利率类型。
String m_cusip	用于债券。9位字符的债券CUSIP或12位的SEDOL。
String m_descAppend	用于债券。包含债券详细描述字符串。
String m_issueDate	用于债券。债券发行的日期。
String m_maturity	用于债券。债券发行人必须偿还债券面值的日期。



String m_nextOptionDate	对债券，仅在债券具有嵌入期权时。
boolean m_nextOptionPartial	对债券，仅在债券具有嵌入期权时。
boolean m_putable	用于债券。值为真或假。如果为真，一定条件下债券可以被卖回给债券发行人。
String m_ratings	用于债券。确认债券发行人的信用评级。信用评级越高，通常投资风险越低。债券评级由Moody's和S&P发布。
String m_nextOptionType	对债券，仅在债券具有嵌入期权时。
String m_notes	对债券，如果在IB数据库中填充了债券。

### ComboLeg

属性	描述
String m_action	你在创建的边的一侧( 买或卖) 。
int m_conId	指定证券的独特合约标识符。
String m_designatedLocation	如果shortSaleSlot == 2，则必须指定designatedLocation。否则留为空白或定单将被拒绝。
String m_exchange	完成的组合定单将被传递的交易所。
int m_openClose	指定定单是开放定单还是关闭定单。有效值是： <ul style="list-style-type: none"> <li>• 0 - 于母证券相同。这是零售客户的唯一选择。</li> <li>• 1 - 开放。该值仅对机构客户有效。</li> <li>• 2 - 关闭。该值仅对机构客户有效。</li> <li>• 不知道 - (3)</li> </ul>
int m_ratio	位你创建的边选择相应的合约数目。帮助确定特定组合定单的比率，参考用户指南中的交互分析部分。
int m_shortSaleSlot	仅用于机构客户。 <ul style="list-style-type: none"> <li>• 0 - 不适用( 例如，零售客户或非卖空边)</li> <li>• 1 - 清算经纪商</li> <li>• 2 - 第三方。如果使用这个值，你必须输入一个指定的地点。</li> </ul>

### OrderComboLeg

属性	描述
double m_price	定单特定边价格。

### Order

属性	描述
<b>定单标识符</b>	
int m_clientId	下达该定单的客户代号。
int m_orderId	该定单代号。
int m_permid	用于确认定单的TWS代号，TWS运行期间保持不变。
<b>主要定单区域</b>	
String m_action	确认边。有效值是：BUY、ELL、SSHORT
double m_auxPrice	这是止损限价单的止损价，和相对定单的抵消额。所有其它情形，指定位零。
double m_lmtPrice	这是用于限价单、止损限价单和相对定单的限价。所有其它情形指定位零。对没有限价的相对定单，也指定为零。
String m_orderType	确认定单类型。  关于支持定单类型的更多信息，请参阅 <a href="#">支持的定单类型</a> 。
long m_totalQuantity	定单数量。
<b>扩展定单区域</b>	
boolean m_allOrNone	0 = no, 1 = yes
boolean m_blockOrder	如果设为真，指定定单为ISE批量定单。
int m_displaySize	公开披露定单尺寸，当下达Iceberg定单时使用。
String m_goodAfterTime	交易“Good After Time( 到时有有效) ”的格式是“YYYYMMDD hh:mm:ss ( 可选时区) “ 如果不适用，使用一个空白字符串。
String m_goodTillDate	使用这个字符串，你必须输入GTD做为有效时间。交易“Good Till Date( 到期前有效) ， ”的格式是“YYYYMMDD hh:mm:ss ( 可选时区) ” 如果不适用，使用一个空白字符串。

boolean hidden	如果设为真，当查看市场深度时，定单将为不可视。这个选择只可以应用于传递到ISLAND交易所的定单。
int m_minQty	确认定单类型的最低数量。
String m_ocaGroup	确认OCA(一取消全)定单组。
int m_ocaType	<p>指示当一个定单或部分定单执行时如何处理OCA定单组中余下的定单。有效值包括：</p> <ul style="list-style-type: none"> <li>• 1 = 和批量一起取消所有剩余定单</li> <li>• 2 = 和批量一起按比率减少剩余定单尺寸</li> <li>• 3 = 不按批量减少相应的剩余定单尺寸</li> </ul> <p>如果你使用值“按批量”，将给你的定单避免超额执行的保护。意味着每次仅传递组中的一个定单，以除去超额执行的可能性。</p>
String m_orderRef	定单参考。意在仅用于机构客户，尽管在多个API客户运行时，所有客户都可能会使用它来确认发送定单的API客户。
boolean m_outsideRth	如果设为真，允许定单在正常交易时段以外触发和执行。
int m_parentId	母定单的定单代号，用于括号单和自动跟踪止损单。
double m_percentOffset	相对定单的抵消额百分比。
boolean overridePercentageConstraints	<p>TWS预设页面上定义的预防约束，帮助保证你的价格和尺寸定单值是合理的。从API发送的定单也使用这些安全约束进行验证，如果违反了任何约束，定单可能会被拒绝。取代验证，设置该参数的值为真。</p> <p>有效值包括：</p> <ul style="list-style-type: none"> <li>• 0 = False(假)</li> <li>• 1 = True(真)</li> </ul>

string m_rule80A	包括的值： <ul style="list-style-type: none"> <li>• Individual = 'I'</li> <li>• Agency = 'A',</li> <li>• AgentOtherMember = 'W'</li> <li>• IndividualPTIA = 'J'</li> <li>• AgencyPTIA = 'U'</li> <li>• AgentOtherMemberPTIA = 'M'</li> <li>• IndividualPT = 'K'</li> <li>• AgencyPT = 'Y'</li> <li>• AgentOtherMemberPT = 'N'</li> </ul>
boolean m_sweepToFill	如果设为真，指定订单为一扫光订单。
String m_tif	有效时间。有效值是：DAY, GTC, IOC, GTD。
bool m_transmit	指定是否订单将由TWS发送。如果设为假，订单将在TWS创建，但不发送。
int m_triggerMethod	指定模拟止损、止损限价和跟踪止损订单是如何被触发的。有效值是： <ul style="list-style-type: none"> <li>• 0 - 默认值。对OTC( 场外) 股票和美国期权将使用“双买价/卖价”功能。所有其它订单将使用“最后价”功能。</li> <li>• 1 - 使用“双买价/卖价”功能，当止损订单的触发是基于两个连续买价或卖价时。</li> <li>• 2 - ”最后价“功能，当止损单的触发是基于最后价时。</li> <li>• 3 双最后价功能。</li> <li>• 4 买价/卖价功能。</li> <li>• 7 最后价或买价/卖价功能。</li> <li>• 8 中点价功能。</li> </ul>
double m_trailStopPrice	仅用于TRAILLIMIT( 跟踪限价) 订单

double m_trailingPercent	<p>指定跟踪止损定单的跟踪额为一个百分比。当使用trailingPercent(跟踪百分比)区域时,遵循下列原则:</p> <ul style="list-style-type: none"> <li>• 这个区域和现有的跟踪额是相互排斥的。也就是,API客户可以发送其中之一,但不能两个都发送。</li> <li>• 这个区域是排在止损价(屏障价格)之后的: deltaNeutralAuxPrice stopPrice trailingPercent scale order attributes(分段定单属性)</li> <li>• 如果API客户的版本是&gt;= 56,该区域也将以openOrder信息的形式发送到API。是在stopPrice区域之后发送的: stopPrice trailingPct basisPoint</li> </ul>
<b>金融顾问区域</b>	
String m_faGroup	交易将被分配到的金融顾问组 -- 不适用时,使用一个空的字符串。
String m_faMethod	金融顾问分配功能将分配交易和 -- 不适用时,使用一个空的字符串。
String m_faPercentage	金融顾问交易分配的百分比 -- 不适用时,使用一个空的字符串。
String m_faProfile	金融顾问分配交易的分配模式 -- 不适用时,使用一个空的字符串。
<b>仅机构(非清算的)</b>	
String m_designatedLocation	仅当shortSaleSlot = 2时使用。
String m_openClose	仅用于机构客户。有效值是O, C。
int m_origin	定单来源。仅用于机构客户。有效值是0 = 客户, 1 = 公司
int m_shortSaleSlot	有效值是1或2。
<b>仅智能传递</b>	
double m_discretionaryAmt	允许的偏离全权委托单限价价格的数额。
boolean m_eTradeOnly	用电子报价交易。 0 = no, 1 = yes

boolean m_firmQuoteOnly	用公司报价交易。 0 = no, 1 = yes
double m_nbboPriceCap	偏离NBBO的最大智能定单。
boolean m_optOutSmartRouting	对直接传递到ASX的定单使用默认退出。除非明确设置为真，否则该属性默认为假。当设为假时，直接传递到ASX的定单将不使用SmartRouting(智能传递)。当设为真，直接传递到ASX的定单将使用SmartRouting。
<b>仅盒顶(BOX)或波动率(VOL)定单</b>	
int m_auctionStrategy	包括的值： <ul style="list-style-type: none"><li>• match(匹配) = 1</li><li>• improvement(改善) = 2</li><li>• transparent(透明) = 3</li></ul> 仅BOX定单。
<b>仅BOX交易所定单</b>	
double m_delta	股票delta。仅BOX定单。
double m_startingPrice	竞价起始价格。仅BOX定单。
double m_stockRefPrice	股票参考价格。参考价格用于波动率定单，计算发送至交易所的限价价格(不管是否选择了连续更新与否)，和价格范围监控。
<b>仅挂钩股票和波动率定单</b>	
double m_stockRangeLower	可接受的底层股票价格范围的低端值。用于BOX和VOL定单的价格过时期权定单和动态管理。
double m_stockRangeUpper	可接受的底层股票价格范围的高端值。用于BOX和VOL定单的价格过时期权定单和动态管理。
<b>仅波动率定单</b>	
boolean m_continuousUpdate	仅VOL(波动率)定单。指定是否TWS将再底层价格变动时自动更新定单的限价价格。
String m_deltaNeutralOrderType	仅VOL(波动率)定单。输入定单类型指示TWS对全部或部分执行的VOL定单发送一个delta中性交易。对没有发送对冲delta定单的情况，指定为NONE(无)。
int m_deltaNeutralAuxPrice	仅VOL(波动率)定单。使用该区域输入一个值，如果deltaNeutralOrderType区域中的值是需要辅助价格的定单类型，比如REL(相对)定单。

int m_referencePriceType	<p>仅VOL(波动率)定单。指定你希望如何TWS来为期权和股票范围价格监控计算限价价格。</p> <p>有效值包括：</p> <ul style="list-style-type: none"> <li>• 1 = 平均NBBO</li> <li>• 2 = 根据行动和权限，为NBB或NBO。</li> </ul>
double m_volatility	<p>波动率的期权价格，由TWS的期权分析来计算。该值的表示为百分比形式，并被用于计算发送到交易所的限价价格。</p>
int m_volatilityType	<p>包括的值：</p> <ul style="list-style-type: none"> <li>• 1 = 日波动率</li> <li>• 2 = 年波动率</li> </ul>
String m_deltaNeutralOpenClose	<p>指定定单是开发还是关闭定单，是否在对冲关系到CFD是使用，是否定单为远离结算。</p>
boolean m_deltaNeutralShortSale	<p>当对冲关系到股票和指数时使用，不论是否为卖空。</p>
int m_deltaNeutralShortSaleSlot	<p>值为1( 结算经纪商持有股票) 或值为2( 由第三方提交) 。如果你使用2，则你必须指定一个deltaNeutralDesignatedLocation。</p>
String m_deltaNeutralDesignatedLocation	<p>仅当deltaNeutralShortSaleSlot = 2时使用。</p>
<b>仅组合定单</b>	
double m_basisPoints	<p>仅EFP定单</p>
int m_basisPointsType	<p>仅EFP( 期转现) 定单</p>
<b>仅分段定单</b>	
boolean m_scaleAutoReset()	<p>用于扩展分段定单。</p>
int m_scaleInitFillQty()	<p>用于扩展分段定单。</p>
int m_scaleInitLevelSize	<p>对分段定单：定义第一个尺寸，或初始尺寸，定单分组。</p>
int m_scaleInitPosition()	<p>用于扩展分段定单。</p>

int m_scalePriceAdjustInterval()	用于扩展分段定单。
double m_scalePriceAdjustValue()	用于扩展分段定单。
double m_scalePriceIncrement	对分段定单：定义分组之间的价格增量。这是必填区域。
double m_scaleProfitOffset()	用于扩展分段定单。
boolean m_scaleRandomPercent()	用于扩展分段定单。
int m_scaleSubsLevelSize	对分段定单：定义后续分段定单组的定单尺寸。和scaleInitLevelSize()一起使用。
<b>仅对冲定单</b>	
String m_hedgeParam	Beta = x用于Beta对冲定单，比率 = y用于成双对冲定单
String m_hedgeType	用于对冲定单。可能的值是： <ul style="list-style-type: none"> <li>• D = Delta</li> <li>• B = Beta</li> <li>• F = FX</li> <li>• P = Pair</li> </ul>
<b>结算信息</b>	
String m_account	账户。仅用于机构客户。
String m_clearingAccount	对IBExecution客户：指定定单的真实受益人。对FUT/FOP定单为要求的区域，用于报告给交易所。
String m_clearingIntent	对IBExecution客户：有效值是：IB，Away，和PTA（后交易分配）。
String m_settlingFirm	仅机构。
<b>仅算法定单</b>	
String m_algoStrategy	关于API算法定单，请参考 <a href="#">IBAlgo参数</a> 。
Vector<TagVlue> m_algoParams	支持IBAlgo参数。
<b>假设</b>	



boolean m_whatIf	用于请求交易前佣金和保证金信息。 如果设为真，保证金和佣金数据将通过openOrder()回调的OrderState()对象接收反馈。
<b>智能组合传递</b>	
Vector<TagValue> m_smartComboRoutingParams	支持 <a href="#">智能组合传递</a> 。
<b>定单组合边</b>	
OrderComboLegs() As Object	持有一个组合定单中所有边的属性。
<b>不保持</b>	
boolean m_notHeld	仅IBDARK定单。传递到IBDARK的定单标签为“仅公布”并在IB的定单册中持有，从其它IB客户传入SmartRouted(智能传递)定单可以对它们进行交易。

### OrderState

属性	描述
double m_commission	显示定单的佣金额。
String m_commissionCurrency	显示佣金值的货币。
String m_equityWithLoan	显示定单将对你的含借贷值股权的影响。
String m_initMargin	显示定单将对你的初始保证金的影响。
String m_maintMargin	显示定单将对你的维持保证金的影响。
double m_maxCommission	和minCommission区域一起使用，用于定义实际定单佣金将落入的可能范围的最高端。
double m_minCommission	和maxCommission区域一起使用，用于定义实际定单佣金将落入的可能范围的最低端。
string m_status	显示定单状态。
String m_warningText	必要时显示警告信息。

### ScannerSubscription

属性	描述
double m_abovePrice	过滤掉价格低于这个值的合约。可以为空白。
int m_aboveVolume	过滤掉交易量低于这个值的合约。可以为空白。
int m_averageOptionVolumeAbove	可以为空白。
double m_belowPrice	过滤掉价格高于这个值的合约。可以为空白。
double m_couponRateAbove	过滤掉利率低于这个值的合约。可以为空白。
double m_couponRateBelow	过滤掉利率高于这个值的合约。可以为空白。
String m_excludeConvertible	过滤掉可转换债券。可以为空白。
String m_instrument	定义扫描的产品类型。
String m_locationCode	地点。
String m_maturityDateAbove	过滤掉到期日早于这个值的合约。可以为空白。
String m_maturityDateBelow	过滤掉到期日迟于这个值的合约。可以为空白。
double m_marketCapAbove	过滤掉市值低于这个值的合约。可以为空白。
double m_marketCapBelow	过滤掉市值高于这个值的合约。可以为空白。
String m_moodyRatingAbove	过滤掉穆迪评级低于这个值的合约。可以为空白。
String m_moodyRatingBelow	过滤掉穆迪评级高于这个值的合约。可以为空白。
int m_numberOfRows	定义查询返回数据行的数目。
String m_scanCode	可以为空白。
String m_scannerSettingPairs	可以为空白。例如，一对使用在“隐含波动率%最大涨幅”扫描的“年，真”将返回年度波动率。
String m_spRatingAbove	过滤掉标普评级低于这个值的合约。可以为空白。
String m_spRatingBelow	过滤掉标普评级高于这个值的合约。可以为空白。

String m_stockTypeFilter	<p>有效值为：</p> <ul style="list-style-type: none"> <li>• CORP = 公司</li> <li>• ADR = 美国存托凭证</li> <li>• ETF = 交易所交易基金</li> <li>• REIT = 房地产投资信托</li> <li>• CEF = 封闭式基金</li> </ul>
--------------------------	--

### UnderComp

属性	描述
int m_conId	指定证券的独特合约标识符。用于Delta-中性组合合约。
double m_delta	底层股票或期货 delta。用于Delta-中性组合合约。
double m_price	底层价格。用于Delta-中性组合合约。

## 下达组合定单

组合定单是一种特殊类型的定单，由多个不同的边组成，但以一个单一交易执行。提交组合定单，比如日历差价、转换和跨式使用BAG证券类型(在Contract对象中定义)。使用API成功的实施API组合定单的关键是要知道任何使用交易平台下达相同的定单。如果你熟悉在TWS中下达组合定单，则比较容易使用API下达相同的定单，因为API仅模仿TWS的操作。

### 举例

此例中，客户下达一个对GOOG的日历差价买单。买进一个日历差价表示：

**边 1: Sell 1 GOOG OPT SEP 18 '09 150.0 CALL (100)**

**边 2: Buy 1 GOOG OPT JAN 21 '11 150.0 CALL (100)**

以下为使用API下达组合定单的步骤要求概括：

- 获取每一边的合约代号( conId) 。通过启动reqContractDetails()方法获取。
- 通过填充相应的区域在ComboLeg对象中包括每个边。
- 实施placeOrder()方法和合约和定单插口客户资源。

### 下达这个组合定单

1. 获取两边定义的合约代号：

```
//First leg

Contract con1 = new Contract();

con1.m_symbol = "GOOG";
con1.m_secType = "OPT";
con1.m_expiry = "200909";
con1.m_strike = 150.0
con1.m_right = "C"
con1.m_multiplier = "100"
con1.m_exchange = "SMART";
con1.m_currency = "USD";

.reqContractDetails(1, con1);

//Second leg

Contract con2 = new Contract();

con2.m_symbol = "GOOG";
con2.m_secType = "OPT";
con2.m_expiry = "201101";
con2.m_strike = 150.0
con2.m_right = "C"
con2.m_multiplier = "100"
con2.m_exchange = "SMART";
con2.m_currency = "USD";
```

```

.reqContractDetails(2, con2);

//All conId numbers are delivered by the ContractDetail()

static public String contractDetails(int reqId, ContractDetails
contractDetails) {

Contract contract = contractDetails.m_summary;

/*Base on the request above,
reqId = 1 is corresponding to the first request or first leg
reqId = 2 is corresponding to the second request or second leg*/

if (reqId == 1)
{ Leg1_conId = contract.m_conId;} // to obtain conId for first leg

if (reqId == 2)
{ Leg2_conId = contract.m_conId;} // to obtain conId for second leg
}

```

- 一旦应用程序获取了每一边的conId值，则将其包括在ComboLeg对象中：

```

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

leg1.m_conId = Leg1_conId;
leg1.m_ratio = 1;
leg1.m_action = "SELL";
leg1.m_exchange = "SMART";
leg1.m_openClose = 0;
leg1.m_shortSaleSlot = 0;
leg1.m_designatedLocation = "";

leg2.m_conId = Leg2_conId;
leg2.m_ratio = 1;
leg2.m_action = "BUY";
leg2.m_exchange = "SMART";
leg2.m_openClose = 0;
leg2.m_shortSaleSlot = 0;
leg2.m_designatedLocation = "";

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

```

- 启动placeOrder()方法和相应的合约和定单对象：

```

Contract contract = new Contract();
Order order = new Order();

contract.m_symbol = "USD"; // For combo order use "USD" as the symbol value
all the time
contract.m_secType = "BAG"; // BAG is the security type for COMBO order

```

```
contract.m_exchange = "SMART";
contract.m_currency = "USD";
contract.m_comboLegs = addAllLegs; //including combo order in contract object

order.m_action = "BUY";
order.m_totalQuantity = 1;
order.m_orderType = "MKT"
.placeOrder(OrderId, contract, order);
```

**Note:** 有关组合定单的更多信息，请参考TWS用户指南中的[关于组合定单](#)题目。

## Java编码样本：合约参数

这一章节包括以下Java编码样本：

- [如何确定期权合约](#)
- [如何确定期货合约](#)
- [如何确定股票](#)

### 如何确定期权合约

#### 例 1 - 确定期权合约的标准方法

```
void onHowToDetermineOption(){

    Contract contract = new Contract();
    Order order = new Order();

    contract.m_symbol = "IBKR";
    contract.m_secType = "OPT";
    contract.m_expiry = "20120316";
    contract.m_strike = 20.0;
    contract.m_right = "P";
    contract.m_multiplier = "100";
    contract.m_exchange = "SMART";
    contract.m_currency = "USD";

    order.m_action = "BUY";
    order.m_totalQuantity = 1;
    order.m_orderType = "LMT";
    order.m_lmtPrice = enteredLmtPrice;

    m_client.placeOrder(GlobalOrderId, contract, order);
}
```

#### 例 2 - 使用 **OCC**( 期权结算公司 ) 期权符号项目确定期权合约

```
void inUsingOptionSymbologyInitiative(){

    Contract contract = new Contract();
    Order order = new Order();

    contract.m_localSymbol = "IBKR 120317P00020000"; //OSI
        contract.m_secType = "OPT";
    contract.m_exchange = "SMART";
    contract.m_currency = "USD";

    order.m_action = "BUY";
    order.m_totalQuantity = 1;
    order.m_orderType = "LMT";
    order.m_lmtPrice = enteredLmtPrice;
}
```

```
m_client.placeOrder(GlobalOrderId, contract, order);  
}
```

## 如何确定期货合约

### 例 1 - 确定期货合约的标准方法

```
void onHowtoDetermineFuture(){  
  
    Contract contract = new Contract();  
    Order order = new Order();  
  
    contract.m_symbol = "ES";  
    contract.m_secType = "FUT";  
    contract.m_expiry = "201109";  
    contract.m_exchange = "GLOBEX";  
    contract.m_currency = "USD";  
  
    order.m_action = "BUY";  
    order.m_totalQuantity = 1;  
    order.m_orderType = "LMT";  
    order.m_lmtPrice = enteredLmtPrice;  
  
    m_client.placeOrder(GlobalOrderId, contract, order);  
}
```

### 例 2 - 使用本地代码确定期货合约

```
void inUsingLocalSymbolForFuture(){  
  
    Contract contract = new Contract();  
    Order order = new Order();  
  
    contract.m_localSymbol = "ESU1";  
    contract.m_secType = "FUT";  
    contract.m_exchange = "GLOBEX";  
    contract.m_currency = "USD";  
  
    order.m_action = "BUY";  
    order.m_totalQuantity = 1;  
    order.m_orderType = "LMT";  
    order.m_lmtPrice = enteredLmtPrice;  
  
    m_client.placeOrder(GlobalOrderId, contract, order);  
}
```

## 如何确定股票

```
void onHowToDetermineStock(){
```



```
Contract contract = new Contract();
Order order = new Order();

contract.m_symbol = "IBKR";
contract.m_secType = "STK";
contract.m_exchange = "SMART";
contract.m_currency = "USD";

order.m_action = "BUY";
order.m_totalQuantity = 100;
order.m_orderType = "LMT";
order.m_lmtPrice = enteredLmtPrice;

m_client.placeOrder(GlobalOrderId, contract, order);
}
```



## 顾问

这一章介绍对拥有金融顾问账户用户的API功能，包括下列题目：

- [金融顾问定单和账户配置](#)
- [支持Excel DDE](#)
- [其它API技术的支持](#)
- [改进的金融顾问执行报告](#)
- [账户组的分配方法](#)
- [金融顾问API定单的Java编码样本](#)

## 金融顾问定单和账户配置

这一节假设用户已经熟悉使用TWS金融顾问账户配置和下达定单。

自从TWS版本821发行后，API的FA功能变得越来越强大，现已放弃的“分配字符串”方法已经被更强大的金融顾问定单分配方法而取代。在这些新方法被启用之前，必须要配置TWS以了解所需的FA定单组，概况和账户别名。现在可以手动通过TWS，或API，或两者进行。

## 支持Excel DDE

从TWS版本824发行开始，DDE定单使用六个扩展定单属性：*Good After Time*( 到时有有效) 、*Good Till Date*( 到时有有效) 、*FA Group*( 金融顾问组) 、*FA Method*( 金融顾问方法) 、*FA Percentage*( 金融顾问百分比) 、和*FA Profile*( 金融顾问概况) 。如果这些属性都不适用于定单的话，可留为空白。对DDE进入，TWS金融顾问账户配置应该使用手动配置。

你可以通过最近发布的Excel API表格的TwsDde.xls DDE的顾问页面上下达FA定单。有关详细信息，请参阅[顾问页面](#)标题。

## 其它API技术的支持

对所有基于ActiveX、Java、或C++的API技术，TWS金融顾问的账户配置是通过两个新方法和一个新事件完成的。方法被称作*replaceFA*，和*requestFA*。事件被叫做*receiveFA*。这些方法和事件属于TWS金融顾问账户配置的以下三部分：创建组、概况、和账户别名。

- **requestFA(int faDataType)** 是由API应用程序调用来请求FA配置数据类型之一的方法。
- **receiveFA(int faDataType, string XML)** 接收来自TWS的请求的数据(通过TWS发送的包括请求数据的事件)。事件包括含有请求信息的XML字符串。
- **replaceFA(int faDataType, string XML)** 可以从API调用，如果API应用程序希望使用一个新的字符串取代之前的FA配置信息。

根据这个新的功能，所有的placeOrder方法，不论是基于ActiveX、Java、或C++，均拥有四个新的属于金融顾问定单下达的参数：faGroup、faMethod、faPercentage、和faProfile。当其中之一或多个新值与定单无关时，只需简单地传递一个空字符串。

## 改进的金融顾问执行报告

当使用TWS版本823和更新的版本时，新的FA定单产生的执行信息将报告包括定单的初始执行，以及分配到不同子账户的信息。下面例子将解释新的报告方法：

- FA账户：通过NTSE买进100股IBM的定单被执行
- FA账户：从FA账户中拿出34股分配给子账户1。信息显示"BUY -34 IBM." ( 买 - 34IBM) 。 负数数量反映的事实是执行报告为减少买进。
- SUB1 Account( 子账户1) : 买34股IBM。
- FA账户：从FA账户中分配33股到子账户2。信息显示"BUY -33 IBM."( 买 - 33IBM) 。
- SUB2 Account( 子账户2) : 买33股IBM。
- FA账户：从FA账户中分配33股到子账户3中。信息显示"BUY -33 IBM."( 买 - 33IBM) 。
- SUB3 Account( 子账户3) : 买33股IBM。

假设100股IBM是在NYSE上由具有三个子账户的金融顾问购得，并且希望每个账户得到均等的数量。将产生以上七个执行信息。

## 账户组的分配方法

注意你输入的方法名称必须与这里给出的绝对一致，否则你的定单不会工作。

### EqualQuantity Method

要求你指定定单尺寸。该方法在组中的所有账户之间平均分配股票。

举例：你发送一个400股的ABC股票定单。如果你的账户组包含四个账户，每个账户收到100股。如果你的账户包括六个账户，每个账户将收到66股，然后每个账户分配1股，知道分完。

### NetLiq Method

要求你指定定单尺寸。该方法基于每个账户的净清算值分配股票。系统基于每个账户的净清算值计算比率并根据这些比率分配股票。

举例：你发送了一个700股的XYZ定单。账户组包括三个账户，A、B和C以及相应的净清算值\$25,000、\$50,000和\$100,000。系统计算的比率为1:2:4并分配100股给客户A，200股给客户B，和400股给客户C。

### AvailableEquity Method

要求你指定定单尺寸。该方法根据当前每个账户中可用的含借贷值股权数额分配股票。系统根据每个账户的含借贷值股权计算比率并根据这些比率分配股票。

举例：你发送了一个700股的XYZ定单。账户组包括三个账户，A、B和C以及相应的可用股权数额为\$25,000、\$50,000和\$100,000。系统计算的比率为1:2:4并分配100股给客户A，200股给客户B，和400股给客户C。

### PctChange Method

该方法仅当你对选取的产品已经持有头寸的情况下工作。不需要指定定单尺寸。因为数量是由系统计算的，在定单被确认之后定单尺寸被显示在数量区域。该方法增加或减少现有的头寸。正百分比将增加头寸，负百分比将减少头寸。

**例1：**假设组中六个账户中的三个持有XYZ股票的做多头寸。客户A有100股，客户B有400股，客户C有200股。你希望增加他们的持有量的50%，所以你在百分比区域输入“50”。系统计算你的定单尺寸需要等于350股。然后分配50股给客户A，200股给客户B，和100股给客户C。

**例2：**你希望平仓组中五个账户中的三个的所有做多头寸。你创建一个卖单和在百分比区域输入“-100”。系统计算组中持有头寸的每个账户的每个头寸的100%，和卖掉所有股票来平仓头寸。

通过这里显示的图表可方便地查看负和正百分比值将如何影响买单和卖单的做多和做空头寸。所有结果，一目了然。

买单	正百分比	负百分比
做多头寸	增加头寸	没有影响
做空头寸	没有影响	减少头寸



卖单	正百分比	负百分比
做多头寸	没有影响	减少头寸
做空头寸	增加头寸	没有影响

## 金融顾问API定单的Java编码样本

在API中从金融顾问(FA)账户下达定单通常有三种方法:

- [为单一管理的账户下达定单。](#)
- [为一个分配方案下达定单。](#)
- [为一个账户组下达定单。](#)

### 为单一管理的账户下达定单

做为FA,你可以为你管理的任何账户下达定单。下面的编码样本将完成这个任务。

```
Contract m_contract = new Contract();

Order m_order = new Order();

/** Stocks */

m_contract.m_symbol = "IBM";

m_contract.m_secType = "STK";

m_contract.m_exchange = "SMART";

m_contract.m_currency = "USD";

m_order.m_orderType = "MKT";

m_order.m_action = "BUY";

m_order.m_totalQuantity = 100;

m_order.m_transmit = true;

// allocate the order for this particular account
```

```
m_order.m_account = "DU74649";

m_client.placeOrder(orderId++, m_contract, m_order);
```

### 为一个分配方案下达定单

做为FA，你可以为分享一个分配方案的那些账户下达定单。下面的编码样本将执行这个任务。

**Note:** 在你自己尝试这个运作之前，你必须熟悉在TWS中设置分配方案和下达定单。

```
Contract m_contract = new Contract();

Order m_order = new Order();

.

.

// allocate the order for this profile

m_order.m_faProfile = "USClients";

m_client.placeOrder(orderId++, m_contract, m_order);
```

### 为账户组下达定单

做为FA，你可以为账户组中的账户下定单。注意，方法属性为账户组下定单时的要求区域。下面的编码样本将完成这个任务。

**Note:** 在你自己尝试这个运作之前，你必须熟悉在TWS中设置账户组和下达定单。

```
Contract m_contract = new Contract();

Order m_order = new Order();

.
```

```
.  
  
// allocate the order for this group  
  
m_order.m_faGroup = "USGroup";  
  
// using the percent change method  
  
m_order.m_faMethod = "PctChange";  
  
m_order.m_faPercentage = "100";  
  
m_client.placeOrder(orderId++, m_contract, m_order);
```

### 改变/更新分配信息

做为FA，你可以用XML格式检索分配信息，和通过将XML格式的配置传递返回来改变或更新分配信息。

下面的编码样本改变组中的一个名称，通过用"MyTestGroup"替代配置文件中第一个出现的"TestGroup"并传递返回文件。

```
public void receiveFA(int faDataType, String xml) {  
  
    switch (faDataType) {  
  
        case EClientSocket.GROUPS:  
  
            faGroupXML = xml ;  
  
            String test = xml.replaceFirst("TestGroup", "MyTestGroup");  
  
            m_client.replaceFA(1, test);  
  
            break ;  
  
        case EClientSocket.PROFILES:
```

```
        faProfilesXML = xml ;

        break ;

    case EClientSocket.ALIASES:

        faAliasesXML = xml ;

        break ;

}
```



## ActiveX for Excel( Excel的ActiveX)

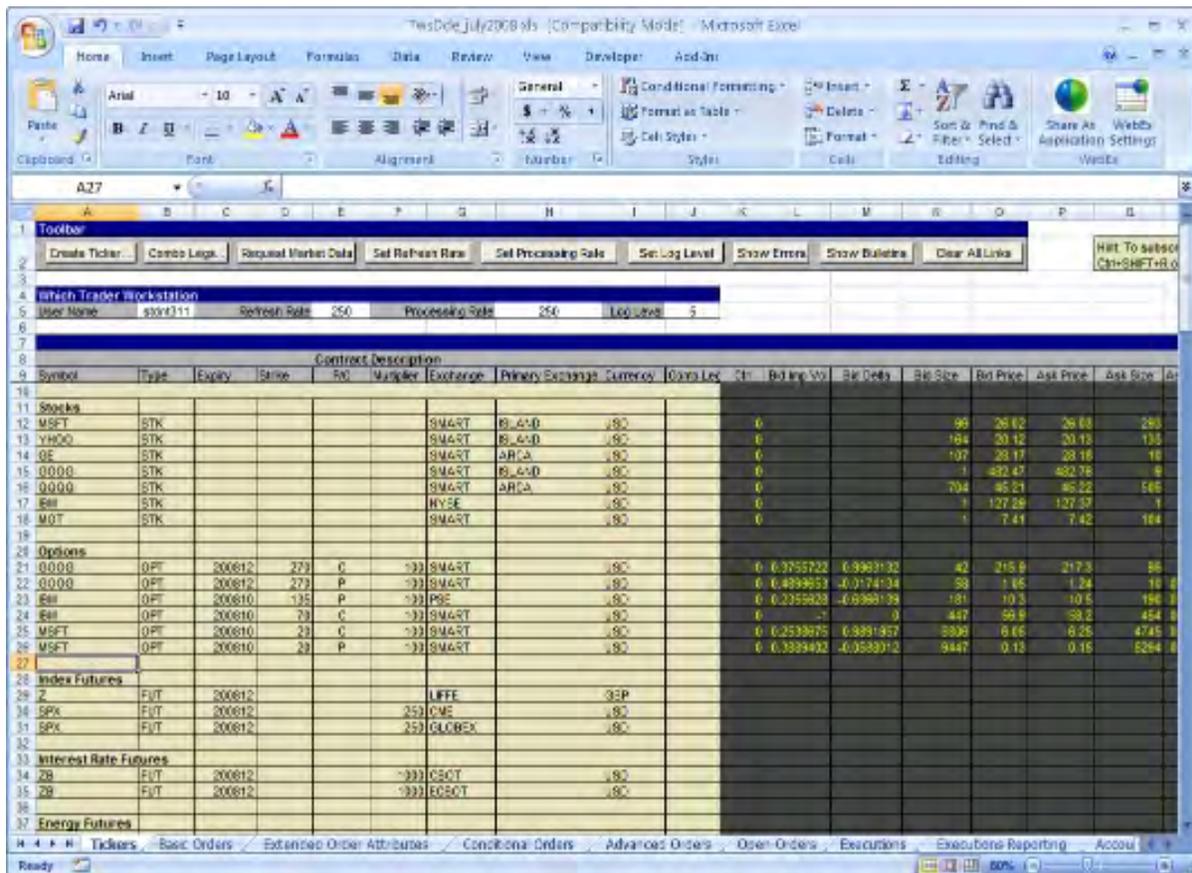
这一章描述ActiveX for Excel样本表格，包括以下题目：

- [开始使用ActiveX for Excel API](#)
- [使用ActiveX for Excel样本表格](#)

ActiveX for Excel样本表格， **TwsActiveX.xls**，复制ActiveX for Excel API表格的功能，但在内部使用ActiveX组件， **Tws.ocx**。使用该表格的益处之一是它可以连接一个在远端运行的TWS或IB网关运行。DDE for Excel API电子表格不能做到这一点。

**Note:** ActiveX for Excel样本电子表格中使用的方法、事件和COM对象是与ActiveX API中使用的一样。查看[ActiveX](#)章节了解关于ActiveX API的完整详细信息。

下图显示的是ActiveX for Excel API样本表格中的代码页面。



## 开始使用ActiveX for Excel API

我们已经创建了一个样本Excel表格，**TwsActiveX.xls**，该表格使用ActiveX控制，**Tws.ocx**。你可以直接使用这个表格和TWS，或用它创建你自己的客户化的TWS API Excel应用程序。非常容易开始：

- [下载API组件](#)，包括ActiveX for Excel样本表格，TwsActiveX.xls。
- 保证：
  - 应用程序服务器处于运行中和配置可以支持ActiveX，或
  - IB网关处于运行中。
- [打开表格](#)并开始使用ActiveX for Excel API。

样本表格目前由几个页面组成，包括了样本数据和行动按钮让你可以方便的获取市场数据，发送定单和查看你的活动。

### 下载API组件和表格

我们建议使用我们提供的样本Excel表格做为开始创建你自己Excel API的ActiveX的起点。遵循以下步骤下载样本表格。

### 安装Excel的ActiveX样本表格

1. 在[IB网站主页](#)，从**交易**菜单中选择API方案。
2. 点击**IB API**按钮，然后在API软件页面上找出适合你操作系统的栏并点击下载最新版本。  
PC用户可以使用**Windows系统Beta**栏下载API的beta测试版本，或通过选择下载前一个版本恢复到前一个正式版本。
3. 在你的计算机中保存安装程序，如果需要，可选择不同的目录。点击**保存**。
4. 关闭你正在运行的任何TWS和Excel版本。
5. 找出你刚刚在计算机中保存的安装程序，然后双击文件开始API安装。
6. 跟随安装向导指令。

**Note:** 在你[使用表格](#)之前，你必须使TWS处于运行之中,并将其配置为支持ActiveX API。请参阅[通过TWS运行API](#)了解详细信息。

### 在64位Windows XP系统下运行ActiveX for Excel API

在64位Windows XP系统下运行ActiveX for Excel API，遵循以下步骤：

1. 安装Microsoft Visual C++ 2005 SP1再发行组件包(x86)。
2. 安装Microsoft Visual J# 2.0再发行组件包。
3. 下载和安装API软件。



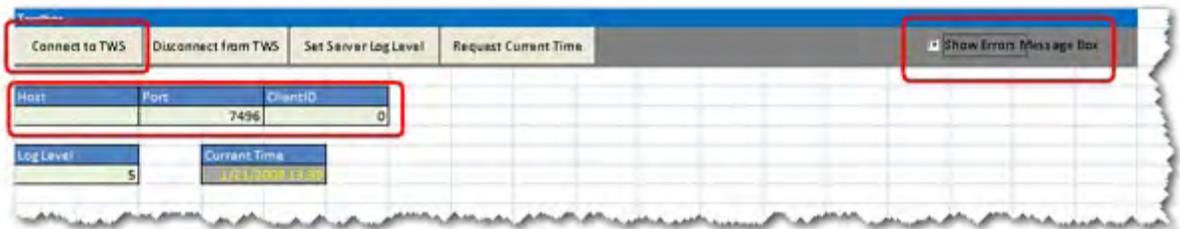
### 打开样本表格

在你下载了样本表格和配置了应用程序以允许ActiveX for Excel API与其连接之后，打开表格并保存为你的个人文件。

**Note:** 不可以有一个以上的API应用程序能够同时进入一个单一实例。API应用程序不需要在运行应用程序的同一台计算机上运行。

### 打开样本表格

1. 打开安装Excel API样本表格的API安装文件夹(通常为C:\Jts\Excel)并双击**TwsActiveX.xls**。
2. 使用不同的文件名保存表格。这样你可以在没有改变原始表格情况下自定义你的表格。
3. 点击**General**标签。
4. 修改*Host*、*Port*和*ClientID*单元中的默认值，然后点击工具栏上的**Connect to TWS(连接TWS)**。
  - 如果你选择**Show Errors Message Box(显示错误信息框)**复选框，当你连接TWS时将显示错误信息。在这种情况下，你必须点击**OK**来取消出现的任何信息。



## 使用ActiveX for Excel样本表格

ActiveX for Excel API样本表格，TwsDde.xls，包括下列页面( 标签)：

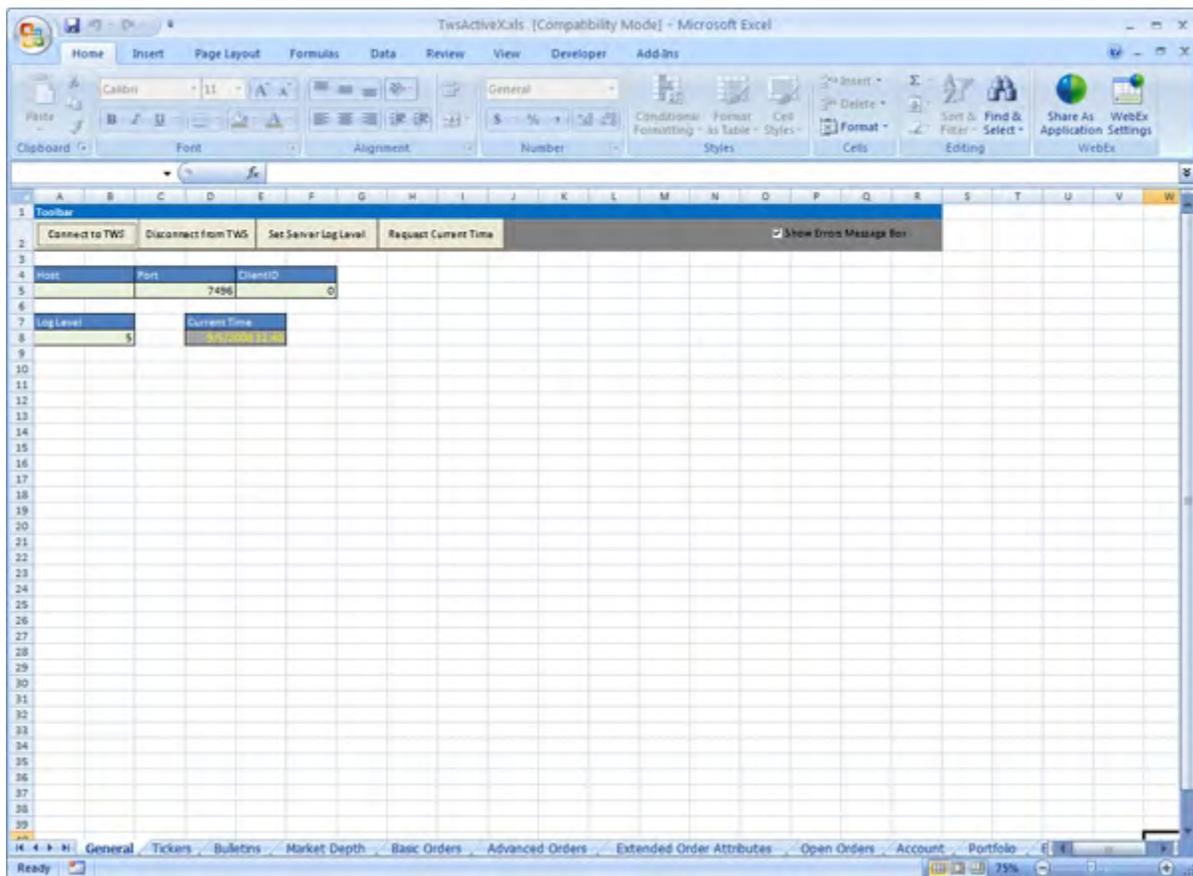
页面	描述
<a href="#"><u>General( 一般)</u></a>	让你和TWS连接和取消连接，设置服务器记录级别和请求当前时间。
<a href="#"><u>Tickers( 代码)</u></a>	让你设置你的代码行和请求市场数据。你可以查看所有资产类型的市场数据，包括EFPs和组合定单。
<a href="#"><u>Bulletins( 公告)</u></a>	让你订阅和查看IB新闻公告。
<a href="#"><u>Market Depth( 市场深度)</u></a>	让你查看选取报价的市场深度。
<a href="#"><u>Basic Orders( 基本定单)</u></a>	让你发送和修改定单、设置组合定单和EFPs、和请求开放定单。
<a href="#"><u>Conditional Orders( 条件定单)</u></a>	让你创建定单的提交是在满足其它条件下的定单，比如一个基于先前执行的定单。
<a href="#"><u>Advanced Orders( 高级定单)</u></a>	让你发送和修改需要使用扩展定单属性的高级定单类型，比如括号单、分段定单和跟踪止损限价单。
<a href="#"><u>Extended Order Attributes( 扩展定单属性)</u></a>	和基本定单、高级定单、条件定单和顾问页面一起使用，你可以使用这个页面改变有效时间、创建隐含或冰山定单和应用许多其它定单属性。
<a href="#"><u>Open Orders( 打开定单)</u></a>	显示所有已发送的定单，包括已经被IB系统接收的定单，和在交易所中工作的定单。
<a href="#"><u>Account( 账户)</u></a>	提供最新的账户信息和显示你的投资组合。
<a href="#"><u>Portfolio( 投资组合)</u></a>	显示所有你当前的头寸。
<a href="#"><u>Executions( 执行)</u></a>	让你查看所有的执行报告，和包括一个你可以用来限制显示结果的过滤框。
<a href="#"><u>Historical Data( 历史数据)</u></a>	请求基于你在查询中输入的数据的产品的历史数据。
<a href="#"><u>Contract Details( 合约详细)</u></a>	让你搜集用于其它活动所需的特定合约的信息，包括conid和支持的合约的定单类型
<a href="#"><u>Bond Contract Details( 债券合约详细)</u></a>	让你搜集用于其它活动所需的特定债券合约的信息，包括债券利率和到期日。

<a href="#">Real Time Bars( 实时柱)</a>	让你请求和查看TWS的实时柱。
<a href="#">Market Scanner( 市场扫描仪)</a>	让你查看市场扫描仪参数和订阅TWS市场扫描仪。
<a href="#">Fundamentals( 基本面)</a>	让你请求和查看来自TWS的基本面数据。
<a href="#">Advisors( 顾问)</a>	让金融顾问发送和修改FA定单。
<a href="#">Log( 记录)</a>	让你查看所有的错误信息。

一般页面

使用一般页面完成：

- 连接TWS。
- 取消连接TWS。
- 设置服务器处理API请求时的记录输入详细级别。
- 请求当前服务器时间。



### 连接TWS

1. 点击工具栏上的 **Connect to TWS( 连接TWS)** 。
  - 如果需要，改变 *Host*、*Port*和 *ClientID*单元中的值。
  - 选择 **Show Errors Message Box( 显示错误信息框)** 在连接TWS时显示出错信息。



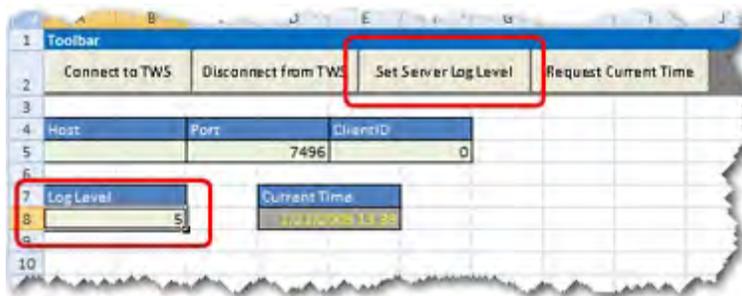
### 取消连接TWS

1. 点击工具栏中的 **Disconnect from TWS( 取消连接TWS)** 。

### 设置服务器记录级别

1. 在 *Log Level( 记录级别)* 单元中输入下列值之一：
  - 1 = SYSTEM( 系统)
  - 2 = ERROR( 错误)
  - 3 = WARNING( 警告)
  - 4 = INFORMATION( 信息)
  - 5 = DETAIL( 详细)

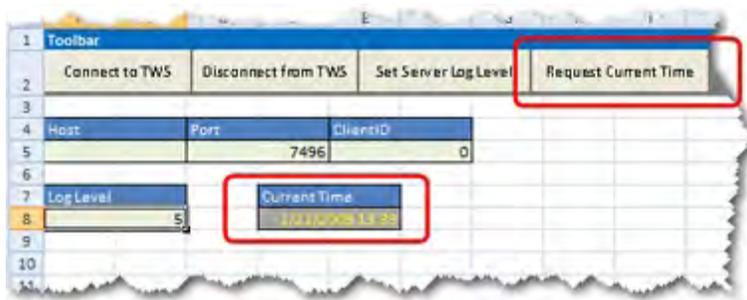
数字越大，详细程度和资源使用越大。



2. 点击工具栏上的 **Set Server Log Level( 设置服务器记录级别)** 。

### 请求当前时间

1. 点击工具栏上的 **Request Current Time( 请求当前时间)** 。



### 一般页面工具栏按钮

一般页面上的工具栏包括的按钮描述如下。

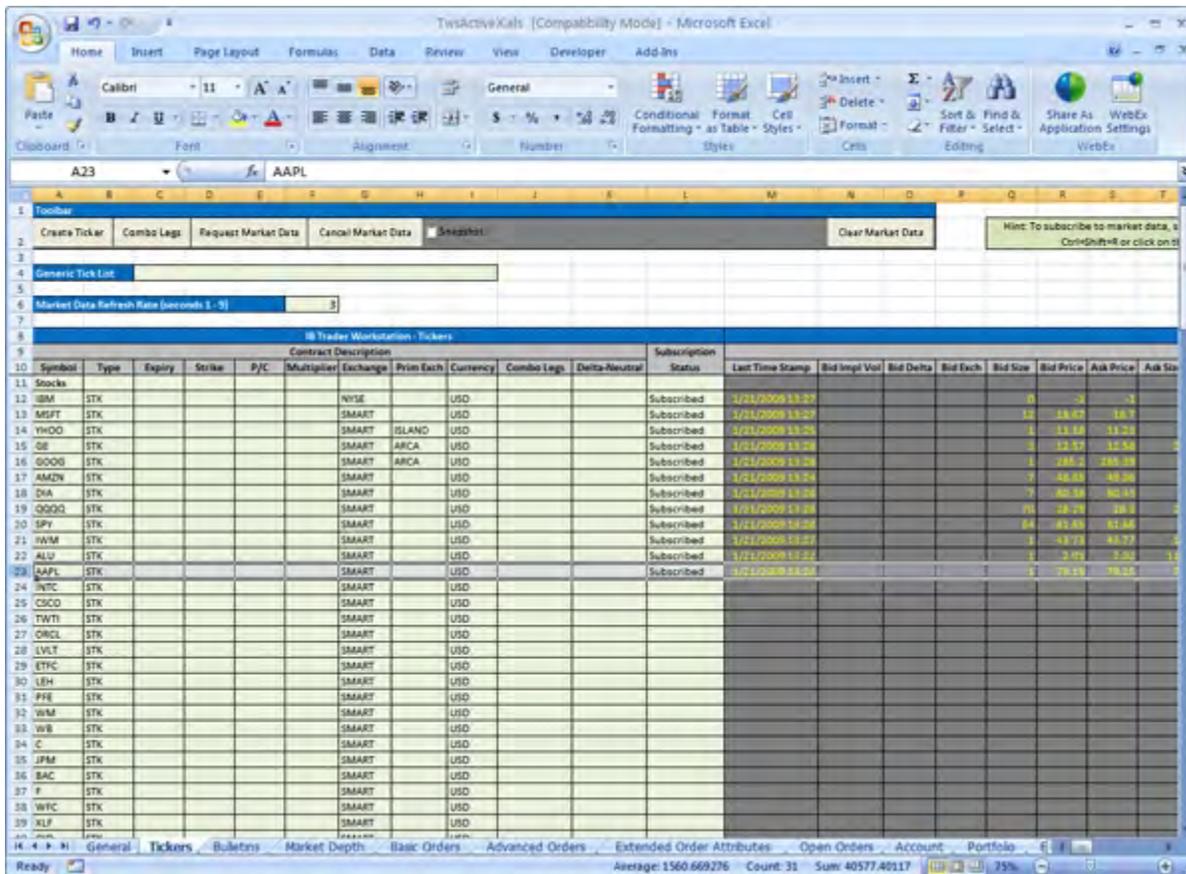
按钮	描述
Connect to TWS( 连接 TWS)	连接 TWS。
Disconnect from TWS( 取消连接 TWS)	取消连接 TWS。
Set Server Log Level( 设置服务器记录级别)	设置 log.txt 记录文件中输入的详细级别。
Request Current Time( 请求当前时间)	请求当前服务器时间。

工具栏还包括 **Show Errors Message Box( 显示错误信息框)** 复选框，当选取后，将在连接 TWS 时显示错误信息。

### 代码页面

使用代码页面完成：

- 创建市场数据( 代码) 行。
- 请求市场数据。
- 创建期权的组合定单。
- 为期货转现货( EFP) 组合定单创建市场数据行。



使用代码页面

**Note:** 确定TWS处于运行状态，和你已经在Excel表格中的所有页面的*Which Trader Workstation?*( 哪个交易平台?) 部分中的*User Name*( 用户名) 区域输入了你的用户名以保证与TWS的正确连接。

使用创建代码( **Create Ticker**) 按钮创建代码

1. 点击表格底部的**Tickers**( 代码) 标签。
2. 点击空白行左侧的行号码来选取该行。你必须选取一个空白行才能够创建代码行。
3. 点击工具栏上的**Create Ticker**( 创建代码) 按钮和在Create Tickers对话框中输入信息。
4. 点击**OK**。

对股票，你只需要指定*Symbol*( 代码) 、*Type*( 类型) 、*Exchange*( 交易所) ( 通常为SMART) 、和*Currency*( 货币) 。



### 在表格上创建代码

1. 在 *Symbol*( 代码) 栏上选取一个空白单元并输入一个代码。
2. 对所有需要的合约描述区域输入数据，例如，如果你输入的是一个股票代码，你就不需要在 *Expiry*( 过期日)、*Strike*( 行使价)、*P/C*和 *Multiplier*( 乘数) 区域内输入任何值。

**Note:** *Exchange*( 交易所) 区域接受下列值：SMART( 对智能定单传递) ，和任何有效的交易所缩写。

### 请求代码的市场数据

1. 点击行号码选择你希望请求市场数据的代码行。
2. 在 *Generic Tick List*( 通用跳动列表) 单元内输入一个由逗号分隔的通用跳动值列表。有关通用跳动值，请参阅 [通用跳动类型](#)。
3. 或者，点击 **Snapshot**( 快照) 复选框来请求市场数据的一个单个快照。
4. 点击工具栏上的 **Request Market Data**( 请求市场数据) 。

获取一组代码的市场数据，按住 **Shift**键选择多个代码行，然后多次点击 **Request Market Data**( 请求市场数据) ，直到所有行显示数据为止。

### 设置刷新率

市场数据刷新率决定刷新与TWS连接的频率。

- 在 *Market Data Refresh Rate*( 市场数据刷新率) 单元内输入刷新率值( 用整数，以秒为单位) 。

默认状态的TWS市场数据每3秒钟更新一次。

### 代码页面工具栏按钮

代码页面上工具栏包括的按钮描述如下。

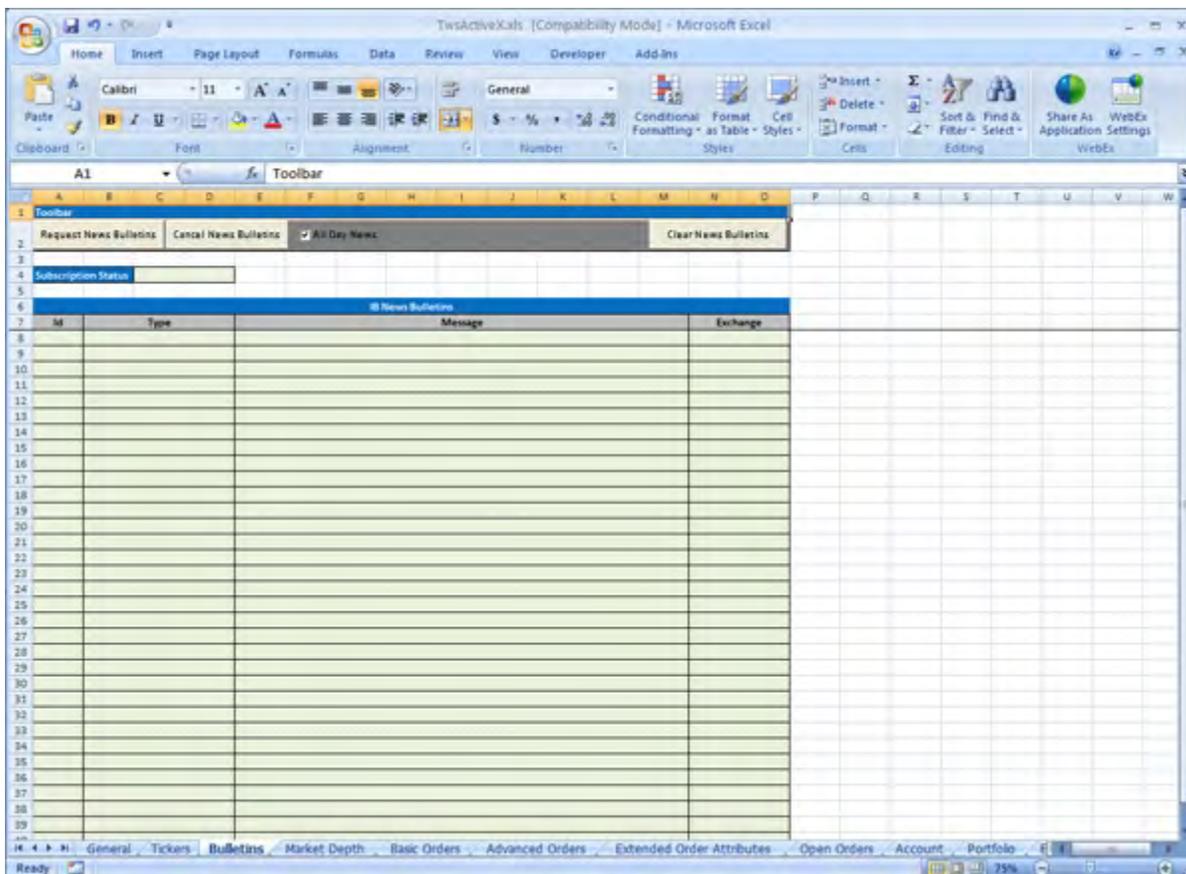
按钮	描述
Create Ticker( 创建动漫)	打开代码框。输入信息创建市场数据行。
Combo Legs( 组合边)	打开组合边框。输入合约详细 分别创建组合定单的每个边。
Request Market Data( 请求市场数据)	选择一个行并点击获取选择合约的市场数据。
Cancel Market Data( 取消市场数据)	取消选取代码的市场数据。
Clear Market Data( 清除市场数据)	清除页面上的所有市场数据。

工具栏还包括 **Snapshot**复选框，当选择后，仅请求市场数据的一个单个快照。

### 公告页面

使用公告页面请求和查看 **IB**新闻公告。只需点击工具栏上的 **Request News Bulletins( 请求新闻公告)** 。新闻公告即在页面的表格中显示。如请求当天所有存在的公告和任何新的公告，选择工具栏上的 **All Day News( 所有当天新闻)** 复选框。如果没有选择这个复选框，你将仅收到新的公告。





公告页面工具栏按钮

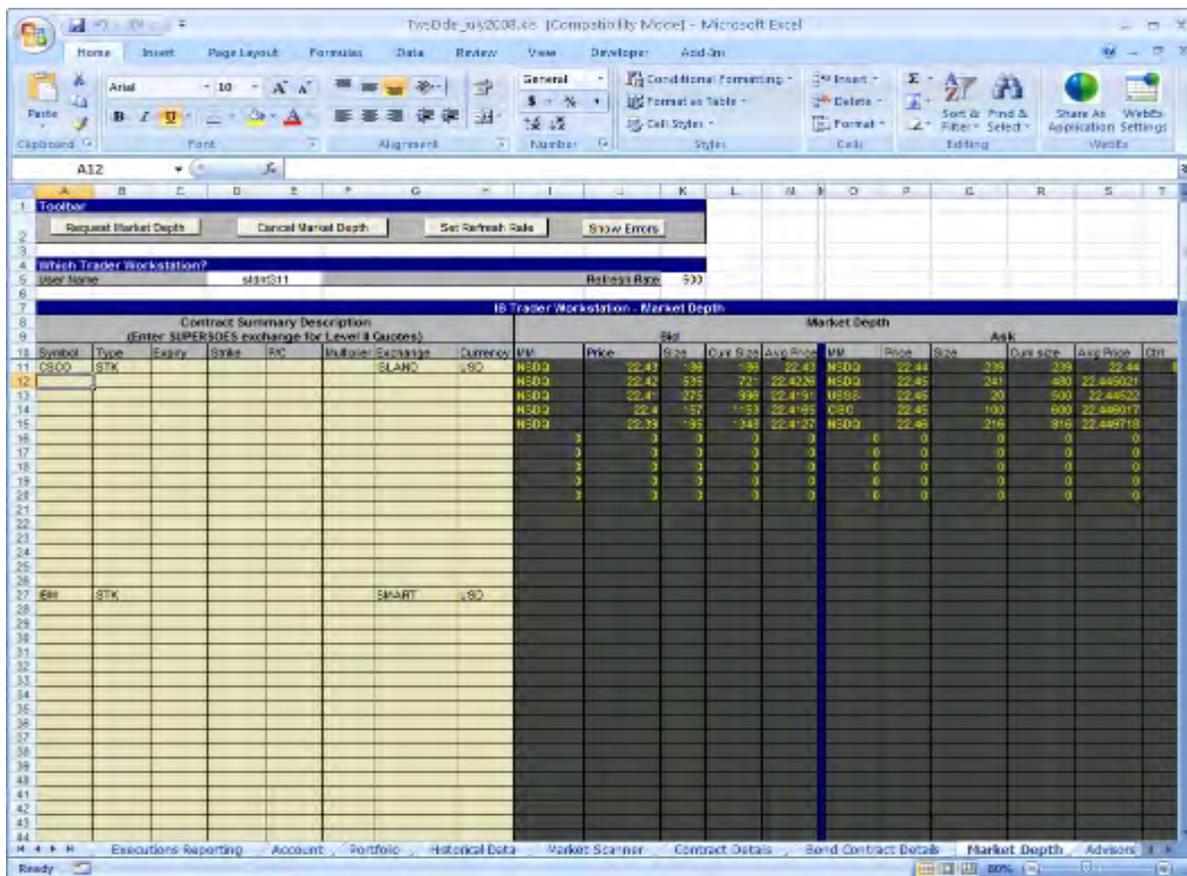
代码页面的工具栏按钮描述如下。

按钮	描述
Request News Bulletins( 请求新闻公告)	请求所有新的新闻公告。
Cancel News Bulletins( 取消新闻公告)	取消接收所有新闻公告。
Clear News Bulletins( 清除新闻公告)	清除页面上的所有新闻公告。

工具栏还包括**All Day News( 所有当天新闻)** 复选框，当选择后，将请求当天所有存在的公告和任何新的公告。

市场深度页面

使用市场深度页面查看选取合约的市场深度。你还可以通过订阅**Open Book Market Data( 开放市场数据)** 查看NYSE挂牌产品的市场深度，订阅**TotalView Market Data**查看NASDAQ挂牌交易产品的市场深度，前题是你需要注册订阅这些数据。



### 使用市场深度页面

**Note:** 确定TWS处于运行状态，你已经将表格与TWS连接。

#### 请求合约的市场深度

1. 点击表格底部的**Market Depth( 市场深度)** 标签打开市场深度页面。
2. 选取你希望请求市场深度的代码，或在一个空白行上输入一个新代码。
3. 点击工具栏上的**Request Market Depth( 请求市场深度)** 按钮。

#### 重新设置市场深度的市场数据刷新率

1. 在**Market Depth Refresh Rate( 市场深度刷新率)** 单元内用秒为单位输入希望的市场数据刷新率。值必须是从1到9的整数。

#### 市场深度页面工具栏按钮

市场深度页面上工具栏包括下列按钮：

按钮	描述
----	----

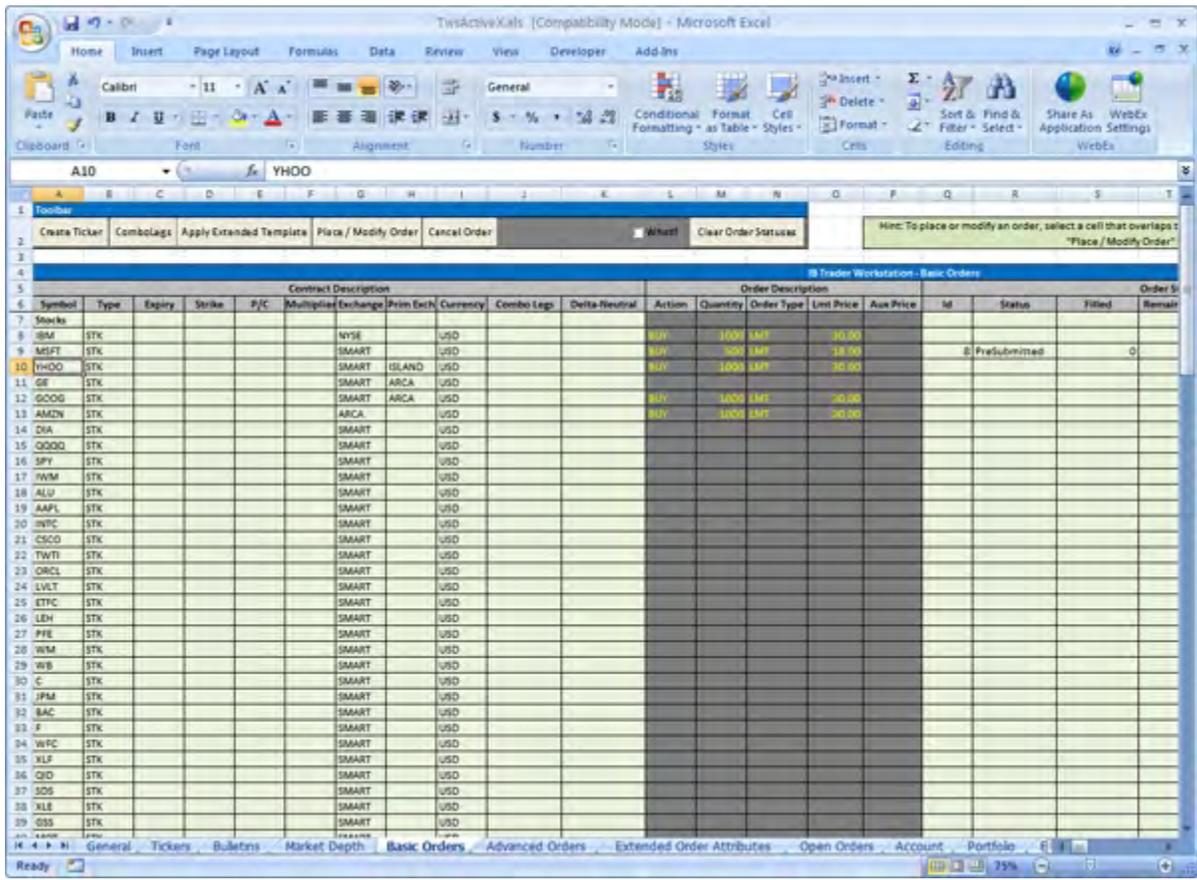
Request Market Depth( 请求市场深度)	查看选取合约的买/卖价深度。
Cancel Market Depth( 取消市场深度)	取消选取合约的市场深度。
Clear Market Depth( 清除市场深度)	删除页面上的所有市场深度。

页面上开包括一个 *Market Depth Refresh Rate( 市场深度刷新率)* 单元，你可以用来以秒为单位( 从1到9的整数) 重新设置市场深度刷新率。默认的刷新率为3秒钟。

**基本定单页面**

使用基本定单页面进行：

- 创建定单。
- 下达“假设”定单以便在下达真正定单之前了解保证金和佣金信息。
- 创建“篮子”定单。
- 修改和取消定单。
- 创建组合定单。



## 下达定单

这一节描述任何在定单页面上下达以下类型的定单：

**Note:** 确定TWS处于运行状态，你已经将表格与TWS连接。

### 下定单

1. 点击表格底部的 **Basic Orders( 基本定单)** 标签。
2. Define a contract by typing a symbol in a blank *Symbol* field, then entering information in the relevant contract description fields.
3. Select a contract and set up the order using the *Order Description* fields.  
你必须定义 *Action( 行动)* ( 买、卖或卖空)、*Quantity( 数量)*、*Order Type( 定单类型)*、*Limit Price( 限价)* ( 除非是市价单)，如需要的话为要求的定单类型定义 *Aux.价格( 辅助价格)*。
4. 如需要，选择合约( 代码行) 和点击工具栏上的 **Apply Extended Template( 应用扩展模版)** 应用扩展定单属性。这将对选择的合约应用你在 [Extended Order Attributes\( 扩展定单属性\)](#) 页面上定义的所有属性。
5. 点击工具栏上的 **Place/Modify Order( 下达/修改定单)** 按钮。

### 下达“篮子”定单

1. 点击表格底部的 **Basic Orders( 基本定单)** 标签。
2. 在一个空白 *Symbol( 代码)* 区域输入一个代码来定义合约，然后在相应的合约描述区域输入信息。
3. 选择合约和使用 *Order Description( 定单描述)* 部分的区域设置定单。
4. 重复第一步和第二步完成其它的定单。
5. 选择一组定单。
  - 选择一组邻近的定单，选中第一个定单，按住 **Shift**键，然后选中组中所有的定单。
  - 选择一组不相邻的定单，按住 **Ctrl**键，同时选择每个定单。
6. 点击 **Place/Modify Order( 下达/修改定单)** 按钮。

### 修改单个定单( 或一组定单)

1. 在基本定单页面上，改变某个定单或一组定单的任何需要的参数。
2. 选择定单或一组定单
  - 选择一组邻近的定单，选中第一个定单，按住 **Shift**键，然后选中组中所有的定单。
  - 选择一组不相邻的定单，按住 **Ctrl**键，同时选择每个定单。
3. 点击 **Place/Modify Order**按钮。

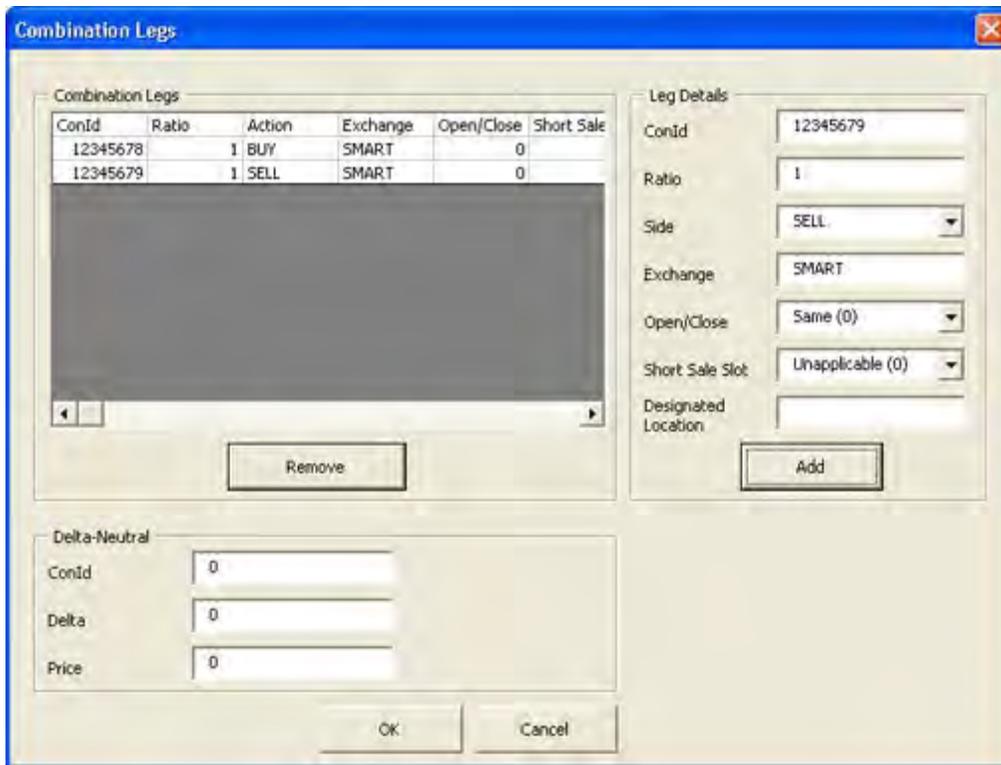
### 下达组合定单

组合定单是一种特殊的定单类型，是由多个不同的边构建组成，但以单个交易执行。

买进一个日历差价，你需要：

### 创建日历差价定单

下面的举例是下达一个ISE交易所的XYZ股票的假设日历差价定单的过程。



1. 使用 [Contract Details\( 合约详细\)](#) 页面获得两个边定义的合约代号。
  - XYZ期权 JUL08 17.5 CALL on ISE的 conid是 "12345678"。
  - XYZ期权 AUG08 17.5 CALL on ISE is "12345679"。
2. 点击 **Basic Orders( 基本定单)** 标签创建组合边定义。点击基本定单页面工具栏的 **Combo Legs( 组合边)** 按钮和输入边信息。 你的边信息被转换成的格式：

[CMBLGS]\_ [NumOfLegs]\_ [Combo Leg Definitions]\_ [CMBLGS]

这里：

- [CMBLGS] 是用于确定边定义的开始和截止的分隔符
- [NumOfLegs] 是边定义的数量
- [Combo Leg Definitions] 定义N边的定义，和每个边的定义，由 [conid]\_ [ratio]\_ [action]\_ [exchange]\_ [openClose]，由此产生的组合子字符串如下：

CMBLGS\_2\_17496957\_1\_BUY\_EMPTY\_0\_15910089\_1\_SELL\_EMPTY\_0\_CMBLGS

3. 组合边定义必须在扩展定单属性之前出现。完整的下达定单DDE请求的字符串如下：

=acctName|ord!id12345?place?BUY\_1\_XYZ\_BAG\_ISE\_LMT\_1\_CMBLGS\_2\_12345678\_1\_BUY\_EMPTY\_0\_12345679\_1\_SELL\_EMPTY\_0\_CMBLGS\_DAY\_EMPTY\_0\_O\_0\_EMPTY\_0\_EMPTY\_0\_0\_EMPTY\_0\_0

如果定单边不构成一个有效组合，将出现下列错误之一：

- 312 = 组合详细无效。
- 313 = '<leg number>'的组合详细无效。
- 314 = 证券类型'BAG'要求组合边详细。
- 315 = 股票组合边限于SMART交易所。

**Note:** 1. 边定义的交易所必须与组合定单的匹配。除了STK边定义以外，必须定义为SMART交易所。  
 2. 零售账户的openClose边定义值永远是'SAME'（例，0）。对机构账户，该值可以是三者之一：（ SAME、OPEN、CLOSE）。

**支持的定单类型**

目前支持的ActiveX for Excel API的定单类型是：

- 限价单 (LMT)
- 市价单 (MKT)
- 触及限价单 (LIT)
- 触及市价单 (MIT)
- 收盘市价单 (MOC)
- 收盘限价单 (LOC)
- 挂钩市价单 (PEGMKT)
- 相对单 (REL)
- 止损单 (STP)
- 止损限价单 (STPLMT)
- 跟踪止损单 (TRAIL)
- 跟踪止损限价单 (TRAILLIMIT)
- 交易量加权平均价单 (VWAP)
- 波动率定单 (VOL)

**基本定单页面工具栏按钮**

基本定单页面上的工具栏包括以下按钮：

按钮	描述
Create Ticker( 创建代码)	打开代码框。输入信息创建市场数据行。

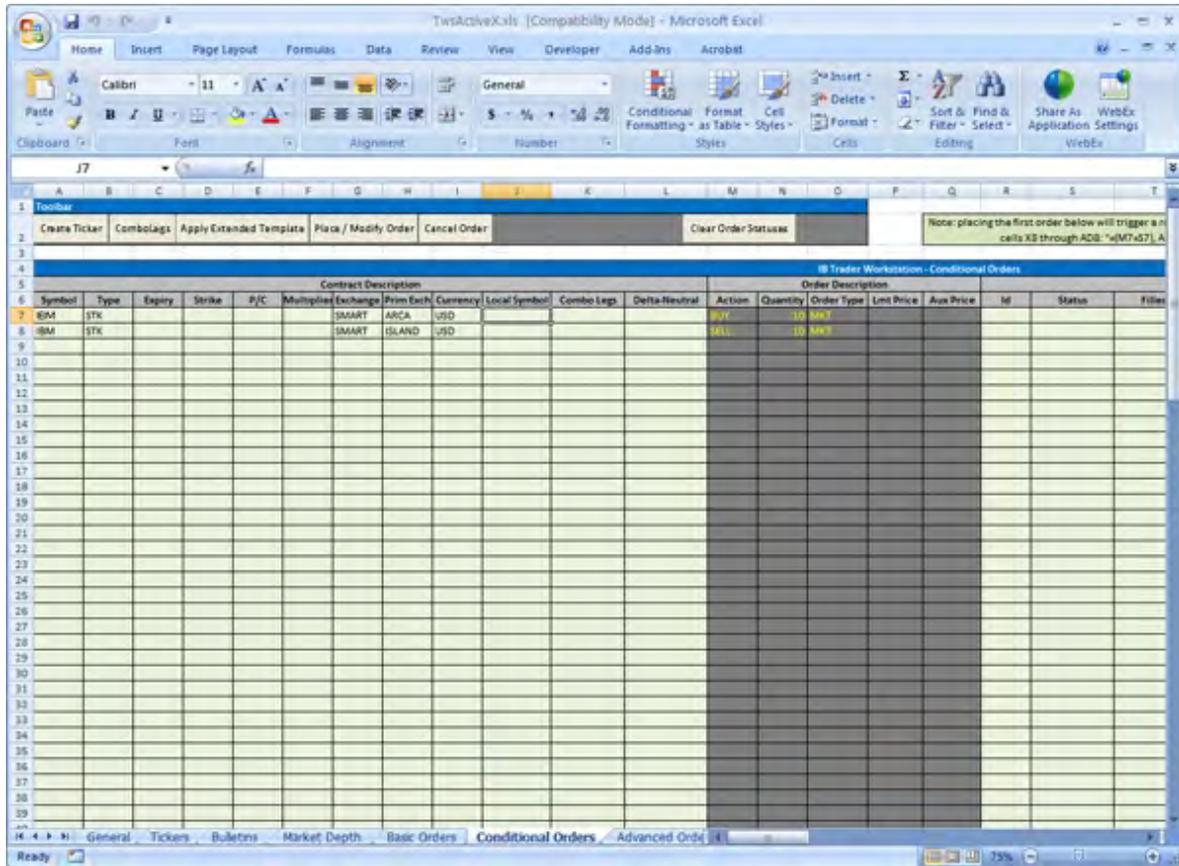
Combo Legs( 组合边)	打开组合边框。输入合约详细创建每个组合定单边。你还可以输入Delta中性信息。
Apply Extended Template( 应用扩展模版)	将当前的值应用到扩展定单属性页面上选中的定单行上。
Place/Modify Orders( 下达/修改定单)	在你完成了定单描述区域和定义了任何扩展属性后，为选取的合约点击创建定单。
Cancel Order( 取消定单)	该按钮取消选取的定单。
Clear Order Statuses( 清除定单状态)	从页面上删除所有定单状态信息。

工具栏上还有一个**WhatIf( 假设)**复选框。当选择后，你将收到保证金和佣金数据，就像是定单被下达了一样，但定单将不会被下达。

### 条件定单页面

使用条件定单页面创建定单的发送是基于满足其它条件的定单，例如，一个基于之前执行或买价或卖价变化的定单。

查看条件描述区域( 蓝色) ，使用页面底部的滚动条向右侧滑动。



### 设置条件定单

**Note:** 确认TWS处于运行状态，你已经在表格所有页面中的*Which Trader Workstation?*(哪个交易平台?)部分的*User Name*(用户名)区域输入了你的用户名以保证与TWS的成功连接。

### 设置条件定单

1. 在 **Conditional Orders**(条件定单)页面上，首先创建你希望当一个满足时被发送的定单，通过 *Contract Description*(合约描述)区域定义合约，然后使用 *Order Description*(定单描述)区域设置定单参数。
2. 在蓝色条件描述区域，使用 *Statement*(描述)区域设置必须满足触发定单的条件。当描述 = TRUE(真)，你的定单将被发送。

样本表格包括一对定单，其中第二个定单的发送是基于第一个定单的完全执行。此种情况下，描述区域触发是T10单元内的值( *Filled*(执行的)区域)必须等于M10区域的值(定单 *Quantity*(数量)区域)。

3. 在 *ADD/MOD*区域中输入 **ADD**，因为你在创建一个一次性定单。
4. 定义其它定单参数如同你在 *Order Description*(定单描述)区域中做的一样。



IB Trader Workstation - Conditional Orders														
Option			Order Description							Order				
Primary Exchange	Currency	Comb Legs	Leave this empty	Action	Quantity	Order Type	Lmt Price	Aux Price	Ctrl	Id	St	Filled	Remaining	
ARCA	USD			BUY	10	MKT				0	id	Fi	10	0
ISLAND	USD			SELL	10	MKT				0	id	Fi	10	0
	usd			buy	200	lmt	81.00							

5. 根据下面表格中的句法在 **Conditional Orders( 条件定单)** 页面上完成必要的区域。

区域	描述
Statement( 描述)	一个返回真或假的Excel功能。当返回真时，定单将被发送；当假时，没有行动。
ADD/MOD	使用ADD用于一次性定单。使用MOD连续检查和修改定单直到定单被完全执行。这个区域启动一个条件定单，仅当定单具有“ADD”或“MOD”标签时才能被启动。
Action( 行动)	买 卖
Quantity( 数量)	输入定单的数量。
Order Type( 定单类型)	参考 <a href="#">支持的定单类型列表。</a>
Lmt Price( 限价)	限价单和止损限价定单类型的限价。
Aux. Price( 辅助价格)	止损和止损限价定单类型的止损额，或相对定单的抵消额。

以上描述的所有区域可能是根据其它区域单元的变量，所有可以创建任何类型的条件定单。

### 条件定单举例

#### 如果获执行的定单

如果获执行的定单是在如果前一个定单执行后执行的定单。创建如果获执行的定单的条件是“如果一个买单获全部执行，输入一个价格为\$50.00的限价卖单”：

区域	值
Statement( 描述)	Filled cell = 100( 执行的单元 = 100)
ADD/MOD	ADD

Action( 行动)	SELL( 卖)
Quantity( 数量)	100
Order Type( 定单类型)	LMT( 限价)
Lmt Price( 限价价格)	50
Aux. Price( 辅助价格)	empty( 空白)

**价格改变定单**

价格改变定单是当一个特定的买价或卖价在大于、小于或等于某个特定价格时被触发。创建一个价格改变定单使用的条件为“如果买价降至81.20以下，发送一个限价价格为\$81.10的200股限价买单：

区域	值
Statement( 描述)	<p>在<b>Tickers( 代码)</b>页面上，将你的光标指向你希望使用的买价区域，然后复制出现在表格顶部公式栏内的值(“=”输入区域)。该值应该是：                      =username tik!id4?bid                      这里“4”代表特定合约的买价。                      粘贴公式栏内(“=”输入区域)在描述区域，并在价格之后添加你的条件，“=”&gt;”或“&lt;”。此例中，公式应为：                      =username tik!id4?bid&lt;81.20</p>
ADD/MOD	ADD
Action( 行动)	BUY( 买)
Quantity( 数量)	200
Order Type( 定单类型)	LMT( 限价)
Lmt Price( 限价价格)	81.10
Aux. Price( 辅助价格)	此例中不使用。

**修改定单( 或篮子定单)**

1. 选择定单或一组定单。
  - 选择一组相邻的定单，选中第一个定单，按住 **Shift**键，然后选中组中最后的定单。
  - 选择一组不相邻的定单，按住 **Ctrl**键，选择每一个定单。
2. 点击 **Place/Modify Order( 下达/修改定单)** 按钮。
3. 改变任何需要的参数，然后点击 **Place/Modify Order**按钮。

**条件定单页面工具栏按钮**

条件定单页面上的工具栏包括下列按钮：

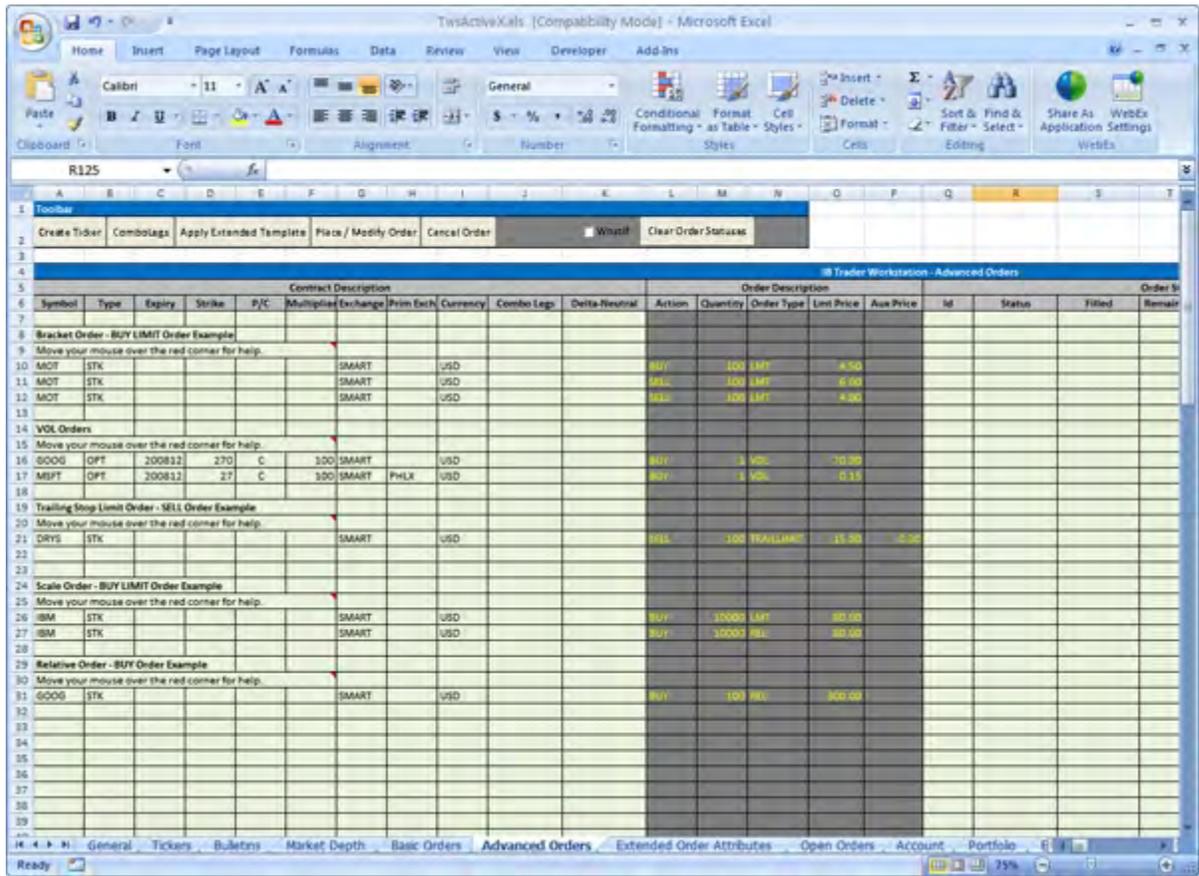
按钮	描述
Combo Legs( 组合边)	打开组合边框。输入合约详细分别创建每个组合定单的边。
Place/Modify Order( 下达/修改定单)	在你完成了定单描述区域和定义了任何扩展属性后，点击创建选取合约的定单。
Apply Extended Template( 应用扩展模版)	应用所有 <a href="#">扩展定单属性页面</a> 上的属性到选取的定单。
Cancel Order( 取消定单)	该按钮取消你选中的定单。
Show Errors( 显示错误)	跳至错误编码区域和显示错误编码。

**高级定单页面**

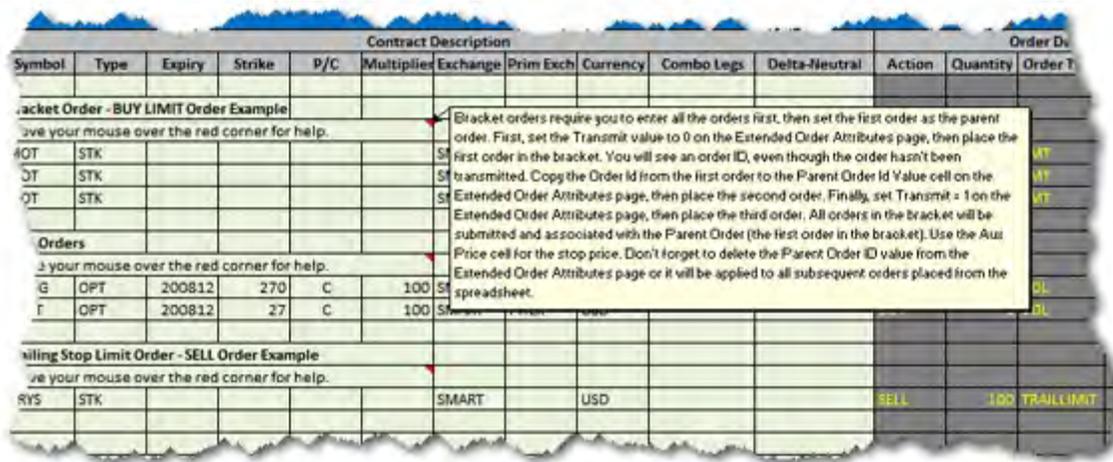
使用高级定单页面创建：

- 括号单
- 波动率单
- 跟踪止损限价单
- 分段单

- 相对单



该页面包括几个含有鼠标提示的样本订单来帮助学习如何下达这些订单。只需将你的鼠标指向页面中单元内的红色三角形即可显示帮助信息。



有关使用单个定单或组定单的扩展定单属性的更多信息，请参阅[对单个定单和组定单应用扩展定单属性](#)

### 下达括号单

ActiveX for Excel样本表格中的括号单要求使用扩展定单属性 *Transmit*( 发送) 和 *Parent Order Id*( 母定单代号) 。你必须在定单被完全设置之前关闭 *Transmit*，你必须指定括号单中的第一个定单为母定单。

### 下达限价括号买单

1. 点击表格底部的 **Advanced Orders( 高级定单)** 标签。
2. 在三个相邻的行上为三个定单输入合约描述和定单描述：
  - 第一个定单应该是一个买进限价单。
  - 的二哥定单应该是卖出止损单。
  - 第三个定单应该是卖出限价单。
3. 点击 **Extended Order Attributes( 扩展定单属性)** 标签。改变 *Transmit* 的值为 **0** ( 扩展定单属性页面上的第13行) 。

这将保证你的定单不会在你完成定单设置之前被发送。

4. 点击 **Advanced Orders** 标签，选中括号单中的第一个定单，然后点击 **Place/Modify Order** 按钮。  
定单没被执行，但系统生成了一个定单代号。
5. 为第一个定单复制定单代号，除去“id”前缀，然后点击 **Extended Order Attributes** 标签并为 *Parent Order Id*( 母定单代号) 粘贴定单代号到 *Value* 区域( 第14行) 。这个值将被应用到所有后续定单中，直到你将其从扩展定单属性页面中删除。

现在括号单的第一个定单成为首要定单。

6. 点击 **Advanced Orders** 标签，选中第二个定单，然后点击 **Place/Modify Order** 按钮。  
定单没有被执行，但现在通过母定单代号扩展定单属性与首要定单相连。
7. 点击 **Extended Order Attributes** 标签并改变 *Transmit* 的值为 **1** ( 13行) 。
8. 点击 **Advanced Orders** 标签，选中括号单中的第三个定单，然后点击 **Place/Modify Order** 按钮。整个括号单被发送。
9. 当你完成下达你的括号单后，进入 **Extended Order Attributes( 扩展定单属性)** 页面和删除你输入的 *Parent Order Id*( 母定单代号) 值。如果你不将其删除，这个值将被应用到所有你在表格中下达的后续的定单中。

### 下达波动率定单

在 ActiveX for Excel 样本表格中，你通过输入下列扩展定单属性下达 VOL 定单。

- Volatility( 波动率)
- Volatility Type( 波动率类型)
- Reference Price Type( 参考价格类型)
- Continuous Update( 连续更新)

- Underlying Range (Low) - optional 『底层范围( 低端) — 可选』
- Underlying Range (High) - optional 『底层范围( 高端) — 可选』
- Hedge Delta Order Type - optional( 对冲Delta定单类型— 可选)
- Hedge Delta Aux Price - optional( 对冲Delta辅助价格— 可选)

### 下达VOL( 波动率) 定单

1. 点击表格底部的**Advanced Orders**标签。
2. 在空白的Symbol区域输入代码定义合约，然后在相对的合约描述区域中输入信息。
3. 选择一个合约和使用*Order Description*( 定单描述) 区域设置定单。
  - 在*Order Type*( 定单类型) 区域中输入**VOL**。
4. 点击**Extended Order Attributes**标签。在*Value* 区域内为以下扩展定单属性输入值：
  - **Volatility** - 这个值代表计算期权限价价格中使用的波动率。以百分比形式输入这个值，而不是显示的市场数据。例如，输入17.12，而不是.1712。
  - **Volatility Type** - 对日波动率输入1或年波动率输入2。
  - **Reference Price Type** - 这个值被用来计算发送到交易所的限价价格和股票范围价格监控。输入1使用平均最佳买卖价；或2使用NBB( 买价)，当买进看涨或卖出看跌，或NBO( 卖价)，当卖出看涨或买进看跌。
  - **Continuous Update** - 输入1当底层股票价格( 或期货价格，对指数期权) 变动时自动更新期权价格。输入0如果你不希望使用该功能。
5. 在**Extended Order Attributes**页面上，在*Value* 区域内为下列可选的扩展定单属性输入值：
  - **Underlying Range (Low)** - 输入一个相对选取的期权定单可接受的股票价格的低端值。如果底层产品的价格跌入到低端股票范围价格以外，期权定单将被取消。
  - **Underlying Range (High)** - 输入一个相对选取的期权定单可接受的股票价格的高端值。如果底层产品的价格涨至高于高端股票范围价格以外，期权定单将被取消。
  - **Hedge Delta Order Type** - 输入LMT、MKT或REL。输入NONE( 无) 如果你不希望使用delta对冲。
  - **Hedge Delta Aux Price** - 如果你输入了LMT或REL做为对冲Delta定单类型，对该属性输入价格做为值。
6. 点击**Advanced Orders**标签，然后选中定单行。
7. 点击**Apply Extended Template**( 应用扩展模版) 按钮。你输入的扩展定单属性的值被应用到定单行并显示在页面的*Extended Order Attributes*部分。
8. 在选中定单行后，点击**Place/Modify Order** 按钮。
9. 当你完成了下达VOL定单后，进入**Extended Order Attributes**页面并删除你输入的VOL定单值。如果你不将其删除，这个值将被应用到所有你在表格中下达的后续的定单中。

### 下达跟踪止损限价单

在TWS中，有四个值组成一个跟踪止损限价单：

- trailing amount( 跟踪额)
- stop price( 止损价)

- limit price( 限价)
- limit offset( 限价抵消)

在ActiveX for Excel API表格中，你输入trailing amount、stop price和limit price。没有用于limit offset值的区域或扩展定单属性。你必须在stop price( *Trail Stop Price*( 跟踪止损价) 扩展定单属性) 中包括limit offset。

### 创建跟踪止损限价单

1. 点击表格底部的**Advanced Orders**标签。
2. 在空白的*Symbol*区域输入代码定义合约，然后在相对的合约描述区域中输入信息。
3. 选择一个合约和使用*Order Description*( 定单描述) 区域设置定单。
  - 在*Action*区域输入**BUY**或**SELL**。
  - 在*Lmt Price*区域输入限价。
  - 在*Order Type*( 定单类型) 区域内输入**TRAILLIMIT**。
  - 在*Aux Price*区域内输入跟踪额。
4. 点击**Extended Order Attributes**标签。指定跟踪止损价格做为扩展定单属性。在跟踪止损价格的*Value*区域输入这个值。
  - 跟踪止损价格值必须包括限价抵消。  
对卖单：

$$\text{Trail Stop Price} = \text{Limit Price} - \text{Trailing Amount} - \text{Limit Offset}$$

对买单：

$$\text{Trail Stop Price} = \text{Limit Price} + \text{Trailing Amount} + \text{Limit Offset}$$

5. 在**Advanced Orders**页面上，选择定单行和点击**Apply Extended Template**按钮。*Trail Stop Price*值被应用到选取的定单并显示在页面的*Extended Order Attributes*部分的*Trail Stop Price*区域。
6. 点击**Place/Modify Order**按钮。
7. 当你完成了下达定单后，进入**Extended Order Attributes**页面和删除你输入的*Trail Stop Price* 值。如果你不将其删除，这个值将被应用到所有你在表格中下达的后续的定单中。

### 下达分段定单

在ActiveX for Excel样本表格中，你可以通过为下面的扩展定单属性输入值来下达分段定单。

- Scale Component Size( 分段组尺寸)
- Scale Price Increment( 分段价格增量)

### 下达分段定单

1. 点击表格底部的**Advanced Orders**( 高级定单) 标签。
2. 在空白的*Symbol*区域中输入代码来定义合约，然后在相应的合约描述区域内输入信息。
3. 选取合约和使用*Order Description*( 定单描述) 区域设置定单。定单类型应为LMT或REL。

4. 点击 **Extended Order Attributes( 扩展定单属性)** 标签。在 *Value* 区域内为下列扩展定单属性输入值：
  - **Scale Component Size** - 输入第一定单组，或起始定单组的尺寸。例如，如果你发送了一个分段组尺寸值为1000的10,000股定单，第一组将为最前面的1000股。
  - **Scale Price Increment** - 输入用于计算分段阶梯中每一组的单位价格的数额。不可以是负数。

**Note:** 截至API发行要点9.41，不支持 **Scale Num Components ( 分段数额组)** 。

5. 在 **Advanced Orders** 页面上，选择定单行并点击 **Apply Extended Template( 应用扩展模版)** 按钮。分段定单值被应用到选取的定单并显示在页面的 *Extended Order Attributes* 部分中。
6. 点击 **Place/Modify Order** 按钮。
7. 当你完成下达定单后，进入 **Extended Order Attributes** 页面并删除你输入的分段定单值。否则的话，这些值将被应用到所有你在表格中下达的后续定单。

### 下达相对定单

在 ActiveX for Excel 样本表格中，你可以通过为 *Percent Offset( 百分比抵消)* 扩展定单属性输入值来下达相对定单。

### 下达相对定单

1. 点击表格底部的 **Advanced Orders( 高级定单)** 标签。
2. 在空白的 *Symbol* 区域中输入代码来定义合约，然后在相应的合约描述区域内输入信息。
3. 选取合约和使用 *Order Description( 定单描述)* 区域设置定单。
  - 输入 **REL** 做为定单类型。
  - 在 *Lmt Price( 限价价格)* 单元内输入价格上限。
4. 点击 **Extended Order Attributes( 扩展定单属性)** 标签。为 *Percent Offset* 扩展定单属性在 *Value* 区域内按十进制形式输入一个百分比。
5. 在 **Advanced Orders** 页面上，选取定单行并点击 **Apply Extended Template( 应用扩展模版)** 按钮。百分比抵消值被应用到选取的定单并显示在页面的 *Extended Order Attributes* 部分中。
6. 点击 **Place/Modify Order** 按钮。
7. 当你完成下达定单后，进入 **Extended Order Attributes** 页面并删除你输入的 *Percent Offset* 值。否则的话，这些值将被应用到所有你在表格中下达的后续定单。

### 高级定单页面工具栏按钮

高级定单页面上的工具栏包括下列按钮：

按钮	描述
Create Ticker( 创建代码)	打开代码框。输入信息创建一个市场数据行。
Combo Legs( 组合边)	打开组合边框。输入合约详细创建组合定单的每个边。



Apply Extended Template( 应用扩展模版)	将扩展定单属性页面上当前的值应用到选取的定单行。
Place/Modify Orders( 下达/修改定单)	在你完成了定单描述区域和定义了任何扩展属性后，点击选取的合约创建定单。
Cancel Order( 取消定单)	该按钮取消选取的定单。
Clear Order Statuses( 清除定单状态)	清除页面上所有定单状态信息。

工具栏上还有一个**WhatIf( 假设如果)** 复选框。当选取后，你将收到保证金和佣金数据，就如同定单已被下达一样，但定单将不会被下达。

### 扩展定单属性页面

扩展定单属性页面包括所有你在发送定单时可以使用的可选属性，比如设置显示尺寸创建冰山定单，给OCA( 一取消全) 组添加定单，和为到时后有效( Good After Time ) 定单设置发送日期等。一旦你在这个页面上定义了属性，你可以使用**Apply Extended Template( 应用扩展模版)** 按钮将它们应用到一个单个定单或选取的定单组，这都可以在定单页面和条件定单页面上进行。属性将填充到页面最右面的*Order Status( 定单状态)* 区域之后的扩展定单属性区域。



**Note:** 你还可以使用这个步骤应用扩展定单属性到条件定单页面的定单中。

#### 应用扩展定单属性到单个定单或定单组。

1. 输入扩展定单属性页面上你希望应用到单个定单或定单组的单个值或多个值。
2. 在定单页面上，选取定单或定单组。
3. 点击 **Apply Extended Template( 应用扩展模版)** 按钮。

扩展定单属性被应用到选取的定单，你在扩展定单属性页面上输入的值被添加到定单页面的 *Extended Order Attributes* 部分的相应的区域中。

当你下达定单或定单组时，你输入的扩展定单属性值将被应用到定单中。

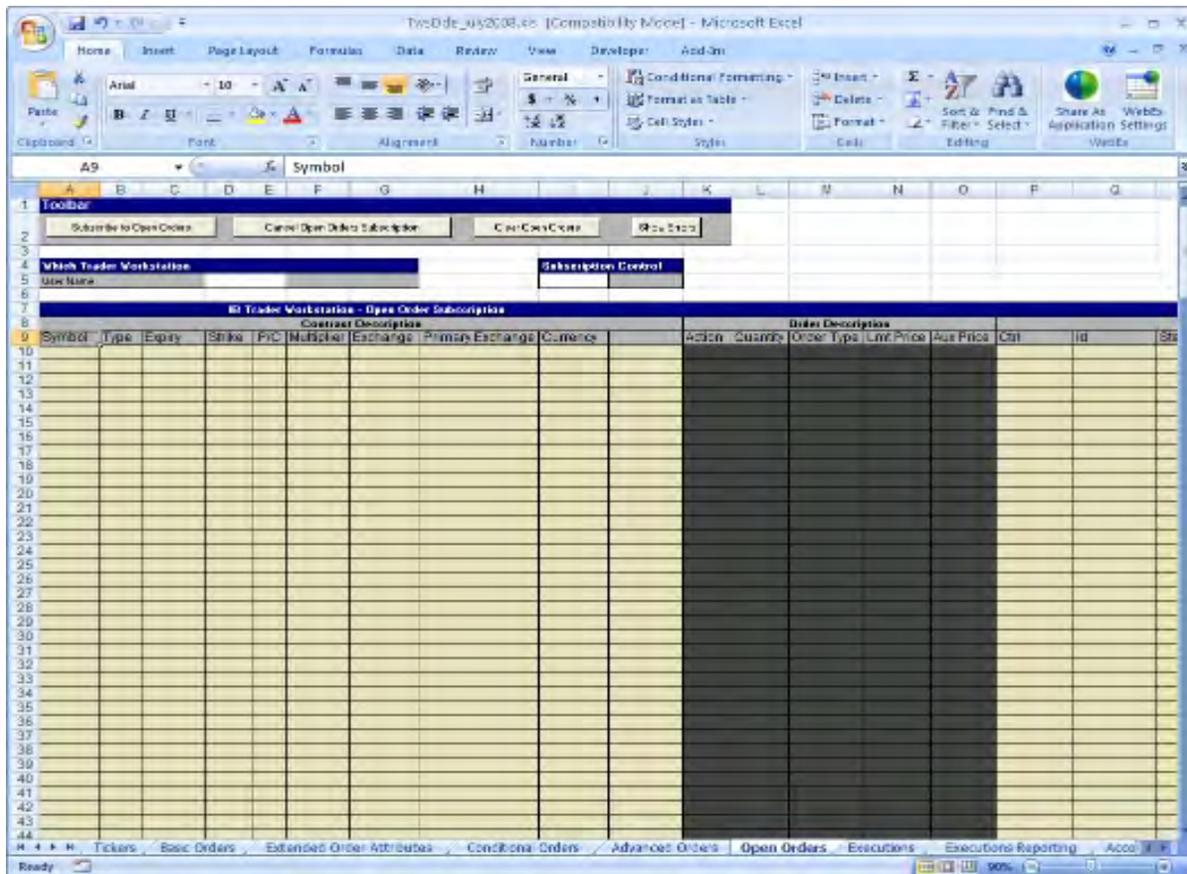
例如，你可能希望分配一个独特的定单参考号码到一个定单组或篮子定单。为实现这个目的，你需要在扩展定单属性页面上为定单参考属性输入数字，然后选取定单页面上定单组的所有定单并点击应用扩展模版( **Apply Extended Template**) 。

4. 从扩展定单属性页面上删除你用于定单的扩展定单属性值。如果你不将这些值取消，这些值仍将被应用到你通过 ActiveX for Excel API 表格下达的所有后续定单中。

#### 开放定单页面

开放定单页面显示所有已发送的定单，包括那些已被 IB 系统接受的定单，和仍在交易所中工作的定单。一旦你订阅了该功能，页面将在每次你发送一个新定单时更新，或者是通过 API 或 TWS。

一旦一个定单被执行，它被保持在开放定单页面上 30 秒，和状态值变为 FILLED( 已执行) 。然后已执行的定单被清除，你可以在执行页面上看到这个定单，如果你订阅了实时执行功能。



### 查看开放定单

**Note:** 确认TWS处于运行状态，你已经将表格与TWS连接。

#### 查看开放定单：

1. 点击表格底部的 **Open Orders( 开放定单)** 标签.
2. 做下列之一：
  - 从当前的ActiveX for Excel表格请求开放定单，点击工具栏上的 **Subscribe to Open Orders( 订阅开放定单)**。
  - 为当前账户请求所有开放定单，点击工具栏上的 **Request All Open Orders( 请求所有开放定单)**。
  - 将所有新创建的TWS定单与当前客户相连，点击工具栏上的 **Request Auto Open Orders( 请求自动开放定单)**。注意，客户代号必须为0。

所有请求的开放定单显示在页面上，包括你在表格上输入的和TWS中的定单。

执行的定单在页面上保留30秒，同时 *Status( 状态)* 区域内出现 *Fill( 执行)* 值。

#### 取消开放定单

1. 点击工具栏上的 **Cancel Open Orders Subscription( 取消开放定单订阅)** 按钮。
2. 点击 **Clear Open Orders( 清除开放定单)** 按钮。

### 开放定单标签工具栏

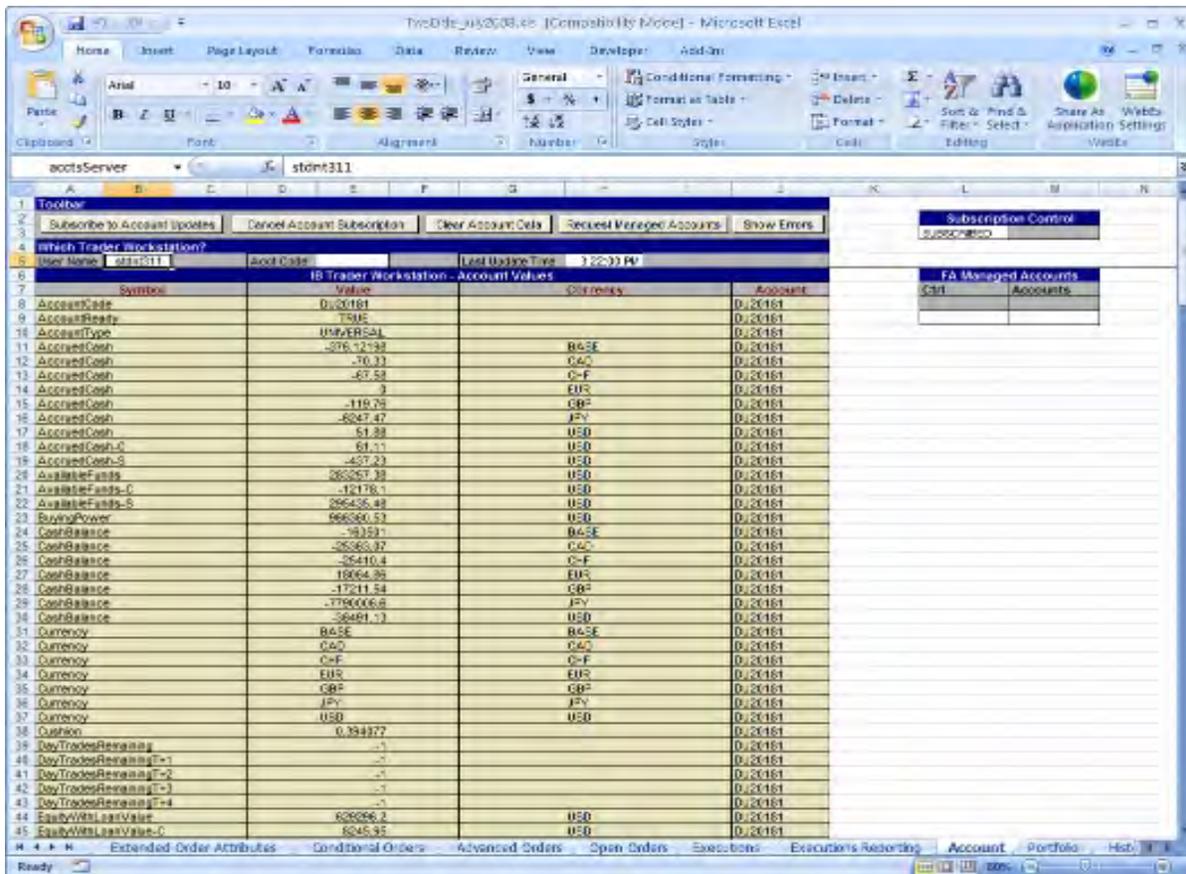
开放定单页面的工具栏包括下列按钮：

按钮	描述
Request Open Orders( 请求开放定单)	查询TWS并返回所有开放定单。一旦你订阅了开放定单，这个页面在每一次有新的开放定单时予以更新。
Request All Open Orders( 请求所有开放定单)	查询TWS和返回当前账户的所有开放定单。一旦你订阅了开放定单，这个页面在每一次有新的开放定单时予以更新。
Request Auto Open Orders( 请求自动开放定单)	查询TWS并使所有新近创建的TWS定单和当前客户相连，客户代号必须是0。
Cancel Auto Open Orders( 取消自动开放定单)	取消新近创建的TWS定单和客户的相连。
Clear Open Orders( 清除开放定单)	删除页面上的所有开放定单。

### 账户页面

使用账户页面进行：

- 查看包括你当前含借贷值股权和可以资金的账户详细信息。
- 查看顾问管理的账户编码列表。
- 金融顾问可以查看FA信息。
- 查看你当前的投资组合。



### 使用账户页面

**Note:** 确认TWS处于运行状态，你已经将表格与TWS连接。

### 查看账户信息

1. 点击表格底部的**Account( 账户)** 标签。
2. 点击工具栏上的**Request Account Updates( 请求账户更新)** 。

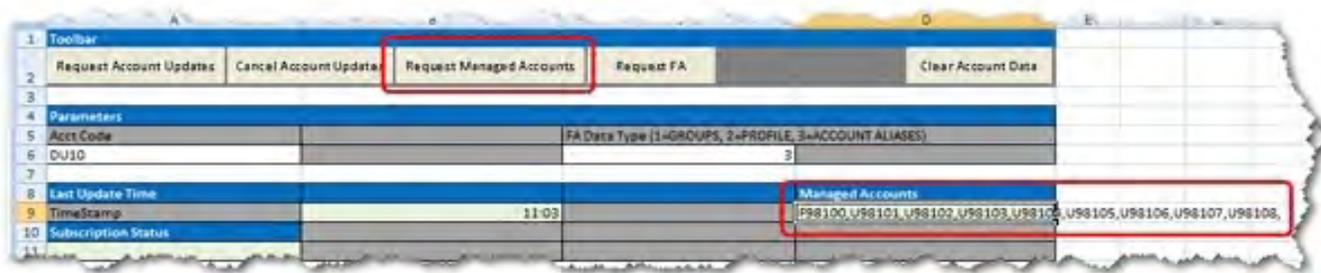
### 删除账户信息

1. 点击表格底部的**Account( 账户)** 标签。
2. 点击工具栏上的**Cancel Account Updates( 取消账户更新)** 停止接收账户更新。
3. 点击工具栏上的**Clear Account Data( 清除账户数据)** 清除页面上的所有数据。

### 请求金融顾问( FA) 管理的账户编码列表。

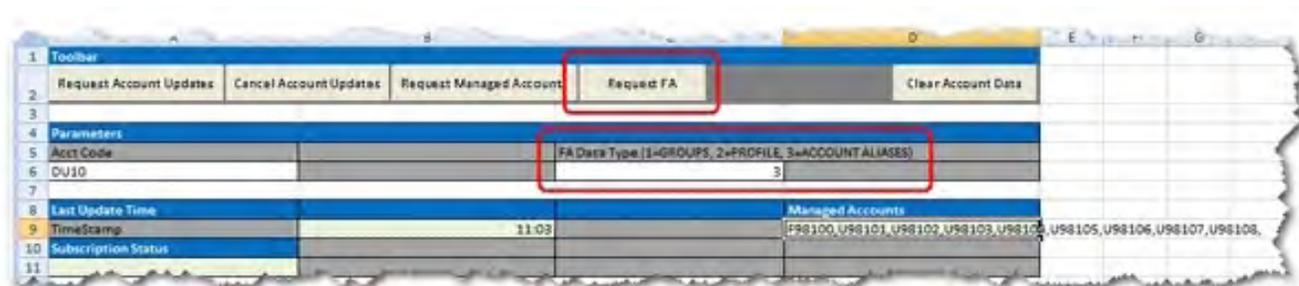
1. 点击表格底部的**Account( 账户)** 标签。
2. 点击工具栏上的**Request Managed Accounts( 请求管理的账户)** 。

一个以逗号分隔的所有管理的账户号码列表显示在**Managed Accounts( 管理的账户)** 单元内。



**请求金融顾问( FA) 信息**

1. 点击表格底部的**Account( 账户)** 标签。
2. 在*Account Code*单元内，输入你希望查看的账户号码。
3. 在*FA Data Type( 金融顾问数据类型)* 单元内，输入一个数字值代表你希望接收的数据类型：
  - o 输入 **1**代表组：
  - o 输入 **2**代表概况。
  - o 数**3**代表账户别名。
4. 点击工具栏上的**Request Managed Accounts( 请求管理的账户)** 。



**账户页面工具栏按钮**

账户页面工具栏包括以下按钮.

按钮	描述
Request Account Updates( 请求账户更新)	每次点击给出特定账户值的数据。账户投资组合部分之前的所有空白行将持有数据。继续点击直到所有的行被数据填充。
Cancel Account Updates( 取消账户更新)	对你持有的每个头寸点击该按钮一次。当你看到行"0's"时，你直到你已经下载了所有当前的头寸。这些值继续以实时更新。
Request Managed Accounts( 请求管理的账户)	对顾问账户，接收一个管理的账户列表并以逗号分隔的列表形式显示在 <i>Management Accounts</i> 单元内。

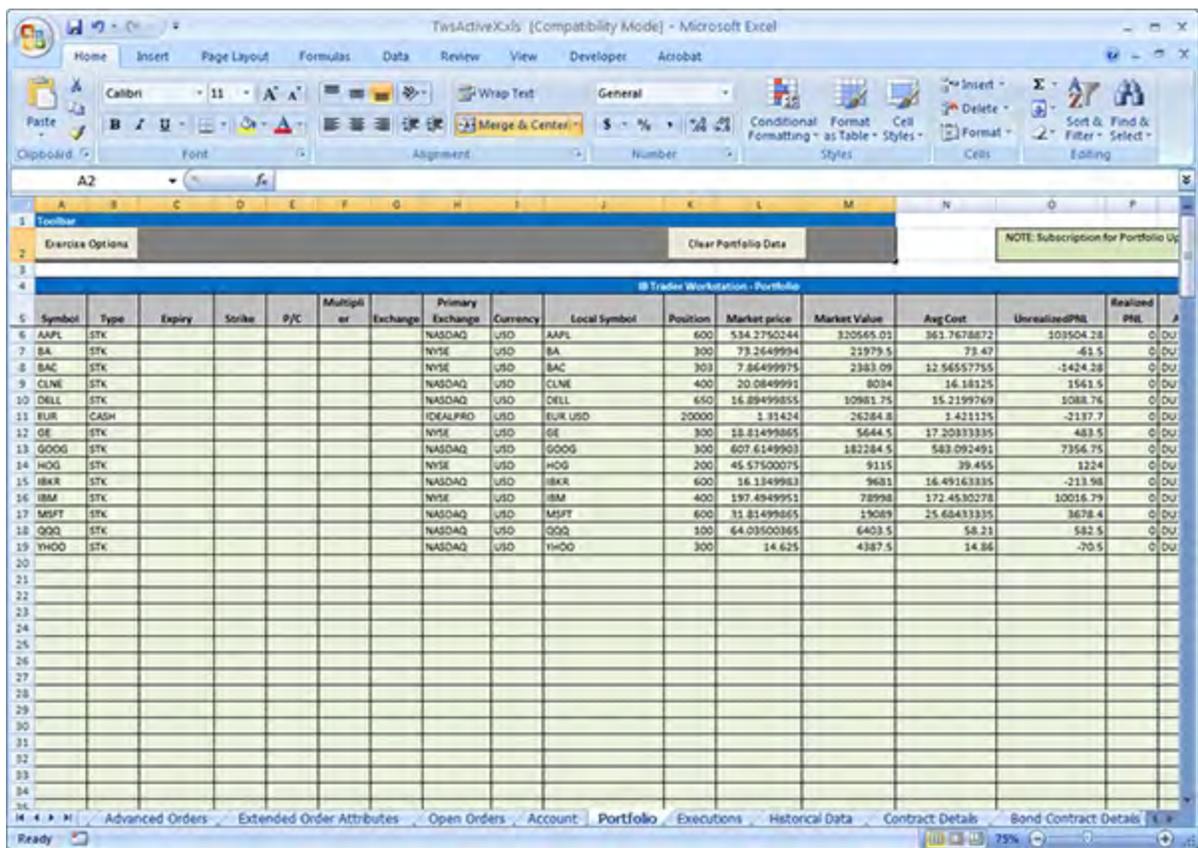
Request FA( 请求FA)	对顾问账户，显示FA Data Type 单元内指定的FA数据类型。
Clear Account Data( 清除账户数据)	清除页面上的所有信息。在你能够清除数据之前，你必须首先取消你的订阅。

### 投资组合页面

使用投资组合页面完成：

- 显示所有你当前的头寸。
- 行使期权。

**Note:** 确认TWS处于运行状态，你已经将表格与TWS连接。



### 查看你的投资组合

#### 查看你的投资组合

1. 点击表格底部的Account( 账户) 标签，然后点击工具栏上的Request Account Updates( 请求账户更新) 。
2. 点击表格底部的Portfolio( 投资组合) 标签查看你的投资组合。



### 删除投资组合信息

1. 点击表格底部的 **Account** 标签，然后点击工具栏上的 **Cancel Account Updates( 取消账户更新)** 停止接收投资组合更新。
2. 点击 **Clear Portfolio Data( 清除投资组合数据)** 按钮清除页面上的所有数据。

### 行使期权

你可以行使期权或让期权在投资组合页面上失效。

### 行使期权或让期权作废

1. 点击表格底部的 **Portfolio( 投资组合)** 标签。
2. 在页面最右侧的 *Exercise Options Parameters( 行使期权参数)* 单元内输入值。
  - *Exercise Action( 行使活动)* - 输入 **1** 行使选取的期权； **2** 让期权失效。
  - *Exercise Quantity( 行使数量)* - 输入你希望行使或让其失效的合约数目。
  - *Override( 取代)* - 输入 **1** 取代系统的自然操作； **2** 不取代。

Exercise Options Parameters			
Status	Exercise Action (1,2)	Exercise Quantity	Override (0,1)
	1	10	0
	1	10	0

3. 点击工具栏中的 **Exercise Options( 行使期权)** 。 *Status( 状态)* 栏更新。

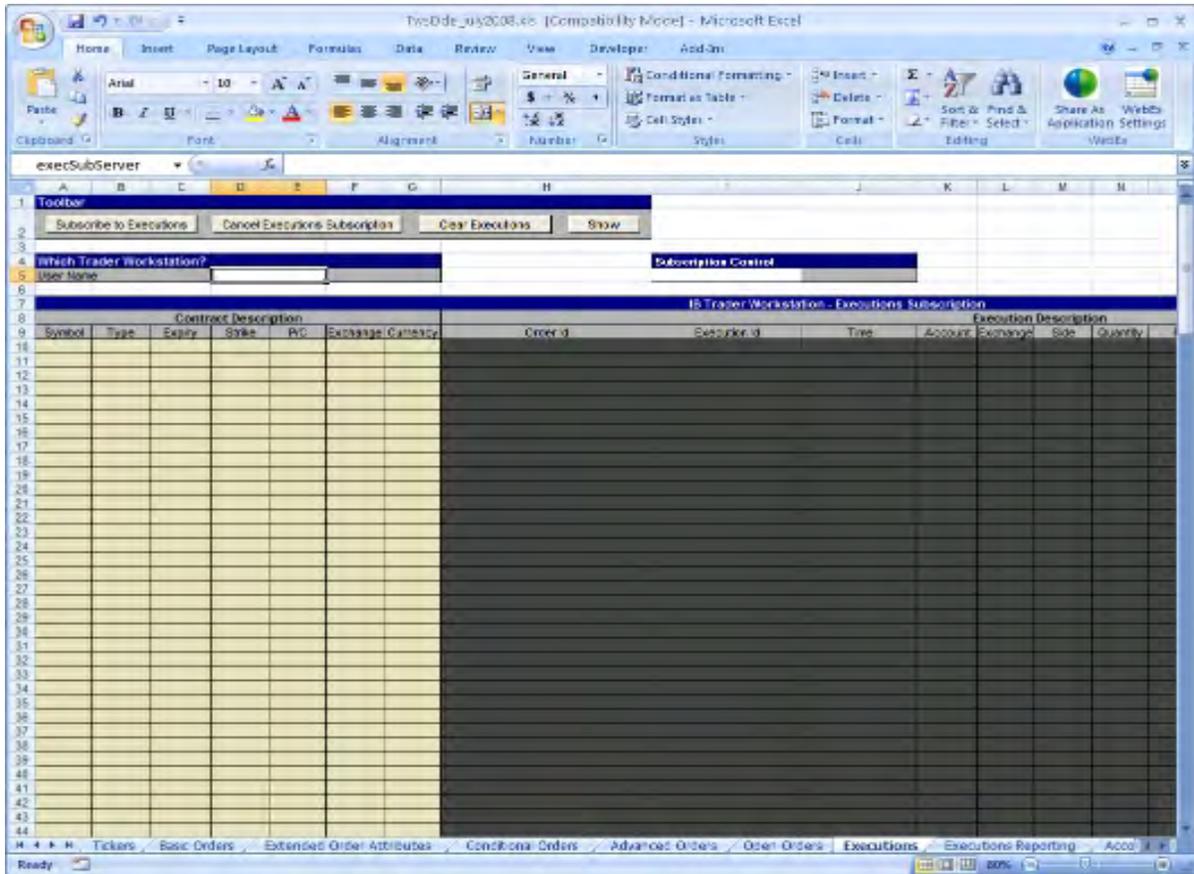
### 投资组合页面工具栏按钮

投资组合页面上的工具栏包括下列按钮。

按钮	描述
Exercise Options( 行使期权)	根据 <i>Exercise Action( 行使行动)</i> 单元内的值，行使选取的期权或让选取的期权失效。
Clear Portfolio Data( 清除投资组合数据)	清除页面上的所有数据。

### 执行页面

当你订阅执行时，执行 ( Executions) 页面显示关于所有已交易的信息。

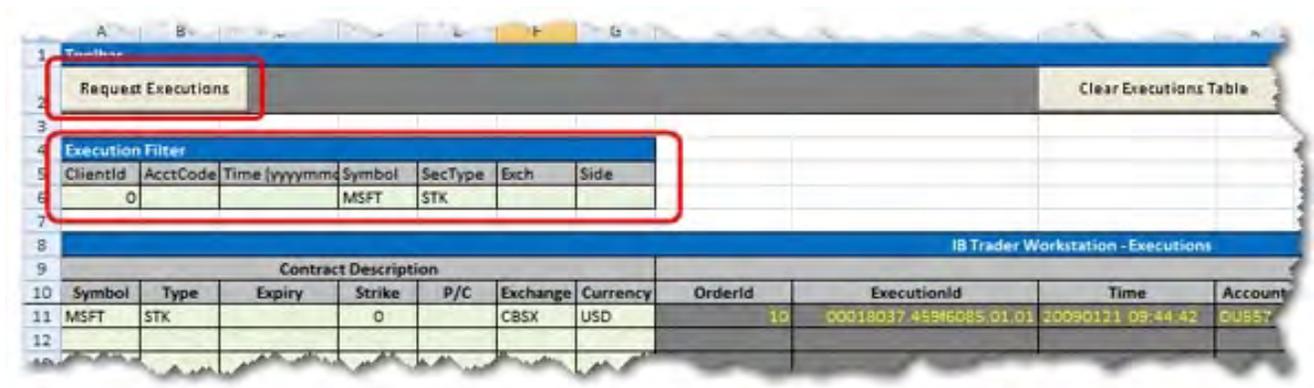


### 查看执行

**Note:** 确认TWS处于运行状态，你已经将表格与TWS连接。

### 查看执行

1. 点击表格底部的**Executions( 执行)** 标签。
2. 还可选择，在**Execution Filter( 执行过滤)** 单元输入值来过滤你的执行。
  - 过滤执行的条件包括客户代号、帐户号码、日期/时间、代码、证券类型、交易所或边。



3. 点击工具栏上的 **Request Executions( 请求执行)** 。

**删除执行数据**

1. 点击工具栏上的 **Clear Executions Table( 清除执行表格)** 。从页面上删除所有数据。

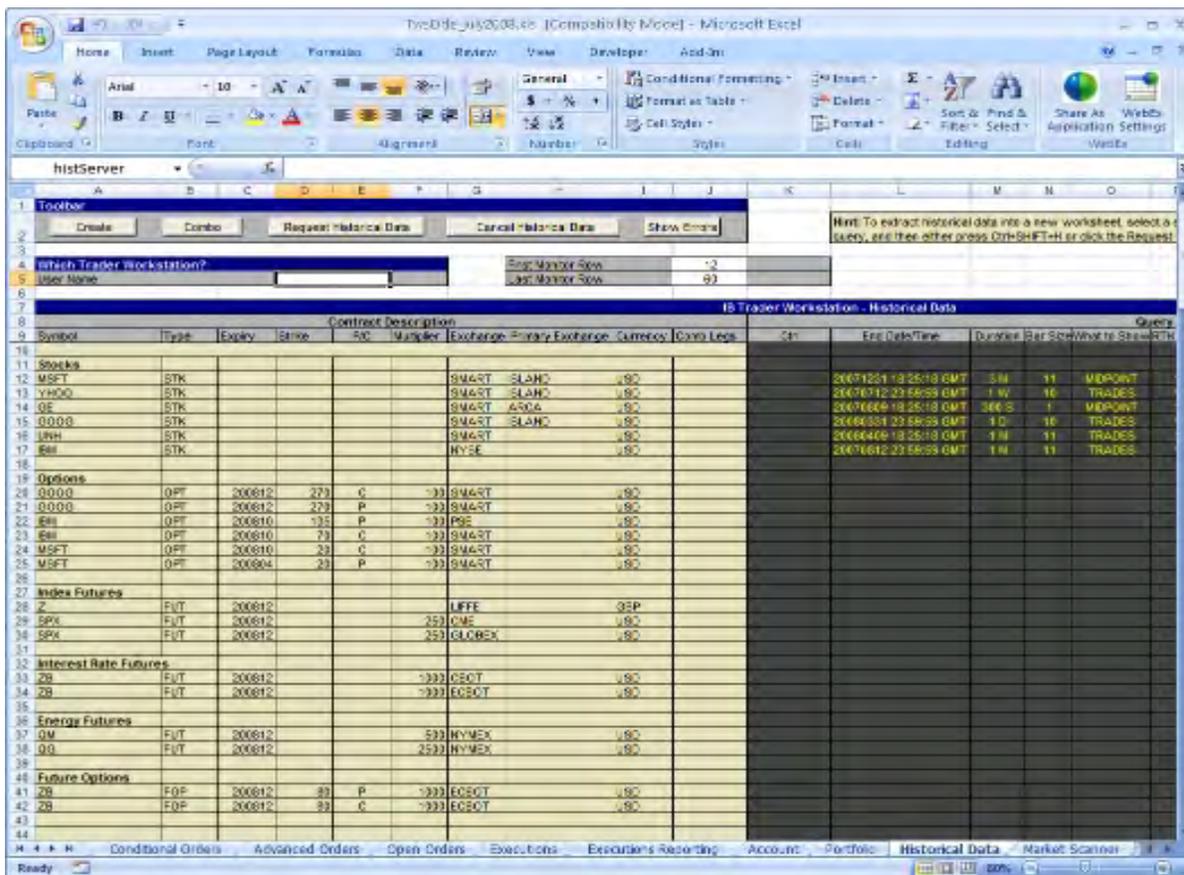
**执行页面工具栏按钮**

执行页面上的工具栏包括下列按钮：

按钮	描述
Request Executions( 请求执行)	查询TWS并返回所有关于有效执行的信息。在你订阅执行之后，这个页面在每次定单执行时更新。
Clear Executions Table( 清除执行表格)	删除页面上的所有执行报告。

**历史数据页面**

使用历史数据页面根据你在查询区域内输入的数据请求产品的历史数据。查询结果显示在另一个表格页面中，如果该页面目前不存在的话，将为查询结果创建一个新的页面。



**Note:** 有关历史数据请求的限制，请参阅[Historical Data Limitations\( 历史数据限制\)](#)。

### 查看历史数据

**Note:** 确认TWS处于运行状态，你已经将表格与TWS连接。

### 请求历史数据

1. 点击表格底部的历史数据( Historical Data) 标签。
2. 在页面的 *Contract Description( 合约描述)* 部分输入信息创建一个代码，或点击工具栏上的 **Create Ticker( 创建代码)** 按钮和 [在代码框中输入要求的信息](#)。
3. 在 *Query Specification( 查询说明)* 区域内输入你的查询参数。关于查询区域的完整描述，请参阅[Historical Data Page Query Specification Fields\( 历史数据页面查询说明区域\)](#)。
4. 选择行，然后点击 **Request Historical Data( 请求历史数据)** 按钮。你的请求状态显示在 *Request Status( 请求状态)* 单元内。

在 *Activate Page( 启动页面)* 单元内，输入 **TRUE( 真)** 在当前窗口的顶部显示结果页面。输入 **FALSE( 假)** 在另一个标签下显示表格页面，而不是在当前窗口的顶部显示。

结果以表格形式在一个新的标签页面上显示，其名称显示在 *Page Name( 页面名称)* 单元内。

### 请求过期合约的历史数据

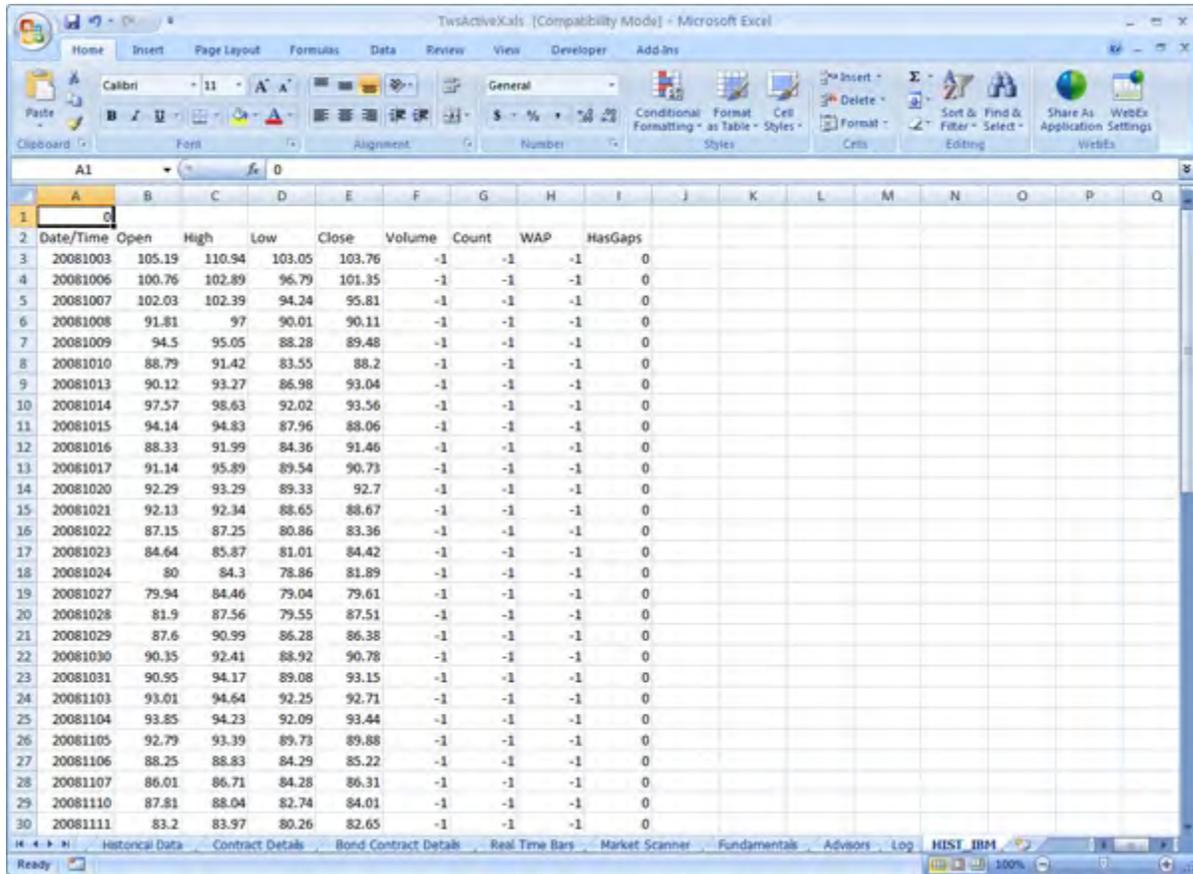
1. 在历史数据页面上，在页面的 *Contract Description( 合约描述)* 部分区域内输入信息创建一个代码，或点击工具栏上的 **Create Ticker( 创建代码)** 按钮和在代码框中输入要求的信息。
2. 在 *Query Specification( 查询说明)* 区域内输入你的查询参数。
3. 在 *Query Specification* 部分的 *Incl Expired( 包括过期的)* 单元内输入 **TRUE**。
4. 择行，然后点击 **Request Historical Data( 请求历史数据)** 按钮。你的请求状态显示在 *Request Status( 请求状态)* 单元内。

在 *Activate Page( 启动页面)* 单元内，输入 **TRUE( 真)** 在当前窗口的顶部显示结果页面。输入 **FALSE( 假)** 在另一个标签下显示表格页面，而不是在当前窗口的顶部显示。

结果以表格形式在一个新的标签页面上显示，其名称显示在 *Page Name( 页面名称)* 单元内。

5. 对过期合约的历史数据查询被限制在合约寿命的最后一年。

下图显示的是一个标准的历史数据结果页面。



历史数据页面查询说明区域

参数	描述
End Date/Time(截止日期/时间)	使用格式 <code>yyymmdd {space}hh:mm:ss{space}tmz</code> 这里允许时区(可选) 出现在结尾的空格之后。
Duration( 区间)	<p>这是请求将要覆盖的时间区间，使用格式 <code>integer {space} unit</code> 说明，这里有效单位是：</p> <ul style="list-style-type: none"> <li>• S(秒)</li> <li>• D(天)</li> <li>• W(星期)</li> <li>• Y(年)</li> </ul> <p>这个单位目前只限制使用一个。 如果没有指定单位，将使用秒。</p>

<p>Bar Size( 柱尺寸)</p>	<p>指定将返回的柱的尺寸。 可以使用下列柱尺寸，用参数值来指定：</p> <table border="1" data-bbox="841 302 1040 1087"> <thead> <tr> <th>柱尺寸字符串</th> <th>整数值</th> </tr> </thead> <tbody> <tr><td>1 秒</td><td>1</td></tr> <tr><td>5 秒</td><td>2</td></tr> <tr><td>15 秒</td><td>3</td></tr> <tr><td>30 秒</td><td>4</td></tr> <tr><td>1 分钟</td><td>5</td></tr> <tr><td>2 分钟</td><td>6</td></tr> <tr><td>3 分钟</td><td>1-6</td></tr> <tr><td>5 分钟</td><td>7</td></tr> <tr><td>15 分钟</td><td>8</td></tr> <tr><td>30 分钟</td><td>9</td></tr> <tr><td>1 小时</td><td>1-0</td></tr> <tr><td>1 天</td><td>1-1</td></tr> </tbody> </table> <p>在查询返回页面上，每个“柱”由表格中的行表示。 如果你指定了一个300秒的区间，一个柱尺寸为“1”(一秒)，你的结果将包括300行，每一行的值等同于一秒，或一秒的柱。注意你还可以使用柱尺寸字符串的整数值或定义柱尺寸的整数值。</p>	柱尺寸字符串	整数值	1 秒	1	5 秒	2	15 秒	3	30 秒	4	1 分钟	5	2 分钟	6	3 分钟	1-6	5 分钟	7	15 分钟	8	30 分钟	9	1 小时	1-0	1 天	1-1
柱尺寸字符串	整数值																										
1 秒	1																										
5 秒	2																										
15 秒	3																										
30 秒	4																										
1 分钟	5																										
2 分钟	6																										
3 分钟	1-6																										
5 分钟	7																										
15 分钟	8																										
30 分钟	9																										
1 小时	1-0																										
1 天	1-1																										
<p>What to Show( 显示什么)</p>	<p>决定数据提取的特性。有效值包括：</p> <ul style="list-style-type: none"> <li>• Trades( 交易)</li> <li>• Midpoint( 中点)</li> <li>• Bid( 买价)</li> <li>• Ask( 卖价)</li> <li>• Bid/Ask( 买/卖价)</li> </ul> <p>所有数据，除了Bid/Ask，包含查询区间的<i>start time</i>( 开始时间)、<i>open</i>( 开盘价)、<i>high</i>( 高价)、<i>low</i>( 低价)、<i>close</i>( 收盘价)、<i>volume</i>( 交易量) 和 <i>weighted average price</i>( 加权平均价) 。</p> <p>对Bid/Ask查询，<i>open</i>和<i>close</i>值分别是时间加权平均买价( <i>time-weighted average bid</i>) 和时间加权平均卖价( <i>time-weighted average offer</i>) 。 这些柱和TWS图表的烛柱是相同的。</p>																										

RTH Only( 仅正常交易时段)	<p>仅正常交易时段。有效值包括：</p> <ul style="list-style-type: none"> <li>• 0 - 返回请求覆盖区间的资源可用数据，包括请求市场处于正常交易时段以外的时间间隔。</li> <li>• 1 - 仅返回请求产品的正常交易时段的数据，不论时间区间是部分或全部处于正常交易时段以外。</li> </ul>
Date Format Style( 数据格式样式)	<p>有效值包括：</p> <ul style="list-style-type: none"> <li>• 1 - 用于返回柱的日期格式是yyyyymmdd{space}{space}hh:mm:dd ( 执行报告使用同样格式) 。</li> <li>• 2 - 返回日期为自1/1/1970 GMT以来的秒数整数。</li> </ul>
Page Name( 页面名称)	<p>结果页面的名称。 出现在表格底部结果页面的标签上。</p>
Activate page( 激活页面)	<p>输入TRUE( 真) 在当前窗口的顶部显示结果页面。 输入FALSE( 假) 在一个新的表格页面上显示结果，而不在当前窗口上部显示。</p>

注意，新页面被加到表格现有标签的右侧。

### 历史数据页面工具栏按钮

历史数据页面上的工具栏包括下列按钮。

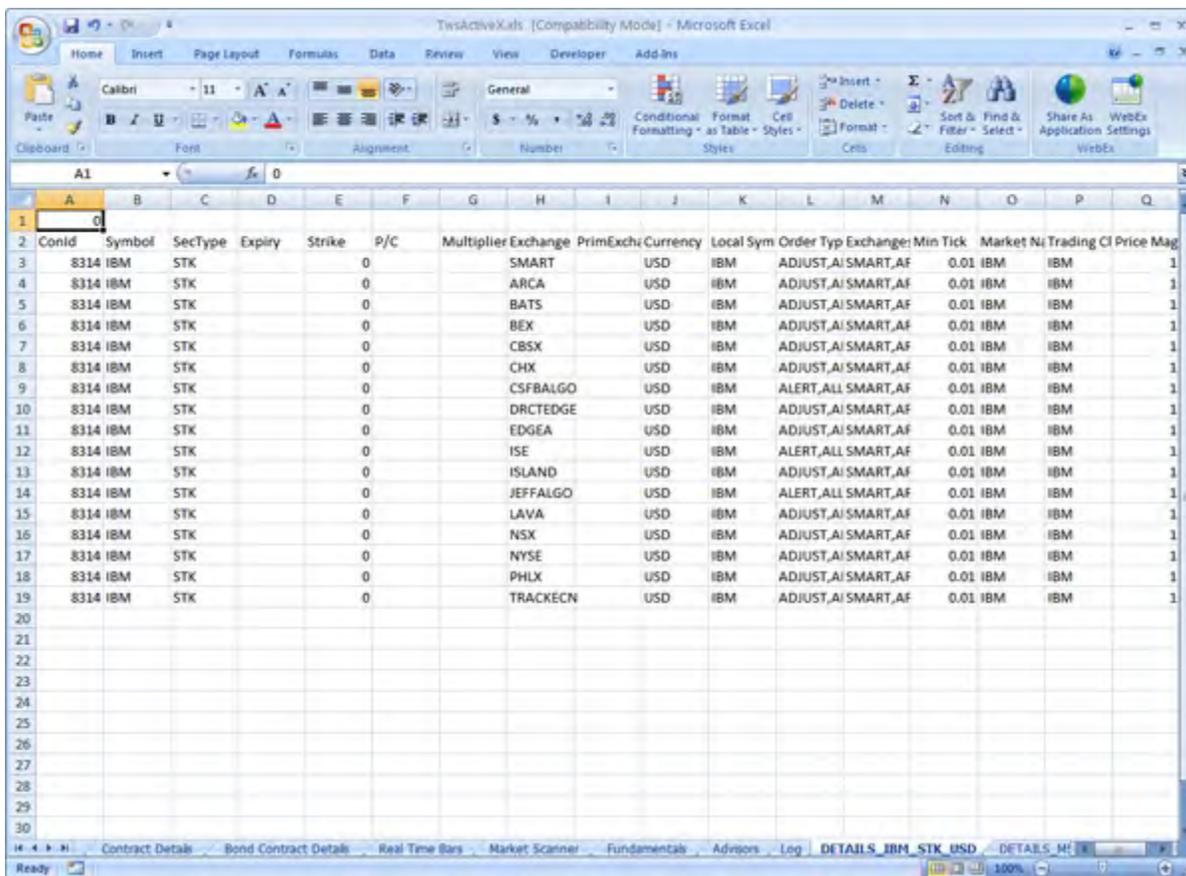
按钮	描述
Create Ticker( 创建代码)	打开 <a href="#">Ticker box( 代码框)</a> 。输入信息创建一个市场数据行。
Combo Legs( 组合边)	打开 <a href="#">Combination Legs box( 组合边框)</a> 。输入合约详细创建组合定单的每一边。
Request Historical Data( 请求历史数据)	对TWS提交你的历史数据请求和在不同的工作页面上显示结果。
Cancel Historical Data( 取消历史数据)	取消历史数据请求。

### 合约详细页面

使用合约详细页面请求合约特定信息，比如支持的定单类型、有效的交易所、合约代码、等等。







合约详细页面工具栏按钮

合约详细页面上的工具栏包括下列按钮：

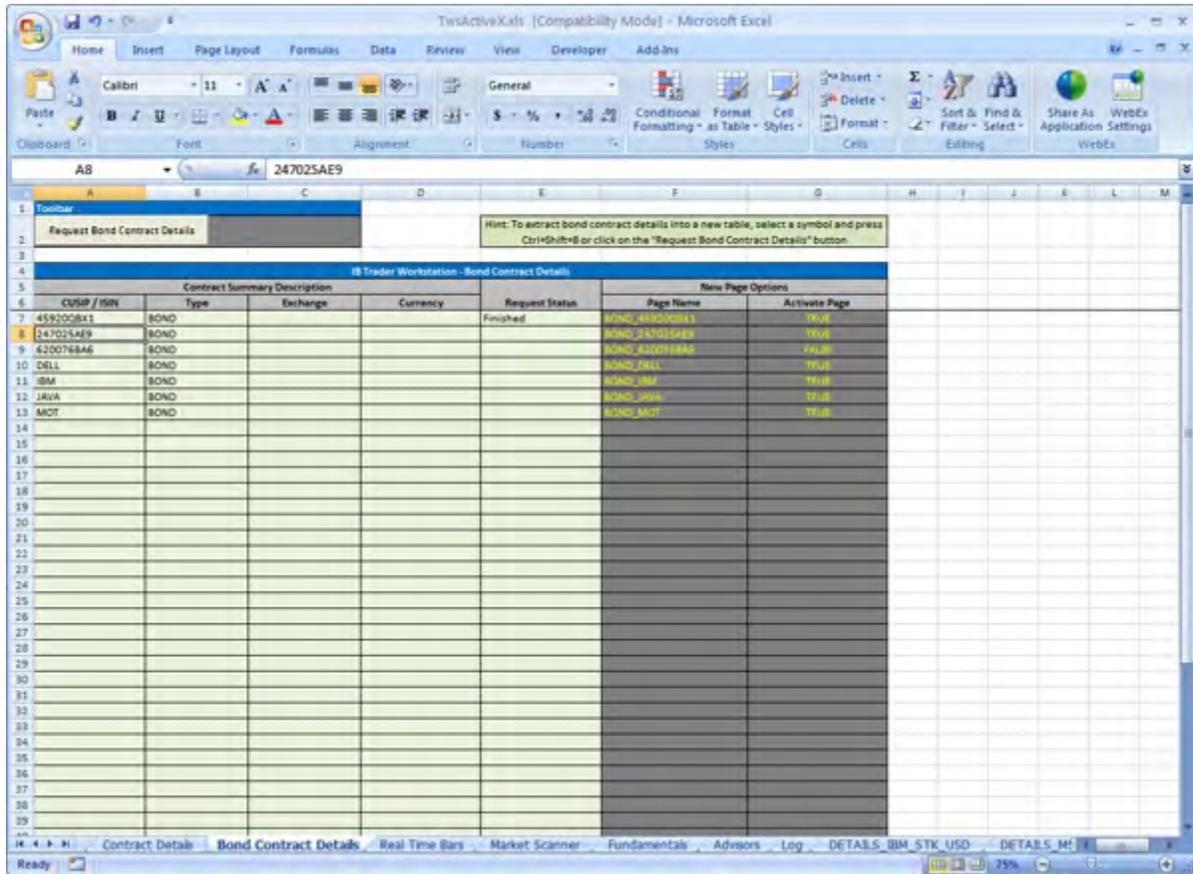
按钮	描述
Request Contract Details( 请求合约详细)	返回选取合约的信息。

债券合约详细页面

使用债券合约详细页面请求债券的特定合约信息，包括利率、评级、债券类型、到期日，等等。

**Note:** 从TWS版本921开始，某些债券合约数据将被限制，将不能从API使用。所有债券合约数据将继续可以通过交易平台使用，仅以下债券合约数据可以通过API使用：

- Contract ID( 合约代码)
- Minimum Tick( 最小跳动)
- CUSIP ( 如果你已经订阅了CUSIP服务)
- Rating( 评级) ( 如果你已经订阅了评级服务)



### 请求债券合约详细

**Note:** 确定TWS处于运行状态，和你已经将表格与TWS连接。

### 请求债券合约详细

1. 点击表格底部的 **Bond Contract Details( 债券合约详细)** 标签打开债券合约详细页面。
2. 选择或输入你希望请求的债券合约详细的代码。
3. 选取该行，然后点击工具栏上的 **Request Bond Contract Details( 请求债券合约详细)**。你请求的状态显示在 *Request Status( 请求状态)* 单元中。

在 *Activate Page( 激活页面)* 单元内，输入 **TRUE** 在当前窗口的顶部显示结果页面。输入 **FALSE** 在表格的不同标签上显示页面，而不在当前窗口的顶部显示。

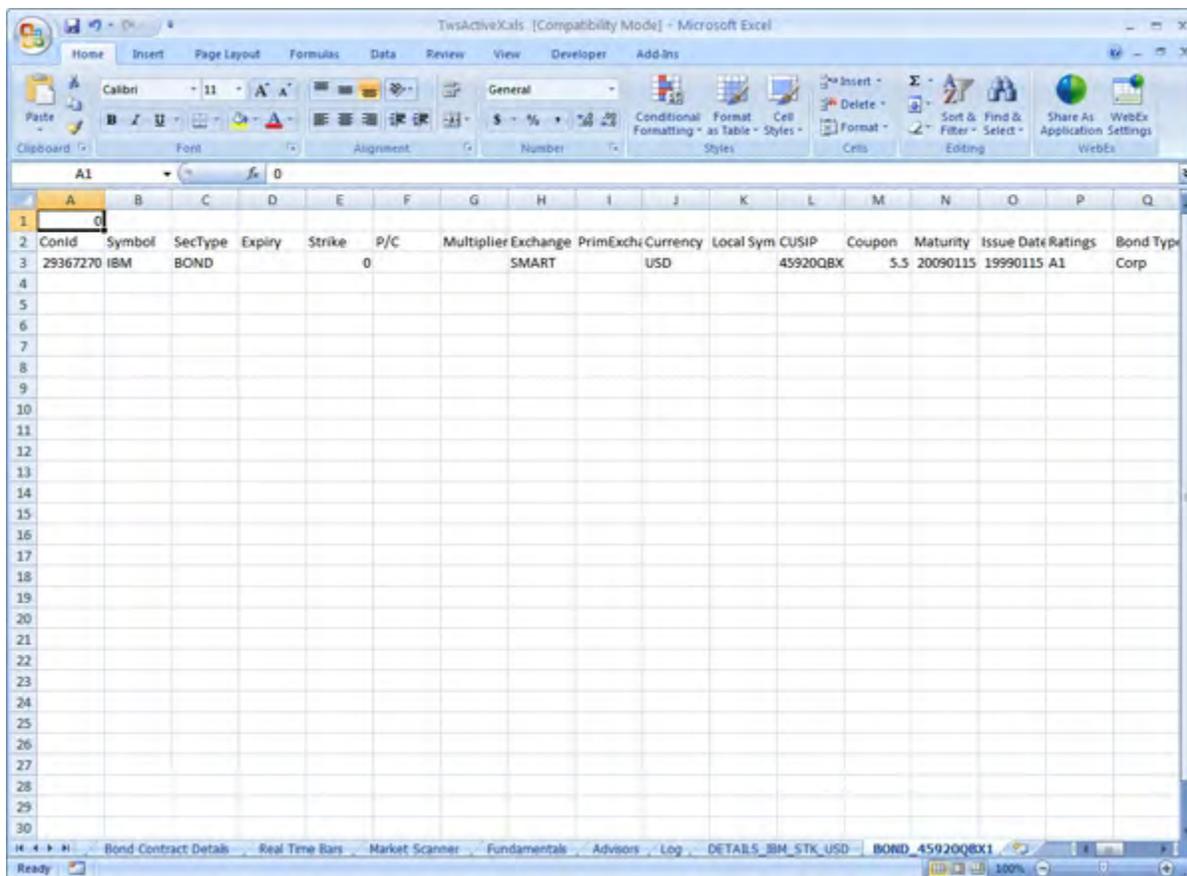
结果显示在表格的一个新标签页面上，其页面名称在 *Page Name( 页面名称)* 单元内指定。

**Note:** 从TWS版本921开始，某些债券合约数据将被限制，将不能从API使用。所有债券合约数据将继续可以通过交易平台使用，仅以下债券合约数据可以通过API使用：

- Contract ID( 合约代码)
- Minimum Tick( 最小跳动)

- CUSIP ( 如果你已经订阅了CUSIP服务)
- Rating( 评级) ( 如果你已经订阅了评级服务)

下面表格显示的是一个典型的合约详细页面。



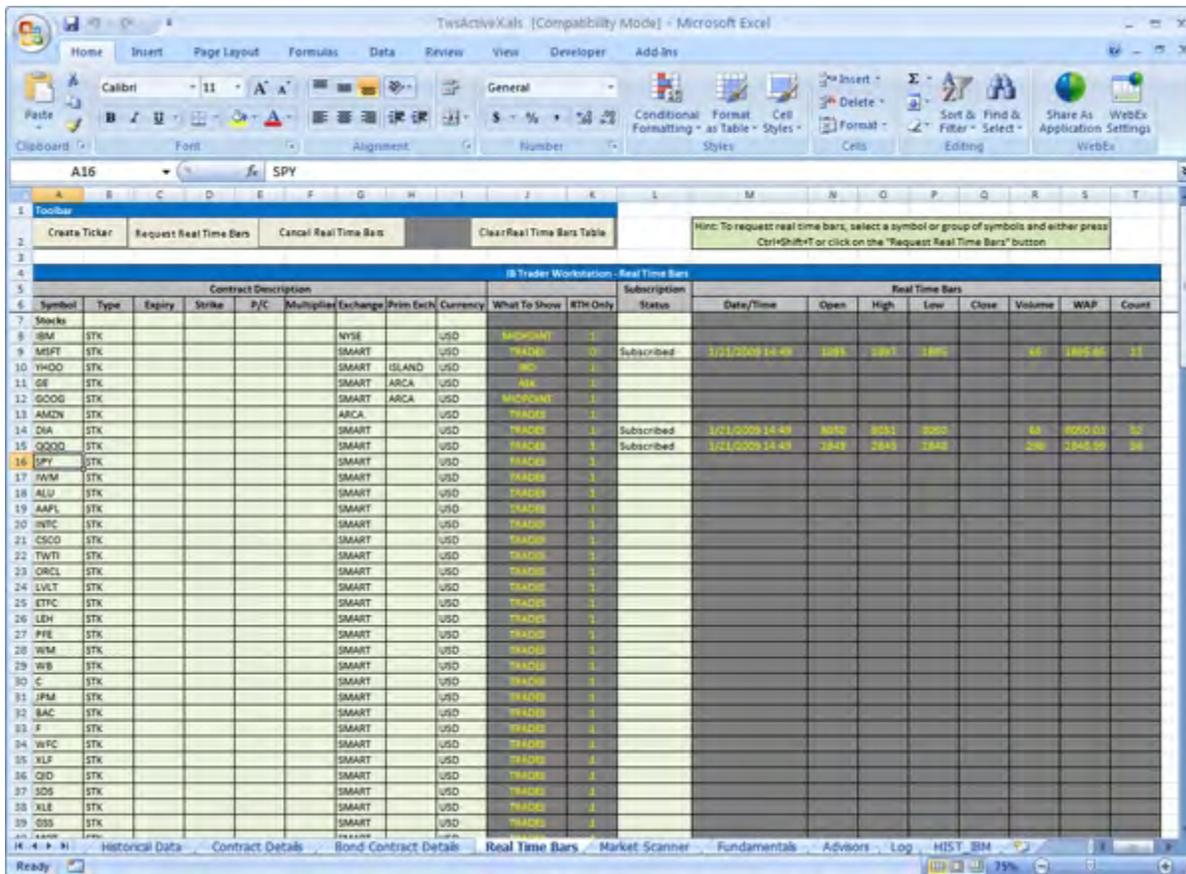
### 债券合约详细页面工具栏按钮

债券合约详细页面的工具栏包括下列按钮：

按钮	描述
Request Bond Contract Details( 请求债券合约详细)	获得选取合约的债券信息数据。

### 实时柱页面

实时柱允许你每五秒钟获取实时市场数据的总结，包括开盘和收盘价，和五秒钟内的高低价( 使用TWS图表术语，我们称这些为五秒期间的“柱”)。你还可以获取显示交易、中点价、买价或卖价的数据。你在实时柱页面请求实时柱，显示如下。



**请求实时柱**

**Note:** 确定TWS处于运行状态，和你已经将表格与TWS连接。

1. 点击表格底部的**Real Time Bars( 实时柱)** 标签打开实时柱页面。
2. 选择或输入你希望请求的实时柱的代码。
3. 对选取的行输入下列信息：
  - 在*What to Show( 显示什么)* 单元中，输入TRADES( 交易)、BID( 买价)、ASK( 卖价)或MIDPOINT( 中点价)。
  - 在*RTH Only( 仅正常交易时段)* 单元中，输入0返回所有请求时段的可用数据，包括请求的市场处于正常交易时段以外的时间间隔的数据。输入1仅返回请求产品的正常交易时段内的数据，尽管请求区间可能部分或全部属于正常交易时段以外。
4. 选取该行，然后点击工具栏上的**Request Real Time Bars( 请求实时柱)**。你请求的状态显示在*Subscription Status( 订阅状态)* 单元内。

结果显示在页面右侧的*Real Time Bars( 实时柱)* 单元内。

**实时柱页面工具栏按钮**

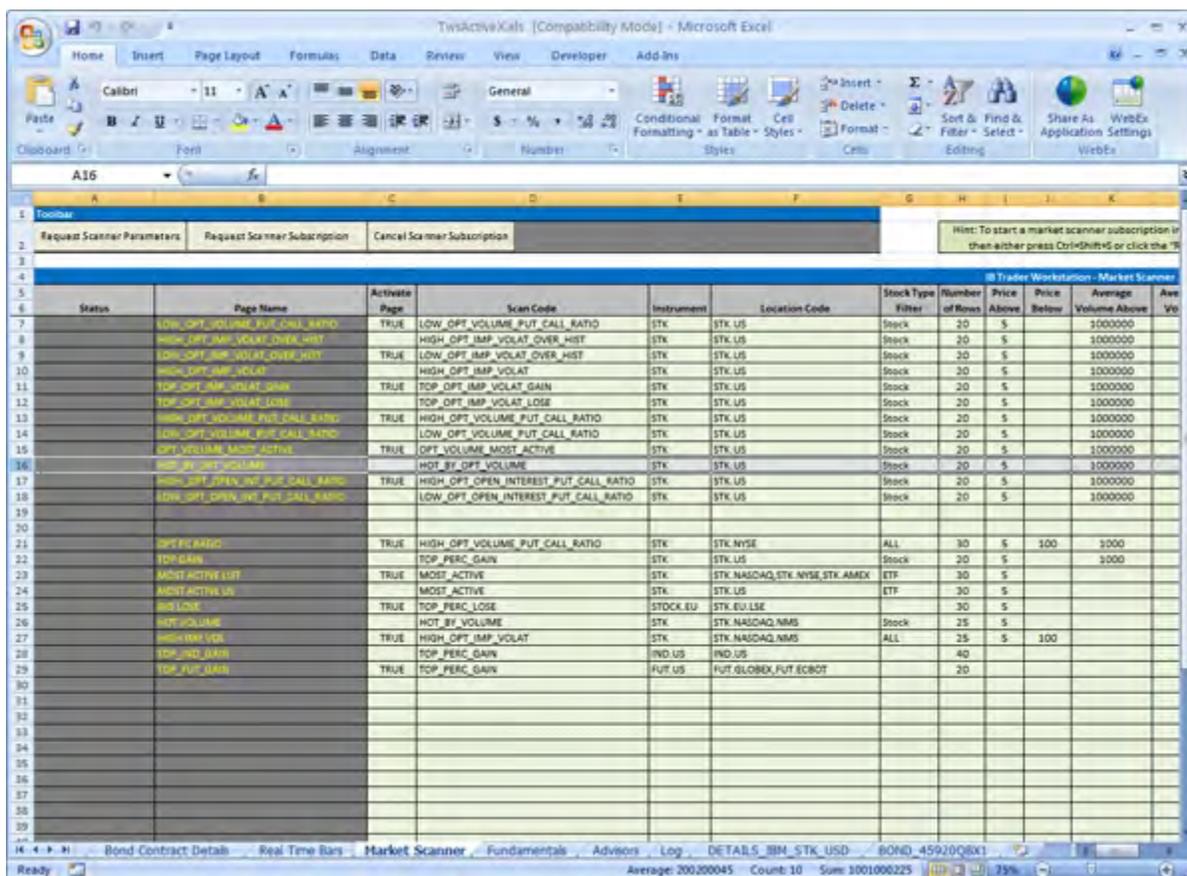
实时柱页面上工具栏包括下列按钮。

按钮	描述
Create Ticker( 创建代码)	打开代码框。输入信息创建一个市场数据行。
Request Real Time Bars( 请求实时柱)	提交给TWS你的实时柱请求和在页面右侧的深灰色单元内显示结果。
Cancel Real Time Bars( 取消实时柱)	取消实时柱请求。
Clear Real Time Bars Table( 清除实时柱表格)	从页面上删除所有实时柱数据。

### 市场扫描仪页面

使用市场扫描仪页面订阅TWS市场扫描仪。这些扫描仪允许你定义条件和设定过滤器以返回满足所有扫描条件最上面的x个底层证券。扫描按实时连续更新。

你可用从这个页面显示市场扫描仪参数。



## 开始市场扫描仪订阅

**Note:** 确定TWS处于运行状态，以及你已经将表格与TWS连接。

### 开始扫描仪订阅

1. 点击表格底部的**Market Scanner( 市场扫描仪)** 标签。
2. 选中一个现有的扫描仪行，或为一个不同的市场扫描仪输入信息：
  - a. 在*Page Name( 页面名称)* 单元内输入扫描结果页面名称。
  - b. 在*Activate Page( 激活页面)* 单元内输入 **TRUE**或**FALSE**。

设定这个单元为 TRUE将强制扫描结果页面在每次更新时弹出到你的应用程序的前面。停止这个运作，只要将这个区域内的值设为 FALSE。

- c. 在页面浅灰色区域输入其它的扫描参数值。你可以从市场扫描仪参数获取所有的扫描编码。
3. 点击工具栏上的**Request Scanner Subscription( 请求扫描仪订阅)** 。一个新的扫描仪页面将被创建，并在完成订阅后显示。

### 市场扫描仪参数

你可以从市场扫描仪页面显示所有的市场扫描仪参数。扫描仪参数以一个XML文件的形式从TWS返回到表格中。

**Note:** 确定TWS处于运行状态，以及你已经将表格与TWS连接。

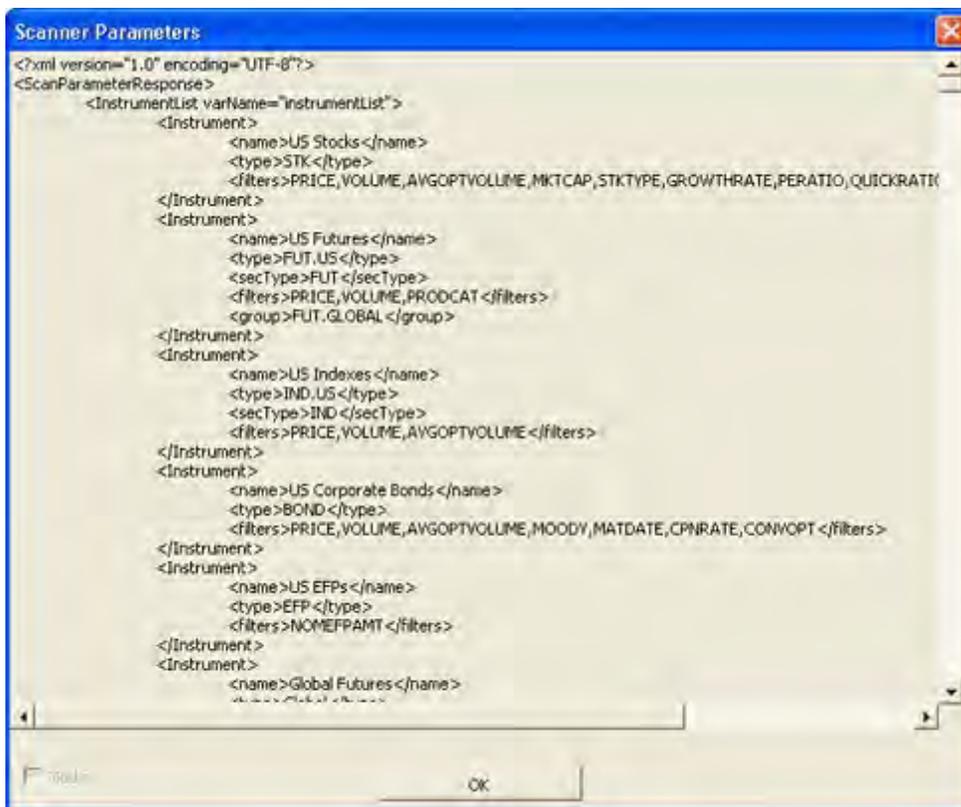
### 查看扫描仪参数

1. 点击表格底部的**Market Scanner( 市场扫描仪)** 标签。
2. 点击工具栏上的**Request Scanner Parameters( 请求扫描仪参数)** 。

整个扫描仪参数的XML文件显示在窗口上。

3. 在你的计算机中将参数保存为一个方便使用的文件，在扫描仪参数窗口中手动选择部分或全部XML文件的内容，然后将其复制和粘贴到一个不同的文字文件中。
4. 点击**OK**关闭扫描仪参数窗口。

扫描仪参数窗口显示在下一个页面上。



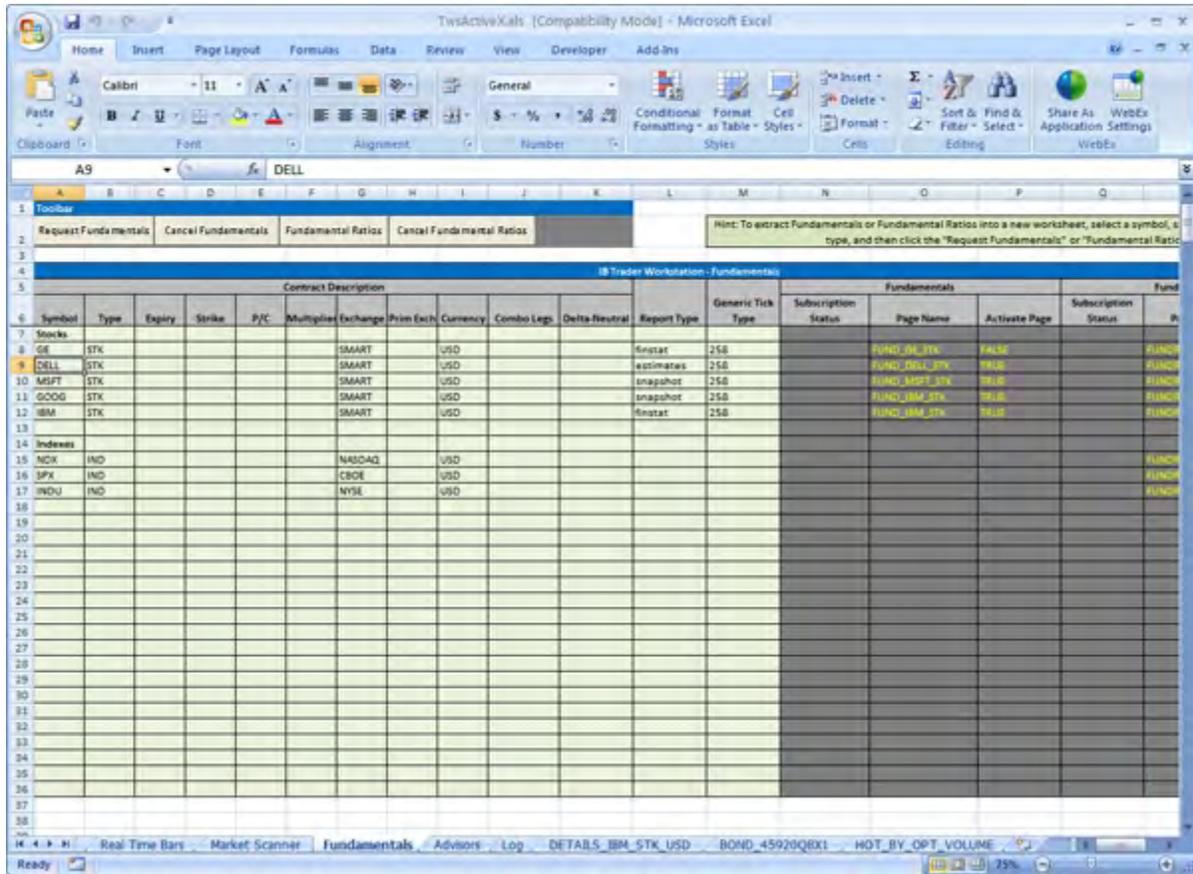
### 市场扫描仪页面工具栏按钮

市场扫描仪页面的工具栏包括下列按钮。

按钮	描述
Request Scanner Parameters( 请求扫描仪参数)	在不同的窗口中以XML文件形式显示所有的扫描仪参数。
Request Scanner Subscription( 请求扫描仪订阅)	为选取的市场扫描仪结果创建和显示一个新的页面。
Cancel Scanner Subscription( 取消扫描仪订阅)	取消市场扫描仪。

### 基本面页面

使用基本面页面接收路透社全球基本面数据和基本面比率。接收这个数据之前必须有通过账户管理中设置的路透社基本面订阅。



### 接收路透社全球基本面数据和基本面比率

**Note:** 确定TWS处于运行状态，以及你已经将表格与TWS连接。

1. 点击表格底部的 **Fundamentals( 基本面)** 标签显示基本面页面。
2. 在 *Contract Description( 合约描述)* 单元中输入关于你希望获得基本面数据或比率的合约信息。
3. 在 *Report Type( 报告类型)* 单元中，输入你希望查看的报告类型：
  - o **finstat** - 财务报告
  - o **estimates** - 估计
  - o **snapshot** - 概括
4. 在 *Generic Tick Type( 一般跳动类型)* 单元内输入 **258** 做为跳动类型的值。有关一般跳动类型值的详细信息，请参阅 [Generic Tick Types\( 一般跳动类型\)](#)。
5. 在页面的 *Fundamentals* 部分输入下列基本面数据信息，或在 *Fundamental Ratios( 基本面比率)* 部分输入基本面比率：
  - a. 基本面数据和比率结果显示在表格中的一个新页面上。在 *Page Name* 单元中输入结果页面的名称。
  - b. 在 *Activate Page( 激活页面)* 单元内输入 **TRUE** 或 **FALSE**。



设定这个单元为TRUE将强制结果页面在每次更新时弹出到你的应用程序的前面。停止这个运作，只要将这个区域内的值设为FALSE。

6. 完成下列之中其一：

- 点击工具栏上的 **Request Fundamentals( 请求基本面)** 查看基本面数据。一个新的结果页面将被创建，并在完成请求后显示。
- 点击工具栏上的 **Fundamentals Ratios( 基本面比率)** 查看基本面比率。一个新的结果页面将被创建，并在完成请求后显示。

你的请求粘贴出现在 *Subscription Status( 订阅状态)* 单元内。

### 基本面页面工具栏按钮

市场扫描仪页面的工具栏包括下列按钮。

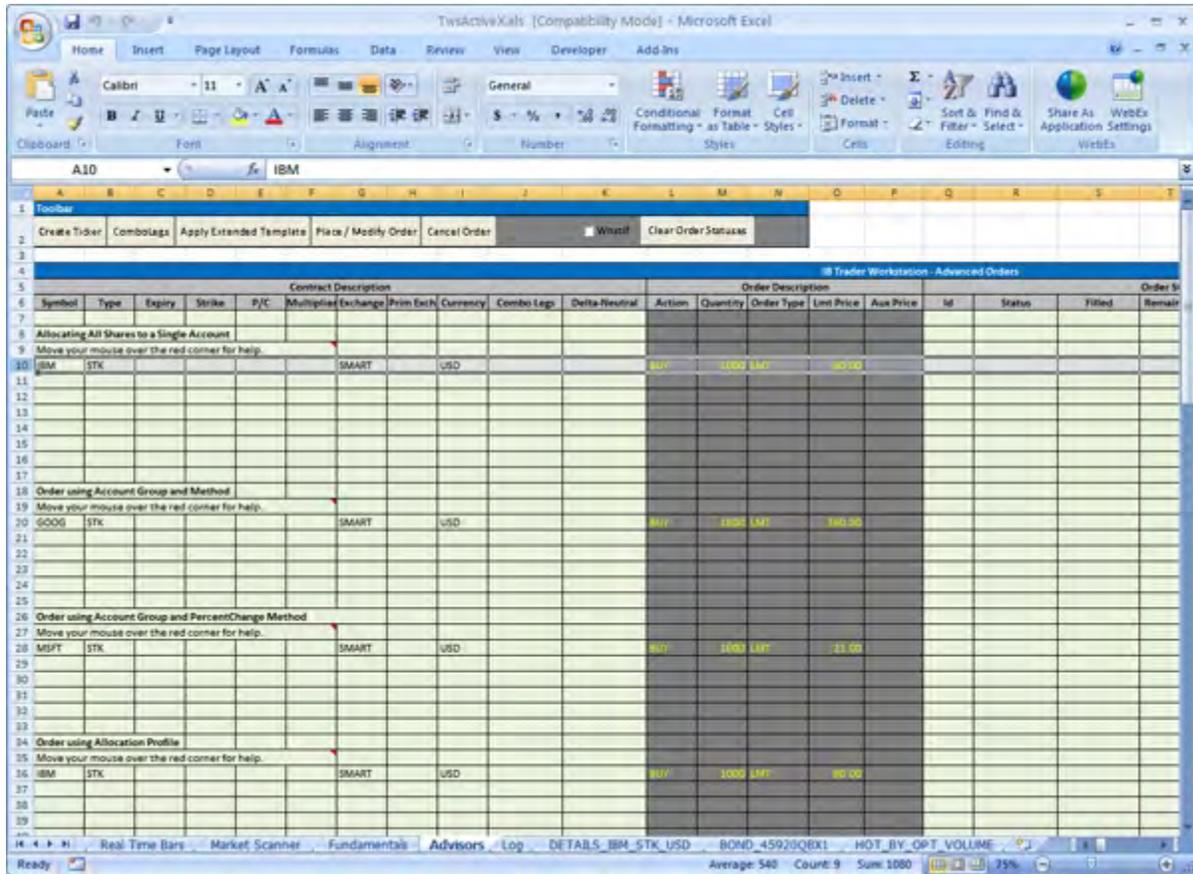
按钮	描述
Request Fundamentals( 请求基本面)	请求基本面数据，该数据显示在表格中的一个新的页面上并基于你在该页面上输入的信息。
Cancel Fundamentals( 取消基本面)	取消基本面数据。
Fundamental Ratios( 基本面比率)	请求基本面比率，该比率显示在表格中的一个新的页面上并基于你在该页面上输入的信息。
Cancel Fundamental Ratios( 取消基本面比率)	取消基本面比率。

### 顾问页面

如果你是一个金融顾问并管理多个账户，使用顾问页面创建FA定单进行：

- 分配股票到一个单个管理的账户
- 使用FA账户组和方法
- 使用分配方案

**Note:** 在你可以通过 ActiveX for Excel API样本表格下达FA定单之前，你在TWS中必须设置你的管理的账户、账户组、方法和分配方案。



### 分配股票到一个单个账户

你可以使用 **Advisors** 页面设置定单并将定单中的所有股票分配给一个单个账户。

**Note:** 确定TWS处于运行状态，和你已经将表格与TWS连接。

#### 分配股票到一个单个账户：

1. 在TWS中创建账户组。
2. 点击表格底部的 **Advisors( 顾问)** 标签。
3. 在 *Contract Description( 合约描述)* 单元内输入合约信息，然后在 *Order Description( 定单描述)* 单元内输入定单信息。
4. 点击 **Extended Order Attributes( 扩展定单属性)** 标签。在 *Value( 值)* 单元内为 *Account( 仅机构账户)* 扩展定单属性输入账户代码。
5. 点击 **Advisors** 标签。
6. 选中定单行，然后点击 **Apply Extended( 应用扩展的)** 按钮将 *Account* 定单属性值应用到定单中。*Account* 值被应用到选取的定单并显示在页面的 *Extended Order Attributes( 扩展定单属性)* 部分。
7. 点击 **Place/Modify Order( 下达/修改定单)** 按钮。

8. 当你完成了分配股票到账户后，从 **Extended Order Attributes** 页面删除 *Account* 值。如果你不删除这个值，它将被应用到所有从 ActiveX for Excel 表格中行动的后续定单中。

或者，如果你通过选择工具栏上 **WhatIf( 假设)** 复选框下达了定单，你可以接收来自该定单的保证金和佣金信息。这种情况下，你的定单并没有被下达。取消选择复选框下达你的定单，而不事先查看保证金和佣金信息。

#### 使用FA账户组和方法下达定单

你还可以使用 **Advisors( 顾问)** 页面使用FA账户组和FA方法设置定单。

#### 使用FA账户组和FA方法下达定单：

1. 在TWS中创建FA账户组和FA方法。
2. 点击表格底部的 **Advisors** 标签。
3. 在 *Contract Description( 合约描述)* 单元内输入合约信息，然后在 *Order Description( 定单描述)* 单元内输入定单信息。
4. 点击 **Extended Order Attributes( 扩展定单属性)** 标签。输入下列扩展定单属性值：
  - FA Group - 输入账户组名称。
  - FA Method - 输入用于这个定单的分配方法名称。
  - FA Percentage - 输入用于这个定单的PctChange分配方法的百分比。该属性仅用于使用该方法的FA组。
5. 点击 **Advisors** 标签。
6. 选中定单行，然后点击 **Apply Extended( 应用扩展的)** 按钮将扩展定单属性应用到定单中。FA Group, FA Method 和 FA Percentage 的值被应用于选取的定单并在页面的 *Extended Order Attributes* 部分内显示。
7. 点击 **Place/Modify Order** 按钮。
8. 当你完成了分配股票到账户后，删除你在 **Extended Order Attributes** 页面中输入的值。如果你不删除这个值，它将被应用到所有从 ActiveX for Excel 表格中行动的后续定单中。

或者，如果你通过选择工具栏上 **WhatIf( 假设)** 复选框下达了定单，你可以接收来自该定单的保证金和佣金信息。这种情况下，你的定单并没有被真正下达。取消选择复选框下达你的定单，而不事先查看保证金和佣金信息。

#### 使用一个分配方案下达定单

你还可以使用 **Advisors** 页面使用FA分配方案设置定单。

#### 使用FA分配方案下达定单：

1. 在TWS中创建FA分配方案。
2. 点击表格底部的 **Advisors** 标签。
3. 在 *Contract Description( 合约描述)* 单元内输入合约信息，然后在 *Order Description( 定单描述)* 单元内输入定单信息。
4. 点击 **Extended Order Attributes( 扩展定单属性)** 标签。在 *Value* 区域为 *FA Profile( FA方案)* 扩展定单属性输入分配方案名称。

5. 点击 **Advisors** 标签。
6. 选中定单行，然后点击 **Apply Extended( 应用扩展的)** 按钮将扩展定单属性应用到定单中。*FA Profile* 的值被应用于选取的定单并在页面的 *Extended Order Attributes* 部分内显示。
7. 点击 **Place/Modify Order**按钮。
8. 当你完成了分配股票到账户后，删除你在 **Extended Order Attributes** 页面中输入的 *FA Profile* 值。如果你不删除这个值，它将被应用到所有从 ActiveX for Excel 表格中行动的后继定单中。

或者，如果你通过选择工具栏上 **WhatIf( 假设)** 复选框下达了定单，你可以接收来自该定单的保证金和佣金信息。这种情况下，你的定单并没有被真正下达。取消选择复选框下达你的定单，而不事先查看保证金和佣金信息。

### 顾问页面工具栏按钮

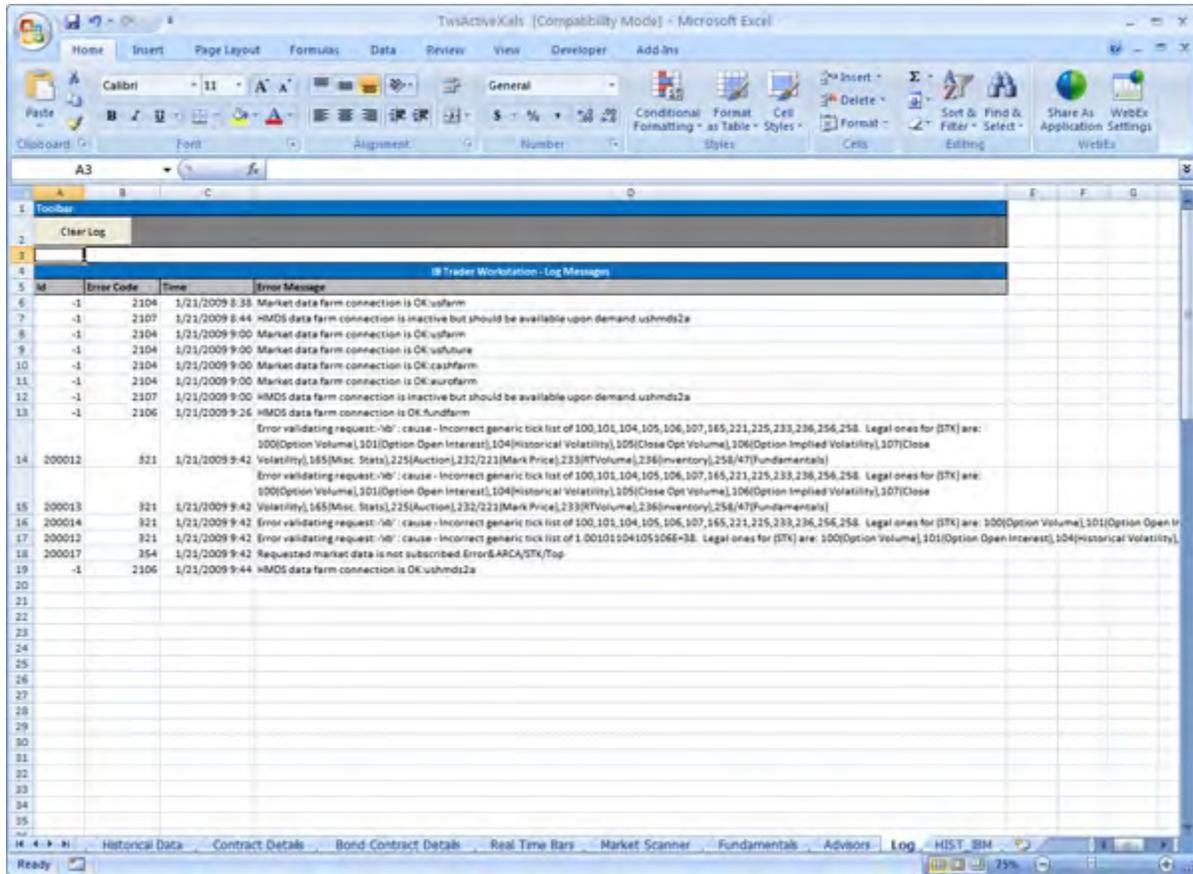
基本定单页面上的工具栏包括下列按钮：

按钮	描述
Create Ticker( 创建代码)	打开代码框。输入信息创建一个市场数据行。
Combo Legs( 组合边)	打开组合边框。输入合约详细创建组合定单的每个边。
Apply Extended Template( 应用扩展模版)	应用扩展定单属性页面上当前的值到选中的定单行。
Place/Modify Orders( 下达/修改定单)	在你完成了定单描述区域和定义了任何扩展属性后，点击创建选取合约的定单。
Cancel Order( 取消定单)	这个按钮取消你选中的定单。
Clear Order Statuses( 清除定单状态)	清除 <i>Order Status( 定单状态)</i> 单元内的所有信息。

### 记录页面

记录页面显示登录进入TWS时接收到的所有错误信息和使用 ActiveX for Excel 表格。你可以点击工具栏上的 **Clear Log( 清除记录)** 按钮清除页面上的所有信息。

下面是一个典型的记录页面举例：





## POSIX

这一章介绍POSIX API。

POSIX API是基于我们的[C++](#) API编码。C++编码是重构的，所有它可以在任何POSIX-兼容的平台下构造。使用这个新的POSIX API创建一个Linux的TWS API，和Windows的非MFC应用程序。

**Note:** 尽管已经保存了之前存在的公共接口，你仍然必须重新编译你的客户应用程序。

我们也包括了一个POSIX测试客户端。POSIX API目录的API安装目录包括：`PosixSocketClient` 和 `TestPosixSocketClient`。POSIX测试客户端是和C++套接字客户端使用相同的方法，另外当套接字的数据可用于读/写时，客户必须调用时还可以有几种额外的方法供选择。参考`TestPosixSocketClient`做为一个举例。请注意，这个测试客户端是非常简化的。对实际的POSIX API应用程序，你将必须使用一个可以管理几个插口和/或非同步事件的系统。

在Windows计算机上运行POSIX测试客户端，请参阅[在Windows计算机上运行POSIX客户端](#)。

## 在 Windows 计算机上运行 POSIX 客户端

### 在 Windows 计算机上运行 POSIX 客户端

1. 在指令提示符下运行 `vcvars32.bat`。下面例子中，`vcvars32.bat` 是位于 `C:\Program Files\Microsoft Visual Studio 8\VC\bin\`。

```
C:\>cd c:\Program Files\Microsoft Visual Studio 8\VC\bin
C:\Program Files\Microsoft Visual Studio 8\VC\bin>vcvars32.bat
```

2. 如果你成功运行 `vcvars32.bat`，指令提示符应该如下：

```
C:\WINDOWS\system32\cmd.exe
C:\>cd c:\Program Files\Microsoft Visual Studio 8\VC\bin
C:\Program Files\Microsoft Visual Studio 8\VC\bin>vcvars32.bat
C:\Program Files\Microsoft Visual Studio 8\VC\bin>"C:\Program Files\Microsoft Visual Studio 8\Commn7\Tools\vcvars32.bat"
Setting environment for using Microsoft Visual Studio 2005 x86 tools.
C:\Program Files\Microsoft Visual Studio 8\VC\bin>_
```

3. 导航进入 `C:\...\TestPosixSocketClient` 在相同的指令提示符窗口。下面举例中，`TestPosixSocketClient` 位于 `C:\IB_API_963`。

```
C:\IB_API_963\TestPosixSocketClient>
```

4. 在指令提示符处输入 `nmake -f Makefile.win`。

```
C:\IB_API_963\TestPosixSocketClient> nmake -f Makefile.win
```

5. 通过运行和 `C:\...\TestPosixSocketClient` 相同目录中的 `PosixSocketClientTest.exe` 运行 POSIX 样本应用程序。

```
C:\IB_API_963\TestPosixSocketClient>PosixSocketClientTest.exe
```



## 参考表

这一章包括下列TWS API参考信息：

- [API消息编码](#)
- [跳动类型](#)
- [普通跳动类型](#)
- [订单类型和IB算法\( IBAlgos\)](#)
- [扩展订单属性](#)
- [可用的市场扫描仪](#)
- [支持的时区](#)
- [智能组合传递](#)

## API消息编码

这一节列出所有的API [Error\( 出错\)](#)、[System\( 系统\)](#) 和 [Warning\( 警告\)](#) 消息编码和它们的描述。

下面列出的消息编码中以冒号(:)结尾的显示额外的信息。

### 出错编码

编码	描述
100	已经超出每秒的最大消息速率。
101	已经达到最多的代码数目。
102	重复代码代号。
103	重复定单代号。
104	不能够修改已执行的定单。
105	被修改的定单与原始定单不匹配。
106	不能发送定单代号：
107	不能发送不完整的定单。
109	价格超出了定单默认框架内设定的百分比。定单将不会被发送。
110	价格不符合该合约的最低价格变量。
111	TIF( 有效时间类型) 和定单类型不相容。
113	Tif选择应该设定为DAY( 日) 对MOC和LOC定单。
114	相对定单仅适用于股票。
115	美国股票的相对定单只可以发送到SMART、SMART_ ECN、INSTINET、或PRIMEX。
116	定单不能被传递到一个不存在的交易所。
117	批量定单尺寸必须至少为50。
118	VWAP定单必须被传递到VWAP交易所。
119	通过VWAP交易所仅可以下达VWAP定单。
120	今天已经太迟下达VWAP定单。
121	定单的BD标识无效。 查看"Destination( 目的地) "和"BD"标识。
122	没有找到定单的请求标签：
123	没有conid记录：
124	没有conid的可用市场规则：

编码	描述
125	买价必须和最佳卖价相等。
126	卖价必须和最佳买价相等。
129	VWAP定单必须至少在开始前三分钟提交。
131	一扫光标识和显示尺寸仅适用于传递到SMART的美国股票，将被忽略。
132	没有一个清算账户不等发送该定单。
133	提交新定单失败。
134	修改定单失败。
135	不能找到定单的代号 =
136	不能取消该定单。
137	VWAP定单只能在开始前三分钟内取消。
138	无法解析代码请求：
139	解析错误：
140	尺寸值应该是整数：
141	价格值应该加倍：
142	机构客户账户没有账户信息
143	请求的代号不是整数。
144	定单尺寸和总的股票分配不匹配。 调整股票分配，右键点击定单并选择“Modify( 修改) > Share Allocation( 股票分配) 。”
145	验证输入区域错误 -
146	无效启动方法。
147	条件合约信息不全。
148	条件定单只能在定单类型设置为限价或市价单时才能发送。
151	没有用户名不能发送该定单。
152	可能不可以为该定单指定“隐藏”定单属性。
153	EFPs只能为限价单。
154	不可以对暂停交易的证券发送定单。
155	sizeOp定单必须拥有用户名和账户。

编码	描述
156	SizeOp定单必须发送到IBSX
157	定单可以为Iceberg(冰山)或Discretionary(委托)。请删除委托额或显示尺寸。
158	你必须指定一个抵消额或一个百分比抵消值。
159	百分比抵消值必须是在0%和100%之间。
160	尺寸值不能为零。
161	当定单不处于可取消状态时试图取消。定单permId =
162	历史市场数据服务错误信息。
163	指定的价格将违反默认定单设置中指定的百分比限制。
164	没有市场数据用来查看价格百分比违反。
165	历史市场数据服务请求信息。
166	HMDS过期合约犯规。
167	VWAP定单时间必须为未来时间。
168	委托额不符合该合约的最低价格变量。

编码	描述
200	没有找到请求的证券定义。
201	定单被拒 - 原因:
202	定单被取消 - 原因:
203	证券<security>不存在或不适用于该账户。

编码	描述
300	找不到EId的代码代号:
301	无效代码行动:
302	错误解析停止代码字符串:
303	无效行动:
304	无效账户值行动:

305	请求解析错误，请求已被忽略。
306	处理DDE请求错误：
307	无效请求题目：
308	无法在TWS中创建‘API’页面，因为现有页面数量已达最大限度。
309	<p>市场深度请求已达到最大数( 3) 。</p> <p><b>Note:</b> TWS目前限制用户的不同的市场深度请求的最大数目是3。同样的限制也适用于API客户，但API客户可以对同一个证券提交多个市场深度请求。</p>
310	不能找到订阅的市场深度的tickerId:
311	原始是无效的。
312	组合详细无效。
313	'<leg number>'边的组合详细无效。
314	证券类型'BAG'需要组合边详细。
315	股票组合边仅限于SMART( 指南) 定单传递。
316	市场深度数已被暂停。请重新订阅。
317	市场深度数据已被重置。请在使用任何新的输入前清空深度在册内容。
319	无效记录级别<log level>
320	当阅读API客户请求时服务器出错。
321	当验证API客户请求时服务器出错。
322	当处理API客户请求时服务器出错。
323	服务器错误：原因 - %s
324	当阅读DDE客户请求时服务器出错( 缺少信息) 。
325	这个交易所和定单类型的组合不支持委托单。
326	无法连接，客户代号已被使用。使用不同的客户代号重试。
327	仅clientId设置为0的API连接可以设置适当的自动绑定TWS定单。
328	跟踪止损定单仅可以被附加到限价单或止损限价单。
329	定单修改失败。不能改变为新的定单类型。

330	仅FA或STL客户可以请求管理账户列表。
331	内部错误。FA或STL没有任何管理账户。
332	定单结构的账户编码无效。
333	无效股票分配句法。
334	到期前有效定单无效
335	无效delta: delta必须是在0和100之间。
336	<p>时间或时区无效。</p> <p>正确的格式是 hh:mm:ss xxx</p> <p>这里 xxx是可选的特定区。例如: 15:59:00 EST</p> <p>注意, 时间和时区之间有一个空格。</p> <p>如果没有指定时区, 将使用当地时间。</p>
337	<p>日期、时间、或时区输入无效。正确格式是yyyymmdd hh:mm:ss xxx</p> <p>这里yyyymmdd和xxx为可选的。例如: 20031126 15:59:00 EST</p> <p>注意, 在日期和时间以及时间和时区之间有一个空格。</p> <p>如果没有指定日期, 将使用当天日期。</p> <p>如果没有指定时区, 将使用当地时区。</p>
338	到时有有效定单在该交易所目前不可用。
339	不再支持期货差价。请使用组合来代替。
340	盒顶竞价策略的无效改善额。
341	<p>无效delta。有效值为从1到100。</p> <p>你可以从定单标签面板的“挂钩股票”部分设置, 或从主菜单选择Page/Layout( 页面/布局) 和添加Delta栏。</p>
342	该交易所不支持挂钩定单。
343	<p>日期、时间、或时区输入无效。正确格式是yyyymmdd hh:mm:ss xxx</p> <p>这里yyyymmdd和xxx为可选的。例如: 20031126 15:59:00 EST</p> <p>注意, 在日期和时间以及时间和时区之间有一个空格。</p> <p>如果没有指定日期, 将使用当天日期。</p> <p>如果没有指定时区, 将使用当地时区。</p>
344	登录的账户不是金融顾问账户。

345	通用组合不支持FA顾问账户。
346	不是机构账户或转移清算账户。
347	卖空交易盘值必须是1( 经纪商持有股票) 或2( 由其它地方交付) 。
348	定单为非卖空交易——类型必须为SSHORT来指定卖空交易盘。
349	通用组合不支持"Good After( 到时后) "属性。
350	最低数量不支持最佳组合定单。
351	“正常交易时段”标识对该定单无效。
352	卖空交易盘值为2( 由其它地方交付) 要求地点。
353	卖空交易盘值是1要求不指定地点。
354	没有订阅请求的市场数据。
355	定单尺寸不符合市场规则。
356	智能组合定单不支持OCA( 一取消全)组。
357	你的客户版本已过期。
358	不支持智能组合子定单。
359	组合定单仅支持减低不和批量一起执行(OCA)。
360	没有假设查看支持智能组合定单。
361	无效触发价格。
362	调整止损价无效。
363	调整止损限价价格无效。
364	调整跟踪额无效。
365	没有找到代码代号的扫描仪订阅：
366	没有找到代码代号的历史数据请求：
367	如果设置VOL定单的波动率，必须为1或2。不要为其它定单类型设置。
368	动态波动率管理的参考价格类型必须是1或2。不要为非VOL定单设置。
369	波动率定单仅适用于美国期权。
370	动态波动率定单必须为SMART( 智能) 传递的，或在价格改善交易所上交易的。

371	VOL定单要求波动率的正浮点值。不要为其它定单类型设置。
372	不能对非VOL定单设置动态VOL属性。
373	只能够为VOL或相对股票定单设置股票范围属性。
374	如果两侧均被设置，低端股票范围属性必须低于高端股票属性。
375	股票范围属性不能为负值。
376	定单不适用于连续更新。期权必须在便宜改道交易所上交易。
377	必须指定有效delta对冲定单辅助价。
378	Delta对冲定单类型要求hedge对冲辅助价被指定。
379	Delta对冲定单类型要求没有hedge对冲辅助价被指定。
380	该定单类型不允许用于delta对冲定单。
381	你的DDE.dll需要更新。
382	指定的价格违反了默认定单设置中指定的跳动限制数量。
383	指定的尺寸违反了默认定单设置中指定的尺寸限制。
384	无效DDE排列请求。
385	API扫描仪订阅的重复代码代号。
386	API历史数据查询的重复代码代号。
387	该交易所和证券类型不支持的定单类型。
388	定单尺寸小于最低要求。
389	提供传递的定单代号不是唯一的。
390	提供传递的定单代号无效。
391	输入的时间或时区无效。正确的格式是hh:mm:ss xxx 这里xxx是可选的特定区。例如：15:59:00 EST。 注意，在时间和时区之间有一个空格。 如果没有指定时区，将使用当地时间。
392	无效定单：合约已过期。
393	卖空交易盘可能仅对delta对冲定单指定。



394	无效处理时间：必须为100和2000之间的毫秒整数。找到：
395	由于系统问题，目前可能不接受具有OCA组的定单。
396	由于系统问题，目前只接受该合约的市价单和限价单申请。
397	由于系统问题，目前只接受该合约的市价单和限价单申请。
398	<> 不能用于条件触发。
399	定单消息错误

编码	描述
400	算法定单错误。
401	长度限制。
402	该合约不允许使用条件。
403	无效止损价。
404	该定单股票不能马上用于卖空。定单将在我们试图锁定股票期间被持有。
405	子定单数量应该等于母定单尺寸。
406	货币 <>为不允许。
407	代码应该仅包括有效的非统一码字符。
408	无效分段定单增量。
409	无效分段定单。你必须指定定单分组尺寸。
410	无效分段定单的后续组尺寸。
411	“Outside Regular Trading Hours( 正常交易时段外) ”标识对该定单无效。
412	合约不能用于交易。
413	假设定单的发送标识应该设置为真。
414	截图市场数据订阅不适用于通用跳动。
415	等到之前的RFQ( 报价请求) 完成后再尝试。
416	RFQ不适用于该合约。定单代号：
417	分段定单的无效初始分组尺寸。

418	无效分段定单利润抵消。
419	缺少分段定单的起始分组尺寸。
420	无效实时查询。
421	无效传递。
422	该定单的账户和清算属性也许不可以改变。
423	跨定单RFQ已过期。THI承诺尺寸不再可用。请打开定单对话框和确认流动性分配。
424	FA定单要求指定分配。
425	FA定单要求手动分配每个账户，因为没有共同的清算指令。请使用定单对话顾问标签输入分配。
426	没有一个账户具有足够的股票。
427	共同基金定单要求指定货币值。
428	共同基金卖单要求指定股数。
429	Delta中性定单仅支持组合( BAG证券类型) 。
430	对不起，指定证券的基本面数据不可用。
431	没有显示什么区域或不正确。
432	佣金不能为负数。
433	多个账户分配分段定单的无效“Restore size after taking profit( 获利后复原尺寸) ”。
434	定单尺寸不能为零。
435	你必须指定一个账户。
436	你必须指定一个分配( 或为单个账户、组、或定义) 。
437	定单只可以在RTH以外由一个标识或允许PreOpen。
438	申请现被锁住。
439	定单处理失败。算法定义没有找到。
440	定单修改失败。不能修改算法。
441	算法属性验证失败。
442	指定的算法不能用于该定单。
443	定单处理失败。不明算法属性。

444	尚未确认波动率组合定单。此时不能提交改变。
445	该定单的RFQ不再有效。
446	缺失分段定单利润抵消。
447	缺少分段价格调整额或间隔。
448	无效分段价格调整间隔。
449	意外的分段价格调整额或间隔。
40	股息时间表查询失败。

编码	描述
501	已经连接。
502	不能连接TWS。通过Configure>API(配置>API)菜单指令确认API已在TWS中启用。
503	你的TWS版本已过期，必须更新。
504	没有连接。
505	致命错误：不明消息代号。
510	请求市场数据 - 发送错误：
511	取消市场数据 - 发送错误：
512	定单 - 发送错误：
513	账户更新请求 - 发送错误：
514	请求执行 - 发送错误：
515	取消定单 - 发送错误：
516	请求开放定单 - 发送错误：
517	未知合约。应用确认合约详细。
518	请求合约数据 - 发送错误：
519	请求市场深度 - 发送错误：
520	取消市场深度 - 发送错误：
521	设置服务器记录级别 - 发送错误：
522	FA信息请求 - 发送错误：
523	FA信息取代 - 发送错误：
524	请求扫描仪订阅 - 发送错误：

525	取消扫描仪订阅 - 发送错误:
526	请求扫描仪参数 - 发送错误:
527	请求历史数据 - 发送错误:
528	取消历史数据 - 发送错误:
529	请求实时柱数据 - 发送错误:
530	取消实时柱数据 - 发送错误:
531	请求当前时间 - 发送错误:

编码	描述
10000	跨货币组合错误。
10001	跨货币交易量错误。
10002	无效非担保边。
10003	不允许IBSX。
10005	只读模式。
10006	缺少母定单。
10007	无效对冲类型。
10008	无效beta值。
10009	无效beta比率。
10010	无效beta对冲定单。
10011	货币不支持智能组合。
10012	无效分配百分比。
10013	智能传递API错误( 要求过滤掉智能传递) 。
10014	PctChange( 百分比改变) 限制。
10015	不允许通过API的交易。
10016	合约不可视。
10017	合约为不可视。
10018	定单使用EV警告。
10019	交易使用EV警告。
10020	显示尺寸应该小于定单尺寸。
10021	Mkt Offset API( 市场抵消API) 的无效leg2( 边2) 。

10022	无效边Prio API。
10023	无效组合显示尺寸API。
10024	无效不启动下一个登录API。
10025	无效leg2 to Mkt time1 API。
10026	无效leg2 to Mkt time2 API。
10027	无效组合传递标签API。

系统信息编码。

编码	描述
1100	IB和TWS之间的连接已经停止。
1101	IB和TWS之间的连接已经恢复 - 数据丢失。*
1102	IB和TWS之间的连接已经恢复 - 数据保存。
1300	TWS套接字端口已被重新设置，该连接已被放弃。 请通过新的端口重新连接 - <port_num>
*必须重新提交市场和账户数据订阅请求	

警告信息编码

编码	描述
2100	从TWS请求了新账户数据。 API客户已从账户数据中取消订阅。
2101	无法订阅账户，因为以下客户订阅了不同的账户。
2102	无法修改该定单，因为它仍处于被处理当中。
2103	市场数据场已断开。
2104	市场数据场已连接。
2105	历史市数据场已断开。
2106	历史数据场已连接。
2107	历史数据场连接已断开，但应依需要获重新连接。
2108	市场数据场连接已断开，但应依需要获重新连接。
2109	定单事件警告：属性“正常交易时段以外”根据定单类型和目的地被忽略。 PlaceOrder现被处理。

---

2110	TWS和服务器之间的连接被切断。连接将被自动恢复。
------	---------------------------

## 跳动类型

下面的表格列出所有可能的tickType参数值，用于下列[ActiveX events](#)、[C++ EWrapper functions](#)、和[Java EWrapper methods](#):

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

跳动值	描述	事件/功能/方法
0	BID_SIZE	tickSize()
1	BID_PRICE	tickPrice()
2	ASK_PRICE	tickPrice()
3	ASK_SIZE	tickSize()
4	LAST_PRICE	tickPrice()
5	LAST_SIZE	tickSize()
6	HIGH	tickPrice()
7	LOW	tickPrice()
8	VOLUME	tickSize()
9	CLOSE_PRICE	tickPrice()
10	BID_OPTION_COMPUTATION	tickOptionComputation() See Note 1 below
11	ASK_OPTION_COMPUTATION	tickOptionComputation() See Note 1 below
12	LAST_OPTION_COMPUTATION	tickOptionComputation() See Note 1 below
13	MODEL_OPTION_COMPUTATION	tickOptionComputation() See Note 1 below
14	OPEN_TICK	tickPrice()
15	LOW_13_WEEK	tickPrice()
16	HIGH_13_WEEK	tickPrice()

跳动值	描述	事件/功能/方法
17	LOW_26_WEEK	tickPrice()
18	HIGH_26_WEEK	tickPrice()
19	LOW_52_WEEK	tickPrice()
20	HIGH_52_WEEK	tickPrice()
21	AVG_VOLUME	tickSize()
22	OPEN_INTEREST	tickSize()
23	OPTION_HISTORICAL_VOL	tickGeneric()
24	OPTION_IMPLIED_VOL	tickGeneric()
25	OPTION_BID_EXCH	NOT USED
26	OPTION_ASK_EXCH	NOT USED
27	OPTION_CALL_OPEN_INTEREST	tickSize()
28	OPTION_PUT_OPEN_INTEREST	tickSize()
29	OPTION_CALL_VOLUME	tickSize()
30	OPTION_PUT_VOLUME	tickSize()
31	INDEX_FUTURE_PREMIUM	tickGeneric()
32	BID_EXCH	tickString()
33	ASK_EXCH	tickString()
34	AUCTION_VOLUME	tickSize()
35	AUCTION_PRICE	tickPrice()
36	AUCTION_IMBALANCE	tickSize()
37	MARK_PRICE	tickPrice()
38	BID_EFP_COMPUTATION	tickEFP()
39	ASK_EFP_COMPUTATION	tickEFP()
40	LAST_EFP_COMPUTATION	tickEFP()
41	OPEN_EFP_COMPUTATION	tickEFP()
42	HIGH_EFP_COMPUTATION	tickEFP()
43	LOW_EFP_COMPUTATION	tickEFP()
44	CLOSE_EFP_COMPUTATION	tickEFP()
45	LAST_TIMESTAMP	tickString()



跳动值	描述	事件/功能/方法
46	SHORTABLE	tickString()
47	FUNDAMENTAL_RATIOS	tickString()
48	RT_VOLUME	tickGeneric()
49	HALTED	See Note 2 below.
50	BIDYIELD	tickPrice() See Note 3 below
51	ASKYIELD	tickPrice() See Note 3 below
52	LASTYIELD	tickPrice() See Note 3 below
53	CUST_OPTION_COMPUTATION	tickOptionComputation()
54	TRADE_COUNT	tickGeneric()
55	TRADE_RATE	tickGeneric()
56	VOLUME_RATE	tickGeneric()

1. 跳动类型 BID\_OPTION\_COMPUTATION, ASK\_OPTION\_COMPUTATION, LAST\_OPTION\_COMPUTATION, and MODEL\_OPTION\_COMPUTATION 当请求时，返回所有 Greeks( delta、gamma、vega、theta )，底层价格和股票和期权参考价格。MODEL\_OPTION\_COMPUTATION 也返回模型隐含波动率。
2. 当合约的交易暂停时，TWS接收一个特殊的跳动：haltedLast=1。当交易恢复时，TWS收到haltedLast=0。跳动类型、HALTED( 暂停)、跳动代号 = 49现在可通过API的正常市场数据获得以标识暂停状态。这个新跳动类型的值可能是：
  - 0 = Not halted( 没有暂停)
  - 1 = Halted( 暂停)。
3. 仅适用于债券合约。

## 通用跳动类型

对所有基于套接字的API技术，包括套接字客户端库、ActiveX和Java，我们提供了几种市场数据跳动，可以做为市场数据请求的一部分来请求。

从API版本9.0开始( 客户版本30) ，版本6的REQ\_MKT\_DATA信息的发送包括一个新的区域，用来指定请求的跳动，如同一个逗号分隔的整数代号列表( 通用跳动类型) 。请求这些跳动将获得答案，如果请求的跳动类型与交易的合约相关。

注意，通过跳动类型不能被指定，如果你选择使用截图市场数据订阅的话。

通用市场数据跳动类型是：

整数代号值	跳动类型	跳动值结果
100	期权交易量( 目前用于股票)	29, 30
101	期权未平仓合约( 目前用于股票)	27, 28
104	历史波动率( 目前用于股票)	23
106	期权隐含波动率( 目前用于股票)	24
162	指数期货溢价	31
165	杂项状态	15, 16, 17, 18, 19, 20, 21
221	市价( 用于TWS中的益/损计算)	37
225	竞价值( 交易量、价格和失调)	34, 35, 36
233	<a href="#">RTVolume</a> - 包括最后交易价、最后交易尺寸、最后交易时间、总交易量、VWAP( 交易量加权平均价) 、和单个交易标识。	48
236	可卖空	46
256	剩余	
258	<a href="#">基本面比率</a>	47
411	实时历史波动率	58
456	<a href="#">IBDividends</a>	59

### 使用 **SHORTABLE Tick**( 可卖空跳动)

在TWS中，有一个SHORTABLE栏，如下图。该栏描述证券可以被卖空的股票数量。

Undrlyng	Exch	Description	Shortable Key	Bid TIF	Bid Size Action	Ask Quantity
IBM	SMAR...	Stock (NYSE)	Green	116.90	1	116.
AGG	SMAR...	Stock (ARCA)	Red	98.98	2	99.
GOOG	SMAR...	Stock (NASDAQ...)	Green	439.99	1	440.
MS	SMAR...	Stock (NYSE)	Red	27.47	12	27.
GS	SMAR...	Stock (NYSE)	Red	129.71	5	129.
YHOO	SMAR...	Stock (NASDAQ...)	Green	18.98	47	18.
SPY	SMAR...	Stock (AMEX)	Red	118.91	97	118.
CSCO	SMAR...	Stock (NASDAQ...)	Green	22.92	271	22.

绿色表示至少有1000股可用于卖空。深绿色表示该合约可以被卖空，但目前没有可用于卖空的股票，并且系统正在搜寻股票。红色表示没有可卖空的股票。

API版本9.30及其后版本，卖空标识支持所有套接字连接。功能相当于TWS平台中的SHORTABLE栏。

当启动reqMktDataEx()/reqMktData()方法时，你必须包括通用ticktype 236参数来获取卖空值。举例：

卖空跳动决定是否如果合约的SHORT SELL( 卖空) 定单将被接受。分析从tickGeneric(int tickerId, int tickType, double value)返回的值如下：

```

if (value > 2.5) { // 3.0

// There are at least 1000 shares available for a short sale

// In TWS, this is identical to GREEN status

}

else if (value > 1.5) { // 2.0

// This contract will be available for short sale if shares can be //
located

// In TWS, this is identical to DARK GREEN status

}
    
```

```

else if (value > 0.5) { // 1.0

    // Not available for short sale
// In TWS, this is identical to RED status

}

else {

    // unknown value

}
    
```

**Note:** 该功能支持服务器版本33( TWS版本872) 。

**FUNDAMENTAL\_RATIOS tickType( 基本面比率跳动类型) 的标签值**

你可使用FUNDAMENTAL\_RATIOS tickType ( 跳动值47) 以格式TAG=VALUE、TAG2=VALUE2、等请求基本面比率。比率的发送使用tickGeneric() callback。下面表格中列出所有FUNDAMENTAL\_RATIOS的标签值。

标签	描述
NPRICE	<b>收盘价</b> 这是产品最后一天交易的收盘价。也被称之为当前价。注意，某些产品可能不会每天都有交易，因此该价可能是来自最后交易日的前一天。
Three_Year_TTM_Growth	3年连续12个月的增长。
TTM_over_TTM	连续12个月的连续12个月。
NHIG	<b>最高价</b> 这个价格是过去12个月内股票交易的最高价。这可能是日内最高价。
NLOW	<b>最低价</b> 这个价格是过去12个月内股票交易的最低价。这可能是日内最低价。
PDATE	<b>定价日</b> 定价日是交易产品最后交易价的日期。
VOL10DAVG	<b>交易量</b> 这是最后10天的累积平均交易量。
MKTCAP	<b>市值</b> 该值的计算是用当前价格乘以总股票发行数。

标签	描述
TTMEPSXCLX	<p><b>不包括非经常性项目的EPS(每股盈利)</b>                      这是连续12个月普通股东的调整后收入除以连续12个月稀释加权平均股票流通数。</p>
AEPSNORM	<p><b>EPS规范化</b>                      这是最近年度期间普通股东的规范化收入除以同期的稀释加权平均股票流通数。</p>
TTMREVPS	<p><b>收入/股份</b>                      该值是连续12个月的总收入除以连续12个月的平均稀释股票流通数。</p> <p><b>Note:</b> 大部分银行和保险公司在发布其初步季度财务业绩时不公布收入。此时，连续12个月的价值将不存在(NA)。</p>
QBVPS	<p><b>每股账面价值(普通股)</b>                      由普通股东的股权除以最近中期截止股票发行数来定义。账面价值(Book Value)是总股东的股权减去优先股和可赎回优先股。</p>
QTANBVPS	<p><b>每股账面价值(有形)</b>                      这是中期有形账面价值除以最近中期截止的股票发行数。有形账面价值是账面价值减去同期的商誉和无形资产。</p>
QCSHPS	<p><b>每股现金</b>                      为总现金加上短期投资然后除以最近中期截止的股票发行数。</p> <p><b>Note:</b> 这不包括可能报告中长期资产下的现金等价物。</p>
TTMCFSHR	<p><b>每股现金流</b>                      这个值是连续12个月现金流除以连续12个月平均股票发行数。现金流的定义是税后收入总合减去优先股息和普通合伙人分配以及折旧、损耗和摊销。</p>
TTMDIVSHR	<p><b>每股股息</b>                      这是过去连续12个月期间支付给普通股东的每股现金股息的中总合。</p>

标签	描述
IAD	<b>股息率</b> 这个值是预期下一个12个月的股息支付总合。通常是最近支付的或发布的现金股息乘以股息支付频率，加上任何经常性的额外股息。
PEEXCLXOR	<b>P/E 扣除非经常性项目</b> 这个比率的计算是用过去四个中期的非经常项目和账户变化之前的持续经营的每股稀释盈利的总合除以当前股价。
APENORM	<b>P/E 规范化</b> 这是当前股价除以最新的年度每股规范化盈利值。
TMPR2REV	<b>股价与销售比率</b> 这是当前股价除以连续12个月的每股销售。如果有最近中期截止的初步盈利发布，将使用该发布中的收入(销售)值来计算连续12个月的每股收入。注意：大部分银行和金融公司在宣布它们的初步中期财务结果时都不包括收入数据。此时，连续12个月的值在最终的中期报告发布之前将不存在( NA)。
PR2TANBK	<b>股价与有形资产账面比率</b> 这是当前股价除以最新的每股年度有形资产账面值。每股有形资产账面值的定义是资产账面值减去商誉和无形资产除以财务年度截止时的股票发行数。
TTMPRCFPS	<b>股价与每股现金流比率</b> 这是当前股价除以连续12个月每股现金流。现金流的定义是税后收入总合减去优先股息和普通合伙人分配以及折旧、损耗和摊销。
PRICE2BK	<b>股价与账面价值比率</b> 这是当前股价除以最近中期的每股账面价值。
QCURRATIO	<b>流动比率</b> 这个比率是最近中期的总流动资产除以同期的总流动负债。注意：这个项目对银行、保险公司和其它不区分流动和长期资产和负债的公司不存在( NA)。

标签	描述
QQUICKRATI	<b>速动比率</b> 也称为酸性测试比率，该比率的定义是最近中期的现金加上短期投资和应收账款除以同期的总流动负债。注意：这个项目对银行、保险公司和其它不区分流动和长期资产和负债的公司不存在( NA)。
QLTD2EQ	<b>长期债务/股权比率</b> 这个比率是最近中期的总长期债务除以同期的总股东股权。
QTOTD2EQ	<b>总债务/总股权比率</b> 这个比率是最近中期的总债务除以同期的总股东股权。注意：这对银行没有意义( NM)。
TTMPAYRAT	<b>派息比率</b> 这个比率是除去连续12个月内以现金股息形式支付给普通股股东的非经常性项目的每股初级/基本盈利的比率。
TTMREV	<b>总收入</b> 这是最近TTM期间报告的所有营业部们的所有收入( 销售) 的总合。注意：大部分银行和保险公司在发布其初步季度财务结果时都不公布收入数据。此时，季度值将不存在( NA)。
TTMEBITD	<b>EBITD</b> 税息折旧前盈利( EBITDA) 是连续12个月的EBIT( 税息前盈利) 加上同期的折旧和摊派费用( 现金流报告)。注意：这个项目仅适用于工业和公共事业公司。
TTMEBT	<b>税前盈利</b> 也称为税前收入和税前盈利，是最近TTM期间的总收入减去总支出加上同期的非营业收入( 支出)。
TTMNIAC	<b>普通股股东的净收入</b> 这是连续12个月用于普通股股东的股息和保留盈利的累计金额。普通股股东的可用收入的计算是连续12个月的税后收入加上相关的少数权益和股权和优先股息，普通合伙人分配和美国GAAP调整。注意：任何为负数的调整( 例如， 优先股票股息) 将从税后收入中减去。

标签	描述
AEBTNORM	<b>税前盈利规范化</b> 这是除去最近年度期间内所有非正常/一次性/特殊支出项目影响的税前收入数目。
ANIACNORM	<b>普通股股东的可用净收入，规范化</b> 这是除去所有非正常/一次性/特殊支出项目影响的用于普通股股东股息和保留盈利的累积年度金额。
TTMGROSMGN	<b>毛利率</b> 这个值衡量除去支付所有直接生产开销后的收入的百分比。它的计算是连续12个月的总收入减去连续12个月的已销售的产品成本除以连续12个月的总收入并乘以100。注意：这个项目仅适用于工业和公共事业公司。
TTMNPMGN	<b>净利润率%</b> 也称为销售回报率，这个值是连续12个月的税后收入除以同期的总收入，并以百分比形式表示。注意：大部分银行和金融公司在发布其初步季度财务业绩时不公布收入数据。此时，连续12个月的值将不存在( NA)。
TTMOPMGN	<b>营业利润率</b> 这个值衡量支付所有的营运支出后的剩余收入的百分比。它的计算是连续12个月的营业收入除以连续12个月的总收入，乘以100。营业收入的定义是总收入减去总营业支出。
APTMGNPCT	<b>税前毛利率</b> 这个值代表最近财务年度的税前收入，以最近财务年度的总收入百分比表示。
TTMROAPCT	<b>平均资产回报率</b> 这个值是连续12个月的税后收入除以平均总资产，以百分比表示。平均总资产的计算是将5个最近季度的总资产相加然后除以5。
TTMROEPCT	<b>平均股权回报率</b> 这个值是连续12个月的普通股股东可用收入除以平均普通股，以百分比表示。平均普通股权的计算是相加5个最近季度的普通股权再除以5。



标签	描述
TTMROI PCT	<b>投资回报率</b> 这个值是连续12个月的税后收入除以平均总长期债务和其它长期负债和股东股权，以百分比表示。
REVCHNGYR	<b>总收入变化%</b> 这个值的计算是最近中期的销售减去一年前的同期销售除以一年前的同期中期销售，乘以100。
TTMREVCHG	<b>总收入变化%</b> 这个值是连续12个月的销售比较一年前相同12个月的销售的百分比变化。这个值的计算是连续12个月的销售减去一年前的同期12个月的销售除以一年前12个月同期的销售，乘以100。
REVTRENDGR	<b>总收入增长率</b> 五年的总收入增长率是过去5年期间的年度复合总收入增长率。
EPSCHNGYR	<b>EPS变化%</b> 这个值的计算是最近中期EPS减去一年前同期的EPS除以一年前同期的EPS，乘以100。注意：两个期间的EPS必须为正值。如果任何之一的EPS值为负，结果即没有意义( NM) 。
TTMEPSCHG	<b>EPS变化%</b> 这个值是连续12个月的EPS比较一年前同期的EPS的百分比变化。这个值的计算是连续12个月的EPS减去一年前12个月EPS除以一年前12个月的EPS，乘以100。注意：如果任何之一的值为负，结果的值即没有意义( NM) 。
EPSTRENDGR	<b>EPS增长率</b> 这个增长率是除去过去5年内的非正常项目和已终止业务的每股盈利的复合年度增长率。注意：如果最近年份或5年前的值为零或负值，则不能计算增长率，将使用“NA”代号( 不存在) 。
DIVGRPCT	<b>增长率保留%—股息</b> 股息增长率是每股股息的复合年度增长率。DIVGR%的计算是用3年的，如果有4年的股息信息的话。

### IBDividends跳动举例

IBDividends通用跳动返回一个以逗号分开股息列表，按下列顺序：

1. 过去12个月的股息总合
2. 下一个12个月的股息总合
3. 下一个股息日期
4. 下一个单个股息额

#### 举例

下面是代码MSFT的IBDividends跳动更新举例：

```
0.83,0.92,20130219,0.23
```

这里

0.83 = 过去12个月的股息总合

0.92 - 下一个12个月的股息总合

20130219 - 下一个股息日期

0.23 - 下一个单个股息额

### RTVolume

RTVolume是普通跳动标签之一，可以做为市场数据请求的一部分来请求。RTVolume返回如下结果：

- 最后交易价
- 最后交易尺寸
- 最后交易时间
- 总交易量
- 交易量加权平均价格
- 单个交易标识(真或假)。

RTVolume是API的相当于交易平台中的打开时间和交易窗口( Time and Sales Window) 和实时查看更新。实施的方式是你必须要在你的市场数据请求的genericTicklist参数中包括233。

你将通过区域值48内的tickString()事件接收RTVolume更新。

#### 举例

下面是对代码AAPL的RTVolume格式的举例：

```
RTVolume=701.28;1;1348075471534;67854;701.46918464>true
```

```
RTVolume=701.26;3;1348075476533;67857;701.46917554>false
```

```
RTVolume=701.27;3;1348075482034;67860;701.46916674>true
```

```
RTVolume=701.27;3;1348075482336;67863;701.46915809>false
```

```
RTVolume=701.25;1;1348075483534;67864;701.46915486>true
```

```
RTVolume=701.24;1;1348075487029;67865;701.46915151>true
```

```
RTVolume=701.25;1;1348075489787;67866;701.46914828>true
```

```
RTVolume=701.32;4;1348075490787;67870;701.46913949>true
```

```
RTVolume=701.32;2;1348075493802;67872;701.46913497>true
```

```
RTVolume=701.29;1;1348075494789;67873;701.46913233>true
```

## 定单类型和IBAlgos( IB算法)

这一节包括下面三个题目：

- [支持的定单类型](#)
- [IBAlgo参数](#)
- [CSFB算法参数](#)

### 支持的定单类型

盈透证券的API技术支持下列定单类型。

API定单仅模拟交易平台( TWS) 的运作。使用API发送相同的定单之前，请测试每个定单类型，保证你可以成功地通过TWS发送每个类型。

定单类型	缩写
限制风险	
括号单	
市价转限价	MTL
有保护市价	MKT PRT
请求报价	QUOTE
止损	STP
止损限价	STP LMT
触发跟踪限价	TRAIL LIT
触发跟踪市价	TRAIL MIT
跟踪止损	TRAIL
跟踪止损限价	TRAIL LIMIT
执行速动	
竞价	
全权委托	
市价	MKT
触发市价	MIT
收盘市价	MOC
开盘市价	MOO
挂钩市价	PEG MKT

订单类型	缩写
相对	REL
一扫光	
价格改善	
盒顶	BOX TOP
价格改善竞价	
大额	
收盘限价	LOC
开盘限价	LOO
触发限价	LIT
挂钩中点	PEG MID
隐私	
隐藏	
冰山/保留	
交易量加权平均价格—担保	VWAP
市场上时间	
全或无	
全数执行或立即取消	
到时/到期后有效	GAT
到时/到期前有效	GTD
取消前有效	GTC
立即执行或取消	IOC
高价交易	
一取消全	OCA
价差	
波动率	VOL
算法交易 (Algos)	
到达价格	
平衡影响和风险	
最小化影响	

订单类型	缩写
交易量百分比	
分段	
时间加权平均价格	
交易量加权平均价格—最大努力	
积累/分配	
IB暗池	

**IBAlgo参数**

从TWS API版本9.6开始，ActiveX、C++和Java API支持美国股票和美国期权的下列IBAlgo订单：

**美国股票**

- [到达价格\(ArrivalPx\)](#)
- [黑冰\(DarkIce\)](#)
- [交易量百分比\(PctVol\)](#)
- [时间加权平均价格\(Twap\)](#)
- [交易量加权平均价格\(Vwap\)](#)

**美国期权**

- [平衡影响和风险\(BalanceImpactRisk\)](#)
- [最小化影响\(MinImpact\)](#)

**美国产品**

- [积累/分配\(AD\)](#)

下面的图片列出所有API支持的IBAlgo策略和参数，除了积累/分配，在[积累/分配\(AD\)](#)中有说明。

Algo Strategy	Corresponding Parameters									
	maxPctVol	pctVol	strategyType	startTime	endTime	allowPastEndTime	noTakeLiq	riskAversion	forceCompletion	displaySize
<b>For US Stocks</b>										
Arrival Price	X			X	X	X		X	X	
Dark Ice				X	X	X				X
Percentage of Volume		X		X	X		X			
TWAP			X	X	X	X				
VWAP	X			X	X		X			
<b>For US Options</b>										
Balance Impact and Risk	X							X	X	
Minimize Impact	X									

### 到达价格 (ArrivalPx)

参数	描述	句法
maxPctVol	最大百分比	范围: “0.01” – “0.5”
riskAversion	紧急/风险规避	“Get Done”, “Aggressive”, “Neutral”, “Passive”
startTime	开始时间	“9:00:00 EST”
endTime	结束时间	“15:00:00 EST”
forceCompletion	争取在交易日结束前( EOD) 完成	“0” 或 “1”
allowPastEndTime	允许截止时间后交易	“0” 或 “1”

#### 到达价格Java编码举例:

```

Contract m_contract = new Contract();

Order m_order = new Order();

Vector<TagValue> m_algoParams = new Vector<TagValue>();

/** Stocks */

m_contract.m_symbol = "MSFT";

m_contract.m_secType = "STK";

m_contract.m_exchange = "SMART";

m_contract.m_currency = "USD";

/** Arrival Price */

m_algoParams.add( new TagValue("maxPctVol","0.01") );

m_algoParams.add( new TagValue("riskAversion","Passive") );
    
```

```

m_algoParams.add( new TagValue("startTime","9:00:00 EST") );

m_algoParams.add( new TagValue("endTime","15:00:00 EST") );

m_algoParams.add( new TagValue("forceCompletion","0") );

m_algoParams.add( new TagValue("allowPastEndTime","1") );

m_order.m_action = "BUY";

m_order.m_totalQuantity = 1;

m_order.m_orderType = "LMT";

m_order.m_lmtPrice = 0.14

m_order.m_algoStrategy = "ArrivalPx";

m_order.m_algoParams = m_algoParams;

m_order.m_transmit = false;

m_client.placeOrder(40, m_contract, m_order);

```

更多信息...

- [到达价格算法](#)

**黑冰(DarkIce)**

参数	描述	句法
displaySize	显示尺寸	
startTime	开始时间	“9:00:00 EST”
endTime	截止时间:	“15:00:00 EST”
allowPastEndTime	允许截止时间后交易:	“0” 或 “1”

更多信息...

- [黑冰算法](#)



**交易量百分比 (PctVol)**

参数	描述	句法
pctVol	交易量百分比	范围：“0.01” – “0.5”
startTime	开始时间	“9:00:00 EST”
endTime	结束时间	“15:00:00 EST”]
noTakeLiq	努力从不减少流动性	“0” 或 “1”

更多信息...

- [交易量百分比算法](#)

### 时间加权平均价格(Twap)

参数	描述	句法
strategyType	交易策略	“Marketable”, “Matching Midpoint”, “Matching Same Side”, “Matching Last”
startTime	开始时间	“9:00:00 EST”
endTime	截止时间	“15:00:00 EST”
allowPastEndTime	允许截止时间后交易	“0” 或 “1”

更多信息...

- [时间加权平均价格算法](#)

### 交易量加权平均价格(Vwap)

参数	描述	句法
maxPctVol	最大百分比	range: “0.01” – “0.5”
startTime	开始时间	“9:00:00 EST”
endTime	截止时间	“15:00:00 EST”
allowPastEndTime	允许截止时间后交易	“0” 或 “1”
noTakeLiq	努力从不减少流动性	“0” 或 “1”
getDone	完成	“0” 或 “1”
noTradeAhead	没有未来交易	“0” 或 “1”
useOddLots	使用零股	“0” 或 “1”

更多信息...

- [交易量加权平均价格算法](#)

### 平衡影响和风险(BalanceImpactRisk)

参数	描述	句法
maxPctVol	最大百分比	range: “0.01” – “0.5”
riskAversion	紧急/风险规避	“Get Done”, “Aggressive”, “Neutral”, “Passive”
forceCompletion	争取在交易日结束前完成	“0” 或 “1”
allowPastEndTime	允许截止时间后交易	“0” 或 “1”

平衡影响和风险Java编码举例:

```
Contract m_contract = new Contract();

Order m_order = new Order();

Vector<TagValue> m_algoParams = new Vector<TagValue>();

/** Options */

m_contract.m_symbol = "C";

m_contract.m_secType = "OPT";

m_contract.m_exchange = "SMART";

m_contract.m_localSymbol = "C      110304C00004500";

/** Balance Impact and Risk (OPT) */

m_algoParams.add( new TagValue("maxPctVol","0.1") );

m_algoParams.add( new TagValue("riskAversion","Aggressive") );

m_algoParams.add( new TagValue("forceCompletion","1") );

m_order.m_action = "BUY";

m_order.m_totalQuantity = 1;

m_order.m_orderType = "LMT";

m_order.m_lmtPrice = 0.14;

m_order.m_algoStrategy = "BalanceImpactRisk";

m_order.m_algoParams = m_algoParams;
```

```
m_order.m_transmit = false;

m_client.placeOrder(45, m_contract, m_order);
```

更多信息...

- [平衡影响和风险算法](#)

**最小化影响 (MinImpact)**

参数	描述	句法
maxPctVol	最大百分比	range: "0.01" – "0.5"

更多信息...

- [最小化影响](#)

**积累/分配 (AD)**

参数	描述	句法
componentSize	数量增量	Cannot exceed the amount of the initial order( 不能超过初始定单量)
timeBetweenOrders	时间间隔	
randomizeTime20	随机时间区间 +/- 20%	"0" 或 "1"
randomizeSize55	随机尺寸 +/- 55%	"0" 或 "1"
giveUp	和清算相关的数字	
catchUp	追赶时间	"0" 或 "1"
waitForFill	发送下一个定单之前等待当前定单的执行	"0" 或 "1"
startTime	开始时间	"9:00:00 EST"
endTime	结束时间:	"15:00:00 EST"

**积累分配Java编码举例**

```
Contract m_contract = newContract();

Order m_order = newOrder();
```

```
Vector<TagValue>m_algoParams = new Vector<TagValue>();

/** Stocks */

m_contract.m_symbol = "IBM";

m_contract.m_secType = "STK";

m_contract.m_exchange = "SMART";

m_contract.m_currency = "USD";

/** Accumulate/Distribute (All) */

m_algoParams.add(newTagValue("componentSize", "100"));

m_algoParams.add(newTagValue("timeBetweenOrders", "60"));

m_algoParams.add(newTagValue("randomizeTime20", "1"));

m_algoParams.add(newTagValue("randomizeSize55", "1"));

m_algoParams.add(newTagValue("giveUp", "1"));

m_algoParams.add(newTagValue("catchUp", "1"));

m_algoParams.add(newTagValue("waitForFill", "1"));

m_algoParams.add(newTagValue("startTime", "20110302-14:30:00 GMT"));

m_algoParams.add(newTagValue("endTime", "20110302-21:00:00 GMT"));

m_order.m_action = "BUY";
```

```
m_order.m_totalQuantity = 700;

m_order.m_orderType = "LMT";

m_order.m_lmtPrice = 140.0;

m_order.m_algoStrategy = "AD";

m_order.m_tif = "DAY";

m_order.m_algoParams = m_algoParams;

m_order.m_transmit = false;

m_client.placeOrder(orderId++, m_contract, m_order);
```

#### 更多信息...

- [积累分配订单类型](#)
- [TWS积累分配](#)

#### CSFB算法参数

ActiveX、C++和Java API支持下列CSFB算法策略：

- [Crossfinder](#)
- [Float](#)
- [Guerilla](#)
- [Work It IW](#)
- [Work It](#)
- [Pathfinder](#)
- [Reserve](#)
- [Strike](#)
- [10B 18](#)
- [Tex](#)
- [TWAP](#)
- [VWAP](#)

下面图片中列出了API支持的所有CSFB算法策略和参数。

	CSFB Algo Corresponding Parameters						
	Abbreviation	StartTime	EndTime	MinPctVolume	MaxPctVolume	DisplaySize	ExecutionStyle
<b>Algo Strategy</b>							
CrossFinder	CROS	X	X		X		X
Float	FLT	X				X	X
Guerilla	GRRL	X			X		X
Work it IW	INIW	X					
Work it	INLN	X	X	X	X		X
Path Finder	PTHF	X	X				
Reserve	RSRV	X				X	
Strike	SNPR	X				X	
10B 18	TENB	X	X		X		
Tex	TEX	X	X		X		X
TWAP	TWAP	X	X		X		
VWAP	VWAP	X	X		X		

**Crossfinder (CROS)**

参数	描述	句法
startTime	开始时间	“9:00:00 EST”
endTime	结束时间:	“15:00:00 EST”
MaxPctVolume	最大百分比交易量	range: “0” – “99”
ExecutionStyle	执行类型	"Normal", "Patient", "Aggressive"

**Crossfinder (CROS) Java编码举例**

```

void onCrossFinderAlgo() {

    Contract con = new Contract();

    con.m_symbol = "IBM";

    con.m_secType = "STK";

    con.m_exchange = "CSFBALGO";

    con.m_currency = "USD";
    
```

```
Vector<TagValue> m_algoParams = new Vector<TagValue>();

m_algoParams.add(new TagValue("StartTime", "13:30:00 EST"));

m_algoParams.add(new TagValue("EndTime", "14:30:00 EST"));

m_algoParams.add(new TagValue("MaxPctVolume", "25")); // Max % Volume

m_algoParams.add(new TagValue("ExecutionStyle", "Normal"));

//possible values for ExecutionStyle: Normal, Patient, Aggressive

Order order = new Order();

order.m_action = "BUY";

order.m_totalQuantity = 100;

order.m_orderType = "LMT";

order.m_lmtPrice = 200.0;

order.m_algoStrategy = "CROS";

order.m_algoParams = m_algoParams;

order.m_transmit = false;

m_client.placeOrder(globalOrderId++, con, order);

}
```



**Float (FLT)**

参数	描述	句法
startTime	开始时间	“9:00:00 EST”
DisplaySize	显示尺寸	“50” (integer)
ExecutionStyle	执行类型	"Normal", "Patient", "Aggressive"

**Float (FLT) Java编码举例**

```

void onFloatCsfbAlgo() {

    Contract con = new Contract();

    con.m_symbol = "IBM";

    con.m_secType = "STK";

    con.m_exchange = "CSFBALGO";

    con.m_currency = "USD";

    Vector<TagValue> m_algoParams = new Vector<TagValue>();

    m_algoParams.add(new TagValue("StartTime", "13:30:00 EST"));

    m_algoParams.add(new TagValue("DisplaySize", "10")); //iceberg

    m_algoParams.add(new TagValue("ExecutionStyle", "Normal"));

    //possible values for ExecutionStyle: Normal, Patient, Aggressive

    Order order = new Order();

    order.m_action = "BUY";
    
```

```

order.m_totalQuantity = 100;

order.m_orderType = "LMT";

order.m_lmtPrice = 200.0;

order.m_algoStrategy = "FLT";

order.m_algoParams = m_algoParams;

order.m_transmit = false;

m_client.placeOrder(globalOrderId++, con, order);

}

```

**Guerilla (GRRL)**

参数	描述	句法
startTime	开始时间	"9:00:00 EST"
MaxPctVolume	最大百分比交易量	range: "0" – "99"
ExecutionStyle	执行类型	"Normal", "Patient", "Aggressive"

**Guerilla (GRRL) Java编码举例**

```

void onGuerillaCsfbAlgo() {

Contract con = new Contract();

con.m_symbol = "IBM";

con.m_secType = "STK";

con.m_exchange = "CSFBALGO";

```

```

con.m_currency = "USD";

Vector<TagValue> m_algoParams = new Vector<TagValue>();

m_algoParams.add(new TagValue("StartTime", "13:30:00 EST"));

m_algoParams.add(new TagValue("MaxPctVolume", "25")); // Max % Volume

m_algoParams.add(new TagValue("ExecutionStyle", "Patient"));

//possible values for ExecutionStyle: Normal, Patient, Aggressive

Order order = new Order();

order.m_action = "BUY";

order.m_totalQuantity = 100;

order.m_orderType = "LMT";

order.m_lmtPrice = 200.0;

order.m_algoStrategy = "GRRL";

order.m_algoParams = m_algoParams;

order.m_transmit = false;

m_client.placeOrder(globalOrderId++, con, order);

}

```

**Work It IW (INIW)**

参数	描述	句法
startTime	开始时间	“9:00:00 EST”

**Work It IW (INIW) Java编码举例**

```
void onWorkItIwCsfbAlgo() {

    Contract con = new Contract();

    con.m_symbol = "IBM";

    con.m_secType = "STK";

    con.m_exchange = "CSFBALGO";

    con.m_currency = "USD";

    Vector<TagValue> m_algoParams = new Vector<TagValue>();

    m_algoParams.add(new TagValue("StartTime", "13:30:00 EST"));

    Order order = new Order();

    order.m_action = "BUY";

    order.m_totalQuantity = 100;

    order.m_orderType = "LMT";

    order.m_lmtPrice = 200.0;

    order.m_algoStrategy = "INIW";

    order.m_algoParams = m_algoParams;

    order.m_transmit = false;
```

```
m_client.placeOrder(globalOrderId++, con, order);

}
```

**Work It (INLN)**

参数	描述	句法
startTime	开始时间	“9:00:00 EST”
endTime	结束时间:	“15:00:00 EST”
MinPctVolume	最低百分比交易量	range: “0” – “99”
MaxPctVolume	最大百分比交易量	range: “0” – “99”
ExecutionStyle	执行类型	"Normal", "Patient", "Aggressive"

**Work It (INLN) Java编码举例**

```
void onWorkItCsfbAlgo() {

    Contract con = new Contract();

    con.m_symbol = "IBM";

    con.m_secType = "STK";

    con.m_exchange = "CSFBALGO";

    con.m_currency = "USD";

    Vector<TagValue> m_algoParams = new Vector<TagValue>();

    m_algoParams.add(new TagValue("StartTime", "13:45:00 EST"));

    m_algoParams.add(new TagValue("EndTime", "15:30:00 EST"));

    m_algoParams.add(new TagValue("MinPctVolume", "15")); // Min % Volume
```

```

m_algoParams.add(new TagValue("MaxPctVolume", "25")); // Max % Volume

m_algoParams.add(new TagValue("ExecutionStyle", "Patient"));

//possible values for ExecutionStyle: Normal, Patient, Aggressive

Order order = new Order();

order.m_action = "BUY";

order.m_totalQuantity = 100;

order.m_orderType = "LMT";

order.m_lmtPrice = 200.0;

order.m_algoStrategy = "INLN";

order.m_algoParams = m_algoParams;

order.m_transmit = false;

m_client.placeOrder(globalOrderId++, con, order);

}
    
```

**Pathfinder (PTHF)**

参数	描述	句法
startTime	开始时间	“9:00:00 EST”
endTime	结束时间:	“15:00:00 EST”

**Pathfinder (PTHF) Java编码举例**

```

void onPathFinderCsfbAlgo() {
    
```

```
Contract con = new Contract();

con.m_symbol = "IBM";

con.m_secType = "STK";

con.m_exchange = "CSFBALGO";

con.m_currency = "USD";

Vector<TagValue> m_algoParams = new Vector<TagValue>();

m_algoParams.add(new TagValue("StartTime", "13:45:00 EST"));

m_algoParams.add(new TagValue("EndTime", "15:30:00 EST"));

Order order = new Order();

order.m_action = "BUY";

order.m_totalQuantity = 100;

order.m_orderType = "LMT";

order.m_lmtPrice = 200.0;

order.m_algoStrategy = "PTHF";

order.m_algoParams = m_algoParams;

order.m_transmit = false;

m_client.placeOrder(globalOrderId++, con, order);

}
```

**Reserve (RSRV)**

参数	描述	句法
startTime	开始时间	“9:00:00 EST”
DisplaySize	显示尺寸	“50” (integer)

**Reserve (RSRV) Java编码举例**

```

void onReserveCsfbAlgo() {

    Contract con = new Contract();

    con.m_symbol = "IBM";

    con.m_secType = "STK";

    con.m_exchange = "CSFBALGO";

    con.m_currency = "USD";

    Vector<TagValue> m_algoParams = new Vector<TagValue>();

    m_algoParams.add(new TagValue("StartTime", "14:40:00 EST"));

    m_algoParams.add(new TagValue("DisplaySize", "50")); //iceberg

    Order order = new Order();

    order.m_action = "BUY";

    order.m_totalQuantity = 100;

    order.m_orderType = "LMT";

    order.m_lmtPrice = 200.0;
    
```



```

order.m_algoStrategy = "RSRV";

order.m_algoParams = m_algoParams;

order.m_transmit = false;

m_client.placeOrder(globalOrderId++, con, order);

}

```

**Strike (SNPR)**

参数	描述	句法
startTime	开始时间	"9:00:00 EST"
DisplaySize	显示尺寸	"50" (integer)

**Strike (SNPR) Java 编码举例**

```

void onStrikeCsfbAlgo() {

Contract con = new Contract();

con.m_symbol = "IBM";

con.m_secType = "STK";

con.m_exchange = "CSFBALGO";

con.m_currency = "USD";

Vector<TagValue> m_algoParams = new Vector<TagValue>();

m_algoParams.add(new TagValue("StartTime", "14:40:00 EST"));

m_algoParams.add(new TagValue("DisplaySize", "50")); //iceberg

```

```

Order order = new Order();

order.m_action = "BUY";

order.m_totalQuantity = 100;

order.m_orderType = "LMT";

order.m_lmtPrice = 200.0;

order.m_algoStrategy = "SNPR";

order.m_algoParams = m_algoParams;

order.m_transmit = false;

m_client.placeOrder(globalOrderId++, con, order);

}
    
```

**10B 18 (TENB) Java 编码举例**

参数	描述	句法
startTime	开始时间	“9:00:00 EST”
endTime	结束时间:	“15:00:00 EST”
MaxPctVolume	最大百分比交易量	range: “0” – “99”

**10B 18 (TENB) Java 编码举例**

```

void on10BCsfbAlgo() {

Contract con = new Contract();

con.m_symbol = "IBM";

con.m_secType = "STK";

con.m_exchange = "CSFBALGO";
    
```

```

con.m_currency = "USD";

Vector<TagValue> m_algoParams = new Vector<TagValue>();

m_algoParams.add(new TagValue("StartTime", "14:40:00 EST"));

m_algoParams.add(new TagValue("EndTime", "15:40:00 EST"));

m_algoParams.add(new TagValue("MaxPctVolume", "35")); // Max % Volume

Order order = new Order();

order.m_action = "BUY";

order.m_totalQuantity = 100;

order.m_orderType = "LMT";

order.m_lmtPrice = 200.0;

order.m_algoStrategy = "TENB";

order.m_algoParams = m_algoParams;

order.m_transmit = false;

m_client.placeOrder(globalOrderId++, con, order);

}
    
```

**Tex (TEX)**

参数	描述	句法
startTime	开始时间	“9:00:00 EST”
endTime	结束时间:	“15:00:00 EST”

参数	描述	句法
MaxPctVolume	最大百分比交易量	range: “0” – “99”
ExecutionStyle	执行类型	"Normal", "Patient", "Aggressive"

**Tex (TEX) Java编码举例**

```

void onTexCsfbAlgo() {

    Contract con = new Contract();

    con.m_symbol = "IBM";

    con.m_secType = "STK";

    con.m_exchange = "CSFBALGO";

    con.m_currency = "USD";

    Vector<TagValue> m_algoParams = new Vector<TagValue>();

    m_algoParams.add(new TagValue("StartTime", "14:40:00 EST"));

    m_algoParams.add(new TagValue("EndTime", "15:40:00 EST"));

    m_algoParams.add(new TagValue("MaxPctVolume", "47")); // Max % Volume

    m_algoParams.add(new TagValue("ExecutionStyle", "Normal"));

    //possible values for ExecutionStyle: Normal, Patient, Aggressive

    Order order = new Order();

    order.m_action = "BUY";

    order.m_totalQuantity = 100;
    
```

```

order.m_orderType = "LMT";

order.m_lmtPrice = 200.0;

order.m_algoStrategy = "TEX";

order.m_algoParams = m_algoParams;

order.m_transmit = false;

m_client.placeOrder(globalOrderId++, con, order);

}

```

**TWAP (TWAP)**

参数	描述	句法
startTime	开始时间	“9:00:00 EST”
endTime	结束时间:	“15:00:00 EST”
MaxPctVolume	最大百分比交易量	range: “0” – “99”

**TWAP (TWAP) Java编码举例**

```

void onTwapCsfbAlgo() {

Contract con = new Contract();

con.m_symbol = "IBM";

con.m_secType = "STK";

con.m_exchange = "CSFBALGO";

con.m_currency = "USD";

Vector<TagValue> m_algoParams = new Vector<TagValue>();
}

```

```

m_algoParams.add(new TagValue("StartTime", "14:40:00 EST"));

m_algoParams.add(new TagValue("EndTime", "15:40:00 EST"));

m_algoParams.add(new TagValue("MaxPctVolume", "48")); // Max % Volume

Order order = new Order();

order.m_action = "BUY";

order.m_totalQuantity = 100;

order.m_orderType = "LMT";

order.m_lmtPrice = 200.0;

order.m_algoStrategy = "TWAP";

order.m_algoParams = m_algoParams;

order.m_transmit = false;

m_client.placeOrder(globalOrderId++, con, order);

}
    
```

**VWAP (VWAP)**

参数	描述	句法
startTime	开始时间	“9:00:00 EST”
endTime	结束时间:	“15:00:00 EST”
MaxPctVolume	最大百分比交易量	range: “0” – “99”

**VWAP (VWAP) Java 编码举例**

```

void onVwapCsfbAlgo() {
    
```

```
Contract con = new Contract();

con.m_symbol = "IBM";

con.m_secType = "STK";

con.m_exchange = "CSFBALGO";

con.m_currency = "USD";

Vector<TagValue> m_algoParams = new Vector<TagValue>();

m_algoParams.add(new TagValue("StartTime", "14:40:00 EST"));

m_algoParams.add(new TagValue("EndTime", "15:40:00 EST"));

m_algoParams.add(new TagValue("MaxPctVolume", "49")); // Max % Volume

Order order = new Order();

order.m_action = "BUY";

order.m_totalQuantity = 100;

order.m_orderType = "LMT";

order.m_lmtPrice = 200.0;

order.m_algoStrategy = "VWAP";

order.m_algoParams = m_algoParams;

order.m_transmit = false;

m_client.placeOrder(globalOrderId++, con, order);
```

}



## 扩展定单属性

下面列出的扩展定单属性可以被用在所有placeOrder功能和Open\_Order( 开放定单) 事件中。

属性	可能的值
string m_tif	Day, GTC, IOC, GTD
string m_ocaGroup	确认—取消全定单组的数目。
string m_account	仅机构。
string m_openClose	仅机构。
int m_origin	仅机构。
string m_orderRef	客户定义的定单代号标签。
boolean m_transmit	指定是否定单将通过TWS发送。如果设为假，定单被创建但不被发送。
int m_parentId	母定单的定单代号，用于括号单、自动止损和跟踪止损单。
boolean m_blockOrder	如果设为真，指定定单为大额定单。
boolean m_sweepToFill	如果设为真，指定定单为一扫光定单。
int m_displaySize	用于下达冰山定单时公开披露定单尺寸。
int m_triggerMethod	<p>指定任何触发模拟止损、止损限价和跟踪止损定单。有效值是：</p> <ul style="list-style-type: none"> <li>• 0 - 默认值。“双买价/卖价”方法将被用于场外( OTC) 股票和美国期权定单。所有其它定单将使用“最后”方法。</li> <li>• 1 - 使用“双买价/卖价”方法，用于止损单的触发是基于两个连续买价或卖价。</li> <li>• 2 - “最后”方法，这里止损单的触发是基于最后价格。</li> <li>• 3 - “双最后”方法，这里止损单的触发是基于最后两个价格。</li> </ul>
boolean m_ignoreRth	如果设为真，允许在正常交易时段以外触发定单。
boolean m_hidden	如果设为真，在查看市场深度时定单将为不可视。仅应用于发送到INet的定单。

属性	可能的值
string m_goodAfterTime	表明应该在设定的时间和日期之后发送交易，使用的格式是YYYYMMDD HH:MM:SS (秒为可选)。如果不适用，使用一个空的字符串。
string m_goodTillDate	表明交易应该在设定的时间和日期达到之前保持工作状态，格式为YYYYMMDD HH:MM:SS(秒为可选)。当使用这个字符串时，你必须将tif设为GTD(直到时间有效)。如果不适用，使用一个空的字符串。
string m_faGroup	交易将被分配到的顾问组。如果不适用，使用一个空的字符串。
string m_faProfile	交易将被分配的顾问分配结构。如果不适用，使用一个空的字符串。
string m_faMethod	交易将被分配的顾问分配方法。如果不适用，使用一个空的字符串。
string m_faPercentage	有关交易分配的顾问百分比。如果不适用，使用一个空的字符串。
string m_primaryExch	澄清智能传递合约的任何疑问，包括目的地的主交易所，和智能目的地。
int m_shortSaleSlot	仅用于机构客户。 <ul style="list-style-type: none"> <li>• 0 - 不适用(例如，零售客户或非sshort边)</li> <li>• 1 - 清算经纪商</li> <li>• 2 - 第三方。如果使用这个值，你必须输入一个目的地地点。</li> </ul>
string m_designatedLocation	仅当shortSaleSlot value = 2时有效。否则保持空白或者定单将被拒绝。
long ocaType	执行后取消和Block = 1 执行后减少和Block = 2 执行后减少和没有Block = 3
int rthOnly	仅正常交易时段。 yes=1, no=0

属性	可能的值
String rule80A	Individual = 'T' Agency = 'A', AgentOtherMember = 'W' IndividualPTIA = 'J' AgencyPTIA = 'U' AgentOtherMemberPTIA = 'M' IndividualPT = 'K' AgencyPT = 'Y' AgentOtherMemberPT = 'N'
String settlingFirm	仅机构
String clearingAccount	定单的真正受益人。为报告给交易所，FUT/FOP定单必须发送这个值。
String clearingIntent	IB、Away、或PTA
int allOrNone	yes=1, no=0
long minQty	确定最低数量定单类型。
double percentOffset	相对定单的百分比抵消。
int eTradeOnly	使用电子报价交易。 yes=1, no=0
int firmQuoteOnly	使用公司报价交易。 yes=1, no=0
double nbboPriceCap	偏离NBBO的最大智能定单的距离。
long auctionStrategy	匹配 = 1 改善 = 2 透明 = 3 仅用于BOX交易所。
double startingPrice	开始价格。仅用于BOX交易所。
double stockRefPrice	股票参考价格。仅用于BOX交易所。
double delta	仅用于BOX交易所。
double stockRangeLower	可接受股票范围的低端值。仅用于BOX交易所。
double stockRangeUpper	可接受股票范围的高端值。仅用于BOX交易所。

属性	可能的值
double m_volatility	由TWSde期权分析计算的期权价格波动。这个值的表达是用百分比形式并用于计算发送给交易所的限价价格。
int m_volatilityType	1 = 日； 2 = 年
m_continuousUpdate	0 = 假； 1 = 真
int m_referencePriceType	1 = 平均； 2 = 买价或卖价
String m_deltaNeutralOrderType	输入一个可接受的定单类型，比如：MKT、LMT、REL等。
double m_deltaNeutralAuxPrice	输入对冲Delta定单类型要求的辅助价格。
int m_scaleNumComponents	对分段定单：定义母定单将被分隔的定单组数目，从而确定每个分组的单位数目。
int m_scaleComponentSize	对分段定单：定义每个分组的单位数目，确定母定单被分成的分组数目。
double m_scalePriceIncrement	对分段定单：定义每个分段组的价格增量。
double m_basisPoints	EFP定单
int basisPointsType	EFP定单

## 可用的市场扫描仪

下面表格中列出的可用的扫描仪(截至2008年7月)。

市场扫描仪(扫描代码)	描述
Low Opt Volume P/C Ratio (LOW_OPT_VOL_PUT_CALL_RATIO)*	看跌期权交易量除以看涨期权交易量, 显示具有最低比率的底层代码。
High Option Imp Vol Over Historical (HIGH_OPT_IMP_VOLAT_OVER_HIST)*	显示隐含和历史波动率之间最大差异的底层合约(股票或指数)。
Low Option Imp Vol Over Historical (LOW_OPT_IMP_VOLAT_OVER_HIST)*	显示隐含和历史波动率之间最小差异的底层合约(股票或指数)。
Highest Option Imp Vol (HIGH_OPT_IMP_VOLAT)*	显示下两个月内到期的近市价执行价期权的具有最高vega-加权隐含波动率的底层合约(股票或指数)。
Top Option Imp Vol % Gainers (TOP_OPT_IMP_VOLAT_GAIN)*	显示当前隐含波动率和昨天收盘前15分钟的平均隐含波动率值之间的最大百分比赢家的底层合约(股票或指数)。
Top Option Imp Vol % Losers (TOP_OPT_IMP_VOLAT_LOSE)*	显示当前隐含波动率和昨天收盘前15分钟的平均隐含波动率值之间的最大百分比输家的底层合约(股票或指数)。
High Opt Volume P/C Ratio (HIGH_OPT_VOLUME_PUT_CALL_RATIO)	看跌期权交易量除以看涨期权交易量和显示具有最高比率的底层代码。
Low Opt Volume P/C Ratio (LOW_OPT_VOLUME_PUT_CALL_RATIO)	看跌期权交易量除以看涨期权交易量, 显示具有最低比率的底层代码。
Most Active by Opt Volume (OPT_VOLUME_MOST_ACTIVE)	按期权交易量降序排列显示最活跃合约。
Hot by Option Volume (HOT_BY_OPT_VOLUME)	显示过去10天内平均最高期权交易量的底层合约。

市场扫描仪( 扫描代码)	描述
High Option Open Interest P/C Ratio (HIGH_OPT_OPEN_INTEREST_PUT_CALL_RATIO)	显示50个具有最高看跌/看涨比率的未平仓合约。
Low Option Open Interest P/C Ratio (LOW_OPT_OPEN_INTEREST_PUT_CALL_RATIO)	给出50个具有最低看跌/看涨比率的未平仓合约。
Top % Gainers (TOP_PERC_GAIN)	合约的最后交易价和前一天收盘价比较具有最高百分比变化的合约。
Most Active (MOST_ACTIVE)	基于TWS使用的单位( 美国股票的Lots; 衍生品和非美国股票的合约) 计算的今天具有最高交易量的合约。 样本表格的两个最活跃扫描: 最活跃列表, 用来显示NASDAQ、NYSE和AMEX市场上的最活跃合约; 最活跃美国, 用来显示美国的最活跃股票。
Top % Losers (TOP_PERC_LOSE)	合约的最后交易价和前一天收盘价比较具有最低百分比变化的合约。
Hot Contracts by Volume (HOT_BY_VOLUME)	合约为: <ul style="list-style-type: none"> <li>• 今天的交易量/最高的平均日交易量。</li> <li>• 平均日交易量( avgDailyVolume) 是合约日交易量的30天日平均值。</li> </ul>
Top % Futures Gainers (TOP_PERC_GAIN)	期货的最后交易价和前一天收盘价比较具有最高百分比增加的合约。

市场扫描仪( 扫描代码)	描述
Hot Contracts by Price (HOT_BY_PRICE)	合约为： <ul style="list-style-type: none"> <li>• (lastTradePrice-prevClose)(最后交易价格 - 前一天收盘价) /avgDailyChange(平均日变化) 是最高的绝对值( 正或负) 。</li> <li>• 平均日变化( avgDailyChange) 的定义为合约的合约的指数移动平均(dailyClose-dailyOpen)( 日收盘 - 日开盘)</li> </ul>
Top Trade Count (TOP_TRADE_COUNT)	日内的最多交易数。
Top Trade Rate (TOP_TRADE_RATE)	过去 60秒内具有最高交易次数的合约( 不考虑交易的尺寸) 。
Top Price Range (TOP_PRICE_RANGE)	今天的高价和低价的最大差，或昨天的收盘价比较，如果超出今天的范围。
Hot by Price Range (HOT_BY_PRICE_RANGE)	最大的价格波动范围( 从最大价格范围计算) 。
Top Volume Rate (TOP_VOLUME_RATE)	每分钟的最高交易量比率。
Lowest Option Imp Vol (LOW_OPT_IMP_VOLAT)	显示下两个月内到期的具有最低vega加权隐含波动率的近市价执行价期权的底层合约( 股票或指数) 。
Most Active by Opt Open Interest (OPT_OPEN_INTEREST_MOST_ACTIVE)	显示具有( 最高数目的未平仓看涨合约) +( 最高数目的未平仓看跌合约) 的头50个底层合约。
Not Open (NOT_OPEN)	今天还没有交易的合约。
Halted (HALTED)	被暂停交易的合约。
Top % Gainers Since Open (TOP_OPEN_PERC_GAIN)	显示最后交易和开盘价之间具有最高百分比价格增加的合约。

市场扫描仪( 扫描代码)	描述
Top % Losers Since Open (TOP_OPEN_PERC_LOSE)	显示最后交易和开盘价之间具有最高百分比价格减少的合约。
Top Close-to-Open % Gainers (HIGH_OPEN_GAP)	显示前一天收盘价和今天开盘价之间具有最高百分比价格增加的合约。
Top Close-to-Open % Losers (LOW_OPEN_GAP)	显示前一天收盘价和今天开盘价之间具有最低百分比价格减少的合约。
Lowest Option Imp Vol (LOW_OPT_IMP_VOLAT)	显示下两个月内到期的具有最低vega加权隐含波动率的近市价执行价期权的底层合约( 股票或指数) 。
Top Option Imp Vol % Gainers (TOP_OPT_IMP_VOLAT_GAIN)	显示当前隐含波动率和昨天收盘前15分钟的平均隐含波动率值之间的最大百分比赢家的底层合约( 股票或指数) 。
Top Option Imp Vol % Losers (TOP_OPT_IMP_VOLAT_LOSE)*	显示当前隐含波动率和昨天收盘前15分钟的平均隐含波动率值之间的最大百分比输家的底层合约( 股票或指数) 。
13-Week High (HIGH_VS_13W_HL)	过去13个星期内的最高价格。
13-Week Low (LOW_VS_13W_HL)	过去13个星期内的最低价格。
26-Week High (HIGH_VS_26W_HL)	过去26个星期内的最高价格。
26-Week Low (LOW_VS_26W_HL)	过去26个星期内的最低价格。
52-Week High (HIGH_VS_52W_HL)	过去52个星期内的最高价格。
52-Week Low (LOW_VS_52W_HL)	过去52个星期内的最低价格。



市场扫描仪( 扫描代码)	描述
EFP - High Synth Bid Rev Yield (HIGH_SYNTH_BID_REV_NAT_YIELD)	选中可用的最高合成EFP利率。这些利率的计算是使用SSF( 单一股票期货) 和底层股票之间的价格差, 以及扣除股息来计算SSF合约期间的年度合成隐含利率。最高利率可能表示存在一个投资机会。
EFP - Low Synth Bid Rev Yield (LOW_SYNTH_BID_REV_NAT_YIELD)	选中可用的最低合成EFP利率。这些利率的计算是使用SSF( 单一股票期货) 和底层股票之间的价格差, 以及扣除股息来计算SSF合约期间的年度合成隐含利率。低利率可能代表一个借贷机会。

**市场扫描仪的产品和地点代码**

TWS API的市场扫描仪支持以下产品和地点代码:

**产品**

- STK - 美国股票
- STOCK.HK - 亚洲股票
- STOCK.EU - 欧洲股票

**地点代码**

- STK.US - 美国股票
- STK.US.MAJOR - 美国股票( 没有粉纸)
- STK.US.MINOR - 美国股票( 仅粉纸)
- STK.HK.SEHK - 香港股票
- STK.HK.ASX - 澳大利亚股票
- STK.EU - 欧洲股票

## 支持的时区

下面列出的是TWS API支持的时区。

时区	描述
GMT	格林威治标准时间
EST	东部标准时间
MST	山地标准时间
PST	太平洋标准时间
AST	大西洋标准时间
JST	日本标准时间
AET	澳大利亚标准时间

## 智能组合传递

这些功能用于传递到智能的担保和非担保组合定单，可用性基于组合类型和定单类型。例如，用户可以指定一次发送的最大尺寸(最大的边组合尺寸)和哪个边应该是发送的第一个边。

通过使用 `smartComboRoutingParams`(要求TWS版本57或之后的版本)，Active X、C++ 和 Java API也支持智能组合传递。`smartComboRoutingParams`与使用边/值对添加参数到组合定单的 `AlgoParams`类似。参数包括下列性能：

- **Priority( 优先)** - 用户可以指定第一个执行的边。

Tag = `LegInPrio`  
 Values = -1, 0 或 1

- **Discretionary Amount( 全权委托金额)** - 当一个边被执行后，我们可以对其它边进行调整的数额可以上至全权委托金额。

Tag = `MaxSegSize`  
 Value = 金额

- **Market-If-Touched Timeout( 触及市价超时)** - 用于触及市价组合定单，我们记录第一个执行边的第一个执行时间( `firstTradeTime`)，和最后部份执行的最后执行市价( `lastTradeTime`)。对这类的定单，你现在可以指定最后执行的超时值和自第一个执行的超时值(以秒计算)。

Tags = `ChangeToMktTime1` 是最后执行之后的超时，`ChangeToMktTime2` 是第一个执行之后的超时。  
 Value = 秒数

- **Market-If-Touched Stop-Loss( 触及市价止损)** - 指定每个组合的绝对止损金额。如果指定了并如果组合的隐含执行价格(基于已经执行的边和当前市场数据)超出组合价格加上止损金额，我们将立即转换限价单( `LMT`)为市价单( `MKT`)，以便完成执行组合定单。如果指定了止损金额，但还未指定超时值，我们将继续努力按计算的 `LMT`价格执行第二个边，直到获得执行或达到止损金额。

Tag = `ChangeToMktOffset`  
 Value = 金额

- **Maximum Leg-In Size( 最大的边数量)** - 指定最大允许的每段的边数量。

Tag = `MaxSegSize`  
 Value = 组合尺寸单位

- **Discretionary Percentage( 全权委托百分比)** - 指定组合价格的百分比。用于分段组合定单，其全权委托额的计算是基于当前的分段级别。当输入的全权委托额是一个百分比形式，API根据组合将其转换为具体金额。该金额将随着价格的变化或分段定单的每个级别而更新。你可以在一个时间为这个参数或者为全权委托额扩展属性输入一个值，但不能同时为两者输入值。

Tag = `DiscretionaryPct`  
 Value = 位于0和100之间的值。



# 索引

## A

about the APIs 24

account details in ActiveX for Excel 355

account information in Excel 75

account information, viewing in ActiveX for Excel 356

Account page

    in ActiveX for Excel 355

    using in Excel 75

Account page in Excel 74

    account values 76

Account page toolbar 76

Account page, using in ActiveX for Excel 356

account values in Excel 76

accountSummary() 154, 217, 283

accountSummaryEnd() 156, 219, 286

Active X 117

Active X events 150

ActiveX

    linking to TWS 118

    placing a combination order 177

    registering third-party controls 119

ActiveX API

    on 64-bit systems 120

ActiveX COM objects 160-163, 165-166, 173-175

ActiveX events 138-143, 145, 148-150, 152-154, 157-159

ActiveX factory methods 136-137

ActiveX for Excel

    Account page 355

    Advanced Orders page 345

    Advisors page 375

    allocating shares to a single account 376

    Basic Orders page 337

    Bond Contract Details page 367

    bracket orders in 347

    Bulletins page 334

    connecting to TWS 330

    Contract Details page 365

    disconnecting from TWS 330

    download API components 326

    Executions page 359

    Extended Order Attributes page 351

    FA orders using account group and method 377

    FA orders using allocation profile 377

    Fundamentals page 373

    General page 329

    getting started 326

    Historical Data page 361

    Log page 378

- Market Depth page 335
- Market Scanner page 371
- Open Orders page 353
- opening sample spreadsheet 327
- placing orders in 338
- Real Time Bars page 369
- relative orders in 350
- requesting current time 330
- scale orders in 349
- setting server log level in 330
- Tickers page 331
- trailing stop limit orders in 348
- ActiveX for Excel historical data
  - expired contracts 362
- ActiveX for Excel on 64-bit Windows 326
- ActiveX for Excel sample spreadsheet
  - using 328
- ActiveX methods 122-124, 127, 129, 134-135
- ActiveX properties 176
- ActiveX sample program 121
- Advanced Orders
  - in ActiveX for Excel 345
- Advanced Orders page
  - in Excel 66
- Advanced Orders page toolbar 70
- advisors 313
  - financial reporting for 317
- Advisors
  - change or update allocation information 322
  - Java code samples for 320
  - place order for a single managed account 320
  - place order for an account group 321
  - place order for an allocation profile 321
- Advisors page 99-100
  - in ActiveX for Excel 375
- Advisors page in Excel 98
- Advisors page toolbar 100
- advisors, Excel DDE support for 99-100, 315
- algos
  - CSFB 420
- allocating shares to a single account in Excel 99
- allocation methods for account groups 318
- allocation profiles in Excel 100
- API
  - about 24
  - recommendations 28
- API components, downloading 326
- API components, using 24
- API logging 29
- API message codes 384
- API overview 23
- API request/server response message identifiers 30
- API settings in TWS 37, 46

- API software
  - downloading 45
  - uninstalling 41
- API, for financial advisor accounts 313
- apply extended order attributes 57
- Apply Extended Template button 352
- Arrival Price Java code sample 413
- available market scanners 443
- available market scanners in Excel 89
- AvailableEquity Method 318
- B**
- bar size settings for historical data 31
- Basic Orders page
  - combination orders 338
  - in ActiveX for Excel 337
  - in Excel 52
  - modifying orders 338
  - placing orders in ActiveX for Excel 338
  - toolbar buttons 340
- Basic Orders page toolbar in Excel 55
- basket orders in Excel 53
- basket orders, in ActiveX for Excel 338
- bond contract details 96
- Bond Contract Details page
  - in ActiveX for Excel 367
- Bond Contract Details page in Excel 95
- Bond Contract Details page toolbar 96
- bond contract details, requesting in ActiveX for Excel 368
- bondContractDetails() 150, 214, 280
- bracket orders
  - in ActiveX for Excel 347
  - in Excel 66
- Bulletins page
  - in ActiveX for Excel 334
  - toolbar buttons 335
- C**
- C++ 181, 202-203
  - Class EClientSocket methods 188
  - Class EWrapper functions 204
  - combinations orders 239
  - linking to TWS 182
  - prebuilt sample application 187
  - running the TestSocketClient from Visual Studio 2008 187
  - using the sample program 187
- C++ SocketClient properties 223
- calcOptionPriceAndGreeks 125
- calculateImpliedVolatility 124, 191, 258
- calculateOptionPrice 191, 258
- calendar spread in Excel 54
- calendar spread order in Excel 339
- calendar spread order, in C++ 239
- cancelAccountSummary() 132, 199, 266
- cancelCalculateImpliedVolatility 124, 191, 258

- cancelCalculateOptionPrice 125, 192, 258
- cancelFundamentalData 137, 203
- cancelFundamentalData() 270
- cancelHistoricalData() 134, 201, 269
- cancelMktData() 124, 191, 257
- cancelMktDepth() 128, 195, 262
- cancelNewsBulletins() 129, 195, 262
- cancelOrder() 126, 192, 259
- cancelPositions() 132, 199, 266
- cancelRealTimeBars() 135, 202, 269
- cancelScannerSubscription() 135, 201, 267
- checkMessages() 190
- Class EClientSocket methods 188-196, 199, 201-202
- Class EWrapper functions 204-209, 211-216, 220-222
- code for DDE for Excel API 102
- code modules in Excel 103
- com.ib.client Java package 253
- combination order, in ActiveX 177
- combination order, in ActiveX for Excel 338
- combination orders in Excel 54
- combination orders, in C++ 239
- combination orders, in Java 306
- ComboLeg 228, 295
- commissionReport 215
- CommissionReport 237, 291
- commissionReport() 152, 281
- common issues and solutions 42
- conditional Orders page 62
- Conditional Orders page 341
  - in Excel 62
- conditional orders in Excel, examples of 64, 343
- Conditional Orders page
  - toolbar buttons 345
- Conditional Orders page toolbar 65
- Conditional Orders page, modifying orders 65, 344
- conditional orders, in Excel 62, 342
- configure TWS 25, 46
- connect() 123
- connecting to TWS
  - using ActiveX for Excel 330
- connectionClosed() 139, 205, 272
- Contract 224, 292
- contract details 94
- Contract Details page
  - in ActiveX for Excel 365
- Contract Details page in Excel 94
- Contract Details page toolbar 94
- contract details, requesting in ActiveX for Excel 366
- contract parameters
  - samples in Java 309
- ContractDetails 226, 293
- contractDetails() 150, 214, 280
- contractDetailsEx() 150



createComboLegList() 136  
createContract() 136  
createExecutionFilter() 136  
createOrder() 136  
createScannerSubscription() 136  
createTagValueList 136  
createUnderComp() 137  
creating a ticker in ActiveX for Excel 332  
creating a ticker in Excel spreadsheet 49  
CSFB algo parameters 420  
current time  
    requesting in ActiveX for Excel 330  
currentTime() 139, 205, 271

**D**

DDE defined 43  
DDE for Excel 43  
    downloading 45  
    getting started with 44  
    macros 103  
    modules 103  
    named ranges in 104  
    open the spreadsheet 47  
    syntax 105  
    viewing the code 102  
DDE for Excel for Advisors 98  
DDE for Excel reference 102  
DDE for Excel spreadsheet pages 48

DDE links  
    removing 51  
DDE syntax 105  
delta-neutral RFQs 39  
determine a futures contract in Java 310  
determine a stock in Java 310  
determine an option contract in Java 309  
disconnect() 123  
disconnecting from TWS  
    using ActiveX for Excel 330  
downloading API components 326  
downloading API software 45  
Dynamic Data Exchange 43

**E**

Eclient Socket methods 260  
EClient Socket methods 255-257, 259-263, 266-267, 269-270  
EClientSocket functions 188  
EClientSocket() 189, 256  
Eclipse  
    running Java test client in 250  
eConnect() 189, 256  
eDisconnect() 189, 256  
Enable DDEclients setting in TWS 46  
EqualQuantity Method 318  
errMsg() 139

- error messages
  - viewing in ActiveX for Excel 378
- error() 205, 272
- EWrapper methods 271
- Excel
  - Advisors page 98
  - Bond Contract Details 95
  - Contract Details page 94
  - Historical Data page 82
  - Market Depth page 96
  - Market Scanner page 86
  - market scanner parameters 87
  - starting market scanner 87
  - viewing your portfolio in 81
- Excel Advanced Order page 70
- Excel API 43
  - getting market data 50
  - supported order types 55
- Excel DDE 325
  - pages 328
  - supported order types 340
- Excel DDE sample spreadsheet, installing 326
- Excel DDE, extended order attributes 351
- Excel DDE, Extended Order attributes page 351
- Excel DDE, financial advisor support 315
- Excel DDE, supported order types 340
- Excel modules 103
- Excel sample spreadsheet 48
  - opening 47
- Excel spreadsheet
  - Advanced Orders page 66
  - Basic Orders page 52
  - Conditional Orders page 62
  - Executions page 72
  - Extended Order Attributes page 56
  - Open Orders page 70
  - placing orders 53
  - removing all links 51
  - setting log detail level 51
  - setting processing rate 51
  - setting refresh rate 50
  - Tickers page 49
- execDetails() 214, 280
- execDetailsEnd() 152, 215, 281
- execDetailsEx() 152
- Execution 223, 290
- Execution page toolbar 73
- execution reporting, for financial advisors 317
- execution reporting, in ActiveX for Excel 359
- execution reports
  - running in Excel 74
- ExecutionFilter 224, 291
- executions
  - viewing in Excel 72

- Executions page
    - in ActiveX for Excel 359
  - Executions page in Excel 72
  - Executions Reporting page in Excel 73
  - executions, viewing in ActiveX for Excel 360
  - exerciseOptions() 193, 260
  - exerciseOptionsEx() 127
  - exercising options
    - in ActiveX for Excel 359
  - expired contracts
    - historical data in Excel 83
  - extended order attributes 439
    - applying to individual or groups of orders 352
    - applying to orders 57
  - extended order attributes in Excel 57
  - Extended Order Attributes page
    - in ActiveX for Excel 351
    - in Excel 56
  - extended order attributes, manually programming
    - in ActiveX for Excel 57, 352
- F**
- FA account groups in Excel 100
  - FA information
    - in ActiveX for Excel 357
  - FA managed account codes in Excel 75
  - FA managed accounts, in ActiveX for Excel 356
    - Portfolio page 358
  - FA orders 100
    - allocating shares to a single account 376
    - allocation profiles 377
    - using account group and method 377
  - FA orders in ActiveX for Excel 375
  - FA page in Excel 98
  - filtering executions in Excel 73
  - financial advisors 313
    - allocation methods for account groups 318
    - execution reporting for 317
    - orders and account configuration 314
  - financial advisors, Excel DDE support for 315
  - financial advisors, support by other API technologies 316
  - fundamental data
    - in ActiveX for Excel 374
    - report types 374
  - fundamental data() 289
  - fundamental ratios
    - in ActiveX for Excel 374
  - FUNDAMENTAL\_RATIOS tickType 402
  - fundamentalData() 159, 222
  - Fundamentals page
    - in ActiveX for Excel 373
- G**
- General page
    - toolbar buttons 331

generic tick types 400

getting started

    DDE for Excel 44

    with ActiveX for Excel

        sample spreadsheet 326

## H

historical data

    duration and bar size settings 31

    viewing in ActiveX for Excel 362

historical data in Excel 82

    query specification fields 84

historical data limitations 31

Historical Data page

    in ActiveX for Excel 361

    query specification fields 363

Historical Data page in Excel 82

Historical Data page toolbar 84

historicalData() 157, 220, 287

## I

IB Gateway

    running the API through 26

IBAlgo parameters 412

IBAlgos 410, 412

IBDividends tick type example 408

IComboLeg 165

IComboLegList 166

ICommissionReport 162

IContract 162

IContractDetails 163

IExecution 160

IExecutionFilter 161

if-filled order, in Excel 64, 343

Index Premium data 40

installing Excel DDE sample spreadsheet 326

instrument codes for market scanners 447

IOrder 166

IOrderComboLeg 173

IOrderState 173

IScannerSubscription 174

isConnected() 189, 256

ITagValue 175

ITagValueList 175

IUnderComp 175

## J

Java 241

    combination orders 306

    linking to TWS 242

    sample program 246

Java API overview 253

Java code samples 309-310

    for FAs 320-322

Java EClient Socket methods 255

Java EWrapper methods 271-276, 278-283, 287-289

java sample program 252  
    classes 252  
Java SocketClient properties 290-293, 295-296,  
    303-305  
Java Test Client 252  
    overview 252  
    running 246  
Java Test Client and Eclipse 250

## L

limitations  
    of historical data requests 31  
linking to TWS, using ActiveX 118  
location codes for market scanners 447  
log detail in Excel 51  
Log page  
    in ActiveX for Excel 378  
logging 29

## M

macros in Excel 103  
managedAccounts() 154, 216, 283  
market data in Excel 50  
market depth  
    requesting in ActiveX for Excel 336  
Market Depth page  
    in ActiveX for Excel 335  
    toolbar buttons in ActiveX for Excel 336  
    using ActiveX for Excel 336  
    using in Excel 97

Market Depth page in Excel 96  
Market Depth page toolbar 98  
Market Scanner page  
    in ActiveX for Excel 371  
Market Scanner page in Excel 86  
Market Scanner page toolbar 89  
market scanner parameters  
    in ActiveX for Excel 372  
market scanner parameters in Excel 87  
market scanner subscription  
    starting in Excel 87  
market scanner subscription, starting in ActiveX for  
    Excel 372  
market scanners 443  
    available in Excel 89  
    instruments and locations codes for 447  
marketDataType() 143, 209, 276  
message codes 384  
modifying orders in ActiveX for Excel 338  
modifying orders in the DDE for Excel API 53  
modifying orders, on Conditional Orders page 65, 344

## N

Name Manager in Excel 104  
named ranges in Excel 104  
NetLiq Method 318  
nextValidId() 145, 211, 278

**O**

- open orders
  - removing in Excel 71
  - viewing in Excel 71
- Open Orders page
  - ActiveX for Excel 353
  - in Excel 70
- Open Orders page toolbar 71
- open orders, viewing in ActiveX for Excel 354
- opening the ActiveX sample program 121
- openOrder() 211, 278
- openOrderEx() 145
- options
  - exercising in ActiveX for Excel 359
- Order 228, 296
- order IDs 35
- order in Excel
  - modifying 53
- order status event 36
- order status for partial fills 36
- order types 410
- order types in Excel DDE 340
- order types, in Excel 340
- OrderComboLeg 296
- orders 33
  - in Excel 66-69
  - placing in ActiveX for Excel 338

- orders and account configuration, for financial advisors 314

- orders in Excel API 53

- orders in Java

- for a single managed account 320

- for an account group 321

- for an allocation profile 321

- orders, modifying in ActiveX for Excel 338

- OrderState 235, 303

- orderStatus() 143, 209, 276

- overview 23

- Java API 253

**P**

- pages 328

- pages in Excel spreadsheet 48

- partial fills and order status 36

- PctChange Method 318

- permId() 145

- placeOrder() 192, 259

- placeOrderEx() 125

- placing orders

- basket 338

- combination order in ActiveX for Excel 338

- conditional orders in Excel 342

- placing orders in ActiveX for Excel 338

- placing orders in Excel 53

- portfolio data in FA managed accounts, in ActiveX for Excel 358

- Portfolio page
  - in FA managed accounts, in ActiveX for Excel 358
- Portfolio page in Excel 80
- Portfolio page toolbar 81
- portfolio, viewing in FA managed accounts, in ActiveX for Excel 358
- position() 156, 219, 286
- positionEnd() 157, 220, 287
- POSIX 381
  - running client on Windows machine 382
- premium data 40
- price-change order, in Excel 64, 344
- processing rate in Excel 51
- Q**
- query specification fields for historical data in Excel 84
- query specification fields, on Historical Data page in ActiveX for Excel 363
- R**
- real time bars
  - in ActiveX for Excel 369
- realtimeBar() 158, 221, 288
- receiveFA() 154, 216, 283
- recommendations for using API 28
- reference 383
- refresh rate in Excel 50
- refresh rate, for market depth in Excel 336
- refresh rate, on ActiveX for Excel Tickers page 333
- registering third-party ActiveX controls 119
- relative orders
  - in Excel 69
- relative orders, in ActiveX for Excel 350
- removing DDE links 51
- replaceFA() 196, 263
- reqAccountSummary 130, 196, 263
- reqAccountUpdates() 128, 194, 261
- reqAllOpenOrders 260
- reqAllOpenOrders() 126, 193
- reqAutoOpenOrders() 126, 193, 260
- reqContractDetails() 194, 261
- reqContractDetailsEx() 128
- reqCurrentTime() 123, 189, 257
- reqExecutions() 194, 261
- reqExecutionsEx() 127
- reqFundamentalData 137, 202
- reqFundamentalData() 270
- reqHistoricalData() 199, 267
- reqHistoricalDataEx() 132
- reqIds() 127, 193
- reqIDs() 260
- reqManagedAccts() 129, 195, 262
- reqMarketDataType() 125, 192, 259
- reqMktData() 190, 257
- reqMktDataEx() 124

reqMktDepth() 195, 262  
reqMktDepthEx() 128  
reqNewsBulletins() 129, 195, 262  
reqOpenOrders() 126, 193, 259  
reqPositions() 132, 199, 266  
reqRealTimeBars() 202, 269  
reqRealTimeBarsEx() 135  
reqScannerParameters() 134, 201, 266  
reqScannerSubscription() 201, 266  
reqScannerSubscriptionEx() 134  
Request for Quote 39  
request market depth in Excel 97  
requestFA() 129, 196, 263  
requesting bond contract details in ActiveX for Excel 368  
requesting bond contract details in Excel 96  
requesting contract details in ActiveX for Excel 366  
requesting contract details in Excel 94  
requesting market data, in ActiveX for Excel 333  
requesting market depth  
    in ActiveX for Excel 336  
Reuters global fundamentals  
    in ActiveX for Excel 373  
RFQs 39  
RTVolume 408  
running execution reports in Excel 74  
running the API through IB Gateway 26

## S

sample program  
    ActiveX 121  
    C++ 187  
    Java 246  
sample spreadsheet  
    Conditional Orders page 341  
    opening 327  
    pages in 328  
sample spreadsheet, installing 326  
scale orders  
    in ActiveX for Excel 349  
    in Excel 69  
scannerData() 220, 287  
scannerDataEnd() 158, 221, 288  
scannerDataEx() 157  
scannerParameters() 157, 220, 287  
ScannerSubscription 236, 304  
server log level  
    setting in ActiveX for Excel 330  
serverVersion() 189, 257  
setLogLevel() 190  
setServerLogLevel() 123, 256  
SHORTABLE tick 401  
smart combo routing 449  
smartCombotRoutingParams 449  
Socket Client Properties, in Java 291



SocketClient Properties 223-224, 226, 228, 235-237

SocketClient properties, in Java API 290

software

    downloading 45

spreadsheet pages 48

starting market scanner in ActiveX for Excel 372

supported order types 410

supported order types in Excel 55

supported time zones 448

## T

tables 383

TAG values for FUNDAMENTAL\_RATIOS 402

TestJavaClient 246

third-party controls, for ActiveX 119

tick types 397

tickEFP() 142, 208, 275

ticker

    creating in Excel 49

ticker, creating in ActiveX for Excel 332

Tickers page

    in ActiveX for Excel 331

    in Excel 49

    requesting market data in ActiveX for Excel 333

    setting the refresh rate in ActiveX for Excel 333

    toolbar buttons in ActiveX for Excel 333

    using 49

Tickers page toolbar 51

Tickers page, using ActiveX for Excel 332

tickGeneric() 141, 207, 274

tickOptionComputation() 140, 206, 273

tickPrice() 139, 272

tickPrice()Class EWrapper Functions 205

tickSize() 140, 206, 273

tickSnapshotEnd() 142, 208, 275

tickString() 141, 207, 274

time zones 448

toolbar

    Historical Data page 84

toolbar buttons

    Advanced Order page 70

    on ActiveX for Excel Advanced Orders page 350

    on ActiveX for Excel Advisors page 378

    on ActiveX for Excel Basic Orders page 340

    on ActiveX for Excel Bond Contract Details page 369

    on ActiveX for Excel Contract Details page 367

    on ActiveX for Excel Executions page 361

    on ActiveX for Excel Fundamentals page 375

    on ActiveX for Excel Historical Data page 365

    on ActiveX for Excel Market Depth page 336

    on ActiveX for Excel Market Scanner page 373

    on ActiveX for Excel Open Orders page 355

- on ActiveX for Excel Portfolio page 359
- on ActiveX for Excel Real Time Bars page 370
- on Conditional Orders page 345
- on FA managed accounts, in ActiveX for Excel Account page 357
- toolbar buttons, on ActiveX for Excel Bulletins page 335
- toolbar buttons, on ActiveX for Excel General page 331
- toolbar buttons, on ActiveX for Excel Tickers page 333
- toolbars
  - Advisors page 100
  - Basic Orders page 55
  - Bond Contract Details 96
  - Conditional Orders page 65
  - Contract Details 94
  - Execution page 73
  - Market Depth page 98
  - Market Scanner page 89
  - Open Orders page 71
  - Portfolio page 81
  - Tickers page 51
- trailing stop limit orders
  - Excel 68
- trailing stop limit orders, in ActiveX for Excel 348
- troubleshooting 42

- TWS
  - linking from Java 242
- TWS API settings 37
- TWS log file 29
- TWS precautionary settings 33
- TWS, configuring for API 25
- TWS, linking from C++ 182
- TWS, linking using ActiveX 118
- TwsConnectionTime() 190, 257
- TwsSocketClient.dll 182
  - linking to 182

## U

- updateMktDepthL2() 282
- UnderComp 237, 305
- uninstalling the API software 41
- updateAccountTime() 149, 213, 279
- updateAccountValue() 145, 212, 278
- updateMktDepth() 152, 215, 281
- updateMktDepthL2() 153, 215
- updateNewsBulletin() 149, 213, 282
- updatePortfolio() 212, 279
- updatePortfolioEx() 148
- using Account page in ActiveX for Excel 356
- using Account page in Excel 75
- using API components 24
- using the ActiveX for Excel sample spreadsheet 328
- using the ActiveX for Excel Tickers page 332

using the Market Depth page in ActiveX for  
Excel 336

using the Tickers page 49

util module in Excel 103

## V

VB\_API\_sample.vbp

opening 121

viewing code in Excel 102

viewing executions in Excel 72

viewing executions, in ActiveX for Excel 360

viewing historical data in ActiveX for Excel 362

viewing historical data in Excel 82

viewing open orders in ActiveX for Excel 354

viewing portfolio data in FA managed accounts,  
in ActiveX for Excel 358

viewing your portfolio in Excel 81

Visual Basic editor 102

Visual Basic sample program, for ActiveX 121

VOL orders

in ActiveX for Excel 347

volatility orders

in Excel 67

## W

winError() 205