

# 元大 Smart API

## 使用手冊

(v 1.0.0)



# 目錄

<b>1. 下載&amp;安裝 元大 Smart API.....</b>	<b>5</b>
1-1 下載&安裝 元大 Smart API .....	5
1-2 開通 API 權限 .....	8
<b>2. 下載&amp;安裝 R &amp; Python .....</b>	<b>9</b>
2-1 R .....	9
2-2 Python .....	9
<b>3. 元大 Smart API 架構說明.....</b>	<b>10</b>
3-1 交易 API .....	10
3-2 行情 API .....	10
3-3 下單 API .....	10
3-4 完整流程.....	11
<b>4. 登入元大 Smart API 進行下單 .....</b>	<b>12</b>
4-1 登入.....	12
4-2 輸入商品代號.....	13
4-3 觀察行情寫入情況.....	13
4-4 測試下單.....	14
<b>5. 使用命令提示字元(cmd.exe)基本操作.....</b>	<b>15</b>
5-1 Command line 簡易操作 .....	15
5-2 行情取用 (tail.exe).....	16
5-3 切換商品 (ChangeProdid.exe) .....	17
5-4 下限價單 (Order.exe).....	17

5-5	下市價單 (Order.exe).....	18
5-6	刪單 (Order.exe).....	18
<b>6.</b>	<b>使用命令提示字元(cmd.exe)進階操作.....</b>	<b>19</b>
6-1	查詢委託回報 (GetAccount.exe).....	19
6-2	查詢未平倉 (OnOpenInterest.exe).....	20
6-3	查詢帳戶資訊 (FutureRights.exe) .....	20
6-4	刪除所有掛單 (CancelALL.exe).....	20
6-5	取得所有未成交之委託單 (GetUnfinished.exe).....	20
6-6	緊急平倉 (MayDay.exe) .....	21
<b>7.</b>	<b>使用 R 串接下單&amp;行情.....</b>	<b>22</b>
7-1	R 呼叫外部指令.....	22
7-2	R 進行基礎操作.....	23
7-2-1	行情取用.....	23
7-2-2	下單.....	24
7-2-3	切換商品.....	24
7-2-4	刪單.....	25
7-3	R 進行進階操作.....	25
7-3-1	查詢委託回報.....	25
7-3-2	查詢未平倉.....	26
7-3-3	查詢帳戶資訊.....	26
7-3-4	刪除所有掛單.....	26
7-3-5	取得所有未成交之委託單.....	27
7-3-6	緊急平倉.....	27
<b>8.</b>	<b>使用 Python 串接下單&amp;行情.....</b>	<b>28</b>
8-1	Python 呼叫外部指令 .....	28

8-2	Python 進行基礎操作 .....	29
8-2-1	行情取用 .....	29
8-2-2	下單 .....	31
8-2-3	切換商品 .....	31
8-2-4	刪單 .....	31
8-3	Python 進行進階操作 .....	32
8-3-1	查詢委託回報 .....	32
8-3-2	查詢未平倉 .....	32
8-3-3	查詢帳戶資訊 .....	33
8-3-4	刪除所有掛單 .....	33
8-3-5	取得所有未成交之委託單 .....	33
8-3-6	緊急平倉 .....	33
9.	附錄 - 商品代碼表 .....	34

# 1. 下載&安裝 元大 Smart API

## 1-1 下載&安裝 元大 Smart API

首先上元大期貨的官方網站(<http://www.yuantafutures.com.tw>)，依以下步驟進行:

1. 點選左側元大 SmartAPI [紅框處]進入說明頁面



2. 點選下載元大 SmartAPI 進行下載



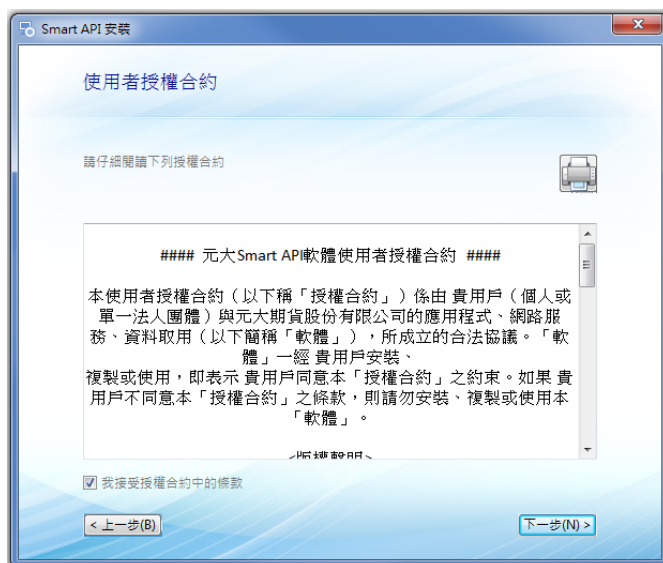
3. 解壓縮該檔案，資料夾中包含了 1 個安裝檔(元大 SmartAPI.exe)、9 個執行檔以及 1 個 README 說明檔。

名稱	修改日期	類型	大小
元大Smart API	2018/4/23 下午 0...	應用程式	27,055 KB
tail	2018/1/3 上午 05...	應用程式	46 KB
Order	2018/3/19 下午 0...	應用程式	7 KB
OnOpenInterest	2018/3/19 下午 0...	應用程式	6 KB
MayDay	2018/3/19 下午 0...	應用程式	5 KB
GetUnfinished	2018/3/19 下午 0...	應用程式	6 KB
GetAccount	2018/3/19 下午 0...	應用程式	6 KB
FutureRights	2018/3/19 下午 0...	應用程式	6 KB
ChangeProdid	2018/3/19 下午 0...	應用程式	6 KB
CancelALL	2018/3/19 下午 0...	應用程式	5 KB
README	2018/4/16 下午 0...	文字文件	1 KB

4. 點選元大 SmartAPI 進行安裝



點選下一步



勾選「接受授權合約」並點選下一步



選擇「典型」安裝



成功完成安裝

## 1-2 開通 API 權限

若要透過元大 SmartAPI 來實做程式交易，需要開通行情 API 與下單 API，開通方式如下：

- 行情 API：請找所屬營業員填寫行情 API 開通申請表
- 下單 API：由以下網址完成線上開通

<https://www.yuantafutures.com.tw/Appointment/pages/member-manage/ESign.aspx>

或由首頁右方[線上服務申請]



接著點選[線上簽署]，也可以進入申請頁面



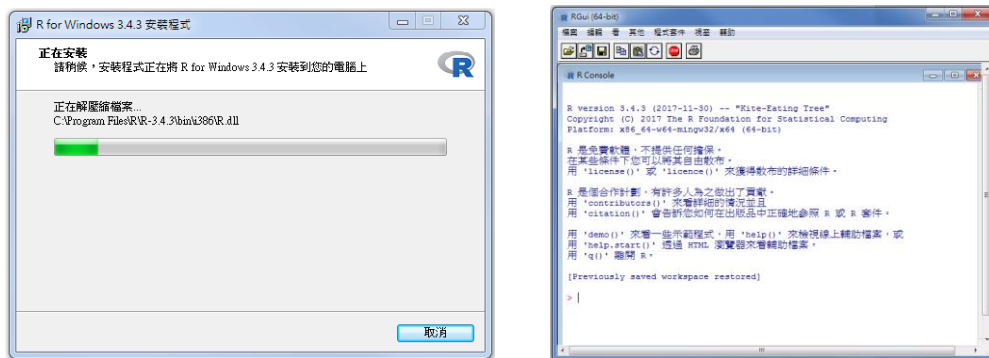
如果下單 API 沒開通，會導致登入時被阻擋，如果行情 API 沒開通，依然可以登入，但是在取用商品報價的時候會有問題，開通完成後即可登入元大 Smart API 進行使用。



## 2. 下載&安裝 R & Python

### 2-1 R

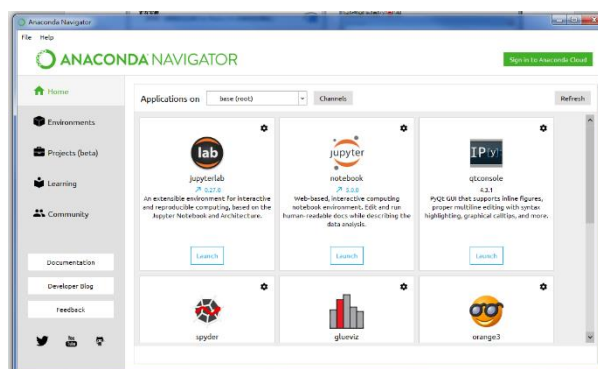
至 R 的官方網站(<https://cran.r-project.org/bin/windows/base/>)下載安裝檔，下載好後開啟安裝檔，選擇繁體中文，使用 default 的設定進行安裝，一直按下一步到結束，安裝好了之後，由桌面或是開始功能表打開 R，若看到以下畫面即代表安裝完成。



### 2-2 Python

Python 的安裝方式有很多種，以下採用 Anaconda 的安裝方式，比較方便並且不容易出問題，首先先上 Anaconda 的官方網站載點(<https://www.anaconda.com/download/>)下載安裝程式，Python 3.6 和 Python 2.7 原則上都可以運作，只不過在後面的程式碼撰寫上會有一點不相同，本說明書以 Python 3.6 為教學範例。

安裝的方式一樣是主程式打開，使用預設值下一步按到底，完成安裝後打開桌面上的 Anaconda Navigator，若看到以下畫面代表安裝成功。



## 3. 元大 Smart API 架構說明

### 3-1 交易 API

API 的全名是應用編程介面(Application Programming Interface)，是為了程式間互相溝通所設計出來的介面，因為要完成交易的整體流程，必須取得行情與下單指令，所以交易 API 本質上是由行情 API 與下單 API 所組成，以下分別描述元大萬用 API 如何實作這兩個功能。

### 3-2 行情 API

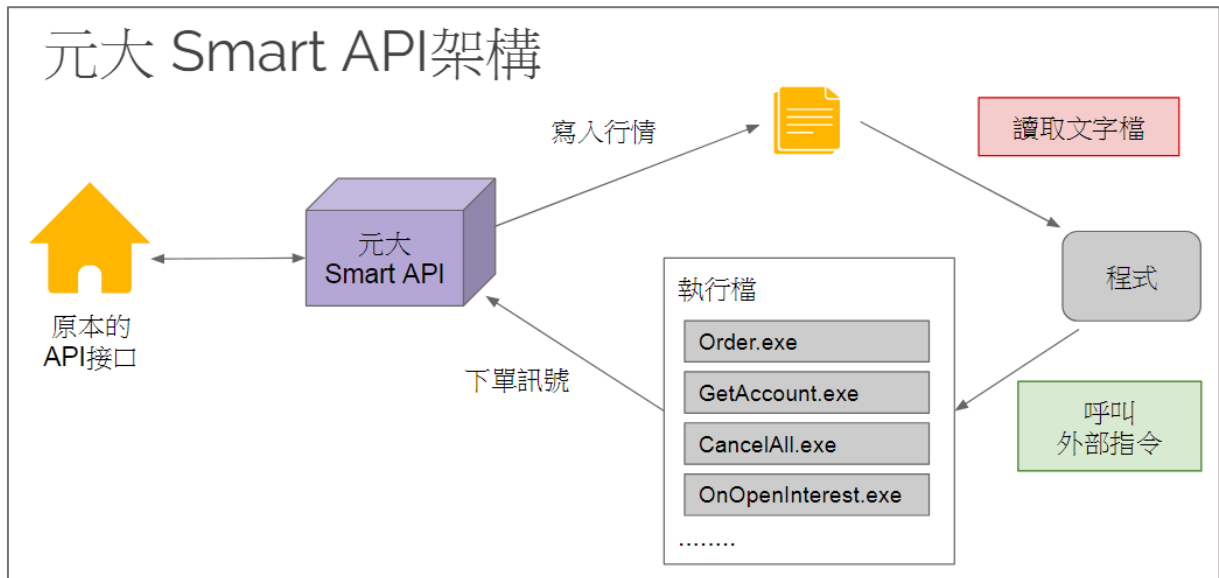
行情 API 是透過下單機取得使用者設定的商品代號後，向行情主機取得行情資料，寫出一個記載著行情的文字檔至指定路徑，當我們的交易程式需要取得行情時，就用讀取該文字檔的方式取得即時的成交、委託行情，完成行情 API 的取用。

### 3-3 下單 API

下單 API 則是透過事先封裝好的執行檔(.exe)是執行下單的動作，以下舉例，若執行 `Order.exe TXFD8 B 10800 1 LMT IOC 1` 指令，透過呼叫 Order.exe 執行檔的同時傳入相對應的參數，就可以執行：`商品代號 TXFD8 的商品下出 10800 的 1 口限價買單`，另外還有市下單、刪單、委託查詢、未平倉查詢、帳務查詢...等等指令，也都有相對應的執行檔，詳細的使用說明請參考第 5 章與第 6 章的說明。

### 3-4 完整流程

因此整體的使用方式如下圖所示：



透過元大 Smart API，可以自動將行情寫程文字檔，讓撰寫策略的程式取用，再透過程式呼叫外部指令的功能去呼叫執行檔，把下單、刪單、委託查詢...等等指令傳送回給元大 Smart API，元大 Smart API 再把相對應的指令回傳給原本的 API 接口，最終完成指令。

因此，本下單機事實上不止 R、Python 可以調用，任何具備讀取文字檔和呼叫外部指令的程式語言均可使用，是一個通用的交易框架。

## 4. 登入元大 Smart API 進行下單

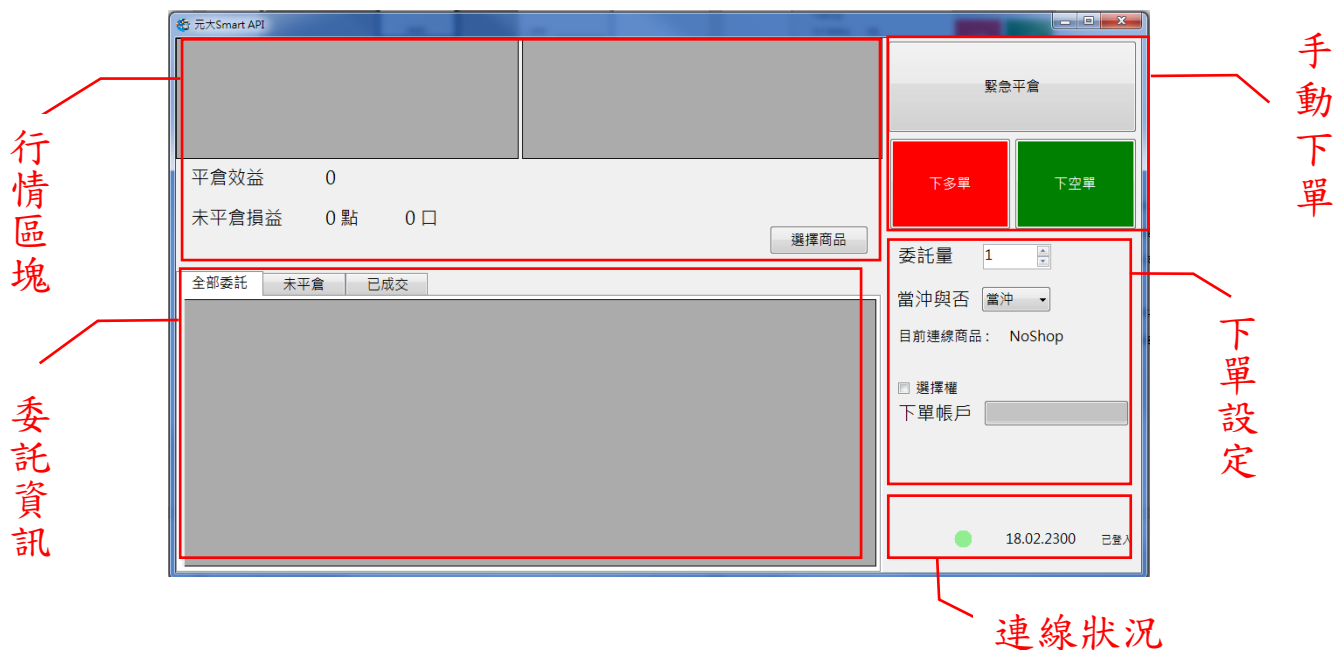
### 4-1 登入

首先打開桌面的元大 Smart API 程式，看到登入畫面：



請輸入身分證字號與密碼(與點金靈、EasyWin 密碼相同)，並勾選報價存檔，路徑就是之後的報價文字檔存檔的位置，以下以存檔路徑 `D:\data` 為例說明，並在閱讀完免責聲明後，勾選已詳閱免責聲明，並按登入，登入後畫面如下。

(注意!! 請先確認有開通行情 API 與下單 API 權限)



## 4-2 輸入商品代號



登入後，點選選擇商品按鈕，並於輸入框內輸入商品代碼(ex:TXFD8)，按下確定，即可於行情區塊中看到該商品當前的行情。若為開盤時間，可以先確認右下角的連線燈號有正常連線(綠燈是連線中;紅燈為斷線) 以及畫面上顯示上一筆成交價的行情有在跳動，則代表成功取得該商品的行情。

## 4-3 觀察行情寫入情況

保持著元大 Smart API 開啟的狀態之下，可以透過路徑 `D:\data\日期\商品代號` 找到行情文字檔，其中存有兩種格式的行情資料：

1. 日期\_Match.txt：代表逐筆成交資料
2. 日期\_UpDn5.txt：代表上下五檔報價資訊

可以用記事本或是 Notepad++等文件編輯軟體打開，看到行情不斷的更新就代表確實有寫入成功，行情格式的詳細說明請參考下一章的行情取用。

## 4-4 測試下單

點選畫面上的下多單或是下空單按鈕，會依據委託量立刻送出該檔商品的市價單，若要測試按鈕是否能運作，請先把帳戶內的保證金清空，否則會立即成交，會看到左下角的委託資訊多了幾筆，畫面如下：

The screenshot displays the 元大Smart API trading platform. The interface includes a top section for product selection (商品名稱, 當前價) showing TXFD8 at 10869. Below this is a section for open orders (平倉效益) showing 0 points. The main area features a table of orders with columns for order number, status, product code, direction, price, quantity, time, and branch code. Three failed orders are highlighted with a red box. On the right, there are buttons for '下多單' (Buy) and '下空單' (Sell), along with fields for '委託量' (Order Quantity) set to 1, '當沖與否' (Day Order) set to '非當沖' (Not Day Order), and '目前連線商品' (Currently Connected Product) set to TXFD8. At the bottom right, there is a '選擇權' (Options) checkbox and a '下單帳戶' (Order Account) field. The status bar at the bottom shows a green dot, the time 18.02.2300, and the user '已登入' (Logged In).

委託序號	狀態	商品代號	多空	價格	口數	時間	分公司代號
2018040900108754S4EO	委託失敗	TXFD8	市賣	0	0	110903	P00
2018040900108001S4EO	委託失敗	TXFD8	市買	0	0	110745	P00
2018040900107960T2EO	委託失敗	TXFD8	市買	0	0	110741	P00

其中包含了委託序號、狀態、商品代號...等等資訊，並且最右側的委託狀態欄位會告訴使用者更詳細的資訊，你可以從裡面看到委託失敗的原因，像是畫面中的委託失敗是因為帳戶內沒放保證金。至於未平倉的部位資訊會顯示在未平倉的頁籤中，當天的成交資訊則會顯示在已成交的頁籤中。

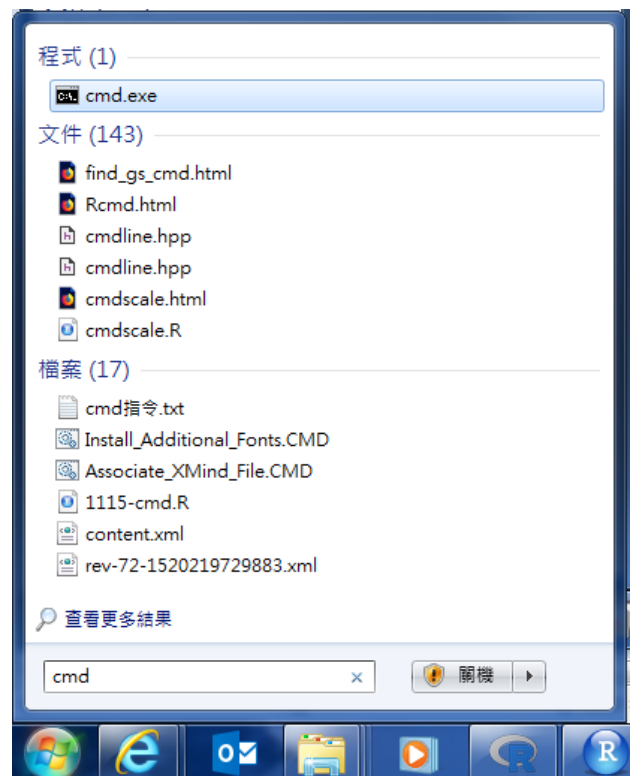
若是盤中行情很大，點擊緊急平倉按鈕，會將原有的部位市價平倉並同時取消其餘未成交的委託單。至於當要操作選擇權契約時，請把右方選擇權的框框打勾，由於選擇權不能下市價單的緣故，當選擇權的框框打勾時，上面的下單按鈕會消失。

## 5. 使用命令提示字元(cmd.exe)基本操作

### 5-1 Command line 簡易操作

以下介紹 Windows Command line 介面的基本操作，首先於開始功能表搜尋程式欄位輸入 cmd，即可看到上方的 cmd.exe 執行檔，點選並執行。

cd path	將工作路徑移到 path 資料下
cd..	將工作路徑退回上一層
tab	自動補齊該路徑下存在的檔名
檔名 a b	執行該檔並傳入 a、b 兩個參數
dir	查看當前目錄下的檔案與資料夾



以下說明以元大 Smart API 資料夾放在桌面為範例，依序輸入：

**cd Desktop\元大 Smart API** 切換到元大 Smart API 資料夾下

**dir** 查看路徑下檔案，確認 Order.exe、CancelALL.exe... 等等執行檔確實存在該路徑下

便可接續以下操作

## 5-2 行情取用 (tail.exe)

行情的取用需透過 tail.exe 執行檔，此執行檔是仿照 Linux shell 的 tail 指令製作而成，搭配傳入檔名的參數即可取用該檔案的最後的第 n 行，若傳入 1 則可以取得該檔案最後一行的數值(也就是最新的成交資料)，以下以檔案路徑放置在 D:\data 為例，輸入：

```
tail.exe -n1 D:\data\20180409\TXFD8\20180409_Match.txt
```

即可取得 2018/3/5 當天的最新一筆成交資料，執行畫面如下：

```
C:\Users\90813\Desktop\元大SmartAPI>tail.exe -n1 D:\data\20180409\TXFD8\20180409_Match.txt
TXFD8,11:28:51.212000,10879,1,108247,10898,10813
```

若想要知道最新一筆的上下 5 檔報價，也可以透過：

```
tail.exe -n1 D:\data\20180409\TXFD8\20180409_UpDn5.txt
```

取得上下五檔的報價資訊如下：

```
C:\Users\90813\Desktop\元大SmartAPI>tail.exe -n1 D:\data\20180409\TXFD8\20180409_UpDn5.txt
TXFD8,11:30:11.395000,10881,4,10880,38,10879,38,10878,52,10877,35,10882,29,10883,65,10884,64,10885,120,10886,39
```

回傳資料的格式說明如下

1. 日期\_Match.txt，表示成交價量揭示訊息。

資料欄位：

---

商品代號,時間,成交價,成交量,總量,最高價,最低價

---

2. 日期\_UpDn5.txt，表示最佳上下五檔價量資訊。

資料欄位：

---

商品代號,時間,下一檔價格,下一檔數量,下兩檔價格,下兩檔數量,下三檔價格,下三檔數量,下四檔價格,下四檔數量,下五檔價格,下五檔數量,上一檔價格,上一檔數量,上兩檔價格,上兩檔數量,上三檔價格,上三檔數量,上四檔價格,上四檔數量,上五檔價格,上五檔數量

---



### 5-3 切換商品 (ChangeProdid.exe)

使用 ChangeProdid.exe 執行檔切換連線中的商品，傳入商品代號做為參數，即可改變連線中的商品，實際執行畫面如下：

```
C:\Users\90813\Desktop\元大Smart API>ChangeProdid.exe TXFC8  
ChangeSucess
```

### 5-4 下限價單 (Order.exe)

使用 Order.exe 執行檔，需要帶入的參數如下

#### 下單程式 Order.exe

1. 商品：期權交易商品，例如2018年第四個月的大台指期TXFD8。
2. 買賣：B買、S賣
3. 價格：委託價格，市價單則任意值即可
4. 量：1至100，100為期交所上限
5. 價格類別：LMT限價單、MKT市價單
6. 委託類別：ROD當日有效、IOC立即成交否則不成交、FOK全部成交否則不成交
7. 是否當沖：1非當沖、0當沖

以下買進1口2018年4月份的10500的限價非當沖單為例，指令如下：

```
Order.exe TXFD8 B 10500 1 LMT ROD 1
```

執行畫面如下：

```
C:\Users\90813\Desktop\元大Smart API>Order.exe TXFD8 B 10500 1 LMT ROD 1  
2018030500218898T2E0
```

回傳的數值為委託序號

需要特別注意的是，若要買賣的商品為選擇權，需要在商品的參數在前面加上 `option,` 的前綴，以買進 1 口 2018 年 4 月份到期履約價為 10700 的買權，限價 30 的非當沖單為例，指令如下：

```
Order.exe option, TXO10700D8 B 30 1 LMT IOC 1
```

## 5-5 下市價單 (Order.exe)

下市價單的方式與限價單一樣，只差在價格可以用任意數值(習慣上是給 0)，並且價格類別改為 MKT、委託類別改為 IOC，以市價買進 1 口 2018 年 4 月份的非當沖單為例，指令如下：

```
Order.exe TXFD8 B 0 1 MKT IOC 1
```

執行畫面如下：

```
C:\Users\90813\Desktop\元大Smart API>Order.exe TXFD8 B 0 1 MKT IOC 1
2018030500219401S8E0
```

與限價單一樣，若要買賣的商品為選擇權，需要在商品的參數在前面加上 `option,` 的前綴。

## 5-6 刪單 (Order.exe)

刪單一樣使用 Order.exe 檔執行，傳入的參數為 `Delete 委託序號`，以下為執行畫面：

```
C:\Users\90813\Desktop\元大Smart API>Order.exe Delete 2018030500218898T2E0
```

所傳入的該筆委託單必須還掛在委託簿上，否則會回傳錯誤。

## 6. 使用命令提示字元(cmd.exe)進階操作

### 6-1 查詢委託回報 (GetAccount.exe)

透過 GetAccount.exe 執行檔即可取得委託的歷史紀錄，有兩種使用方式，參數可以輸入委託序號以查詢單筆委託單，也可以輸入 ALL 以查詢當日所有委託記錄(僅限於當下連線的商品委託記錄)，指令如下：

1. GetAccount.exe 2018030500218898T2EO
2. GetAccount.exe ALL

實際執行畫面如下

- 單筆委託查詢

```
C:\Users\90813\Desktop\元大Smart API>GetAccount.exe 2018030500218898T2EO
2018030500218898T2EO,委託失敗,TXFD8,限買,10500,1,164510,P00,9851503,,,,,,,,,2 261
中檔:該客戶無保證金-委託失敗!
```

- 全部委託查詢

```
C:\Users\90813\Desktop\元大Smart API>GetAccount.exe ALL
2018030500219401S8EO,委託失敗,TXFD8,市買,0,0,164956,P00,9851503,,,,,,,,,2 261中檔
:該客戶無保證金-委託失敗!
2018030500218898T2EO,委託失敗,TXFD8,限買,10500,1,164510,P00,9851503,,,,,,,,,2 261
中檔:該客戶無保證金-委託失敗!
2018030500216572T0EO,委託失敗,TXFD8,限買,10500,1,161414,P00,9851503,,,,,,,,,2 261
中檔:該客戶無保證金-委託失敗!
2018030500130669S6EO,委託失敗,TXFD8,市賣,0,0,104638,P00,9851503,,,,,,,,,2 261中檔
:該客戶無保證金-委託失敗!
2018030500130435T2EO,委託失敗,TXFD8,市買,0,0,104619,P00,9851503,,,,,,,,,2 274中檔
:客戶未申請當沖-委託失敗!
```

回傳的欄位如下：

---

委託序號,狀態,商品代號,多空,價格,口數,時間,分公司代號,交易帳號,交易所,委託書號,異動變更前量,異動變更後量,成交序號,子帳號,營業員編號,委託狀態

---

## 6-2 查詢未平倉 (OnOpenInterest.exe)

透過 OnOpenInterest.exe 查詢目前連線商品的未平倉資訊，不需傳入參數，直接呼叫即可，執行畫面如下：

```
C:\Users\90813\Desktop\元大Smart API>OnOpenInterest.exe
```

## 6-3 查詢帳戶資訊 (FutureRights.exe)

透過 FutureRights.exe 查詢目前的帳務資訊，會回傳包含國內、國外以及各種損益的帳務資訊，實際執行畫面如下：

```
C:\Users\90813\Desktop\元大Smart API>FutureRights.exe
0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,
0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,17:11:22,0.00,0.00,0.00,0.00,0.00,0
.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00
```

回傳的欄位如下：

---

權益數,全額原始保證金,可運用保證金,權益比率,當沖風險指標,風險指標,前日餘額,存/提,手續費,期交稅,本日期貨平倉損益淨額+到期履約損益,權利金收入與支出,本日餘額,債券實物交割款,有價證券抵繳總額,昨日未平倉損益,未沖銷期貨浮動損益,權益數 2,原始保證金,維持保證金,可運用保證金 2,可動用(出金)保證金(含抵委),債券實物不足交割款,期貨多空減收保證金,風險更新時間,未沖銷買方選擇權市值+未沖銷賣方選擇權市值,權益總值,追繳保證金,委託保證金,委託權利金,委託抵繳保證金,債券待交割保證金,加收保證金,Y\_TOTAL\_VALUE,DSP\_EQUITY,DSP\_IM,DSP\_RISK\_RATE,DSP\_FLOATMARGIN, DSP\_FLOATPREMIUM

---

## 6-4 刪除所有掛單 (CancelALL.exe)

透過 CancellALL.exe 刪除目前連線商品的還存在委託簿的所有掛單(僅連線商品)，不需要帶入其他參數，執行的畫面如下：

```
C:\Users\90813\Desktop\元大Smart API>CancelALL.exe
AllCancelSuccess
```

## 6-5 取得所有未成交之委託單 (GetUnfinished.exe)

透過 GetUnfinished.exe 取得目前連線商品在委託簿上還未成交之委託單(僅連線商品)，若委託簿上已無未成交之委託單，回傳 over，執行畫面如下：

```
C:\Users\90813\Desktop\元大Smart API>GetUnfinished.exe  
over
```

## 6-6 緊急平倉 (MayDay.exe)

透過 MayDay.exe 依序執行:

1. 取消所有的委託單
2. 未平倉的部位以市價單強制平倉

(請注意，以上指令中指令、MayDay.exe 指令、GetAccount.exe ALL 指令、OnOpenInterest.exe 指令、CancelALL.exe 指令以及 GetUnfinished.exe 指令，這些指令只對於正在連線中的商品作用)

## 7. 使用 R 串接下單&行情

### 7-1 R 呼叫外部指令

在 R 中可以透過內建函數 `system()` 或是 `system2()` 去呼叫外部指令，兩者的使用方式略有不同，以下範例我們使用 `system2()` 這個函數去呼叫外部指令，例如要在 command line 介面呼叫開啟記事本的指令，R 的語法如下：

```
system2("notepad.exe", wait = F, invisible = F)
```

其中 `wait=F` 參數是讓進程直接跳離呼叫出來的程式，而 `invisible=F` 則是讓視窗顯示出來，執行後可以看到記事本自動打開了

`system2` 的重要參數說明如下：

- 1：執行檔(路徑+檔名) [必要]
- 2：args 代表呼叫執行檔時傳入的參數 [非必要]
- 3：stdout 代表是否要回傳執行後的結果 [非必要]

因此使用時的語法大致為 `system2("路徑+檔名", args="傳入參數", stdout=T)`，上面呼叫記事本的範例之所以可以不加路徑是因為 `Notepad.exe` 所在的資料夾 `C:\Windows\system32` 已存在系統的環境變數 `PATH` 中，因此可以直接呼叫，下面的範例為了保險起見，我們連同檔案的路徑也一併傳入指令中。

另外有別於 Windows 的作業系統都是用反斜線(backslash; `"\"`)當作路徑區隔，因為 R 的設計是從 Unix/Linux 作業系統的體系出來的，因此他的資料夾路徑符號是用斜線(slash; `"/"`)當作路徑區隔，若使用者複製 Windows 下的檔案路徑貼到 R 中，務必要注意把反斜線(`"\"`)替換成斜線(`"/"`)。

## 7-2 R 進行基礎操作

先把元大 Smart API 資料夾下所包含的執行檔路徑先儲存在一個字串變數中，以供後續指令調用，以資料夾放置在桌面為範例，指令為：

```
ExecPath = "C:/Users/90813/Desktop/元大 Smart API/"
```

另外，使用 `paste0()` 函數可以將多個字串直接連接在一起，`paste()` 函數則是會將多個字串以空格的方式連接在一起。以下分別示範在 R 中呼叫下單執行檔的操作方式。

### 7-2-1 行情取用

語法：

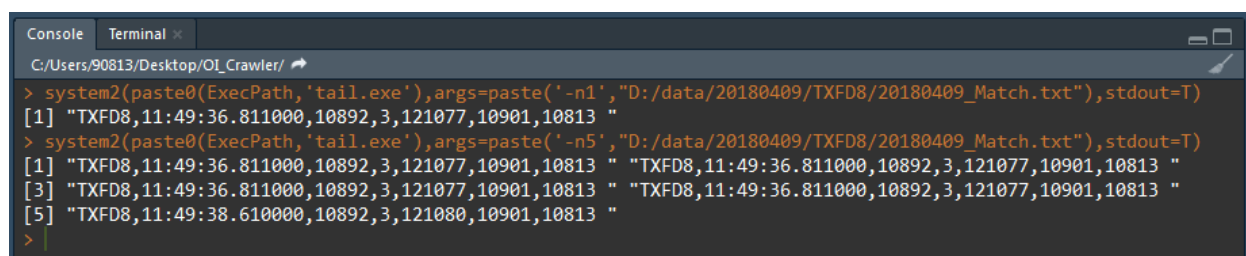
```
#取得最後 1 行
```

```
system2(paste0(ExecPath,'tail.exe'),args=paste('-n1',"D:/data/20180409/TXFD8/20180409_Match.txt"),stdout=T)
```

```
#取得最後 5 行
```

```
system2(paste0(ExecPath,'tail.exe'),args=paste('-n5',"D:/data/20180409/TXFD8/20180409_Match.txt"),stdout=T)
```

實際執行畫面：



```
Console Terminal x
C:/Users/90813/Desktop/OI_Crawler/
> system2(paste0(ExecPath,'tail.exe'),args=paste('-n1',"D:/data/20180409/TXFD8/20180409_Match.txt"),stdout=T)
[1] "TXFD8,11:49:36.811000,10892,3,121077,10901,10813 "
> system2(paste0(ExecPath,'tail.exe'),args=paste('-n5',"D:/data/20180409/TXFD8/20180409_Match.txt"),stdout=T)
[1] "TXFD8,11:49:36.811000,10892,3,121077,10901,10813 " "TXFD8,11:49:36.811000,10892,3,121077,10901,10813 "
[3] "TXFD8,11:49:36.811000,10892,3,121077,10901,10813 " "TXFD8,11:49:36.811000,10892,3,121077,10901,10813 "
[5] "TXFD8,11:49:38.610000,10892,3,121080,10901,10813 "
>
```

## 7-2-2 下單

語法：

#限價單

```
system2(paste0(ExecPath,'Order.exe'),args=paste("TXFD8","B",'11100',1,'LMT',"ROD",1),stdout = T)
```

#市價單

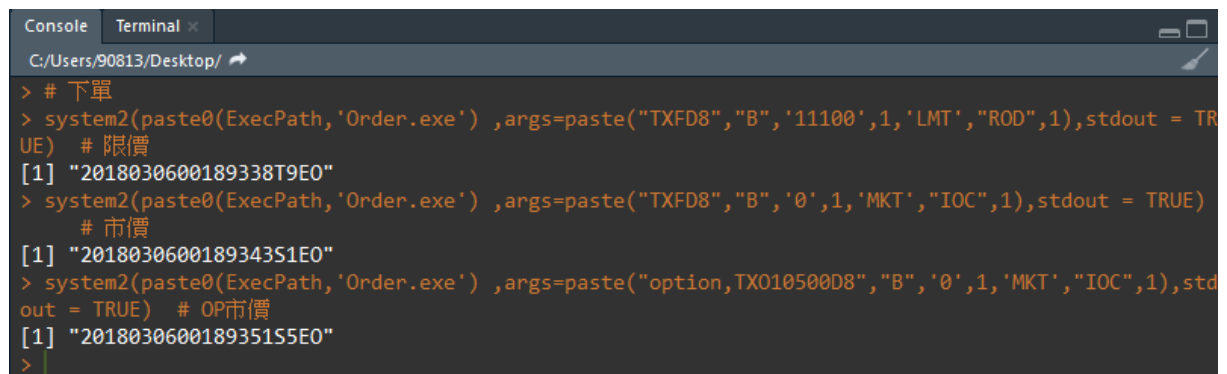
```
system2(paste0(ExecPath,'Order.exe'),args=paste("TXFD8","B",'0',1,'MKT',"IOC",1),stdout = T)
```

#選擇權(市價單)

```
system2(paste0(ExecPath,'Order.exe'),args=paste("option,TXO10500D8","B",'0',1,'MKT',"IOC",1),stdou
```

```
t = TRUE) # OP 市價
```

實際執行畫面：



```
Console Terminal x
C:/Users/90813/Desktop/
> # 下單
> system2(paste0(ExecPath,'Order.exe'),args=paste("TXFD8","B",'11100',1,'LMT',"ROD",1),stdout = TRUE) # 限價
[1] "2018030600189338T9E0"
> system2(paste0(ExecPath,'Order.exe'),args=paste("TXFD8","B",'0',1,'MKT',"IOC",1),stdout = TRUE)
# 市價
[1] "2018030600189343S1E0"
> system2(paste0(ExecPath,'Order.exe'),args=paste("option,TXO10500D8","B",'0',1,'MKT',"IOC",1),stdout = TRUE) # OP市價
[1] "2018030600189351S5E0"
>
```

## 7-2-3 切換商品

語法：`system2(paste0(ExecPath,"ChangeProdid.exe"),args="TXFC8")`

實際執行畫面：



```
Console Terminal x
C:/Users/90813/Desktop/ ↗
> # 切换商品
> system2(paste0(ExecPath,"ChangeProdid.exe"),args="TXFC8")
ChangeSucess
> |
```

#### 7-2-4 刪單

語法：`system2(paste0(ExecPath,'Order.exe'),args=paste("Delete",OrderNo),stdout = TRUE)` #刪單

實際執行畫面：

```
Console Terminal x
C:/Users/90813/Desktop/ ↗
> OrderNo = "2018030600183490T9E0"
> system2(paste0(ExecPath,'Order.exe'),args=paste("Delete",OrderNo),stdout = TRUE) # 刪單
[1] "|90000005|刪改必須提供委託書號!"
>
>
```

變數 OrderNo 需要透過下單時回傳的委託序號來儲存，在刪單的時候才可以調用，畫面中的委託序號是委託失敗的單子，已不在委託簿中，因此回傳刪改必須提供委託書號!，否則回傳刪單成功。

### 7-3 R 進行進階操作

#### 7-3-1 查詢委託回報

查詢委託回報有兩種用法，一種是依據傳入的委託序號查詢單筆，另一種是傳入 ALL 參數以查詢當日所有委託記錄。

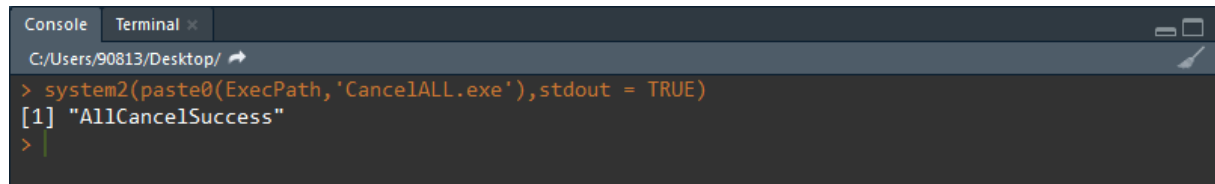
語法：`system2(paste0(ExecPath,'GetAccount.exe'),args=paste(OrderNo),stdout = TRUE)` #單筆

`system2(paste0(ExecPath,'GetAccount.exe'),args=paste("ALL"),stdout = TRUE)` #全部

實際執行畫面：



實際執行畫面：

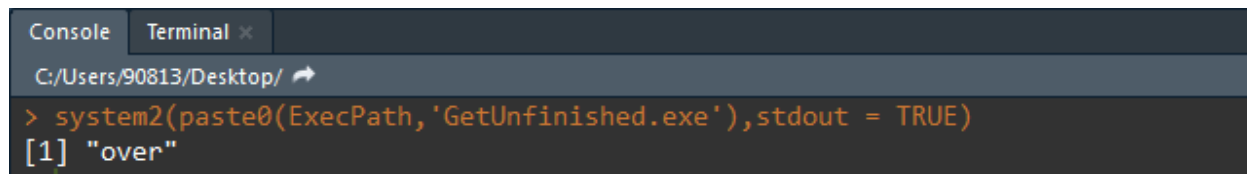


```
Console Terminal x
C:/Users/90813/Desktop/
> system2(paste0(ExecPath,'CancelALL.exe'),stdout = TRUE)
[1] "AllCancelSuccess"
>
```

### 7-3-5 取得所有未成交之委託單

語法：`system2(paste0(ExecPath,'GetUnfinished.exe'),stdout = TRUE)`

實際執行畫面：



```
Console Terminal x
C:/Users/90813/Desktop/
> system2(paste0(ExecPath,'GetUnfinished.exe'),stdout = TRUE)
[1] "over"
```

### 7-3-6 緊急平倉

語法：`system2(paste0(ExecPath,'MayDay.exe'),stdout = TRUE)`

實際執行畫面：

注意 MayDay.exe 指令由於涉及一連串的平倉動作，並且是強制平倉，因此只能在開盤使用，若收盤時用會進入持續等待回應的狀態。

## 8. 使用 Python 串接下單&行情

### 8-1 Python 呼叫外部指令

在 Python 中是透過 subprocess 套件中 `check_output()` 函數去呼叫外部指令，因此在腳本一開始需要先引用 subprocess 套件，語法為 `import subprocess as sp`，引用過後在後續的程式中就可以使用 `sp` 來當作 subprocess 的代名詞，以下示範若要在 command line 介面呼叫開啟記事本的指令，Python 的語法如下：

```
sp.check_output(["notepad.exe"])
```

執行後可以看到記事本自動打開，其中 `check_output()` 接收的參數是一個陣列(List)，函數會將陣列中的第一個元素做為檔案名稱，其餘陣列中的元素則做為傳入的參數使用，上面呼叫記事本的範例之所以可以不加路徑是因為 Notepad.exe 所在的資料夾 C:\Windows\system32 已存在系統的環境變數 PATH 中，因此可以直接呼叫，下面的範例為了保險起見，我們連同檔案的路徑也一併傳入指令中。

使用時的語法大致為 `sp.checkout( ["路徑+檔名", 參數 1, 參數 2..., 參數 n] )`，其中檔案的路徑有別於 Windows 的作業系統都是用反斜線(backslash; "\\")當作路徑區隔，Python 的設計是從 Unix/Linux 作業系統的體系出來的，因此他的資料夾路徑符號是用斜線(slash; "/" )當作路徑區隔，若使用者複製 Windows 下的檔案路徑貼到 Python 中，務必要注意把反斜線("\\")替換成斜線("/")。

由於使用的是 Python 3，因此若有回傳值需要透過 `decode()` 函數使用 Big 5 的編碼方式將回傳的編碼狀態(bytes)轉為 Unicode，再透過 `strip()` 函數去除結尾的 `\r\n` 字符，完整語法如下：

```
sp.check_output([路徑+檔名, 參數 1, 參數 2,..., 參數 n]).decode("big5").strip('\r\n')
```

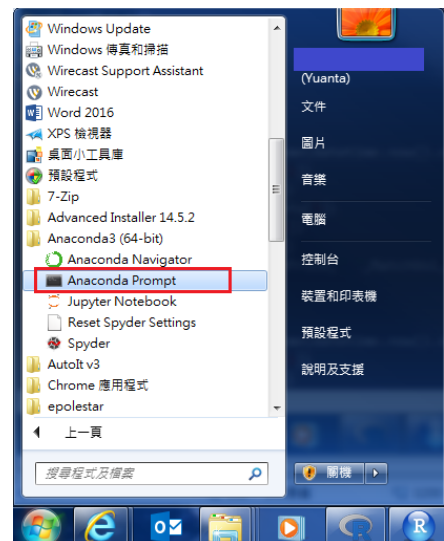
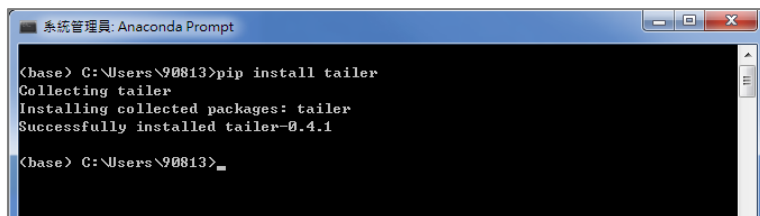
## 8-2 Python 進行基礎操作

先把元大 Smart API 資料夾下所包含的執行檔路徑先儲存在一個字串變數中，以供後續指令調用，以資料夾放置在桌面為範例，指令為：

```
ExecPath = "C:/Users/90813/Desktop/元大 Smart API/"
```

### 8-2-1 行情取用

Python 的行情取用方式和 R 略為不同，執行起來效率比 R 更好，原因是 Python 不僅可以透過 `tailer` 模塊中的 `tail()` 函數去取得文件的最後一行，更可以使用 `follow()` 函數去監控檔案是否有更新，只有在更新的時候去取用新的資料，就不會在檔案沒有更新時不斷去消耗資源讀資料，首先需要先安裝 `tailer` 模塊，使用原生 Python 3 的如果已有安裝 `pip` 這個模塊管理的模塊，可以用 `pip install tailer` 完成安裝，本範例使用 Anaconda 中的 `spyder` 開發環境，因此要開啟 `Anaconda Prompt` 的指令介面，用開始功能表的 `Anaconda` 資料夾中找到 `Anaconda Prompt`，開啟後一樣輸入 `pip install tailer` 即可安裝完畢，安裝完成畫面如下。



以下描述如何使用 tailer 模塊:

- 引用模塊

語法: `import tailer`

- 取得最後 1 筆(tail 函數)

語法: `tailer.tail(open('D:/data/20180409/TXFD8/20180409_Match.txt'),3)[-2]`

實際執行畫面:

```
In [258]: tailer.tail(open('D:/data/20180409/TXFD8/20180409_Match.txt'),3)[-2]
Out[258]: 'TXFD8,11:58:37.692000,10892,1,122745,10901,10813 '
```

- 監控行情檔案的變動(follow 函數)

語法: `for line in tailer.follow(open('D:/data/20180409/TXFD8/20180409_Match.txt')):`

`print(line)`

```
In [260]: for line in tailer.follow(open('D:/data/20180409/
TXFD8/20180409_Match.txt')):
...:     print(line)
...:
...:
TXFD8,12:00:45.195000,10896,1,123702,10901,10813
TXFD8,12:00:45.195000,10896,1,123702,10901,10813
TXFD8,12:00:45.195000,10896,1,123702,10901,10813
TXFD8,12:00:45.195000,10896,1,123702,10901,10813
TXFD8,12:00:45.195000,10896,1,123702,10901,10813
TXFD8,12:00:45.195000,10896,1,123702,10901,10813
TXFD8,12:00:45.195000,10896,1,123702,10901,10813
TXFD8,12:00:45.195000,10896,1,123702,10901,10813
TXFD8,12:00:45.195000,10896,1,123702,10901,10813
TXFD8,12:00:45.195000,10896,1,123702,10901,10813
TXFD8,12:00:45.195000,10896,1,123702,10901,10813
TXFD8,12:00:48.615000,10896,1,123703,10901,10813
TXFD8,12:00:48.615000,10896,1,123703,10901,10813
TXFD8,12:00:48.615000,10896,1,123703,10901,10813
TXFD8,12:00:48.906000,10896,2,123705,10901,10813
```

## 8-2-2 下單

語法：

#市價單

```
sp.check_output([ExecPath+"Order.exe","TXFD8","S","0","1","MKT","IOC","1"]).decode("big5").strip('\r\n')
```

#限價單

```
sp.check_output([ExecPath+"Order.exe","TXFD8","S","11100","1","LMT","IOC","1"]).decode("big5").strip('\r\n')
```

實際執行畫面：

```
In [88]: sp.check_output([ExecPath
+"Order.exe","TXFD8","S","0","1","MKT","IOC","1"]).decode("big5").strip('\r\n')
Out[88]: '2018030700142887T8EO'

In [89]: sp.check_output([ExecPath
+"Order.exe","TXFD8","S","11100","1","LMT","IOC","1"]).decode("big5").strip('\r\n')
Out[89]: '2018030700142902T3EO'
```

## 8-2-3 切換商品

語法：`sp.check_output([ExecPath+"ChangeProdid.exe","TXFC8"]).decode("big5")`

實際執行畫面：

```
In [100]: sp.check_output([ExecPath+"ChangeProdid.exe","TXFC8"]).decode("big5")
Out[100]: 'ChangeSucess'
```

## 8-2-4 刪單

語法：

```
sp.check_output([ExecPath+"Order.exe","Delete",'2018030700143973T3EO']).decode("big5")
```

實際執行畫面：

```
In [108]: OrderNo = "2018030700143973T3EO"

In [109]: sp.check_output([ExecPath+"Order.exe","Delete",OrderNo]).decode("big5")
Out[109]: '|90000005|刪改必須提供委託書號!\r\n'
```

變數 OrderNo 需要透過下單時回傳的委託序號來儲存，在刪單的時候才可以調用，畫面中的委託序號是委託失敗的單子，已不在委託簿中，因此回傳刪改必須提供委託書號!，否則回傳刪單成功。

## 8-3 Python 進行進階操作

### 8-3-1 查詢委託回報

查詢委託回報有兩種用法，一種是依據傳入的委託序號查詢單筆，另一種是傳入 ALL 參數以查詢當日所有委託記錄。

語法：

```
sp.check_output([ExecPath+"GetAccount.exe",OrderNo]).decode("big5").strip('\r\n') #單筆
```

```
sp.check_output([ExecPath+"GetAccount.exe","ALL"]).decode("big5").strip('\r\n') #全部
```

實際執行畫面：

```
In [113]: sp.check_output([ExecPath+"GetAccount.exe",OrderNo]).decode("big5").strip('\r\n') #單筆
Out[113]: '2018030700143973T3EO,委託失敗,TXFD8,市賣,0,0,105655,P00,9851503,,,,,,,,,2 261中樑:該客戶無
保證金-委託失敗!'

In [114]: sp.check_output([ExecPath+"GetAccount.exe","ALL"]).decode("big5").strip('\r\n') #全部
Out[114]: '2018030700143973T3EO,委託失敗,TXFD8,市賣,0,0,105655,P00,9851503,,,,,,,,,2 261中樑:該客戶無
保證金-委託失敗!\n2018030700142902T3EO,委託失敗,TXFD8,限賣,11100,1,105345,P00,9851503,,,,,,,,,2 261中
樑:該客戶無保證金-委託失敗!\n2018030700142887T8EO,委託失敗,TXFD8,市賣,0,0,105342,P00,9851503,,,,,,,,,2
261中樑:該客戶無保證金-委託失敗!\n\r\nover'
```

### 8-3-2 查詢未平倉

語法：`sp.check_output([ExecPath+"OnOpenInterest.exe"]).decode("big5").strip('\r\n')`

實際執行畫面：

```
In [115]: sp.check_output([ExecPath+"OnOpenInterest.exe"]).decode("big5").strip('\r\n')
Out[115]: ''
```



### 8-3-3 查詢帳戶資訊

語法：`sp.check_output([ExecPath+"FutureRights.exe"]).decode("big5").strip('\r\n')`

實際執行畫面：

```
In [117]: sp.check_output([ExecPath+"FutureRights.exe"]).decode("big5").strip('\r\n')  
Out[117]:  
'0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.0  
0.00,0.00,0.00,0.00,0.00,11:22:56,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.00,0.0  
0'
```

### 8-3-4 刪除所有掛單

語法：`sp.check_output([ExecPath+"CancelALL.exe"]).decode("big5").strip('\r\n')`

實際執行畫面：

```
In [118]: sp.check_output([ExecPath+"CancelALL.exe"]).decode("big5").strip('\r\n')
Out[118]: 'AllCancelSuccess'
```

### 8-3-5 取得所有未成交之委託單

語法：`sp.check_output([ExecPath+"GetUnfinished.exe"]).decode("big5").strip('\r\n')`

實際執行畫面：

```
In [119]: sp.check_output([ExecPath+"GetUnfinished.exe"]).decode("big5").strip('\r\n')
Out[119]: 'over'
```

### 8-3-6 緊急平倉

語法：`sp.check_output([ExecPath+"MayDay.exe"]).decode("big5").strip('\r\n')`

實際執行畫面：

```
In [10]: sp.check_output([ExecPath+"MayDay.exe"]).decode("big5").strip('\r\n')
Out[10]: 'MaydaySuccess'
```

## 9. 附錄 - 商品代碼表

有關可以使用的商品代碼，請參考元大國內行情 API 的使用說明文件 (<http://easywin.yuanta futures.com.tw/api/download.html>)，文件中 2-3 的註冊商品代碼