策略王COM元件使用說明

-國內報價-

# 函式範例

## SKQuoteLib (國內報價)

*函式*

1. 連線

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| 1. **報價連線** | [SKQuoteLib\_EnterMonitorLONG](#_SKQuoteLib_EnterMonitorLONG) | LONG index 對應 |
| 1. **報價斷線** | [SKQuoteLib\_LeaveMonitor](#_SKQuoteLib_LeaveMonitor) |  |
| 1. **取得報價連線狀態** | [SKQuoteLib\_IsConnected](#_SKQuoteLib_IsConnected) |  |
| 1. **取得連線狀態** | [SKQuoteLib\_GetQuoteStatus](#_SKQuoteLib_GetQuoteStatus) | (僅限連線成功後使用) |
| 1. **主機時間查詢** | [SKQuoteLib\_RequestServerTime](#_SKQuoteLib_RequestServerTime) |  |

1. 大盤資訊

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| **取得大盤買賣** | [SKQuoteLib\_GetMarketBuySellUpDown](#_SKQuoteLib_GetMarketBuySellUpDown) | 不支援盤中零股。 |

1. 商品清單

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| **國內商品清單查詢** | [SKQuoteLib\_RequestStockList](#_SKQuoteLib_RequestStockList) | 盤中零股上市5、盤中零股上櫃 6  客製化期貨-9、客製化選擇權-10 |

1. 個股資訊

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| **取得商品物件** | [SKQuoteLib\_GetStockByNoLONG](#_SKQuoteLib_GetStockByNoLONG) | LONG index 對應  不適用盤中零股 |
| **取得商品物件** | [SKQuoteLib\_GetStockByMarketAndNo](#_SKQuoteLib_GetStockByMarketAndNo_1) | 適用盤中零股 |

1. 即時報價

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| 1. **訂閱報價** | [SKQuoteLib\_RequestStocks](#_SKQuoteLib_RequestStocks) | 不支援盤中零股。 |
| 1. **取得商品物件** | [SKQuoteLib\_GetStockByIndexLONG](#_SKQuoteLib_GetStockByIndexLONG) | LONG index 對應 |
| 1. **訂閱指定市場別及指定商品即時報價** | [SKQuoteLib\_RequestStocksWithMarketNo](#_SKQuoteLib_RequestStocksWithMarketN) | 適用盤中零股 |
| 1. **取消訂閱報價** | [SKQuoteLib\_CancelRequestStocks](#_SKQuoteLib_CancelRequestStocks_1) | 不支援盤中零股。 |
| 1. **取得證券市價特殊值** | [SKQuoteLib\_GetMarketPriceTS](#_SKQuoteLib_GetMarketPriceTS) |  |

1. 五檔&成交明細

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| 1. **訂閱Ticks及五檔** | [SKQuoteLib\_RequestTicks](#_SKQuoteLib_RequestTicks) | 不支援盤中零股。  [OnNotifyTicksLONG](#_OnNotifyTicksLONG_1)、[OnNotifyHistoryTicksLONG](#_OnNotifyHistoryTicksLONG)、[OnNotifyBest5LONG](#_OnNotifyBest5LONG) |
| 1. **取得Tick** | [SKQuoteLib\_GetTickLONG](#_SKQuoteLib_GetTickLONG) | LONG index 對應 |
| 1. **取得五檔** | [SKQuoteLib\_GetBest5LONG](#_SKQuoteLib_GetBest5LONG) | LONG index 對應 |
| 1. **取得即時成交明細Ticks&Best5** | [SKQuoteLib\_RequestTicksWithMarketNo](#_SKQuoteLib_RequestTicksWithMarketNo) | 適用盤中零股 |
| 1. **訂閱即時成交明細** | [SKQuoteLib\_RequestLiveTick](#_SKQuoteLib_RequestLiveTick) | 包含訂閱，僅即時成交明細 |
| 1. **取消訂閱Tick及五檔** | [SKQuoteLib\_CancelRequestTicks](#_SKQuoteLib_CancelRequestTicks) | 不支援盤中零股。 |

1. 技術分析

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| **MACD** | |  | | --- | | 訂閱[SKQuoteLib\_RequestMACD](#_SKQuoteLib_RequestMACD) | | 取得[SKQuoteLib\_GetMACDLONG](#_SKQuoteLib_GetMACDLONG) | | 目前只提供證券市場，不含盤中零股  LONG index 對應  不支援盤中零股。 |
| **布林通道** | |  | | --- | | 訂閱[SKQuoteLib\_RequestBoolTunel](#_SKQuoteLib_RequestBoolTunel) | | 取得[SKQuoteLib\_GetBoolTunelLONG](#_SKQuoteLib_GetBoolTunelLONG) | | 目前只提供證券市場，不含盤中零股  LONG index 對應  不支援盤中零股。 |
| **K線查詢** | |  | | --- | | **指定區間**[SKQuoteLib\_RequestKLineAMByDate](#_SKQuoteLib_RequestKLineAMByDate) | | (新) [SKQuoteLib\_RequestKLineAM](#_SKQuoteLib_RequestKLineAM) | | [SKQuoteLib\_RequestKLine](#_SKQuoteLib_RequestKLine_1) | | 不支援盤中零股。  不支援盤中零股。  國內期權可選盤別 |

1. 風險計算(選擇權)

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| **風險參數Delta** | [SKQuoteLib\_Delta](#_SKQuoteLib_Delta) |  |
| **風險參數Gamma** | [SKQuoteLib\_Gamma](#_SKQuoteLib_Gamma) |  |
| **風險參數Vega** | [SKQuoteLib\_Vega](#_SKQuoteLib_Vega) |  |
| **風險參數Theta** | [SKQuoteLib\_Theta](#_SKQuoteLib_Theta) |  |
| **風險參數Rho** | [SKQuoteLib\_Rho](#_SKQuoteLib_Rho) |  |

1. 交易資訊(期選)

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| **訂閱期貨商品資訊** | [SKQuoteLib\_RequestFutureTradeInfo](#_SKQuoteLib_RequestFutureTradeInfo) |  |
| **選擇權資訊** | [SKQuoteLib\_GetStrikePrices](#_SKQuoteLib_GetStrikePrices) |  |

*事件*

1. 連線

|  |  |  |
| --- | --- | --- |
| **功能** | **事件名稱** | **備註** |
| **連線狀態** | [OnConnection](#_OnConnection) | 避免在此處直接進行RequestStocks and RequestTicks 等報價訂閱 |
| **主機時間** | [OnNotifyServerTime](#_OnNotifyServerTime) |  |

1. 大盤資訊

|  |  |  |
| --- | --- | --- |
| **功能** | **事件名稱** | **備註** |
| **大盤張筆** | [OnNotifyMarketTot](#_OnNotifyMarketTot) |  |
| **大盤買賣** | [OnNotifyMarketBuySell](#_OnNotifyMarketBuySell) |  |
| **大盤漲跌家數** | |  | | --- | | [OnNotifyMarketHighLowNoWarrant](#_OnNotifyMarketHighLowNoWarrant) | | [OnNotifyMarketHighLow](#_OnNotifyMarketHighLow) | | (同時回傳含權證及不含權證)  (包含權證) |

1. 商品清單

|  |  |  |
| --- | --- | --- |
| **功能** | **事件名稱** | **備註** |
| **國內市場商品清單(含類別代碼)** | |  | | --- | | [OnNotifyCommodityListWithTypeNo](#_OnNotifyCommodityListWithTypeNo) | | [OnNotifyStockList](#_OnNotifyStockList) | | (上市、上櫃、期貨、選擇權、  興櫃) |

1. 即時報價

|  |  |  |
| --- | --- | --- |
| **功能** | **事件名稱** | **備註** |
| **報價更新通知** | [OnNotifyQuoteLONG](#_OnNotifyQuoteLONG) | LONG index 對應 |
| **整零價差即時行情** | [OnNotifyOddLotSpreadDeal](#_OnNotifyOddLotSpreadDeal) |  |

1. 五檔&成交明細

|  |  |  |
| --- | --- | --- |
| **功能** | **事件名稱** | **備註** |
| **即時五檔** | |  | | --- | | [OnNotifyBest5LONG](#_OnNotifyBest5LONG) | | [OnNotifyBest5Emerging](#_OnNotifyBest5Emerging) **※**此事件索取的興櫃商品五檔單位為”股” | | LONG index 對應 |
| **即時Tick** | [OnNotifyTicksLONG](#_OnNotifyTicksLONG_1) | LONG index 對應 |
| **回補Tick** | [OnNotifyHistoryTicksLONG](#_OnNotifyHistoryTicksLONG) | LONG index 對應 |

1. 技術分析

|  |  |  |
| --- | --- | --- |
| **功能** | **事件名稱** | **備註** |
| **MACD** | [OnNotifyMACDLONG](#_OnNotifyMACDLONG) | LONG index 對應 |
| **布林通道** | [OnNotifyBoolTunelLONG](#_OnNotifyBoolTunelLONG_1) | LONG index 對應 |
| **K線資料** | [OnNotifyKLineData](#_OnNotifyKLineData_1) |  |
| **歷史KLine回補完成** | [OnKLineComplete](#_OnKLineComplete) |  |

1. 交易資訊(期選)

|  |  |  |
| --- | --- | --- |
| **功能** | **事件名稱** | **備註** |
| **期貨商品交易資訊** | [OnNotifyFutureTradeInfoLONG](#_OnNotifyFutureTradeInfoLONG) | LONG index 對應 |
| **選擇權資訊** | [OnNotifyStrikePrices](#_OnNotifyStrikePrices) | 選擇權 |

### 連線

#### SKQuoteLib\_EnterMonitorLONG

|  |  |  |
| --- | --- | --- |
| (LONG index)與報價伺服器建立連線。（含盤中零股市場商品） | | |
| **宣告** | Long SKQuoteLib\_EnterMonitorLONG(); | |
| **參數** | 無 |  |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **備註** | 連線狀態由 [OnConnection](#_OnConnection) 事件回傳。  需先簽署證券或期貨API下單聲明書，方可使用 | |

|  |
| --- |
| Python範例程式碼\_Quote.py類別ConnectForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # buttonSKQuoteLib\_EnterMonitorLONG          self.buttonSKQuoteLib\_EnterMonitorLONG = tk.Button(self)          self.buttonSKQuoteLib\_EnterMonitorLONG["text"] = "連線報價主機"          self.buttonSKQuoteLib\_EnterMonitorLONG["command"] = self.buttonSKQuoteLib\_EnterMonitorLONG\_Click          self.buttonSKQuoteLib\_EnterMonitorLONG.grid(row=1, column=1) |

|  |
| --- |
| 全域變數 |
| # 群益API元件導入Python code內用的物件宣告  m\_pSKCenter = comtypes.client.CreateObject(sk.SKCenterLib,interface=sk.ISKCenterLib)  m\_pSKQuote = comtypes.client.CreateObject(sk.SKQuoteLib,interface=sk.ISKQuoteLib) |

|  |
| --- |
| Current Block(放置於def createWidgets(self): 的同一層) |
| def buttonSKQuoteLib\_EnterMonitorLONG\_Click(self):          # 與報價伺服器連線          nCode= m\_pSKQuote.SKQuoteLib\_EnterMonitorLONG()          msg = "【SKQuoteLib\_EnterMonitorLONG】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

#### SKQuoteLib\_LeaveMonitor

|  |  |  |
| --- | --- | --- |
| 中斷所有Solace伺服器連線。 | | |
| **宣告** | Long SKQuoteLib\_LeaveMonitor(); | |
| **參數** | 無 |  |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **備註** | 會中斷所有Solace連線，包含報價與回報，但不會中斷模擬平台回報連線與公告  ＊斷線後，重新連線即還原設定：  限制:『單一SKQuoteLib物件，僅一個訂閱Tick報價功能(10檔商品)及一個RequestStocks訂閱即時報價功能(100檔)』。 | |

|  |
| --- |
| Python範例程式碼\_Quote.py類別ConnectForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # buttonSKQuoteLib\_LeaveMonitor          self.buttonSKQuoteLib\_LeaveMonitor = tk.Button(self)          self.buttonSKQuoteLib\_LeaveMonitor["text"] = "斷線報價主機(ALL)"          self.buttonSKQuoteLib\_LeaveMonitor["command"] = self.buttonSKQuoteLib\_LeaveMonitor\_Click          self.buttonSKQuoteLib\_LeaveMonitor.grid(row=2, column=1) |

|  |
| --- |
| 全域變數 |
| # 群益API元件導入Python code內用的物件宣告  m\_pSKCenter = comtypes.client.CreateObject(sk.SKCenterLib,interface=sk.ISKCenterLib)  m\_pSKQuote = comtypes.client.CreateObject(sk.SKQuoteLib,interface=sk.ISKQuoteLib) |

|  |
| --- |
| Current Block(放置於def createWidgets(self): 的同一層) |
| def buttonSKQuoteLib\_LeaveMonitor\_Click(self):          # 中斷所有Solace伺服器連線          nCode= m\_pSKQuote.SKQuoteLib\_LeaveMonitor()          msg = "【SKQuoteLib\_LeaveMonitor】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

#### SKQuoteLib\_IsConnected

|  |  |
| --- | --- |
| 檢查目前報價的連線狀態。 | |
| **宣告** | Long SKQuoteLib\_IsConnected(); |
| **參數** | 無 |
| **回傳值** | 0表示斷線。1表示連線中。2表示下載中。  錯誤代碼可參考對照表。 |
| **備註** | 請同時接收通知事件[OnConnection](#_OnConnection) |

|  |
| --- |
| Python範例程式碼\_Quote.py類別ConnectForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # buttonSKQuoteLib\_IsConnected          self.buttonSKQuoteLib\_IsConnected = tk.Button(self)          self.buttonSKQuoteLib\_IsConnected["text"] = "檢查連線狀態"          self.buttonSKQuoteLib\_IsConnected["command"] = self.buttonSKQuoteLib\_IsConnected\_Click          self.buttonSKQuoteLib\_IsConnected.grid(row=3, column=1) |

|  |
| --- |
| 全域變數 |
| # 群益API元件導入Python code內用的物件宣告  m\_pSKCenter = comtypes.client.CreateObject(sk.SKCenterLib,interface=sk.ISKCenterLib)  m\_pSKQuote = comtypes.client.CreateObject(sk.SKQuoteLib,interface=sk.ISKQuoteLib) |

|  |
| --- |
| Current Block(放置於def createWidgets(self): 的同一層) |
| def buttonSKQuoteLib\_IsConnected\_Click(self):          # 檢查目前報價的連線狀態          nCode= m\_pSKQuote.SKQuoteLib\_IsConnected()          if nCode == 0:              msg = "斷線"          elif nCode == 1:              msg = "連線中"          elif nCode == 2:              msg = "下載中"          else:              msg = "出錯啦"          msg = "【SKQuoteLib\_IsConnected】" + msg          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

#### SKQuoteLib\_GetQuoteStatus

|  |  |  |
| --- | --- | --- |
| 查詢報價連線狀態(是否超過報價連線限制,連線數資訊)。 | | |
| **宣告** | Long SKQuoteLib\_GetQuoteStatus([in,out] LONG\* pnConnectionCount,  [in,out] VARIANT\_BOOL\* pbIsOutLimit); | |
| **參數** | pnConnectionCount | 連線數:   * 當pbIsOutLimit**連線數超過限制**為true , 連線數為目前限制最大可使用連線數 * 當pbIsOutLimit**連線數超過限制**為false, 連線數為先前已使用連線數(不含當次新連線)   當pnConnectionCount 帶入任一數值，函式庫會回傳連線數，並回傳給呼叫端。 |
| pbIsOutLimit | 報價連線數是超過限制  當pbIsOutLimit帶入false，函式庫會回傳是否超過連線數布林值，並回傳給呼叫端。 |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **備註** | 限當次報價連線成功後，查詢用  一張含有 文字, 螢幕擷取畫面, 數字, 字型 的圖片  自動產生的描述  ＊EX：若最大連線數為2, 且目前連線超過限制,則回傳:2,True. | |

|  |
| --- |
| Python範例程式碼\_Quote.py類別ConnectForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # buttonSKQuoteLib\_GetQuoteStatus          self.buttonSKQuoteLib\_GetQuoteStatus = tk.Button(self)          self.buttonSKQuoteLib\_GetQuoteStatus["text"] = "連線數資訊/限制"          self.buttonSKQuoteLib\_GetQuoteStatus["command"] = self.buttonSKQuoteLib\_GetQuoteStatus\_Click          self.buttonSKQuoteLib\_GetQuoteStatus.grid(row=4, column=1) |

|  |
| --- |
| 全域變數 |
| # 群益API元件導入Python code內用的物件宣告  m\_pSKCenter = comtypes.client.CreateObject(sk.SKCenterLib,interface=sk.ISKCenterLib)  m\_pSKQuote = comtypes.client.CreateObject(sk.SKQuoteLib,interface=sk.ISKQuoteLib) |

|  |
| --- |
| Current Block(放置於def createWidgets(self): 的同一層) |
| def buttonSKQuoteLib\_GetQuoteStatus\_Click(self):          pnConnectionCount = 0          pbIsOutLimit = False          # 查詢報價連線狀態(是否超過報價連線限制,連線數資訊)          pnConnectionCount, pbIsOutLimit, nCode= m\_pSKQuote.SKQuoteLib\_GetQuoteStatus(pnConnectionCount, pbIsOutLimit)          msg = "【SKQuoteLib\_GetQuoteStatus】"  + "連線數:" + str(pnConnectionCount) + "超過限制:" + str(pbIsOutLimit)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

#### SKQuoteLib\_RequestServerTime

|  |  |  |
| --- | --- | --- |
| 要求報價主機傳送目前時間。  目前solace固定每十秒會自動更新時間 | | |
| **宣告** | Long SKQuoteLib\_RequestServerTime(); | |
| **參數** | 無 |  |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **相關通知事件** | [OnNotifyServerTime](#_OnNotifyServerTime) | |
| **備註** |  | |

|  |
| --- |
| Python範例程式碼\_Quote.py類別ConnectForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # buttonSKQuoteLib\_RequestServerTime          self.buttonSKQuoteLib\_RequestServerTime = tk.Button(self)          self.buttonSKQuoteLib\_RequestServerTime["text"] = "報價主機現在時間"          self.buttonSKQuoteLib\_RequestServerTime["command"] = self.buttonSKQuoteLib\_RequestServerTime\_Click          self.buttonSKQuoteLib\_RequestServerTime.grid(row=0, column=1) |

|  |
| --- |
| 全域變數 |
| # 群益API元件導入Python code內用的物件宣告  m\_pSKCenter = comtypes.client.CreateObject(sk.SKCenterLib,interface=sk.ISKCenterLib)  m\_pSKQuote = comtypes.client.CreateObject(sk.SKQuoteLib,interface=sk.ISKQuoteLib) |

|  |
| --- |
| Current Block(放置於def createWidgets(self): 的同一層) |
| def buttonSKQuoteLib\_RequestServerTime\_Click(self):          # 要求報價主機傳送目前時間。          nCode= m\_pSKQuote.SKQuoteLib\_RequestServerTime()          msg = "【SKQuoteLib\_RequestServerTime】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

### 大盤資訊

#### SKQuoteLib\_GetMarketBuySellUpDown

|  |  |  |
| --- | --- | --- |
| 要求傳送上市與上櫃大盤資訊(成交數,買賣數,漲跌家數)。 | | |
| **宣告** | Long SKQuoteLib\_GetMarketBuySellUpDown(); | |
| **參數** | 無 |  |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **相關通知事件** | 1. **大盤張筆**[OnNotifyMarketTot](#_OnNotifyMarketTot) 2. **大盤買賣**[OnNotifyMarketBuySell](#_OnNotifyMarketBuySell) 3. **大盤漲跌家數**[OnNotifyMarketHighLow](#_OnNotifyMarketHighLow) 4. **大盤漲跌家數(『含權證家數』、 『不含權證家數』)**[OnNotifyMarketHighLowNoWarrant](#_OnNotifyMarketHighLowNoWarrant) | |
| **備註** | 不支援盤中零股。 | |

|  |
| --- |
| Python範例程式碼\_Quote.py類別MarketForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # buttonSKQuoteLib\_GetMarketBuySellUpDown          self.buttonSKQuoteLib\_GetMarketBuySellUpDown = tk.Button(self)          self.buttonSKQuoteLib\_GetMarketBuySellUpDown["text"] = "取得大盤資訊 (左)上市 (右)上櫃"          self.buttonSKQuoteLib\_GetMarketBuySellUpDown["command"] = self.buttonSKQuoteLib\_GetMarketBuySellUpDown\_Click          self.buttonSKQuoteLib\_GetMarketBuySellUpDown.grid(row=0, column=1)          # labelMarketTotPtr          tk.Label(self, text = "目前第x筆資料").grid(row=1, column=1)          # labelMarketTotPtr          self.labelMarketTotPtr = tk.Label(self, text = "")          self.labelMarketTotPtr.grid(row=1, column=2)          global labelMarketTotPtr          labelMarketTotPtr = self.labelMarketTotPtr            # labelnTotv          tk.Label(self, text = "成交值(億)").grid(row=2, column=1)          # labelnTotv          self.labelnTotv = tk.Label(self, text = "")          self.labelnTotv.grid(row=2, column=2)          global labelnTotv          labelnTotv = self.labelnTotv            # labelnTots          tk.Label(self, text = "成交張數").grid(row=3, column=1)          # labelnTots          self.labelnTots = tk.Label(self, text = "")          self.labelnTots.grid(row=3, column=2)          global labelnTots          labelnTots = self.labelnTots            # labelnTotc          tk.Label(self, text = "成交筆數").grid(row=4, column=1)          # labelnTotc          self.labelnTotc = tk.Label(self, text = "")          self.labelnTotc.grid(row=4, column=2)          global labelnTotc          labelnTotc = self.labelnTotc            # labelnUp          tk.Label(self, text = "成交上漲家數").grid(row=5, column=1)          # labelnUp          self.labelnUp = tk.Label(self, text = "")          self.labelnUp.grid(row=5, column=2)          global labelnUp          labelnUp = self.labelnUp            # labelnDown          tk.Label(self, text = "成交下跌家數").grid(row=6, column=1)          # labelnDown          self.labelnDown = tk.Label(self, text = "")          self.labelnDown.grid(row=6, column=2)          global labelnDown          labelnDown = self.labelnDown            # labelnHigh          tk.Label(self, text = "成交漲停家數").grid(row=7, column=1)          # labelnHigh          self.labelnHigh = tk.Label(self, text = "")          self.labelnHigh.grid(row=7, column=2)          global labelnHigh          labelnHigh = self.labelnHigh            # labelnLow          tk.Label(self, text = "成交跌停家數").grid(row=8, column=1)          # labelnLow          self.labelnLow = tk.Label(self, text = "")          self.labelnLow.grid(row=8, column=2)          global labelnLow          labelnLow = self.labelnLow            # labelnNoChange          tk.Label(self, text = "平盤家數").grid(row=9, column=1)          # labelnNoChange          self.labelnNoChange = tk.Label(self, text = "")          self.labelnNoChange.grid(row=9, column=2)          global labelnNoChange          labelnNoChange = self.labelnNoChange            # labelMarketTotTime          tk.Label(self, text = "大盤成交時間").grid(row=1, column=3)          # labelMarketTotTime          self.labelMarketTotTime = tk.Label(self, text = "")          self.labelMarketTotTime.grid(row=1, column=4)          global labelMarketTotTime          labelMarketTotTime = self.labelMarketTotTime            # labelnBs          tk.Label(self, text = "成交買進張數").grid(row=2, column=3)          # labelnBs          self.labelnBs = tk.Label(self, text = "")          self.labelnBs.grid(row=2, column=4)          global labelnBs          labelnBs = self.labelnBs            # labelnSs          tk.Label(self, text = "成交賣出張數").grid(row=3, column=3)          # labelnSs          self.labelnSs = tk.Label(self, text = "")          self.labelnSs.grid(row=3, column=4)          global labelnSs          labelnSs = self.labelnSs            # labelnBc          tk.Label(self, text = "成交買進筆數").grid(row=4, column=3)          # labelnBc          self.labelnBc = tk.Label(self, text = "")          self.labelnBc.grid(row=4, column=4)          global labelnBc          labelnBc = self.labelnBc            # labelnSc          tk.Label(self, text = "成交賣出筆數").grid(row=5, column=3)          # labelnSc          self.labelnSc = tk.Label(self, text = "")          self.labelnSc.grid(row=5, column=4)          global labelnSc          labelnSc = self.labelnSc            # labelnUpNoW          tk.Label(self, text = "(不含權證)成交上漲家數").grid(row=6, column=3)          # labelnUpNoW          self.labelnUpNoW = tk.Label(self, text = "")          self.labelnUpNoW.grid(row=6, column=4)          global labelnUpNoW          labelnUpNoW = self.labelnUpNoW            # labelnDownNoW          tk.Label(self, text = "(不含權證)成交下跌家數").grid(row=7, column=3)          # labelnDownNoW          self.labelnDownNoW = tk.Label(self, text = "")          self.labelnDownNoW.grid(row=7, column=4)          global labelnDownNoW          labelnDownNoW = self.labelnDownNoW            # labelnHighNoW          tk.Label(self, text = "(不含權證)成交漲停家數").grid(row=8, column=3)          # labelnHighNoW          self.labelnHighNoW = tk.Label(self, text = "")          self.labelnHighNoW.grid(row=8, column=4)          global labelnHighNoW          labelnHighNoW = self.labelnHighNoW            # labelnLowNoW          tk.Label(self, text = "(不含權證)成交跌停家數").grid(row=9, column=3)          # labelnLowNoW          self.labelnLowNoW = tk.Label(self, text = "")          self.labelnLowNoW.grid(row=9, column=4)          global labelnLowNoW          labelnLowNoW = self.labelnLowNoW            # labelnNoChangeNoW          tk.Label(self, text = "(不含權證)平盤家數").grid(row=10, column=3)          # labelnNoChangeNoW          self.labelnNoChangeNoW = tk.Label(self, text = "")          self.labelnNoChangeNoW.grid(row=10, column=4)          global labelnNoChangeNoW          labelnNoChangeNoW = self.labelnNoChangeNoW                # labelMarketTotPtr2          tk.Label(self, text = "目前第x筆資料").grid(row=1, column=5)          # labelMarketTotPtr2          self.labelMarketTotPtr2 = tk.Label(self, text = "")          self.labelMarketTotPtr2.grid(row=1, column=6)          global labelMarketTotPtr2          labelMarketTotPtr2 = self.labelMarketTotPtr2            # labelnTotv2          tk.Label(self, text = "成交值(億)").grid(row=2, column=5)          # labelnTotv2          self.labelnTotv2 = tk.Label(self, text = "")          self.labelnTotv2.grid(row=2, column=6)          global labelnTotv2          labelnTotv2 = self.labelnTotv2            # labelnTots2          tk.Label(self, text = "成交張數").grid(row=3, column=5)          # labelnTots2          self.labelnTots2 = tk.Label(self, text = "")          self.labelnTots2.grid(row=3, column=6)          global labelnTots2          labelnTots2 = self.labelnTots2            # labelnTotc2          tk.Label(self, text = "成交筆數").grid(row=4, column=5)          # labelnTotc2          self.labelnTotc2 = tk.Label(self, text = "")          self.labelnTotc2.grid(row=4, column=6)          global labelnTotc2          labelnTotc2 = self.labelnTotc2            # labelnUp2          tk.Label(self, text = "成交上漲家數").grid(row=5, column=5)          # labelnUp2          self.labelnUp2 = tk.Label(self, text = "")          self.labelnUp2.grid(row=5, column=6)          global labelnUp2          labelnUp2 = self.labelnUp2            # labelnDown2          tk.Label(self, text = "成交下跌家數").grid(row=6, column=5)          # labelnDown2          self.labelnDown2 = tk.Label(self, text = "")          self.labelnDown2.grid(row=6, column=6)          global labelnDown2          labelnDown2 = self.labelnDown2            # labelnHigh2          tk.Label(self, text = "成交漲停家數").grid(row=7, column=5)          # labelnHigh2          self.labelnHigh2 = tk.Label(self, text = "")          self.labelnHigh2.grid(row=7, column=6)          global labelnHigh2          labelnHigh2 = self.labelnHigh2            # labelnLow2          tk.Label(self, text = "成交跌停家數").grid(row=8, column=5)          # labelnLow2          self.labelnLow2 = tk.Label(self, text = "")          self.labelnLow2.grid(row=8, column=6)          global labelnLow2          labelnLow2 = self.labelnLow2            # labelnNoChange2          tk.Label(self, text = "平盤家數").grid(row=9, column=5)          # labelnNoChange2          self.labelnNoChange2 = tk.Label(self, text = "")          self.labelnNoChange2.grid(row=9, column=6)          global labelnNoChange2          labelnNoChange2 = self.labelnNoChange2            # labelMarketTotTime2          tk.Label(self, text = "大盤成交時間").grid(row=1, column=7)          # labelMarketTotTime2          self.labelMarketTotTime2 = tk.Label(self, text = "")          self.labelMarketTotTime2.grid(row=1, column=8)          global labelMarketTotTime2          labelMarketTotTime2 = self.labelMarketTotTime2            # labelnBs2          tk.Label(self, text = "成交買進張數").grid(row=2, column=7)          # labelnBs2          self.labelnBs2 = tk.Label(self, text = "")          self.labelnBs2.grid(row=2, column=8)          global labelnBs2          labelnBs2 = self.labelnBs2            # labelnSs2          tk.Label(self, text = "成交賣出張數").grid(row=3, column=7)          # labelnSs2          self.labelnSs2 = tk.Label(self, text = "")          self.labelnSs2.grid(row=3, column=8)          global labelnSs2          labelnSs2 = self.labelnSs2            # labelnBc2          tk.Label(self, text = "成交買進筆數").grid(row=4, column=7)          # labelnBc2          self.labelnBc2 = tk.Label(self, text = "")          self.labelnBc2.grid(row=4, column=8)          global labelnBc2          labelnBc2 = self.labelnBc2            # labelnSc2          tk.Label(self, text = "成交賣出筆數").grid(row=5, column=7)          # labelnSc2          self.labelnSc2 = tk.Label(self, text = "")          self.labelnSc2.grid(row=5, column=8)          global labelnSc2          labelnSc2 = self.labelnSc2            # labelnUpNoW2          tk.Label(self, text = "(不含權證)成交上漲家數").grid(row=6, column=7)          # labelnUpNoW2          self.labelnUpNoW2 = tk.Label(self, text = "")          self.labelnUpNoW2.grid(row=6, column=8)          global labelnUpNoW2          labelnUpNoW2 = self.labelnUpNoW2            # labelnDownNoW2          tk.Label(self, text = "(不含權證)成交下跌家數").grid(row=7, column=7)          # labelnDownNoW2          self.labelnDownNoW2 = tk.Label(self, text = "")          self.labelnDownNoW2.grid(row=7, column=8)          global labelnDownNoW2          labelnDownNoW2 = self.labelnDownNoW2            # labelnHighNoW2          tk.Label(self, text = "(不含權證)成交漲停家數").grid(row=8, column=7)          # labelnHighNoW2          self.labelnHighNoW2 = tk.Label(self, text = "")          self.labelnHighNoW2.grid(row=8, column=8)          global labelnHighNoW2          labelnHighNoW2 = self.labelnHighNoW2            # labelnLowNoW2          tk.Label(self, text = "(不含權證)成交跌停家數").grid(row=9, column=7)          # labelnLowNoW2          self.labelnLowNoW2 = tk.Label(self, text = "")          self.labelnLowNoW2.grid(row=9, column=8)          global labelnLowNoW2          labelnLowNoW2 = self.labelnLowNoW2            # labelnNoChangeNoW2          tk.Label(self, text = "(不含權證)平盤家數").grid(row=10, column=7)          # labelnNoChangeNoW2          self.labelnNoChangeNoW2 = tk.Label(self, text = "")          self.labelnNoChangeNoW2.grid(row=10, column=8)          global labelnNoChangeNoW2          labelnNoChangeNoW2 = self.labelnNoChangeNoW2 |

|  |
| --- |
| 全域變數 |
| # 群益API元件導入Python code內用的物件宣告  m\_pSKCenter = comtypes.client.CreateObject(sk.SKCenterLib,interface=sk.ISKCenterLib)  m\_pSKQuote = comtypes.client.CreateObject(sk.SKQuoteLib,interface=sk.ISKQuoteLib) |

|  |
| --- |
| Current Block(放置於def createWidgets(self): 的同一層) |
| def buttonSKQuoteLib\_GetMarketBuySellUpDown\_Click(self):          # 要求傳送上市與上櫃大盤資訊(成交數,買賣數,漲跌家數)          nCode= m\_pSKQuote.SKQuoteLib\_GetMarketBuySellUpDown()          msg = "【SKQuoteLib\_GetMarketBuySellUpDown】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

### 商品清單

#### SKQuoteLib\_RequestStockList

|  |  |  |
| --- | --- | --- |
| 根據市場別編號，取得國內各市場代碼所包含的商品基本資料相關資訊。 | | |
| **宣告** | Long SKQuoteLib\_RequestStockList([in] SHORT sMarketNo); | |
| **參數** | sMarketNo | 市場別代碼。 |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **相關通知事件** | [OnNotifyCommodityListWithTypeNo](#_OnNotifyCommodityListWithTypeNo)  [OnNotifyStockList](#_OnNotifyStockList) | |
| **備註** | ＊請先[SKQuoteLib\_EnterMonitorLONG](#_SKQuoteLib_EnterMonitorLONG)，須等[OnConnection](#_OnConnection)收到SK\_SUBJECT\_CONNECTION\_STOCKS\_READY後，再進行商品檔查詢。  市場編號分別為：  上市 0、上櫃 1、期貨 2、選擇權 3、興櫃 4、  盤中零股-上市5、盤中零股-上櫃6  客製化期貨-9、客製化選擇權-10  若未開立證券帳戶或未簽署證券API下單同意書，將無法查詢  ”上市 0、上櫃 1、興櫃 4”、盤中零股-上市5、盤中零股-上櫃6。（錯誤代碼3031）   * 若未開立期貨帳戶或未簽署期貨API下單同意書，將無法查詢”期貨 2、選擇權 3”。（錯誤代碼3031） | |

|  |
| --- |
| Python範例程式碼\_Quote.py類別ProductForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # comboBoxSKQuoteLib\_RequestStockList          tk.Label(self, text = "請選擇市場別").grid(row=1, column=1)              #輸入框          self.comboBoxSKQuoteLib\_RequestStockList = ttk.Combobox(self, state='readonly')          self.comboBoxSKQuoteLib\_RequestStockList['values'] = Config.comboBoxSKQuoteLib\_RequestStockList          self.comboBoxSKQuoteLib\_RequestStockList.grid(row=1, column=2)          global comboBoxSKQuoteLib\_RequestStockList          comboBoxSKQuoteLib\_RequestStockList = self.comboBoxSKQuoteLib\_RequestStockList          # buttonSKQuoteLib\_RequestStockList          self.buttonSKQuoteLib\_RequestStockList = tk.Button(self)          self.buttonSKQuoteLib\_RequestStockList["text"] = "商品清單送出"          self.buttonSKQuoteLib\_RequestStockList["command"] = self.buttonSKQuoteLib\_RequestStockList\_Click          self.buttonSKQuoteLib\_RequestStockList.grid(row=1, column=3) |

|  |
| --- |
| Config.py |
| # ProductForm  comboBoxSKQuoteLib\_RequestStockList =(      "上市", "上櫃", "期貨","選擇權", "興櫃", "盤中零股-上市","盤中零股-上櫃", "客製化期貨", "客製化選擇權"  ) |
| 全域變數 |
| # 群益API元件導入Python code內用的物件宣告  m\_pSKCenter = comtypes.client.CreateObject(sk.SKCenterLib,interface=sk.ISKCenterLib)  m\_pSKQuote = comtypes.client.CreateObject(sk.SKQuoteLib,interface=sk.ISKQuoteLib) |

|  |
| --- |
| Current Block(放置於def createWidgets(self): 的同一層) |
| def buttonSKQuoteLib\_RequestStockList\_Click(self):          selectValue = comboBoxSKQuoteLib\_RequestStockList.get()          if (selectValue == "上市"):              sMarketNo = 0          elif (selectValue == "上櫃"):              sMarketNo = 1          elif (selectValue == "期貨"):              sMarketNo = 2          elif (selectValue == "選擇權"):              sMarketNo = 3          elif (selectValue == "興櫃"):              sMarketNo = 4          elif (selectValue == "盤中零股-上市"):              sMarketNo = 5          elif (selectValue == "盤中零股-上櫃"):              sMarketNo = 6          elif (selectValue == "客製化期貨"):              sMarketNo = 9          elif (selectValue == "客製化選擇權"):              sMarketNo = 10          # 根據市場別編號，取得國內各市場代碼所包含的商品基本資料相關資訊          nCode= m\_pSKQuote.SKQuoteLib\_RequestStockList(sMarketNo)          msg = "【SKQuoteLib\_RequestStockList】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

### 個股資訊

#### SKQuoteLib\_GetStockByNoLONG

|  |  |  |
| --- | --- | --- |
| (LONG index)根據商品代號，取回商品報價的相關資訊。 | | |
| **宣告** | Long SKQuoteLib\_GetStockByNoLONG ([in] BSTR bstrStockNo, [in,out] struct **SKSTOCKLONG\*** pSKStock); | |
| **參數** | bstrStockNo | 商品代號，例如 6005。 |
| pSKStock | SKCOM元件中的**[SKSTOCKLONG](#_SKSTOCKLONG_(國內報價商品物件_LONG)**物件，將物件帶入此欄位中。 |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **備註** | 僅適用現股，不適用盤中零股-上市及盤中零股-上櫃  ＊須等收到SK\_SUBJECT\_CONNECTION\_STOCKS\_READY後，方可進行。  ＊須使用[SKQuoteLib\_EnterMonitorLONG](#_SKQuoteLib_EnterMonitorLONG)登入，方可執行此函式。  未訂閱即時報價，EX:只先訂成交明細SKQuoteLib\_ReqeustTicks時，執行此功能，僅可取得商品基本資料(商品名稱、昨收價，非即時性質欄位) | |

|  |
| --- |
| Python範例程式碼\_Quote.py類別ProductForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # textBoxSKQuoteLib\_GetStockByNoLONG          tk.Label(self, text = "商品代碼").grid(row=2, column=1)              #輸入框          self.textBoxSKQuoteLib\_GetStockByNoLONG = tk.Entry(self)          self.textBoxSKQuoteLib\_GetStockByNoLONG.grid(row=2, column=2)          global textBoxSKQuoteLib\_GetStockByNoLONG          textBoxSKQuoteLib\_GetStockByNoLONG = self.textBoxSKQuoteLib\_GetStockByNoLONG            # buttonSKQuoteLib\_GetStockByNoLONG          self.buttonSKQuoteLib\_GetStockByNoLONG = tk.Button(self)          self.buttonSKQuoteLib\_GetStockByNoLONG["text"] = "個股資訊"          self.buttonSKQuoteLib\_GetStockByNoLONG["command"] = self.buttonSKQuoteLib\_GetStockByNoLONG\_Click          self.buttonSKQuoteLib\_GetStockByNoLONG.grid(row=2, column=3) |

|  |
| --- |
| 全域變數 |
| # 群益API元件導入Python code內用的物件宣告  m\_pSKCenter = comtypes.client.CreateObject(sk.SKCenterLib,interface=sk.ISKCenterLib)  m\_pSKQuote = comtypes.client.CreateObject(sk.SKQuoteLib,interface=sk.ISKQuoteLib) |

|  |
| --- |
| Current Block(放置於def createWidgets(self): 的同一層) |
| def buttonSKQuoteLib\_GetStockByNoLONG\_Click(self):            pSKStock = sk.SKSTOCKLONG()          # (LONG index)根據商品代號，取回商品報價的相關資訊          pSKStock, nCode= m\_pSKQuote.SKQuoteLib\_GetStockByNoLONG(textBoxSKQuoteLib\_GetStockByNoLONG.get(), pSKStock)          msg = "【SKQuoteLib\_GetStockByNoLONG】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end')          msg = ("類股別" + str(pSKStock.sTypeNo) +          "市埸代碼" + str(pSKStock.bstrMarketNo) +          "商品代碼" + str(pSKStock.bstrStockNo) +          "商品名稱" + str(pSKStock.bstrStockName) +          "最高價" + str(pSKStock.nHigh / 100.0) +          "開盤價" + str(pSKStock.nOpen / 100.0) +          "最低價" + str(pSKStock.nLow / 100.0) +          "成交價" + str(pSKStock.nClose / 100.0) +          "單量" + str(pSKStock.nTickQty) +          "昨收、參考價" + str(pSKStock.nRef / 100.0) +          "買價" + str(pSKStock.nBid / 100.0) +          "買量" + str(pSKStock.nBc) +          "賣價" + str(pSKStock.nAsk / 100.0) +          "賣量" +  str(pSKStock.nAc) +          "買盤量(即外盤量)" +  str(pSKStock.nTBc) +          "賣盤量(即內盤量)" +  str(pSKStock.nTAc) +          "總量" +  str(pSKStock.nTQty) +          "昨量" + str(pSKStock.nYQty) +          "漲停價" +  str(pSKStock.nUp / 100.0) +          "跌停價" +  str(pSKStock.nDown / 100.0) +          "揭示 0:一般 1:試算(試撮)" +  str(pSKStock.nSimulate) +          "[限證券整股商品]可否當沖 0:一般 1:可先買後賣現股當沖2:可先買後賣和先賣後買現股當沖" + str(pSKStock.nDayTrade) +          "交易日(YYYYMMDD)" +  str(pSKStock.nTradingDay) +          "[證券] 整股、盤中零股揭示註記   0:現股 ; 1:盤中零股" +  str(pSKStock.nTradingLotFlag))          richTextBoxMessage.insert('end', msg + "\n")          richTextBoxMessage.see('end') |

##### SKQuoteLib\_GetStockByMarketAndNo

|  |  |  |
| --- | --- | --- |
| (LONG index)根據市場別編號與商品代號，取回商品報價的相關資訊。 | | |
| **宣告** | Long SKQuoteLib\_GetStockByMarketAndNo ([in]SHORT sMarketNo,[in] BSTR bstrStockNo, [in,out] struct **SKSTOCKLONG\*** pSKStock); | |
| **參數** | sMarketNo | 市場別編號 |
| bstrStockNo | 商品代號，例如 6005。 |
| pSKStock | SKCOM元件中的**[SKSTOCKLONG](#_SKSTOCKLONG_(國內報價商品物件_LONG)**物件，將物件帶入此欄位中。 |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **備註** | 適用盤中零股  ＊須等收到SK\_SUBJECT\_CONNECTION\_STOCKS\_READY後，方可進行。  未訂閱即時報價[SKQuoteLib\_RequestStocksWithMarketNo](#_SKQuoteLib_RequestStocksWithMarketN)  ，執行此功能，僅可取得商品基本資料(商品名稱、昨收價，非即時性質欄位) | |

### 即時報價

#### SKQuoteLib\_RequestStocks

|  |  |  |
| --- | --- | --- |
| 此功能不支援盤中零股。  訂閱指定商品即時報價  要求伺服器針對 bstrStockNos 內的商品代號訂閱商品報價通知動作。 | | |
| **宣告** | Long SKQuoteLib\_RequestStocks([**in,out**] SHORT\* psPageNo, [in] BSTR bstrStockNos); | |
| **參數** | psPageNo | 請固定帶1。 |
| bstrStockNos | 欲訂閱的商品代號，一筆以上的資料時，每檔商品代號以”,”做區隔。 |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **相關通知事件** | [OnNotifyQuoteLONG](#_OnNotifyQuoteLONG) | |
| **備註** | ＊請先[SKQuoteLib\_EnterMonitorLONG](#_SKQuoteLib_EnterMonitorLONG)，須等[OnConnection](#_OnConnection)收到SK\_SUBJECT\_CONNECTION\_STOCKS\_READY後，方可進行訂閱商品報價。  \*因應檔數限制，一個SKQuoteLib物件，僅可擇一使用一個即時報價訂閱  ([SKQuoteLib\_RequestStocks](#_SKQuoteLib_RequestTicks)功能或 [SKQuoteLib\_RequestStocksWithMarketNo](#_SKQuoteLib_RequestTicksWithMarketNo)功  能)。  可重新連線即還原限制與設定。  舉例說明：  psPageNo = 1 時，bstrStockNos = “1101,1102,1103,1104,1108,1109,1110”，表示向伺服器索取這七檔股票的報價。  ~~如果帶入的股票數超過100檔，則僅以100檔處理，並不會回傳錯誤。~~  如果帶入的股票代號不存在，則會直接略過不處理，也不會回傳錯誤。  特殊行情：  部分期選商品有T+1盤別交易，若不需要T+1盤的資訊，可在商品代號最後加上AM以取得純AM盤行情，需注意委託商品時加上AM是無法下單的，委託請使用原始商品代號  例：TX00的純AM盤行情為TX00AM  一般用戶目前PageNo上限為1 | |

修改比較表:

|  |  |  |
| --- | --- | --- |
| 修改前 |  | 修改後V2.13.54 |
| 當psPageNo=-1時帶入，函式庫會指定一個新的編號，並回傳給呼叫端。 | 參數psPageNo | 請固定帶1。 |
|  | 備註欄位  新增盤後交易介紹及參考資料 | 註:AM代表國內期選早盤行情，可參考臺灣期貨交易所股份有限公司 各商品規格。  查看交易時間  <https://www.taifex.com.tw/cht/2/tX>  T+1盤：  盤後交易介紹  <https://www.taifex.com.tw/cht/4/aHIntroduction> |

|  |
| --- |
| Python範例程式碼\_Quote.py類別RequestForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # textBoxpsPageNo2          tk.Label(self, text = "Page").grid(row=0, column=1)              #輸入框          self.textBoxpsPageNo2 = tk.Entry(self)          self.textBoxpsPageNo2.grid(row=1, column=2)          global textBoxpsPageNo2          textBoxpsPageNo2 = self.textBoxpsPageNo2            # textBoxStockNos          tk.Label(self, text = "請輸入商品代號(每檔以,做區隔)").grid(row=1, column=3)              #輸入框          self.textBoxStockNos = tk.Entry(self)          self.textBoxStockNos.grid(row=1, column=4)          global textBoxStockNos          textBoxStockNos = self.textBoxStockNos            # textBoxStockNos          tk.Label(self, text = "(不含盤中零股)").grid(row=2, column=1)          # buttonSKQuoteLib\_CancelRequestStocks          self.buttonSKQuoteLib\_CancelRequestStocks = tk.Button(self)          self.buttonSKQuoteLib\_CancelRequestStocks["text"] = "取消訂閱"          self.buttonSKQuoteLib\_CancelRequestStocks["command"] = self.buttonSKQuoteLib\_CancelRequestStocks\_Click          self.buttonSKQuoteLib\_CancelRequestStocks.grid(row=2, column=2)          # buttonSKQuoteLib\_RequestStocks          self.buttonSKQuoteLib\_RequestStocks = tk.Button(self)          self.buttonSKQuoteLib\_RequestStocks["text"] = "訂閱"          self.buttonSKQuoteLib\_RequestStocks["command"] = self.buttonSKQuoteLib\_RequestStocks\_Click          self.buttonSKQuoteLib\_RequestStocks.grid(row=2, column=3)            # comboBoxSKQuoteLib\_RequestStocksWithMarketNo          tk.Label(self, text = "請選擇市場別").grid(row=3, column=1)              #輸入框          self.comboBoxSKQuoteLib\_RequestStocksWithMarketNo = ttk.Combobox(self, state='readonly')          self.comboBoxSKQuoteLib\_RequestStocksWithMarketNo['values'] = Config.comboBoxSKQuoteLib\_RequestStocksWithMarketNo          self.comboBoxSKQuoteLib\_RequestStocksWithMarketNo.grid(row=3, column=2)          global comboBoxSKQuoteLib\_RequestStocksWithMarketNo          comboBoxSKQuoteLib\_RequestStocksWithMarketNo = self.comboBoxSKQuoteLib\_RequestStocksWithMarketNo            # buttonSKQuoteLib\_RequestStocksWithMarketNo          self.buttonSKQuoteLib\_RequestStocksWithMarketNo = tk.Button(self)          self.buttonSKQuoteLib\_RequestStocksWithMarketNo["text"] = "盤中零股訂閱"          self.buttonSKQuoteLib\_RequestStocksWithMarketNo["command"] = self.buttonSKQuoteLib\_RequestStocksWithMarketNo\_Click          self.buttonSKQuoteLib\_RequestStocksWithMarketNo.grid(row=3, column=3) |

|  |
| --- |
| Config.py |
| # RequestForm  comboBoxSKQuoteLib\_RequestStocksWithMarketNo =(      "盤中零股-上市(5)", "盤中零股-上櫃(6)", "客製化期貨-9", "客製化選擇權-10"  ) |
| 全域變數 |
| # 群益API元件導入Python code內用的物件宣告  m\_pSKCenter = comtypes.client.CreateObject(sk.SKCenterLib,interface=sk.ISKCenterLib)  m\_pSKQuote = comtypes.client.CreateObject(sk.SKQuoteLib,interface=sk.ISKQuoteLib) |

|  |
| --- |
| Current Block(放置於def createWidgets(self): 的同一層) |
| def buttonSKQuoteLib\_CancelRequestStocks\_Click(self):            # 取消訂閱SKQuoteLib\_RequestStocks的報價通知，並停止更新商品報價          nCode= m\_pSKQuote.SKQuoteLib\_CancelRequestStocks(textBoxStockNos.get())          msg = "【SKQuoteLib\_CancelRequestStocks】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end')        def buttonSKQuoteLib\_RequestStocks\_Click(self):          psPageNo = int(textBoxpsPageNo2.get())          # 訂閱指定商品即時報價(註冊)          psPageNo, nCode= m\_pSKQuote.SKQuoteLib\_RequestStocks(psPageNo, textBoxStockNos.get())          msg = "【SKQuoteLib\_RequestStocks】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end')        def buttonSKQuoteLib\_RequestStocksWithMarketNo\_Click(self):          psPageNo = int(textBoxpsPageNo2.get())          selectedValue = comboBoxSKQuoteLib\_RequestStocksWithMarketNo.get()          if (selectedValue == "盤中零股-上市(5)"):              sMarketNo = 5          elif (selectedValue == "盤中零股-上櫃(6)"):              sMarketNo = 6          elif (selectedValue == "客製化期貨-9"):              sMarketNo = 9          elif (selectedValue == "客製化選擇權-10"):              sMarketNo = 10          # 訂閱指定市場別及指定商品即時報價          # 要求伺服器針對sMarketNo市場別、 bstrStockNos 內的商品代號訂閱商品報價通知動作          psPageNo, nCode= m\_pSKQuote.SKQuoteLib\_RequestStocksWithMarketNo(psPageNo, sMarketNo, textBoxStockNos.get())          msg = "【SKQuoteLib\_RequestStocksWithMarketNo】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

##### SKQuoteLib\_GetStockByIndexLONG

|  |  |  |
| --- | --- | --- |
| 請先訂閱即時報價([SKQuoteLib\_ReqeustStocks](#_SKQuoteLib_RequestTicks)),方可取得商品報價  未訂閱即時報價,僅可取得商品基本資料  (LONG index)根據市場別編號與系統所編的索引代碼，取回商品報價的及商品相關資訊。 | | |
| **宣告** | Long SKQuoteLib\_GetStockByIndexLONG([in] SHORT sMarketNo, [in] LONG nIndex, [in,out] struct **SKSTOCKLONG\*** pSKStock); | |
| **參數** | sMarketNo | 市場別代碼。 |
| nIndex | 系統所編的索引代碼。 |
| pSKStock | SKCOM元件中的 [**SKSTOCKLONG**](#_SKSTOCKLONG_(國內報價商品物件_LONG)物件，將物件帶入此欄位中。 |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **備註** | 市場編號分別為：**與SKSTOCKLONG物件內之BSTR bstrMarketNo 型態不同**。   |  |  | | --- | --- | | 上市 0x00  上櫃 0x01  期貨 0x02  選擇權 0x03  興櫃 0x04 | 盤中零股-上市 0x05  盤中零股-上櫃 0x06  客製化期貨 0x09  客製化選擇權 0x0A |   ＊須等收到SK\_SUBJECT\_CONNECTION\_STOCKS\_READY後，方可進行。  ＊須使用[SKQuoteLib\_EnterMonitorLONG](#_SKQuoteLib_EnterMonitorLONG)登入，方可執行此函式。  ＊依國內或海外市場，可能需搭配MarketNo  ＊未訂閱即時報價，EX:只先訂成交明細[SKQuoteLib\_ReqeustTicks](#_SKQuoteLib_RequestTicks)時，執行此功能，僅可取得商品基本資料(商品名稱、昨收價，非即時性質欄位)  一張含有 文字, 螢幕擷取畫面, 字型, 電子藍 的圖片  自動產生的描述 | |

#### SKQuoteLib\_RequestStocksWithMarketNo

|  |  |  |
| --- | --- | --- |
| 訂閱指定市場別及指定商品即時報價  要求伺服器針對sMarketNo市場別、 bstrStockNos 內的商品代號訂閱商品報價通知動作。 | | |
| **宣告** | Long SKQuoteLib\_RequestStocksWithMarketNo([**in,out**] SHORT\* psPageNo, [**in**] SHORT sMarketNo, [in] BSTR bstrStockNos); | |
| **參數** | psPageNo | 請固定帶1。 |
| sMarketNo | (目前提供)  盤中零股-上市(5)、盤中零股-上櫃(6)市場代碼  客製化期貨-9、客製化選擇權-10 |
| bstrStockNos | 欲訂閱的商品代號，一筆以上的資料時，每檔商品代號以”,”做區隔。 |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **相關通知事件** | [OnNotifyQuoteLONG](#_OnNotifyQuoteLONG)  [OnNotifyOddLotSpreadDeal](#_OnNotifyOddLotSpreadDeal) | |
| **備註** | 適用盤中零股  ＊請先[SKQuoteLib\_EnterMonitorLONG](#_SKQuoteLib_EnterMonitorLONG)，須等[OnConnection](#_OnConnection)收到SK\_SUBJECT\_CONNECTION\_STOCKS\_READY後，方可進行訂閱商品報價。  \*因應檔數限制，一個SKQuoteLib物件，僅可擇一使用一個即時報價訂閱([SKQuoteLib\_RequestStocks](#_SKQuoteLib_RequestStocks)功能或 SKQuoteLib\_RequestStocksWithMarketNo功能)。  可重新連線即還原限制與設定。 | |

在函式「[SKQuoteLib\_RequestStocks](#_SKQuoteLib_RequestStocks)」有範例程式碼。

#### SKQuoteLib\_CancelRequestStocks

|  |  |  |
| --- | --- | --- |
| 取消訂閱SKQuoteLib\_RequestStocks的報價通知，並停止更新商品報價。 | | |
| **宣告** | Long SKQuoteLib\_CancelRequestStocks([in] BSTR bstrStockNos); | |
| **參數** | bstrStockNos | 欲解除訂閱的商品代號，一筆以上的資料時，每檔股票代號以”,”做區隔。 |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **備註** | 不支援盤中零股。 | |

在函式「[SKQuoteLib\_RequestStocks](#_SKQuoteLib_RequestStocks)」有範例程式碼。

#### SKQuoteLib\_GetMarketPriceTS

|  |  |
| --- | --- |
| 取得證券市場逐筆交易價格欄位為市價時之特殊值。 | |
| **宣告** | Long SKQuoteLib\_GetMarketPriceTS(); |
| **參數** | 無 |
| **回傳值** | 市價:-2147483647 |
| **備註** | 證券即時報價，凡買價、賣價為「市價」時判斷依據  如下圖範例所示：  D:\document\策略王_API_包版文件\109\0322-+\市價_即時報價+2207.jpg |

在事件「[OnNotifyQuoteLONG](#_OnNotifyQuoteLONG)」有範例程式碼。

### 五檔&成交明細

#### SKQuoteLib\_RequestTicks

|  |  |  |
| --- | --- | --- |
| 此功能不支援盤中零股。  訂閱要求傳送成交明細以及五檔。 | | |
| **宣告** | Long SKQuoteLib\_RequestTicks([in,out] SHORT\* psPageNo, [in] BSTR bstrStockNo); | |
| **參數** | psPageNo | 請從0開始。 |
| bstrStockNo | 索取的商品代號，**一個Page僅能索取一檔**。 |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **相關通知事件** | 1. 即時Tick由[OnNotifyTicksLONG](#_OnNotifyTicksLONG_1)事件通知。 2. Tick回補由 [OnNotifyHistoryTicksLONG](#_OnNotifyHistoryTicksLONG)事件通知。 3. 最佳五檔由[OnNotifyBest5LONG](#_OnNotifyBest5LONG)事件通知。 | |
| **備註** | ＊請先[SKQuoteLib\_EnterMonitorLONG](#_SKQuoteLib_EnterMonitorLONG)，須等[OnConnection](#_OnConnection)收到SK\_SUBJECT\_CONNECTION\_STOCKS\_READY後，方可進行訂閱商品成交明細及最佳五檔。  特殊行情：  部分期選商品有T+1盤別交易，若不需要T+1盤的資訊，可在商品代號最後加上AM以取得純AM盤行情，需注意委託商品時加上AM是無法下單的，委託請使用原始商品代號  例：TX00的純AM盤行情為TX00AM  **\*盤中零股：請使用[SKQuoteLib\_RequestTicksWithMarketNo](#_SKQuoteLib_RequestTicksWithMarketNo)** | |

|  |
| --- |
| Python範例程式碼\_Quote.py類別TicksForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # textBoxpsPageNo          tk.Label(self, text = "Page").grid(row=0, column=1)              #輸入框          self.textBoxpsPageNo = tk.Entry(self)          self.textBoxpsPageNo.grid(row=0, column=2)          global textBoxpsPageNo          textBoxpsPageNo = self.textBoxpsPageNo            # textBoxTicks          tk.Label(self, text = "輸入商品代碼(僅1檔)").grid(row=0, column=3)              #輸入框          self.textBoxTicks = tk.Entry(self)          self.textBoxTicks.grid(row=0, column=4)          global textBoxTicks          textBoxTicks = self.textBoxTicks            # buttonSKQuoteLib\_CancelRequestTicks          self.buttonSKQuoteLib\_CancelRequestTicks = tk.Button(self)          self.buttonSKQuoteLib\_CancelRequestTicks["text"] = "取消訂閱"          self.buttonSKQuoteLib\_CancelRequestTicks["command"] = self.buttonSKQuoteLib\_CancelRequestTicks\_Click          self.buttonSKQuoteLib\_CancelRequestTicks.grid(row=1, column=1)            # buttonSKQuoteLib\_RequestTicks          self.buttonSKQuoteLib\_RequestTicks = tk.Button(self)          self.buttonSKQuoteLib\_RequestTicks["text"] = "訂閱"          self.buttonSKQuoteLib\_RequestTicks["command"] = self.buttonSKQuoteLib\_RequestTicks\_Click          self.buttonSKQuoteLib\_RequestTicks.grid(row=1, column=2)            # comboBoxSKQuoteLib\_RequestTicksWithMarketNosMarketNo          tk.Label(self, text = "請選擇市場別").grid(row=2, column=1)              #輸入框          self.comboBoxSKQuoteLib\_RequestTicksWithMarketNosMarketNo = ttk.Combobox(self, state='readonly')          self.comboBoxSKQuoteLib\_RequestTicksWithMarketNosMarketNo['values'] = Config.comboBoxSKQuoteLib\_RequestTicksWithMarketNosMarketNo          self.comboBoxSKQuoteLib\_RequestTicksWithMarketNosMarketNo.grid(row=2, column=2)          global comboBoxSKQuoteLib\_RequestTicksWithMarketNosMarketNo          comboBoxSKQuoteLib\_RequestTicksWithMarketNosMarketNo = self.comboBoxSKQuoteLib\_RequestTicksWithMarketNosMarketNo            # buttonSKQuoteLib\_RequestTicksWithMarketNo          self.buttonSKQuoteLib\_RequestTicksWithMarketNo = tk.Button(self)          self.buttonSKQuoteLib\_RequestTicksWithMarketNo["text"] = "盤中零股訂閱"          self.buttonSKQuoteLib\_RequestTicksWithMarketNo["command"] = self.buttonSKQuoteLib\_RequestTicksWithMarketNo\_Click          self.buttonSKQuoteLib\_RequestTicksWithMarketNo.grid(row=2, column=3) |

|  |
| --- |
| Config.py |
| # TicksForm  comboBoxSKQuoteLib\_RequestTicksWithMarketNosMarketNo =(      "盤中零股-上市(5)", "盤中零股-上櫃(6)", "客製化期貨-9", "客製化選擇權-10"  ) |
| 全域變數 |
| # 群益API元件導入Python code內用的物件宣告  m\_pSKCenter = comtypes.client.CreateObject(sk.SKCenterLib,interface=sk.ISKCenterLib)  m\_pSKQuote = comtypes.client.CreateObject(sk.SKQuoteLib,interface=sk.ISKQuoteLib) |

|  |
| --- |
| Current Block(放置於def createWidgets(self): 的同一層) |
| def buttonSKQuoteLib\_CancelRequestTicks\_Click(self):          # 取消訂閱RequestTicks的成交明細及五檔          nCode= m\_pSKQuote.SKQuoteLib\_CancelRequestTicks(textBoxTicks.get())          msg = "【SKQuoteLib\_CancelRequestTicks】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end')      def buttonSKQuoteLib\_RequestTicks\_Click(self):          psPageNo = int(textBoxpsPageNo.get())          # 此功能不支援盤中零股。訂閱要求傳送成交明細以及五檔          psPageNo, nCode= m\_pSKQuote.SKQuoteLib\_RequestTicks(psPageNo, textBoxTicks.get())          msg = "【SKQuoteLib\_RequestTicks】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end')        def buttonSKQuoteLib\_RequestTicksWithMarketNo\_Click(self):          psPageNo = int(textBoxpsPageNo.get())          selectedValue = comboBoxSKQuoteLib\_RequestTicksWithMarketNosMarketNo.get()          if (selectedValue == "盤中零股-上市(5)"):              sMarketNo = 5          elif (selectedValue == "盤中零股-上櫃(6)"):              sMarketNo = 6          elif (selectedValue == "客製化期貨-9"):              sMarketNo = 9          elif (selectedValue == "客製化選擇權-10"):              sMarketNo = 10          # 適用盤中零股，訂閱要求傳送成交明細以及五檔          psPageNo, nCode= m\_pSKQuote.SKQuoteLib\_RequestTicksWithMarketNo(psPageNo, sMarketNo, textBoxTicks.get())          msg = "【SKQuoteLib\_RequestTicksWithMarketNo】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

##### SKQuoteLib\_GetTickLONG

|  |  |  |
| --- | --- | --- |
| (LONG index)(需**先訂閱即時成交明細RequestTicks**)取得成交明細資訊。 | | |
| **宣告** | Long SKQuoteLib\_GetTickLONG([in] SHORT sMarketNo, [in] LONG nIndex, [in] LONG nPtr, [in,out] struct **SKTICK\*** pSKTick); | |
| **參數** | sMarketNo | 市場別代碼。 |
| nIndex | 系統所編的索引代碼。 |
| nPtr | 第幾筆成交明細，又稱成交明細順序。可以根據此 Ptr 取得**指定第幾筆**成交明細資訊。 |
| pSKTick | SKCOM元件中的 [**SKTICK**](#_SKTICK(國內報價TICK物件))物件，將物件帶入此欄位中。 |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **備註** | 避免在[SKQuoteLib\_OnNotifyHistoryTicksLONG](#_OnNotifyHistoryTicksLONG)、[SKQuoteLib\_OnNotifyTicksLONG](#_OnNotifyTicksLONG_1)通知事件裡進行SKQuoteLib\_GetTickLONG  ＊須使用[SKQuoteLib\_EnterMonitorLONG](#_SKQuoteLib_EnterMonitorLONG)登入，方可執行此函式。  \*Ptr: 第幾筆成交明細，由0開始  一張含有 文字, 螢幕擷取畫面, 字型, Rectangle 的圖片  自動產生的描述 | |

##### SKQuoteLib\_GetBest5LONG

|  |  |  |
| --- | --- | --- |
| (LONG index)(需先訂閱最佳五檔RequestTicks)取得最佳五檔價格資訊。 | | |
| **宣告** | Long SKQuoteLib\_GetBest5LONG([in] SHORT sMarketNo, [in] LONG nIndex, [in,out] struct **SKBEST5\*** pSKBest5); | |
| **參數** | sMarketNo | 市場別代碼。 |
| nIndex | 系統所編的索引代碼。 |
| pSKStock | SKCOM元件中的 [**SKBEST5**](#_SKBEST5(五檔價格物件))物件，將物件帶入此欄位中。 |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **備註** | 避免在[SKQuoteLib\_OnNotifyBest5LONG](#_OnNotifyBest5LONG)通知事件裡進行[SKQuoteLib\_GetBest5LONG](#_SKQuoteLib_GetBest5LONG)  ＊須使用[SKQuoteLib\_EnterMonitorLONG](#_SKQuoteLib_EnterMonitorLONG)登入，方可執行此函式。 | |

#### SKQuoteLib\_RequestTicksWithMarketNo

|  |  |  |
| --- | --- | --- |
| 訂閱要求傳送成交明細以及五檔。 | | |
| **宣告** | Long SKQuoteLib\_RequestTicksWithMarketNo([in,out] SHORT\* psPageNo, [in] SHORT sMarketNo ,[in] BSTR bstrStockNo); | |
| **參數** | psPageNo | 請從0開始。 |
| sMarketNo | 盤中零股-上市(5)、盤中零股- 上櫃(6)市場代碼  客製化期貨-9、客製化選擇權-10 |
| bstrStockNo | 索取的商品代號，**一個Page僅能索取一檔**。 |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **相關通知事件** | 1. 即時Tick由[OnNotifyTicksLONG](#_OnNotifyTicksLONG_1)事件通知。 2. Tick回補由 [OnNotifyHistoryTicksLONG](#_OnNotifyHistoryTicksLONG)事件通知。 3. 最佳五檔由[OnNotifyBest5LONG](#_OnNotifyBest5LONG)事件通知。 | |
| **備註** | 適用盤中零股  ＊請先[SKQuoteLib\_EnterMonitorLONG](#_SKQuoteLib_EnterMonitorLONG)，須等[OnConnection](#_OnConnection)收到SK\_SUBJECT\_CONNECTION\_STOCKS\_READY後，方可進行訂閱商品成交明細及最佳五檔。 | |

|  |
| --- |
| C#範例程式碼\_QuoteForm.cs |
| UI示意圖 |
| |  | | --- | |  | | 拉出1個ComboBox，命名(Name)為「comboBoxSKQuoteLib\_RequestTicksWithMarketNosMarketNo」。  拉出1個Button，命名(Name)為「buttonSKQuoteLib\_RequestTicksWithMarketNo」。 | |
| 全域變數 |
| // 宣告物件  SKCenterLib m\_pSKCenter = new SKCenterLib(); // 登入&環境設定物件  SKQuoteLib m\_pSKQuote = new SKQuoteLib(); // 國內報價物件 |
| Constructor |
| //comboBoxSKQuoteLib\_RequestTicksWithMarketNosMarketNo  {  comboBoxSKQuoteLib\_RequestTicksWithMarketNosMarketNo.Items.Add("盤中零股-上市(5)");  comboBoxSKQuoteLib\_RequestTicksWithMarketNosMarketNo.Items.Add("盤中零股-上櫃(6)");  comboBoxSKQuoteLib\_RequestTicksWithMarketNosMarketNo.Items.Add("客製化期貨-9");  comboBoxSKQuoteLib\_RequestTicksWithMarketNosMarketNo.Items.Add("客製化選擇權-10");  } |

|  |
| --- |
| Current Block (可放在buttonSKQuoteLib\_RequestTicksWithMarketNo\_Click的Click事件裡) |
| // 適用盤中零股，訂閱要求傳送成交明細以及五檔  short psPageNo = short.Parse(textBoxpsPageNo.Text);  short sMarketNo = 5; //盤中零股-上市(5)、盤中零股- 上櫃(6)市場代碼  string selectedValue = comboBoxSKQuoteLib\_RequestTicksWithMarketNosMarketNo.Text;  if (selectedValue == "盤中零股-上市(5)") sMarketNo = 5;  else if (selectedValue == "盤中零股-上櫃(6)") sMarketNo = 6;  else if (selectedValue == "客製化期貨-9") sMarketNo = 9;  else if (selectedValue == "客製化選擇權-10") sMarketNo = 10;  int nCode = m\_pSKQuote.SKQuoteLib\_RequestTicksWithMarketNo(psPageNo, sMarketNo, textBoxTicks.Text);  // 取得回傳訊息  string msg = "【SKQuoteLib2\_RequestTicksWithMarketNo】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode);  richTextBoxMethodMessage.AppendText(msg + "\n"); |
| 執行成功畫面 |
|  |

#### SKQuoteLib\_RequestLiveTick

|  |  |  |
| --- | --- | --- |
| 訂閱與要求傳送**即時**成交明細。  (本功能不會訂閱最佳五檔，亦不包含歷史Ticks) | | |
| **宣告** | Long SKQuoteLib\_RequestLiveTick([in,out] SHORT\* psPageNo, [in] BSTR bstrStockNo); | |
| **參數** | psPageNo | 請從0開始。 |
| bstrStockNo | 索取的商品代號，**一個Page僅能索取一檔**。 |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **相關通知事件** | 即時Tick由[OnNotifyTicksLONG](#_OnNotifyTicksLONG_1)事件通知。 | |
| **備註** | ＊請先[SKQuoteLib\_EnterMonitorLONG](#_SKQuoteLib_EnterMonitorLONG)，須等[OnConnection](#_OnConnection)收到SK\_SUBJECT\_CONNECTION\_STOCKS\_READY後，再進行商品檔查詢。  與[SKQuoteLib\_RequestTicks](#_SKQuoteLib_RequestTicks)不同，**不會**回補歷史成交明細資料，請擇一使用;因無回補歷史成交明細資料，[SKQuoteLib\_GetTickLONG](#_SKQuoteLib_GetTickLONG)僅查閱，**訂閱後**之即時成交明細資料。 | |

作法類同「[SKQuoteLib\_RequestTicks](#_SKQuoteLib_RequestTicks)」，請擇一使用。

#### SKQuoteLib\_CancelRequestTicks

|  |  |  |
| --- | --- | --- |
| 取消訂閱RequestTicks的成交明細及五檔。 | | |
| **宣告** | Long SKQuoteLib\_CancelRequestTicks([in] BSTR bstrStockNo); | |
| **參數** | bstrStockNo | 欲解除訂閱的商品代號，**一次僅能解訂閱一檔**。 |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **備註** | 不支援盤中零股。 | |

在函式「[SKQuoteLib\_RequestTicks](#_SKQuoteLib_RequestTicks)」有範例程式碼。

### 技術分析

#### MACD

##### SKQuoteLib\_RequestMACD

|  |  |  |
| --- | --- | --- |
| 要求傳送商品技術指標MACD。(平滑異同平均線) | | |
| **宣告** | Long SKQuoteLib\_RequestMACD([in,out] SHORT\* psPageNo, [in] BSTR bstrStockNo); | |
| **參數** | psPageNo | 每一次送出要求的動作都需要向報價伺服器指定一個  特定的 Page 編號，**當代入 Page 為 50 會取消該檔的MACD**。  (請參考[SKQuoteLib\_RequestStocks](#_SKQuoteLib_RequestStocks)觀念說明)。 |
| bstrStockNo | 索取的商品代號，**一個Page僅能索取一檔**。 |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **相關通知事件** | [OnNotifyMACDLONG](#_OnNotifyMACDLONG) | |
| **備註** | 目前只提供證券市場，不含盤中零股。 | |

|  |
| --- |
| Python範例程式碼\_Quote.py類別KLineForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # textBoxSKQuoteLib\_RequestMACDpsPageNo          tk.Label(self, text = "Page").grid(row=0, column=1)              #輸入框          self.textBoxSKQuoteLib\_RequestMACDpsPageNo = tk.Entry(self)          self.textBoxSKQuoteLib\_RequestMACDpsPageNo.grid(row=0, column=2)          global textBoxSKQuoteLib\_RequestMACDpsPageNo          textBoxSKQuoteLib\_RequestMACDpsPageNo = self.textBoxSKQuoteLib\_RequestMACDpsPageNo            # textBoxbstrStockNo          tk.Label(self, text = "商品代碼").grid(row=0, column=3)              #輸入框          self.textBoxbstrStockNo = tk.Entry(self)          self.textBoxbstrStockNo.grid(row=0, column=4)          global textBoxbstrStockNo          textBoxbstrStockNo = self.textBoxbstrStockNo            # buttonSKQuoteLib\_RequestMACD          self.buttonSKQuoteLib\_RequestMACD = tk.Button(self)          self.buttonSKQuoteLib\_RequestMACD["text"] = "MACD"          self.buttonSKQuoteLib\_RequestMACD["command"] = self.buttonSKQuoteLib\_RequestMACD\_Click          self.buttonSKQuoteLib\_RequestMACD.grid(row=1, column=1)            # buttonSKQuoteLib\_RequestBoolTunel          self.buttonSKQuoteLib\_RequestBoolTunel = tk.Button(self)          self.buttonSKQuoteLib\_RequestBoolTunel["text"] = "BoolTunel"          self.buttonSKQuoteLib\_RequestBoolTunel["command"] = self.buttonSKQuoteLib\_RequestBoolTunel\_Click          self.buttonSKQuoteLib\_RequestBoolTunel.grid(row=2, column=1)            # comboBoxsKLineType          tk.Label(self, text = "K線種類").grid(row=3, column=1)              #輸入框          self.comboBoxsKLineType = ttk.Combobox(self, state='readonly')          self.comboBoxsKLineType['values'] = Config.comboBoxsKLineType          self.comboBoxsKLineType.grid(row=3, column=2)          global comboBoxsKLineType          comboBoxsKLineType = self.comboBoxsKLineType            # comboBoxsOutType          tk.Label(self, text = "輸出格式").grid(row=3, column=3)              #輸入框          self.comboBoxsOutType = ttk.Combobox(self, state='readonly')          self.comboBoxsOutType['values'] = Config.comboBoxsOutType          self.comboBoxsOutType.grid(row=3, column=4)          global comboBoxsOutType          comboBoxsOutType = self.comboBoxsOutType            # comboBoxsTradeSession          tk.Label(self, text = "全盤/AM盤").grid(row=3, column=5)              #輸入框          self.comboBoxsTradeSession = ttk.Combobox(self, state='readonly')          self.comboBoxsTradeSession['values'] = Config.comboBoxsTradeSession          self.comboBoxsTradeSession.grid(row=3, column=6)          global comboBoxsTradeSession          comboBoxsTradeSession = self.comboBoxsTradeSession          # textBoxbstrStartDate          tk.Label(self, text = "起始日期").grid(row=3, column=7)              #輸入框          self.textBoxbstrStartDate = tk.Entry(self)          self.textBoxbstrStartDate.grid(row=3, column=8)          global textBoxbstrStartDate          textBoxbstrStartDate = self.textBoxbstrStartDate            # textBoxbstrEndDate          tk.Label(self, text = "結束日期").grid(row=3, column=9)              #輸入框          self.textBoxbstrEndDate = tk.Entry(self)          self.textBoxbstrEndDate.grid(row=3, column=10)          global textBoxbstrEndDate          textBoxbstrEndDate = self.textBoxbstrEndDate            # textBoxsMinuteNumber          tk.Label(self, text = "幾分K(分線)").grid(row=3, column=11)              #輸入框          self.textBoxsMinuteNumber = tk.Entry(self)          self.textBoxsMinuteNumber.grid(row=3, column=12)          global textBoxsMinuteNumber          textBoxsMinuteNumber = self.textBoxsMinuteNumber            # buttonSKQuoteLib\_RequestKLineAMByDate          self.buttonSKQuoteLib\_RequestKLineAMByDate = tk.Button(self)          self.buttonSKQuoteLib\_RequestKLineAMByDate["text"] = "KLine"          self.buttonSKQuoteLib\_RequestKLineAMByDate["command"] = self.buttonSKQuoteLib\_RequestKLineAMByDate\_Click          self.buttonSKQuoteLib\_RequestKLineAMByDate.grid(row=4, column=1) |

|  |
| --- |
| Config.py |
| # KLineForm  comboBoxsKLineType =(      "分線", "日線", "週線", "月線"  )  comboBoxsOutType =(      "舊版", "新版"  )  comboBoxsTradeSession =(      "全盤", "AM盤"  ) |
| 全域變數 |
| # 群益API元件導入Python code內用的物件宣告  m\_pSKCenter = comtypes.client.CreateObject(sk.SKCenterLib,interface=sk.ISKCenterLib)  m\_pSKQuote = comtypes.client.CreateObject(sk.SKQuoteLib,interface=sk.ISKQuoteLib) |

|  |
| --- |
| Current Block(放置於def createWidgets(self): 的同一層) |
| def buttonSKQuoteLib\_RequestMACD\_Click(self):          psPageNo = int(textBoxSKQuoteLib\_RequestMACDpsPageNo.get())          # 要求傳送商品技術指標MACD。(平滑異同平均線)          psPageNo, nCode= m\_pSKQuote.SKQuoteLib\_RequestMACD(psPageNo, textBoxbstrStockNo.get())          msg = "【SKQuoteLib\_RequestMACD】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end')        def buttonSKQuoteLib\_RequestBoolTunel\_Click(self):          psPageNo = int(textBoxSKQuoteLib\_RequestMACDpsPageNo.get())          # 要求傳送商品布林通道BoolTunel          psPageNo, nCode= m\_pSKQuote.SKQuoteLib\_RequestBoolTunel(psPageNo, textBoxbstrStockNo.get())          msg = "【SKQuoteLib\_RequestBoolTunel】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end')        def buttonSKQuoteLib\_RequestKLineAMByDate\_Click(self):          if (comboBoxsKLineType.get() == "分線"):              sKLineType = 0          elif (comboBoxsKLineType.get() == "日線"):              sKLineType = 4          elif (comboBoxsKLineType.get() == "週線"):              sKLineType = 5          elif (comboBoxsKLineType.get() == "月線"):              sKLineType = 6          if (comboBoxsOutType.get() == "舊版"):              sOutType = 0          elif (comboBoxsOutType.get() == "新版"):              sOutType = 1            if (comboBoxsTradeSession.get() == "全盤"):              sTradeSession = 0          elif (comboBoxsTradeSession.get() == "AM盤"):              sTradeSession = 1          # （僅提供歷史資料）向報價伺服器提出，取得單一商品技術分析資訊需求，可選AM盤或全盤，可指定日期區間，分K時可指定幾分K          nCode= m\_pSKQuote.SKQuoteLib\_RequestKLineAMByDate(textBoxbstrStockNo.get(), sKLineType, sOutType, sTradeSession, textBoxbstrStartDate.get(), textBoxbstrEndDate.get(), int(textBoxsMinuteNumber.get()))          msg = "【SKQuoteLib\_RequestKLineAMByDate】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

##### SKQuoteLib\_GetMACDLONG

|  |  |  |
| --- | --- | --- |
| (LONG index)取得商品技術指標MACD資訊。(平滑異同平均線) | | |
| **宣告** | Long SKQuoteLib\_GetMACDLONG([in] SHORT sMarketNo, [in] LONG nIndex, [in,out] struct **SKMACD\*** pSKMACD); | |
| **參數** | sMarketNo | 市場別代碼。 |
| nIndex | 系統所編的索引代碼。 |
| pSKStock | SKCOM元件中的**[SKMACD](#_SKMACD(國內報價SKMACD物件))**物件，將物件帶入此欄位中。 |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **備註** | 目前只提供證券市場  ＊須使用[SKQuoteLib\_EnterMonitorLONG](#_SKQuoteLib_EnterMonitorLONG)登入，方可執行此函式。 | |

在事件「[OnNotifyMACDLONG](#_OnNotifyMACDLONG)」有範例程式碼。

#### BoolTunel

##### SKQuoteLib\_RequestBoolTunel

|  |  |  |
| --- | --- | --- |
| 要求傳送商品布林通道BoolTunel。 | | |
| **宣告** | Long SKQuoteLib\_RequestBoolTunel([in,out] SHORT\* psPageNo, [in] BSTR bstrStockNo); | |
| **參數** | psPageNo | 每一次送出要求的動作都需要向報價伺服器指定一個  特定的 Page 編號，**當代入 Page 為 50 會取消該檔的BoolTunel**。  (請參考[SKQuoteLib\_RequestStocks](#_SKQuoteLib_RequestStocks)觀念說明)。 |
| bstrStockNo | 索取的商品代號，**一個Page僅能索取一檔**。 |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **相關通知事件** | [OnNotifyBoolTunelLONG](#_OnNotifyBoolTunelLONG_1) | |
| **備註** | 目前只提供證券市場，不含盤中零股。 | |

在函式「[SKQuoteLib\_RequestMACD](#_SKQuoteLib_RequestMACD)」有範例程式碼。

##### SKQuoteLib\_GetBoolTunelLONG

|  |  |  |
| --- | --- | --- |
| (LONG index)取得商品BoolTunel資訊。 | | |
| **宣告** | Long SKQuoteLib\_GetBoolTunelLONG([in] SHORT sMarketNo, [in] LONG nIndex, [in,out] struct **SKBoolTunel\*** pSKBoolTunel); | |
| **參數** | sMarketNo | 市場別代碼。 |
| nIndex | 系統所編的索引代碼。 |
| pSKBoolTunel | SKCOM元件中的**[SKBoolTunel](#_SKBoolTunel_(國內報價SKBoolTunel物件))**物件，將物件帶入此欄位中。 |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **備註** | 目前只提供證券市場  ＊須使用[SKQuoteLib\_EnterMonitorLONG](#_SKQuoteLib_EnterMonitorLONG)登入，方可執行此函式。 | |

在事件「[OnNotifyBoolTunelLONG](#_OnNotifyBoolTunelLONG_1)」有範例程式碼。

#### Kline

##### SKQuoteLib\_RequestKLineAMByDate

|  |  |  |
| --- | --- | --- |
| （僅提供歷史資料）向報價伺服器提出，取得單一商品技術分析資訊需求，可選AM盤或全盤，可指定日期區間，分K時可指定幾分K。 | | |
| **宣告** | Long SKQuoteLib\_RequestKLineAMByDate([in] BSTR bstrStockNo, [in] SHORT sKLineType, [in] SHORT sOutType, [in] SHORT sTradeSession, [in] BSTR bstrStartDate, [in] BSTR bstrEndDate, [in] SHORT sMinuteNumber); | |
| **參數** | bstrStockNo | 商品代號，例如 6005。 |
| sKLineType | 0 = 分線。  4 = 日線。  5 = 週線。  6 = 月線。 |
| sOutType | 0 = 舊版輸出格式。  1 = 新版輸出格式。 |
| sTradeSession | 僅國內期權技術分析有效。  0 = 全盤技術分析  1 = AM盤技術分析 |
| bstrStartDate | 起始日期，格式為：YYYYMMDD (ex:20201001) |
| bstrEndDate | 結束日期，格式為：YYYYMMDD (ex:20201010) |
| sMinuteNumber | 指定幾分K (ex : 1=1分K, 3=3分K)  **sKLineType = 0時, 此參數才有意義** |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **相關通知事件** | [OnNotifyKLineData](#_OnNotifyKLineData_1) | |
| **備註** | 不支援盤中零股。   * 證券:該查詢以整股為主計算，不包含「鉅額交易、盤中零股、盤後零股」 * 國內期選—全盤:   EX: 指定日期區間為2023/9/5全盤K LINE，將包含2023/9/4 17:25開盤後~2023/9/5 05:00，及2023/9/5 8:45~16:15。   * 國內期選—AM盤:當日T盤     \*若您未開立證券帳戶，無法訂閱或取得證券上市櫃、興櫃市場商品及上市櫃指數資料(含歷史K線)。  \*若您未開立期貨帳戶，無法訂閱或取得期貨選擇權相關市場商品資料(含歷史K線)。 | |

比較表:

|  |  |  |
| --- | --- | --- |
| 修改前 | 說明 | V2.13.45 |
| N/A | 備註  新增說明—證券市場 | 該查詢以整股為主計算，不包含「鉅額交易、盤中零股、盤後零股」 |
| N/A | 新增說明—國內期選盤別 | 全盤及AM盤 |

在函式「[SKQuoteLib\_RequestMACD](#_SKQuoteLib_RequestMACD)」有範例程式碼。

###### SKQuoteLib\_RequestKLineAM

|  |  |  |
| --- | --- | --- |
| （僅提供歷史資料）向報價伺服器提出，取得單一商品技術分析資訊需求，可選AM盤或全盤。 | | |
| **宣告** | Long SKQuoteLib\_RequestKLineAM([in] BSTR bstrStockNo, [in] SHORT sKLineType, [in] BSTR sOutType, [in] SHORT sTradeSession); | |
| **參數** | bstrStockNo | 商品代號，例如 6005。 |
| sKLineType | 0 = 1分鐘線。  4 =完整日線。  5 =週線。  6 =月線。 |
| sOutType | 0=舊版輸出格式。  1=新版輸出格式。 |
| sTradeSession | 僅國內期權技術分析有效。  0=全盤技術分析  1=AM盤技術分析 |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **相關通知事件** | [OnNotifyKLineData](#_OnNotifyKLineData_1) | |
| **備註** | * 國內期選—全盤:   EX: 2023/9/5全盤K LINE，將包含2023/9/4 17:25開盤後~2023/9/5 05:00，及2023/9/5 8:45~16:15。   * 國內期選—AM盤:當日T盤   \*若您未開立期貨帳戶，無法訂閱或取得期貨選擇權相關市場商品資料(含歷史K線)。  目前僅提供1分鐘K，其他5分鐘、30分鐘K Line可自行以1分鐘K Line為基礎組成，目前未提供288日K Line。 | |

比較表:

|  |  |  |
| --- | --- | --- |
| 修改前 | 說明 | V2.13.45 |
| N/A | 新增說明—國內期選盤別 | 全盤及AM盤 |

###### SKQuoteLib\_RequestKLine

|  |  |  |
| --- | --- | --- |
| （僅提供歷史資料）向報價伺服器提出，取得單一商品技術分析資訊需求。 | | |
| **宣告** | Long SKQuoteLib\_RequestKLine([in] BSTR bstrStockNo, [in] SHORT sKLineType, [in] BSTR sOutType); | |
| **參數** | bstrStockNo | 商品代號，例如 6005。 |
| sKLineType | 0 = 1分鐘線。  4 =完整日線。  5 =週線。  6 =月線。 |
| sOutType | 0=舊版輸出格式。  1=新版輸出格式。 |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **相關通知事件** | [OnNotifyKLineData](#_OnNotifyKLineData_1) | |
| **備註** | * 不支援盤中零股。 * 證券:該查詢以整股為主計算，不包含「鉅額交易、盤中零股、盤後零股」 * 國內期選—全盤:   EX: 指定日期區間為2023/9/5全盤K LINE，將包含2023/9/4 17:25開盤後~2023/9/5 05:00，及2023/9/5 8:45~16:15。   * 國內期選—AM盤:當日T盤   \*若您未開立證券帳戶，無法訂閱或取得證券上市櫃、興櫃市場商品及上市櫃指數資料(含歷史K線)。  \*若您未開立期貨帳戶，無法訂閱或取得期貨選擇權相關市場商品資料(含歷史K線)。  solace目前僅提供1分鐘K，其他5分鐘、30分鐘K Line可自行以1分鐘K Line為基礎組成，目前未提供288日K Line。  僅輸出AM盤，若需全盤請使用[SKQuoteLib\_RequestKLineAM](#_SKQuoteLib_RequestKLineAM) | |

比較表:

|  |  |  |
| --- | --- | --- |
| 修改前 | 說明 | V2.13.45 |
| N/A | 備註  新增說明—證券市場 | 該查詢以整股為主計算，不包含「鉅額交易、盤中零股、盤後零股」 |
| N/A | 新增說明—國內期選盤別 | 全盤及AM盤 |

### 風險計算(選擇權)

#### SKQuoteLib\_Delta

|  |  |  |
| --- | --- | --- |
| 輸入  1. nCallPut 商品買賣權別(Call: 0, Put: 1)  2. S 指數  3. K StrikePrice(履約價)  4. R RisklessRate(無風險利率)  5. T RemainMaturity(剩餘天數)  6. sigma  得到Delta值 | | |
| **宣告** | Long SKQuoteLib\_Delta([in] SHORT nCallPut, [in] DOUBLE S, [in] DOUBLE K, [in] DOUBLE R, [in] DOUBLE T, [in] DOUBLE sigma, [out] DOUBLE\* dDelta); | |
| **參數** | nCallPut | 商品買賣權別( Call: 0 , Put: 1) |
| S | 指數 |
| K | StrikePrice(履約價) |
| R | RisklessRate(無風險利率) |
| T | RemainMaturity(剩餘天數) |
| sigma | sigma |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |

|  |
| --- |
| Python範例程式碼\_Quote.py類別DeltaForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # textBoxS          tk.Label(self, text = "指數").grid(row=0, column=1)              #輸入框          self.textBoxS = tk.Entry(self)          self.textBoxS.grid(row=0, column=2)          global textBoxS          textBoxS = self.textBoxS            # textBoxK          tk.Label(self, text = "履約價").grid(row=0, column=3)              #輸入框          self.textBoxK = tk.Entry(self)          self.textBoxK.grid(row=0, column=4)          global textBoxK          textBoxK = self.textBoxK            # textBoxR          tk.Label(self, text = "無風險利率").grid(row=1, column=1)              #輸入框          self.textBoxR = tk.Entry(self)          self.textBoxR.grid(row=1, column=2)          global textBoxR          textBoxR = self.textBoxR            # textBoxT          tk.Label(self, text = "剩餘天數").grid(row=1, column=3)              #輸入框          self.textBoxT = tk.Entry(self)          self.textBoxT.grid(row=1, column=4)          global textBoxT          textBoxT = self.textBoxT            # textBoxsigma          tk.Label(self, text = "sigma").grid(row=2, column=1)              #輸入框          self.textBoxsigma = tk.Entry(self)          self.textBoxsigma.grid(row=2, column=2)          global textBoxsigma          textBoxsigma = self.textBoxsigma            # comboBoxnCallPut          tk.Label(self, text = "買賣權別").grid(row=2, column=3)              #輸入框          self.comboBoxnCallPut = ttk.Combobox(self, state='readonly')          self.comboBoxnCallPut['values'] = Config.comboBoxnCallPut          self.comboBoxnCallPut.grid(row=2, column=4)          global comboBoxnCallPut          comboBoxnCallPut = self.comboBoxnCallPut          # buttonSKQuoteLib\_Gamma          self.buttonSKQuoteLib\_Gamma = tk.Button(self)          self.buttonSKQuoteLib\_Gamma["text"] = "計算"          self.buttonSKQuoteLib\_Gamma["command"] = self.buttonSKQuoteLib\_Gamma\_Click          self.buttonSKQuoteLib\_Gamma.grid(row=3, column=1)          # labelDelta          tk.Label(self, text = "Delta:").grid(row=0, column=5)          # labelDelta          self.labelDelta = tk.Label(self, text = "")          self.labelDelta.grid(row=0, column=6)          global labelDelta          labelDelta = self.labelDelta          # labelGamma          tk.Label(self, text = "Gamma:").grid(row=1, column=5)          # labelGamma          self.labelGamma = tk.Label(self, text = "")          self.labelGamma.grid(row=1, column=6)          global labelGamma          labelGamma = self.labelGamma            # labelVega          tk.Label(self, text = "Vega:").grid(row=2, column=5)          # labelVega          self.labelVega = tk.Label(self, text = "")          self.labelVega.grid(row=2, column=6)          global labelVega          labelVega = self.labelVega            # labelTheta          tk.Label(self, text = "Theta:").grid(row=3, column=5)          # labelTheta          self.labelTheta = tk.Label(self, text = "")          self.labelTheta.grid(row=3, column=6)          global labelTheta          labelTheta = self.labelTheta            # labelRho          tk.Label(self, text = "Rho:").grid(row=4, column=5)          # labelRho          self.labelRho = tk.Label(self, text = "")          self.labelRho.grid(row=4, column=6)          global labelRho          labelRho = self.labelRho |

|  |
| --- |
| Config.py |
| # DeltaForm  comboBoxnCallPut =(      "Call", "Put"  ) |
| 全域變數 |
| # 群益API元件導入Python code內用的物件宣告  m\_pSKCenter = comtypes.client.CreateObject(sk.SKCenterLib,interface=sk.ISKCenterLib)  m\_pSKQuote = comtypes.client.CreateObject(sk.SKQuoteLib,interface=sk.ISKQuoteLib) |

|  |
| --- |
| Current Block(放置於def createWidgets(self): 的同一層) |
| def buttonSKQuoteLib\_Gamma\_Click(self):          # 輸入1~5得到 Gamma值          Gamma, nCode= m\_pSKQuote.SKQuoteLib\_Gamma(int(textBoxS.get()), int(textBoxK.get()), int(textBoxR.get()), int(textBoxT.get()), int(textBoxsigma.get()))          labelGamma.config(text = Gamma)          msg = "【SKQuoteLib\_Gamma】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          # 輸入1~5得到 Vega值          Vega, nCode= m\_pSKQuote.SKQuoteLib\_Vega(int(textBoxS.get()), int(textBoxK.get()), int(textBoxR.get()), int(textBoxT.get()), int(textBoxsigma.get()))          labelVega.config(text = Vega)          msg = "【SKQuoteLib\_Vega】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          if (comboBoxnCallPut.get() == "Call"):              nCallPut = 0          elif (comboBoxnCallPut.get() == "Put"):              nCallPut = 1          # 輸入1~6得到 Delta值          Delta, nCode= m\_pSKQuote.SKQuoteLib\_Delta(nCallPut, int(textBoxS.get()), int(textBoxK.get()), int(textBoxR.get()), int(textBoxT.get()), int(textBoxsigma.get()))          labelDelta.config(text = Delta)          msg = "【SKQuoteLib\_Delta】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          # 輸入1~6得到 Theta值          Theta, nCode= m\_pSKQuote.SKQuoteLib\_Theta(nCallPut, int(textBoxS.get()), int(textBoxK.get()), int(textBoxR.get()), int(textBoxT.get()), int(textBoxsigma.get()))          labelTheta.config(text = Theta)          msg = "【SKQuoteLib\_Theta】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          # 輸入1~6得到 Rho值          Rho, nCode= m\_pSKQuote.SKQuoteLib\_Rho(nCallPut, int(textBoxS.get()), int(textBoxK.get()), int(textBoxR.get()), int(textBoxT.get()), int(textBoxsigma.get()))          labelRho.config(text = Rho)          msg = "【SKQuoteLib\_Rho】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

#### SKQuoteLib\_Gamma

|  |  |  |
| --- | --- | --- |
| 輸入  1. S 指數  2. K StrikePrice(履約價)  3. R RisklessRate(無風險利率)  4. T RemainMaturity(剩餘天數)  5. sigma  得到Gamma值 | | |
| **宣告** | Long SKQuoteLib\_Gamma([[in] DOUBLE S, [in] DOUBLE K, [in] DOUBLE R, [in] DOUBLE T, [in] DOUBLE sigma, [out] DOUBLE\* dGamma); | |
| **參數** | S | 指數 |
| K | StrikePrice(履約價) |
| R | RisklessRate(無風險利率) |
| T | RemainMaturity(剩餘天數) |
| sigma | sigma |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |

在「[SKQuoteLib\_Delta](#_SKQuoteLib_Delta)」範例程式碼。

#### SKQuoteLib\_Vega

|  |  |  |
| --- | --- | --- |
| 輸入  1. S 台指期成交價  2. K StrikePrice(履約價)  3. R RisklessRate(無風險利率)  4. T RemainMaturity(剩餘天數)  5. v 波動率=sigma  得到Vega值 | | |
| **宣告** | Long SKQuoteLib\_Vega([[in] DOUBLE S, [in] DOUBLE K, [in] DOUBLE R, [in] DOUBLE T, [in] DOUBLE sigma, [out] DOUBLE\* dVega); | |
| **參數** | S | 台指期成交價 |
| K | StrikePrice(履約價) |
| R | RisklessRate(無風險利率) |
| T | RemainMaturity(剩餘天數) |
| sigma | V波動率=sigma |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |

在「[SKQuoteLib\_Delta](#_SKQuoteLib_Delta)」範例程式碼。

#### SKQuoteLib\_Theta

|  |  |  |
| --- | --- | --- |
| 輸入  1. nCallPut 商品買賣權別(Call: 0,Put: 1)  2. S 台指期成交價  3. K StrikePrice(履約價)  4. R RisklessRate(無風險利率)  5. T RemainMaturity(剩餘天數)  6. v 波動率=sigma  得到Theta值 | | |
| **宣告** | Long SKQuoteLib\_Theta(([[in] SHORT nCallPut, [in] DOUBLE S, [in] DOUBLE K, [in] DOUBLE R, [in] DOUBLE T, [in] DOUBLE v, [out] DOUBLE\* dTheta); | |
| **參數** | nCallPut | 商品買賣權別( Call: 0 , Put: 1) |
| S | 台指期成交價 |
| K | StrikePrice(履約價) |
| R | RisklessRate(無風險利率) |
| T | RemainMaturity(剩餘天數) |
| sigma | V波動率=sigma |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |

在「[SKQuoteLib\_Delta](#_SKQuoteLib_Delta)」範例程式碼。

#### SKQuoteLib\_Rho

|  |  |  |
| --- | --- | --- |
| 輸入  1. nCallPut 商品買賣權別(Call: 0, Put: 1)  2. S 台指期成交價  3. K StrikePrice(履約價)  4. R RisklessRate(無風險利率)  5. T RemainMaturity(剩餘天數)  6. v 波動率=sigma  得到Rho值 | | |
| **宣告** | Long SKQuoteLib\_Rho(([[in] SHORT nCallPut, [in] DOUBLE S, [in] DOUBLE K, [in] DOUBLE R, [in] DOUBLE T, [in] DOUBLE v, [out] DOUBLE\* dRho); | |
| **參數** | nCallPut | 商品買賣權別( Call: 0 , Put: 1) |
| S | 指數 |
| K | StrikePrice(履約價) |
| R | RisklessRate(無風險利率) |
| T | RemainMaturity(剩餘天數) |
| sigma | V波動率=sigma |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |

在「[SKQuoteLib\_Delta](#_SKQuoteLib_Delta)」範例程式碼。

### 交易資訊(期選)

#### SKQuoteLib\_RequestFutureTradeInfo

|  |  |  |
| --- | --- | --- |
| 取得報價函式庫註冊接收**期貨**商品的交易資訊。 | | |
| **宣告** | Long SKQuoteLib\_RequestFutureTradeInfo([in] SHORT\* psPageNo, [in] BSTR bstrStockNo); | |
| **參數** | psPageNo | 每一次送出要求的動作都需要向報價伺服器指定一個  特定的 Page 編號。  (請參考[SKQuoteLib\_RequestStocks](#_SKQuoteLib_RequestStocks)觀念說明)。 |
| bstrStockNo | 索取的商品代號，**一個Page僅能索取一檔**。 |
|  |  |
| **相關通知事件** | [OnNotifyFutureTradeInfoLONG](#_OnNotifyFutureTradeInfoLONG) | |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。  若未簽署期貨API下單同意書，將無法查詢。（錯誤代碼3031） | |

|  |
| --- |
| Python範例程式碼\_Quote.py類別TradeForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # textBoxSKQuoteLib\_RequestFutureTradeInfopsPageNo          tk.Label(self, text = "Page").grid(row=0, column=1)              #輸入框          self.textBoxSKQuoteLib\_RequestFutureTradeInfopsPageNo = tk.Entry(self)          self.textBoxSKQuoteLib\_RequestFutureTradeInfopsPageNo.grid(row=0, column=2)          global textBoxSKQuoteLib\_RequestFutureTradeInfopsPageNo          textBoxSKQuoteLib\_RequestFutureTradeInfopsPageNo = self.textBoxSKQuoteLib\_RequestFutureTradeInfopsPageNo            # textBoxSKQuoteLib\_RequestFutureTradeInfobstrStockNo          tk.Label(self, text = "請輸入商品代號(僅1檔)").grid(row=0, column=3)              #輸入框          self.textBoxSKQuoteLib\_RequestFutureTradeInfobstrStockNo = tk.Entry(self)          self.textBoxSKQuoteLib\_RequestFutureTradeInfobstrStockNo.grid(row=0, column=4)          global textBoxSKQuoteLib\_RequestFutureTradeInfobstrStockNo          textBoxSKQuoteLib\_RequestFutureTradeInfobstrStockNo = self.textBoxSKQuoteLib\_RequestFutureTradeInfobstrStockNo            # buttonSKQuoteLib\_RequestFutureTradeInfo          self.buttonSKQuoteLib\_RequestFutureTradeInfo = tk.Button(self)          self.buttonSKQuoteLib\_RequestFutureTradeInfo["text"] = "期貨"          self.buttonSKQuoteLib\_RequestFutureTradeInfo["command"] = self.buttonSKQuoteLib\_RequestFutureTradeInfo\_Click          self.buttonSKQuoteLib\_RequestFutureTradeInfo.grid(row=1, column=1) |

|  |
| --- |
| 全域變數 |
| # 群益API元件導入Python code內用的物件宣告  m\_pSKCenter = comtypes.client.CreateObject(sk.SKCenterLib,interface=sk.ISKCenterLib)  m\_pSKQuote = comtypes.client.CreateObject(sk.SKQuoteLib,interface=sk.ISKQuoteLib) |

|  |
| --- |
| Current Block(放置於def createWidgets(self): 的同一層) |
| def buttonSKQuoteLib\_RequestFutureTradeInfo\_Click(self):          psPageNo = ctypes.c\_short(int(textBoxSKQuoteLib\_RequestFutureTradeInfopsPageNo.get()))          # 取得報價函式庫註冊接收期貨商品的交易資訊          nCode= m\_pSKQuote.SKQuoteLib\_RequestFutureTradeInfo(psPageNo, textBoxSKQuoteLib\_RequestFutureTradeInfobstrStockNo.get())          msg = "【SKQuoteLib\_RequestFutureTradeInfo】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

#### SKQuoteLib\_GetStrikePrices

|  |  |  |
| --- | --- | --- |
| 取得報價函式庫選擇權交易商品資訊。 | | |
| **宣告** | Long SKQuoteLib\_GetStrikePrices() | |
| **參數** | 無 |  |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。  若未簽署期貨API下單同意書，將無法查詢。（錯誤代碼3031） | |
| **相關通知事件** | [OnNotifyStrikePrices](#_OnNotifyStrikePrices) | |

|  |
| --- |
| Python範例程式碼\_Quote.py類別TradeForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # buttonSKQuoteLib\_GetStrikePrices          self.buttonSKQuoteLib\_GetStrikePrices = tk.Button(self)          self.buttonSKQuoteLib\_GetStrikePrices["text"] = "選擇權"          self.buttonSKQuoteLib\_GetStrikePrices["command"] = self.buttonSKQuoteLib\_GetStrikePrices\_Click          self.buttonSKQuoteLib\_GetStrikePrices.grid(row=2, column=1) |

|  |
| --- |
| 全域變數 |
| # 群益API元件導入Python code內用的物件宣告  m\_pSKCenter = comtypes.client.CreateObject(sk.SKCenterLib,interface=sk.ISKCenterLib)  m\_pSKQuote = comtypes.client.CreateObject(sk.SKQuoteLib,interface=sk.ISKQuoteLib) |

|  |
| --- |
| Current Block(放置於def createWidgets(self): 的同一層) |
| def buttonSKQuoteLib\_GetStrikePrices\_Click(self):          # 取得報價函式庫選擇權交易商品資訊          nCode= m\_pSKQuote.SKQuoteLib\_GetStrikePrices()          msg = "【SKQuoteLib\_GetStrikePrices】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

# 事件

## 連線

### OnConnection

|  |  |  |
| --- | --- | --- |
| 接收連線狀態。 | | |
| **宣告** | void OnConnection([in] LONG nKind, [in] LONG nCode); | |
| **參數** | nKind | SK\_SUBJECT\_CONNECTION\_CONNECTED 表示為連線事件。  SK\_SUBJECT\_CONNECTION\_DISCONNECT 表示為斷線事件。  SK\_SUBJECT\_CONNECTION\_STOCKS\_READY 表示為商品基本資料已下載完成。  其他錯誤代碼，可參考代碼定義表 |
| nCode | 表示事件所獲得的代碼，代碼為0值即表示執行正確，非0表示例外事件。 |
| **備註** | 避免在此處直接進行[EnterMonitorLONG](#_SKQuoteLib_EnterMonitorLONG)、[LeaveMonitor](#_SKQuoteLib_LeaveMonitor)、[RequestStocks](#_SKQuoteLib_RequestStocks) and [RequestTicks](#_SKQuoteLib_RequestTicks) 等報價連線、斷線、訂閱，若全市場商品未下載完成，即無法進行商品訂閱。 | |

|  |
| --- |
| Python範例程式碼\_Quote.py |
| UI程式碼(放置於 def createWidgets(self): 的下一層)/UI示意圖 |
| # richTextBoxMessage          self.richTextBoxMessage = tk.Listbox(self, height=5)          self.richTextBoxMessage.grid(column = 0, row = 10, columnspan=5, sticky = tk.E + tk.W)          global richTextBoxMessage          richTextBoxMessage = self.richTextBoxMessage |

|  |
| --- |
| 事件(可放在事件SKQuoteLibEvent註冊物件底下一層) |
| def OnConnection(self, nKind, nCode):          msg = "【OnConnection】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nKind) + "\_" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMessage.insert('end', msg + "\n")          richTextBoxMessage.see('end') |

### OnNotifyServerTime

|  |  |  |
| --- | --- | --- |
| 事件回傳查詢主機時間的結果。 | | |
| **宣告** | void OnNotifyServerTime([in] SHORT sHour, [in] SHORT sMinute, [in] SHORT sSecond, [in] LONG nTotal); | |
| **參數** | sHour | 時 |
| sMinute | 分 |
| sSecond | 秒 |
| nTotal | 總秒數 |
| **備註** |  | |

|  |
| --- |
| Python範例程式碼\_Quote.py |
| UI程式碼(放置於 def createWidgets(self): 的下一層)/UI示意圖 |
| # richTextBoxMessage          self.richTextBoxMessage = tk.Listbox(self, height=5)          self.richTextBoxMessage.grid(column = 0, row = 10, columnspan=5, sticky = tk.E + tk.W)          global richTextBoxMessage          richTextBoxMessage = self.richTextBoxMessage |

|  |
| --- |
| 事件(可放在事件SKQuoteLibEvent註冊物件底下一層) |
| def OnNotifyServerTime(self, sHour, sMinute, sSecond, nTotal):          msg = "【OnNotifyServerTime】" + str(sHour) + ":" + str(sMinute) + ":" + str(sSecond) + "總秒數:" + str(nTotal)          richTextBoxMessage.insert('end', msg + "\n")          richTextBoxMessage.see('end') |

## 大盤資訊

### OnNotifyMarketTot

|  |  |  |
| --- | --- | --- |
| 透過呼叫[SKQuoteLib\_GetMarketBuySellUpDown](#_SKQuoteLib_GetMarketBuySellUpDown)後，  事件回傳大盤成交張筆資料。 | | |
| **宣告** | void OnNotifyMarketTot([in] SHORT sMarketNo, [in] SHORT sPtr, [in] LONG nTime, [in] LONG nTotv, [in] LONG nTots, [in] LONG nTotc); | |
| **參數** | sMarketNo | 市場別代號 (0x00 上市, 0x01 上櫃) |
| sPtr | 目前第幾筆資料。  可以根據此 index 取得該筆資訊。 |
| nTime | 大盤成交時間。Ex. lTime = 92000 ，為早上九點20分(09:20)。 |
| nTotv | 大盤成交值(億)。Ex. lTotv = 88542，為885.42 (億)。 |
| nTots | 大盤成交張數。 |
| nTotc | 大盤成交筆數。 |
| **備註** |  | |

|  |
| --- |
| Python範例程式碼\_Quote.py |
| UI程式碼(放置於 def createWidgets(self): 的下一層)/UI示意圖 |
| # richTextBoxMessage          self.richTextBoxMessage = tk.Listbox(self, height=5)          self.richTextBoxMessage.grid(column = 0, row = 10, columnspan=5, sticky = tk.E + tk.W)          global richTextBoxMessage          richTextBoxMessage = self.richTextBoxMessage |

|  |
| --- |
| 事件(可放在事件SKQuoteLibEvent註冊物件底下一層) |
| def OnNotifyMarketTot(self, sMarketNo, sPtr, nTime, nTotv, nTots, nTotc):          if (sMarketNo == 0): #上市              labelMarketTotPtr.config(text = str(sPtr))              labelMarketTotTime.config(text = str(nTime / 10000) + "點" + str((nTime % 10000) / 100) + "分" + str(nTime % 100) + "秒")              labelnTotv.config(text = str(nTotv / 100.00))              labelnTots.config(text = str(nTots))              labelnTotc.config(text = str(nTotc))          else: #上櫃              labelMarketTotPtr2.config(text = str(sPtr))              labelMarketTotTime2.config(text = str(nTime / 10000) + "點" + str((nTime % 10000) / 100) + "分" + str(nTime % 100) + "秒")              labelnTotv2.config(text = str(nTotv / 100.00))              labelnTots2.config(text = str(nTots))              labelnTotc2.config(text = str(nTotc))      def OnNotifyMarketBuySell(self, sMarketNo, sPtr, nTime, nBc, nSc, nBs, nSs):          if (sMarketNo == 0): #上市              labelnBs.config(text = str(nBs))              labelnSs.config(text = str(nSs))              labelnBc.config(text = str(nBc))              labelnSc.config(text = str(nSc))          else: #上櫃              labelnBs2.config(text = str(nBs))              labelnSs2.config(text = str(nSs))              labelnBc2.config(text = str(nBc))              labelnSc2.config(text = str(nSc))      def OnNotifyMarketHighLowNoWarrant(self, sMarketNo, sPtr, nTime, nUp, nDown, nHigh, nLow, nNoChange, nUpNoW, nDownNoW, nHighNoW, nLowNoW, nNoChangeNoW):          if (sMarketNo == 0): #上市              labelnUp.config(text = str(nUp))              labelnDown.config(text = str(nDown))              labelnHigh.config(text = str(nHigh))              labelnLow.config(text = str(nLow))              labelnNoChange.config(text = str(nNoChange))              labelnUpNoW.config(text = str(nUpNoW))              labelnDownNoW.config(text = str(nDownNoW))              labelnHighNoW.config(text = str(nHighNoW))              labelnLowNoW.config(text = str(nLowNoW))              labelnNoChangeNoW.config(text = str(nNoChangeNoW))          else: #上櫃              labelnUp2.config(text = str(nUp))              labelnDown2.config(text = str(nDown))              labelnHigh2.config(text = str(nHigh))              labelnLow2.config(text = str(nLow))              labelnNoChange2.config(text = str(nNoChange))              labelnUpNoW2.config(text = str(nUpNoW))              labelnDownNoW2.config(text = str(nDownNoW))              labelnHighNoW2.config(text = str(nHighNoW))              labelnLowNoW2.config(text = str(nLowNoW))              labelnNoChangeNoW2.config(text = str(nNoChangeNoW)) |

#### OnNotifyMarketBuySell

|  |  |  |
| --- | --- | --- |
| 透過呼叫[SKQuoteLib\_GetMarketBuySellUpDown](#_SKQuoteLib_GetMarketBuySellUpDown)後，  事件回傳大盤成交買賣張筆數資料。 | | |
| **宣告** | void OnNotifyMarketBuySell([in] SHORT sMarketNo, [in] SHORT sPtr, [in] LONG nTime, [in] LONG nBc, [in] LONG nSc, [in] LONG nBs, [in] LONG nSs); | |
| **參數** | sMarketNo | 市場別代號 (0x00 上市, 0x01 上櫃) |
| sPtr | 目前第幾筆資料。  可以根據此 index 取得該筆資訊。 |
| nTime | 大盤成交時間。Ex. lTime = 92000 ，為早上九點20分(09:20)。 |
| nBc | 大盤成交買進筆數 |
| nSc | 大盤成交賣出筆數 |
| nBs | 大盤成交買進張數 |
| nSs | 大盤成交賣出張數 |
| **備註** |  | |

#### OnNotifyMarketHighLowNoWarrant

|  |  |  |
| --- | --- | --- |
| 透過呼叫[SKQuoteLib\_GetMarketBuySellUpDown](#_SKQuoteLib_GetMarketBuySellUpDown)後，  事件回傳大盤成交上漲下跌家數資料(包含『含權證家數』、 『不含權證家數』)。 | | |
| **宣告** | void OnNotifyMarketHighLowNoWarrant([in] SHORT sMarketNo, [in] LONG nPtr, [in] LONG nTime, [in] LONG nUp, [in] LONG nDown, [in] LONG nHigh, [in] LONG nLow, [in] LONG nUnChange, [in] LONG nUpNoW, [in] LONG nDownNoW, [in] LONG nHighNoW, [in] LONG nLowNoW, [in] LONG nUnChangeNoW); | |
| **參數** | sMarketNo | 市場別代號 (0x00 上市, 0x01 上櫃) |
| sPtr | 目前第幾筆資料。  可以根據此 index 取得該筆資訊。 |
| nTime | 大盤成交時間。Ex. lTime = 92000 ，為早上九點20分(09:20)。 |
| nUp | 大盤成交上漲家數 |
| nDown | 大盤成交下跌家數 |
| nHigh | 大盤成交漲停家數 |
| nLow | 大盤成交跌停家數 |
| nNoChange | 大盤平盤家數 |
| nUpNoW | (不含權證)大盤成交上漲家數 |
| nDownNoW | (不含權證)大盤成交下跌家數 |
| nHighNoW | (不含權證)大盤成交漲停家數 |
| nLowNoW | (不含權證)大盤成交跌停家數 |
| nUnChangeNoW | (不含權證)大盤平盤家數 |
| **備註** |  | |

##### OnNotifyMarketHighLow

|  |  |  |
| --- | --- | --- |
| 透過呼叫[SKQuoteLib\_GetMarketBuySellUpDown](#_SKQuoteLib_GetMarketBuySellUpDown)後，  事件回傳大盤成交上漲下跌家數資料。 | | |
| **宣告** | void OnNotifyMarketHighLow([in] SHORT sMarketNo, [in] SHORT sPtr, [in] LONG nTime, [in] SHORT sUp, [in] SHORT sDown, [in] SHORT sHigh, [in] SHORT sLow, [in] SHORT sNoChange); | |
| **參數** | sMarketNo | 市場別代號 (0x00 上市, 0x01 上櫃) |
| sPtr | 目前第幾筆資料。  可以根據此 index 取得該筆資訊。 |
| nTime | 大盤成交時間。Ex. lTime = 92000 ，為早上九點20分(09:20)。 |
| sUp | 大盤成交上漲家數 |
| sDown | 大盤成交下跌家數 |
| sHigh | 大盤成交漲停家數 |
| sLow | 大盤成交跌停家數 |
| sNoChange | 大盤平盤家數 |
| **備註** |  | |

## 商品清單

### OnNotifyCommodityListWithTypeNo

|  |  |  |
| --- | --- | --- |
| 事件回傳指定國內市場－含類別代碼及類別中文名稱之商品清單。 | | |
| **宣告** | void OnNotifyCommodityListWithTypeNo([in] SHORT sMarketNo, [in] BSTR bstrCommodityData); | |
| **參數** | sMarketNo | 市場別代號 |
| bstrCommodityData | **開頭含類別代碼，**格式為「%」、類別代碼「%」類別中文名稱，**結尾為「%」。**  EX: %1%水泥%   |  |  | | --- | --- | | 0 | **類別代碼**。例如，1、2。 | | 1 | **類別中文名稱**。例如，水泥、食品等。 |   接著為商品資料(商品代碼及商品中文名稱)  以上市0 – 水泥類股為例，以「,」分隔每一欄，欄位依序為：  [商品代碼],[商名名稱],[最後交易日],[交易所商品代碼];   |  |  | | --- | --- | | 0 | Product商品代碼。  例如，1101。 | | 1 | Name中文名稱。例如，台泥。 | | 2 | LastTradeDay 最後交易日，目前僅提供期貨及選擇權。  例如，20180719  當為0時代表沒有最後交易日或為不可交易商品。 | | 3 | OrderID 交易所商品代碼，目前僅提供期貨及選擇權。  例如，TXFI9。 |   另以「；」做為下一筆資料分隔，  而各類別結尾（水泥、食品、塑膠等）則以「換行」做分隔。  以上市商品-代碼0  **%1%水泥%**1101,台泥,0,;1102,亞泥,0,;1103,嘉泥,0,;1104,環泥,0,;1108,幸福,0,;1109,信大,0,;1110,東泥,0,;  **%2%食品%**1201,味全,0,;1203,味王,0,; 1210,大成,0,; 1213,大飲,0,;1215,卜蜂,0,;….; |
| **備註** | 當全部資料已經全部回傳完畢，將回傳一筆以「##」開頭的內容，表示查詢結束。 | |

|  |
| --- |
| Python範例程式碼\_Quote.py |
| UI程式碼(放置於 def createWidgets(self): 的下一層)/UI示意圖 |
| # richTextBoxMessage          self.richTextBoxMessage = tk.Listbox(self, height=5)          self.richTextBoxMessage.grid(column = 0, row = 10, columnspan=5, sticky = tk.E + tk.W)          global richTextBoxMessage          richTextBoxMessage = self.richTextBoxMessage |

|  |
| --- |
| 事件(可放在事件SKQuoteLibEvent註冊物件底下一層) |
| def OnNotifyCommodityListWithTypeNo(self, sMarketNo, bstrCommodityData):          msg = "【OnNotifyCommodityListWithTypeNo】" + str(sMarketNo) + "\_" + bstrCommodityData          richTextBoxMessage.insert('end', msg + "\n")          richTextBoxMessage.see('end') |

#### OnNotifyStockList

|  |  |  |
| --- | --- | --- |
| 事件回傳指定國內市場－各類股商品清單。 | | |
| **宣告** | void OnNotifyStockList([in] SHORT sMarketNo, [in] BSTR bstrStockData); | |
| **參數** | sMarketNo | 市場別代號 |
| bstrStockData | 商品資料(商品代碼及商品中文名稱)  以上市0 – 水泥類股為例，以「,」分隔每一欄，欄位依序為：  [商品代碼],[商名名稱],[最後交易日],[交易所商品代碼];   |  |  | | --- | --- | | 0 | Product商品代碼。  例如，1101。 | | 1 | Name中文名稱。例如，台泥。 | | 2 | LastTradeDay 最後交易日，目前僅提供期貨及選擇權  例如，20180719  當為0時代表沒有最後交易日或為不可交易商品。 | | 3 | OrderID 交易所商品代碼，目前僅提供期貨及選擇權。  例如，TXFI9。 |   另以「；」做為下一筆資料分隔，  而各類股結尾（水泥、食品、塑膠等）則以「換行」做分隔。  1101,台泥,0,;1102,亞泥,0,;1103,嘉泥,0,;1104,環泥,0,;1108,幸福,0,;1109,信大,0,;1110,東泥,0,;  1201,味全,0,;1203,味王,0,; 1210,大成,0,; 1213,大飲,0,;1215,卜蜂,0,;….; |
| **備註** | 當全部資料已經全部回傳完畢，將回傳一筆以「##」開頭的內容，表示查詢結束。 | |

## 即時報價

### OnNotifyQuoteLONG

|  |  |  |
| --- | --- | --- |
| (LONG index)當有索取的個股報價異動時，將透過此事件通知應用程式處理。 | | |
| **宣告** | void OnNotifyQuoteLONG([in] SHORT sMarketNo, [in] LONG nIndex); | |
| **參數** | sMarketNo | 報價有異動的商品市場別。 |
| nIndex | 系統所編的索引代碼。 |
| **備註** | 透過以上兩個參數，可以再使用[SKQuoteLib\_GetStockByIndexLONG](#_SKQuoteLib_GetStockByIndexLONG) 取出報價的內容。  \*開發人員可於OnUpDateDataRow(SKSTOCK pStock) 取得Stock揭示值，另外自行接手處理  \*依國內或海外市場，可能需搭配MarketNo  ＊須使用[SKQuoteLib\_EnterMonitorLONG](#_SKQuoteLib_EnterMonitorLONG)登入，該事件才會被觸發。  一張含有 文字, 螢幕擷取畫面, 字型, 電子藍 的圖片  自動產生的描述 | |

|  |
| --- |
| Python範例程式碼\_Quote.py |
| UI程式碼(放置於 def createWidgets(self): 的下一層)/UI示意圖 |
| # richTextBoxMessage          self.richTextBoxMessage = tk.Listbox(self, height=5)          self.richTextBoxMessage.grid(column = 0, row = 10, columnspan=5, sticky = tk.E + tk.W)          global richTextBoxMessage          richTextBoxMessage = self.richTextBoxMessage |

|  |
| --- |
| 事件(可放在事件SKQuoteLibEvent註冊物件底下一層) |
| def OnNotifyQuoteLONG(self, sMarketNo, nIndex):          pSKStock = sk.SKSTOCKLONG()          pSKStock, nCode= m\_pSKQuote.SKQuoteLib\_GetStockByIndexLONG(sMarketNo, nIndex, pSKStock)          if (pSKStock.nBid == m\_pSKQuote.SKQuoteLib\_GetMarketPriceTS()):              nBidValue = "市價"          else:              nBidValue = pSKStock.nBid / 100.0          if (pSKStock.nBid == m\_pSKQuote.SKQuoteLib\_GetMarketPriceTS()):              nAskValue = "市價"          else:              nAskValue = pSKStock.nAsk / 100.0          msg = ("【OnNotifyQuoteLONG】" + " 商品代碼" + str(pSKStock.bstrStockNo) +          " 名稱" + str(pSKStock.bstrStockName) +          " 開盤價" + str(pSKStock.nOpen / 100.0) +          " 成交價" + str(pSKStock.nClose / 100.0) +          " 最高" + str(pSKStock.nHigh / 100.0) +          " 最低" + str(pSKStock.nLow / 100.0) +          " 漲停價" + str(pSKStock.nUp / 100.0) +          " 跌停價" + str(pSKStock.nDown / 100.0) +          " 買盤量(外盤)" + str(pSKStock.nTBc) +          " 賣盤量(內盤)" + str(pSKStock.nTAc) +          " 總量" + str(pSKStock.nTQty) +          " 昨收(參考價)" + str(pSKStock.nRef / 100.0) +          " 昨量" + str(pSKStock.nYQty) +          " 買價" + str(nBidValue) +          " 買量" + str(pSKStock.nBc) +          " 賣價" + str(nAskValue) +          " 賣量" + str(pSKStock.nAc) )            richTextBoxMessage.insert('end', msg + "\n")          richTextBoxMessage.see('end') |

### OnNotifyOddLotSpreadDeal

|  |  |  |
| --- | --- | --- |
| 事件回傳證券市場－整零價差即時行情 | | |
| **宣告** | void OnNotifyOddLotSpreadDeal([in] SHORT sMarketNo, [in] BSTR bstrStockNo, [in] LONG nDealPrice, [in] LONG nDigit); | |
| **參數** | sMarketNo | 市場別代號 |
| bstrStockNo | 商品代碼。 |
| nDealPrice | 整零成交價差(負數則含-負號) |
| nDigit | 小數位數 |
| **備註** | 目前只提供證券市場；實際價差，需再透過小數位數自行計算。  \*公式:  整零價差=整股成交價-零股成交價。  若整股當日無成交價，以昨收價計算。  若零股當日無成交價，不計算整零價差。 | |

V2.13.46 修正文件宣告內容敘述 (移除[in] CHAR cSignDeal)

|  |
| --- |
| Python範例程式碼\_Quote.py |
| UI程式碼(放置於 def createWidgets(self): 的下一層)/UI示意圖 |
| # richTextBoxMessage          self.richTextBoxMessage = tk.Listbox(self, height=5)          self.richTextBoxMessage.grid(column = 0, row = 10, columnspan=5, sticky = tk.E + tk.W)          global richTextBoxMessage          richTextBoxMessage = self.richTextBoxMessage |

|  |
| --- |
| 事件(可放在事件SKQuoteLibEvent註冊物件底下一層) |
| def OnNotifyOddLotSpreadDeal(self, sMarketNo, bstrStockNo, nDealPrice, sDigit):          while sDigit != 0:              sDigit -=1              nDealPrice /= 10          msg = "【OnNotifyOddLotSpreadDeal】" + " 市場別代號" + str(sMarketNo)+ " 商品代碼" + str(bstrStockNo)+ " 整零成交價差(負數則含-負號)" + str(nDealPrice)          richTextBoxMessage.insert('end', msg + "\n")          richTextBoxMessage.see('end') |

## 五檔&成交明細

### OnNotifyBest5LONG

|  |  |  |
| --- | --- | --- |
| (LONG index)當有索取的個股五檔價格有所異動，即透過向此註冊的事件進行處理。此函式會回傳所異動的個股五檔價格。 | | |
| **事件宣告** | void OnNotifyBest5LONG([in] SHORT sMarketNo, [in] LONG nStockidx, [in] LONG nBestBid1, [in] LONG nBestBidQty1, [in] LONG nBestBid2, [in] LONG nBestBidQty2, [in] LONG nBestBid3, [in] LONG nBestBidQty3, [in] LONG nBestBid4, [in] LONG nBestBidQty4, [in] LONG nBestBid5, [in] LONG nBestBidQty5, [in] LONG nExtendBid, [in] LONG nExtendBidQty, [in] LONG nBestAsk1, [in] LONG nBestAskQty1 , [in] LONG nBestAsk2, [in] LONG nBestAskQty2, [in] LONG nBestAsk3, [in] LONG nBestAskQty3, [in] LONG nBestAsk4, [in] LONG nBestAskQty4, [in] LONG nBestAsk5, [in] LONG nBestAskQty5, [in] LONG nExtendAsk, [in] LONG nExtendAskQty, [in] LONG nSimulate); | |
| **參數** | sMarketNo | 市埸代碼。 |
| nStockidx | 系統所編的索引代碼。 |
| nBestBid1 | 第一檔買價。 |
| nBestBidQty1 | 第一檔買量。 |
| nBestBid2 | 第二檔買價。 |
| nBestBidQty2 | 第二檔買量。 |
| nBestBid3 | 第三檔買價。 |
| nBestBidQty3 | 第三檔買量。 |
| nBestBid4 | 第四檔買價。 |
| nBestBidQty4 | 第四檔買量。 |
| nBestBid5 | 第五檔買價。 |
| nBestBidQty5 | 第五檔買量。 |
| nExtendBid | 衍生一檔買價。 |
| nExtendBidQty | 衍生一檔買量。 |
| nBestAsk1 | 第一檔賣價。 |
| nBestAskQty1 | 第一檔賣量。 |
| nBestAsk2 | 第二檔賣價。 |
| nBestAskQty2 | 第二檔賣量。 |
| nBestAsk3 | 第三檔賣價。 |
| nBestAskQty3 | 第三檔賣量。 |
| nBestAsk4 | 第四檔賣價。 |
| nBestAskQty4 | 第四檔賣量。 |
| nBestAsk5 | 第五檔賣價。 |
| nBestAskQty5 | 第五檔賣量。 |
| nExtendAsk | 衍生一檔賣價。 |
| nExtendAskQty | 衍生一檔賣量。 |
| nSimulate | 0:一般揭示 1:試算揭示 |
| **相關函式** | [SKQuote\_RequestTicks](#_SKQuoteLib_RequestStocks) 相關通知事件。 | |
| **備註** | 未進行揭示處理，開發人員需自行接手處理(判斷接收的五檔報價為一般或試算揭示)。  ＊盤中零股: 因盤中仍會進入試撮，請留意是否收取『試撮資料』，即試算揭示。  ＊須使用[SKQuoteLib\_EnterMonitorLONG](#_SKQuoteLib_EnterMonitorLONG)登入，該事件才會被觸發。 | |

|  |
| --- |
| Python範例程式碼\_Quote.py |
| UI程式碼(放置於 def createWidgets(self): 的下一層)/UI示意圖 |
| # richTextBoxMessage          self.richTextBoxMessage = tk.Listbox(self, height=5)          self.richTextBoxMessage.grid(column = 0, row = 10, columnspan=5, sticky = tk.E + tk.W)          global richTextBoxMessage          richTextBoxMessage = self.richTextBoxMessage |

|  |
| --- |
| 事件(可放在事件SKQuoteLibEvent註冊物件底下一層) |
| def OnNotifyBest5LONG(self, sMarketNo, nStockidx, nBestBid1, nBestBidQty1, nBestBid2,  nBestBidQty2, nBestBid3, nBestBidQty3, nBestBid4, nBestBidQty4, nBestBid5, nBestBidQty5, nExtendBid, nExtendBidQty, nBestAsk1, nBestAskQty1, nBestAsk2, nBestAskQty2, nBestAsk3, nBestAskQty3, nBestAsk4, nBestAskQty4, nBestAsk5, nBestAskQty5, nExtendAsk, nExtendAskQty, nSimulate):          if (nSimulate == 0):              labelnSimulate = "一般揭示"          elif (nSimulate == 1):              labelnSimulate = "試算揭示"          msg = ("【OnNotifyBest5LONG】" + " 揭示" + labelnSimulate +                 " 買量一檔" + str(nBestBidQty1)+                 " 買價一檔" + str(nBestBid1 / 100.0)+                  " 賣價一檔" + str(nBestAsk1 / 100.0)+                 " 賣量一檔" + str(nBestAskQty1)+                  " 買量二檔" + str(nBestBidQty2)+                 " 買價二檔" + str(nBestBid2 / 100.0)+                  " 賣價二檔" + str(nBestAsk2 / 100.0)+                 " 賣量二檔" + str(nBestAskQty2)+                  " 買量三檔" + str(nBestBidQty3)+                 " 買價三檔" + str(nBestBid3 / 100.0)+                  " 賣價三檔" + str(nBestAsk3 / 100.0)+                 " 賣量三檔" + str(nBestAskQty3)+                  " 買量四檔" + str(nBestBidQty4)+                 " 買價四檔" + str(nBestBid4 / 100.0)+                  " 賣價四檔" + str(nBestAsk4 / 100.0)+                 " 賣量四檔" + str(nBestAskQty4)+                  " 買量五檔" + str(nBestBidQty5)+                 " 買價五檔" + str(nBestBid5 / 100.0)+                  " 賣價五檔" + str(nBestAsk5 / 100.0)+                 " 賣量五檔" + str(nBestAskQty5) )          richTextBoxMessage.insert('end', msg + "\n")          richTextBoxMessage.see('end') |

### OnNotifyBest5Emerging

|  |  |  |
| --- | --- | --- |
| (LONG index)當有索取的個股五檔價格有所異動，即透過向此註冊的事件進行處理。此函式會回傳所異動的個股五檔價格。  **※**此事件索取的興櫃商品五檔單位為”股” | | |
| **事件宣告** | void OnNotifyBest5Emerging([in] SHORT sMarketNo, [in] LONG nStockidx, [in] LONG nBestBid1, [in] LONG nBestBidQty1, [in] LONG nBestBid2, [in] LONG nBestBidQty2, [in] LONG nBestBid3, [in] LONG nBestBidQty3, [in] LONG nBestBid4, [in] LONG nBestBidQty4, [in] LONG nBestBid5, [in] LONG nBestBidQty5, [in] LONG nExtendBid, [in] LONG nExtendBidQty, [in] LONG nBestAsk1, [in] LONG nBestAskQty1 , [in] LONG nBestAsk2, [in] LONG nBestAskQty2, [in] LONG nBestAsk3, [in] LONG nBestAskQty3, [in] LONG nBestAsk4, [in] LONG nBestAskQty4, [in] LONG nBestAsk5, [in] LONG nBestAskQty5, [in] LONG nExtendAsk, [in] LONG nExtendAskQty, [in] LONG nSimulate); | |
| **參數** | sMarketNo | 市埸代碼。 |
| nStockidx | 系統所編的索引代碼。 |
| nBestBid1 | 第一檔買價。 |
| nBestBidQty1 | 第一檔買量。 |
| nBestBid2 | 第二檔買價。 |
| nBestBidQty2 | 第二檔買量。 |
| nBestBid3 | 第三檔買價。 |
| nBestBidQty3 | 第三檔買量。 |
| nBestBid4 | 第四檔買價。 |
| nBestBidQty4 | 第四檔買量。 |
| nBestBid5 | 第五檔買價。 |
| nBestBidQty5 | 第五檔買量。 |
| nExtendBid | 衍生一檔買價。 |
| nExtendBidQty | 衍生一檔買量。 |
| nBestAsk1 | 第一檔賣價。 |
| nBestAskQty1 | 第一檔賣量。 |
| nBestAsk2 | 第二檔賣價。 |
| nBestAskQty2 | 第二檔賣量。 |
| nBestAsk3 | 第三檔賣價。 |
| nBestAskQty3 | 第三檔賣量。 |
| nBestAsk4 | 第四檔賣價。 |
| nBestAskQty4 | 第四檔賣量。 |
| nBestAsk5 | 第五檔賣價。 |
| nBestAskQty5 | 第五檔賣量。 |
| nExtendAsk | 衍生一檔賣價。 |
| nExtendAskQty | 衍生一檔賣量。 |
| nSimulate | 0:一般揭示 1:試算揭示 |
| **相關函式** | [SKQuote\_RequestTicks](#_SKQuoteLib_RequestTicks) 相關通知事件。 | |
| **備註** | 未進行揭示處理，開發人員需自行接手處理(判斷接收的五檔報價為一般或試算揭示)。  ＊盤中零股: 因盤中仍會進入試撮，請留意是否收取『試撮資料』，即試算揭示。  ＊須使用[SKQuoteLib\_EnterMonitorLONG](#_SKQuoteLib_EnterMonitorLONG)登入，該事件才會被觸發。 | |

### OnNotifyTicksLONG

|  |  |  |
| --- | --- | --- |
| (LONG index)當有索取的個股成交明細有所異動，即透過向此註冊事件回傳所異動的個股成交明細。 | | |
| **宣告** | void OnNotifyTicksLONG([in] SHORT sMarketNo, [in] LONG nIndex, [in] LONG nPtr, [in] LONG nDate, [in] LONG nTimehms, [in] LONG nTimemillismicros, [in] LONG nBid, [in] LONG nAsk, [in] LONG nClose, [in] LONG nQty, [in] LONG nSimulate); | |
| **參數** | sMarketNo | 報價有異動的商品市場別。 |
| nIndex | 系統所編的索引代碼。 |
| nPtr | 表示資料的位址(Key) |
| nDate | 交易日期。(YYYYMMDD) |
| nTimehms | 時間1。(時：分：秒) |
| nTimemillismicros | 時間2。(‘毫秒"微秒) |
| nBid | 買價。 |
| nAsk | 賣價。 |
| nClose | 成交價。 |
| nQty | 成交量。 |
| nSimulate | 0:一般揭示 1:試算揭示 |
| **備註** | * + 1. nTimehms提供單位：時分秒hh:mm:ss。     2. nTimemillismicros提供單位：毫秒微秒’millissecond’’micorsecond，預設提供毫秒微秒，開發人員自行決定是否顯示。(目前solace只提供證券商品)     3. 未進行揭示處理，開發人員需自行接手處理(判斷收的tick 為一般或試算揭示)。     4. T盤切換T+1盤，不保留前一盤資料。開發者需自行清除前一盤資料。     5. 避免在[SKQuoteLib\_OnNotifyTicksLONG](#_OnNotifyTicksLONG_1)通知事件裡進行[SKQuoteLib\_GetTickLONG](#_SKQuoteLib_GetTickLONG)()、[SKQuoteLib\_GetStockByIndexLONG](#_SKQuoteLib_GetStockByIndexLONG)(只可取得基本資料)   \*依國內或海外市場，可能需搭配MarketNo  ＊須使用[SKQuoteLib\_EnterMonitorLONG](#_SKQuoteLib_EnterMonitorLONG)登入，該事件才會被觸發。  一張含有 文字, 螢幕擷取畫面, 字型, 電子藍 的圖片  自動產生的描述 | |

|  |
| --- |
| Python範例程式碼\_Quote.py |
| UI程式碼(放置於 def createWidgets(self): 的下一層)/UI示意圖 |
| # richTextBoxMessage          self.richTextBoxMessage = tk.Listbox(self, height=5)          self.richTextBoxMessage.grid(column = 0, row = 10, columnspan=5, sticky = tk.E + tk.W)          global richTextBoxMessage          richTextBoxMessage = self.richTextBoxMessage |

|  |
| --- |
| 事件(可放在事件SKQuoteLibEvent註冊物件底下一層) |
| def OnNotifyTicksLONG(self, sMarketNo, nIndex, nPtr, nDate, nTimehms,  nTimemillismicros, nBid, nAsk, nClose, nQty, nSimulate):          msg = ("【OnNotifyTicksLONG】" + " 交易日期。(YYYYMMDD)" + str(nDate) +                 " 時間1。(時：分：秒)" + str(nTimehms)+                 " 時間2。(毫秒微秒)" + str(nTimemillismicros)+                  " 買價" + str(nBid / 100.0)+                 " 賣價" + str(nAsk / 100.0)+                  " 成交價" + str(nClose / 100.0)+                 " 成交量" + str(nQty) )          richTextBoxMessage.insert('end', msg + "\n")          richTextBoxMessage.see('end') |

#### OnNotifyHistoryTicksLONG

|  |  |  |
| --- | --- | --- |
| (LONG index)當**首次**索取個股成交明細，此事件會**回補當天Tick**。 | | |
| **宣告** | void OnNotifyHistoryTicksLONG([in] SHORT sMarketNo, [in] LONG nIndex, [in] LONG nPtr, [in] LONG nDate, [in] LONG nTimehms, [in] LONG nTimemillismicros, [in] LONG nBid, [in] LONG nAsk, [in] LONG nClose, [in] LONG nQty, [in] LONG nSimulate); | |
| **參數** | sMarketNo | 報價有異動的商品市場別。 |
| nIndex | 系統所編的索引代碼。 |
| nPtr | 表示資料的位址(Key) |
| nDate | 交易日期。(YYYYMMDD) |
| nTimehms | 時間1。(時：分：秒’毫秒"微秒) |
| nTimemillismicros | 時間2。(’毫秒"微秒) |
| nBid | 買價。 |
| nAsk | 賣價。 |
| nClose | 成交價。 |
| nQty | 成交量。 |
| nSimulate | 0:一般揭示 1:試算揭示 |
| **備註** | * + 1. nTimehms提供單位：時分秒hh:mm:ss。     2. nTimemillismicros提供單位：毫秒微秒’millissecond’’micorsecond，預設提供毫秒微秒，開發人員自行決定是否顯示。(目前solace只提供證券商品)     3. 未進行揭示處理，開發人員需自行接手處理(判斷收的tick 為一般或試算揭示)。     4. T盤切換T+1盤，不保留前一盤資料，開發者需自行清除前一盤資料。     5. 避免在[SKQuoteLib\_OnNotifyHistoryTicksLONG](#_OnNotifyHistoryTicksLONG)通知事件裡進行[SKQuoteLib\_GetTickLONG](#_SKQuoteLib_GetTickLONG)()、[SKQuoteLib\_GetStockByIndexLONG](#_SKQuoteLib_GetStockByIndexLONG)(只可取得基本資料)     6. 盤中零股: 因盤中仍會進入試撮，請留意是否收取『試撮資料』，即試算揭示.   \*依國內或海外市場，可能需搭配MarketNo  ＊須使用[SKQuoteLib\_EnterMonitorLONG](#_SKQuoteLib_EnterMonitorLONG)登入，該事件才會被觸發。 | |

## 技術分析

### OnNotifyMACDLONG

|  |  |  |
| --- | --- | --- |
| (LONG index)事件回傳證券市場－技術分析平滑異同平均線MACD數值。（日線－完整） | | |
| **宣告** | void OnNotifyMACDLONG([in] SHORT sMarketNo, [in] LONG nStockidx, [in] BSTR bstrMACD, [in] BSTR bstrDIF, [in] BSTR bstrOSC); | |
| **參數** | sMarketNo | 市場別代號 |
| nStockidx | 系統所編的索引代碼。 |
| bstrMACD | MACD平滑異同平均線 |
| bstrDIF | DIF |
| bstrOSC | DIF-MACD |
| **備註** | 目前只提供證券市場(不含盤中零股)  ＊須使用[SKQuoteLib\_EnterMonitorLONG](#_SKQuoteLib_EnterMonitorLONG)登入，該事件才會被觸發。 | |

|  |
| --- |
| Python範例程式碼\_Quote.py |
| UI程式碼(放置於 def createWidgets(self): 的下一層)/UI示意圖 |
| # richTextBoxMessage          self.richTextBoxMessage = tk.Listbox(self, height=5)          self.richTextBoxMessage.grid(column = 0, row = 10, columnspan=5, sticky = tk.E + tk.W)          global richTextBoxMessage          richTextBoxMessage = self.richTextBoxMessage |

|  |
| --- |
| 事件(可放在事件SKQuoteLibEvent註冊物件底下一層) |
| def OnNotifyMACDLONG(self, sMarketNo, nStockidx, bstrMACD, bstrDIF, bstrOSC):          pSKMACD = sk.SKMACD()          pSKMACD.bstrStockNo = textBoxbstrStockNo.get()          pSKMACD.bstrMACD = bstrMACD          pSKMACD.bstrDIF = bstrDIF          pSKMACD.bstrOSC = bstrOSC          pSKMACD, nCode = m\_pSKQuote.SKQuoteLib\_GetMACDLONG(sMarketNo, nStockidx, pSKMACD)          msg = ("【OnNotifyMACDLONG】" + "市場別代號:" + str(sMarketNo) +          "系統所編的索引代碼" + str(nStockidx) +          "MACD平滑異同平均線:" + str(pSKMACD.bstrMACD) +          "DIF:" + str(pSKMACD.bstrDIF) +          "OSC:" + str(pSKMACD.bstrOSC))          richTextBoxMessage.insert('end', msg + "\n")          richTextBoxMessage.see('end') |

### OnNotifyBoolTunelLONG

|  |  |  |
| --- | --- | --- |
| (LONG index)事件回傳技術分析－布林通道。（日線－完整） | | |
| **宣告** | void OnNotifyBoolTunelLONG([in] SHORT sMarketNo, [in] LONG nStockidx, [in] BSTR bstrAVG, [in] BSTR bstrUBT, [in] BSTR bstrLBT); | |
| **參數** | sMarketNo | 市場別代號 |
| nStockidx | 系統所編的索引代碼。 |
| bstrAVG | 均線 |
| bstrUBT | 通道上端 |
| bstrLBT | 通道下端 |
| **備註** | 目前只提供證券市場(不含盤中零股)  ＊須使用[SKQuoteLib\_EnterMonitorLONG](#_SKQuoteLib_EnterMonitorLONG)登入，該事件才會被觸發。 | |

|  |
| --- |
| Python範例程式碼\_Quote.py |
| UI程式碼(放置於 def createWidgets(self): 的下一層)/UI示意圖  一張含有 文字, 螢幕擷取畫面, 字型 的圖片  自動產生的描述 |
| # richTextBoxMessage          self.richTextBoxMessage = tk.Listbox(self, height=5)          self.richTextBoxMessage.grid(column = 0, row = 10, columnspan=5, sticky = tk.E + tk.W)          global richTextBoxMessage          richTextBoxMessage = self.richTextBoxMessage |

|  |
| --- |
| 事件(可放在事件SKQuoteLibEvent註冊物件底下一層) |
| def OnNotifyBoolTunelLONG(self, sMarketNo, nStockidx, bstrAVG, bstrUBT, bstrLBT):          pBoolTunel = sk.SKBoolTunel()          pBoolTunel.bstrStockNo = textBoxbstrStockNo.get()          pBoolTunel.bstrAVG = bstrAVG          pBoolTunel.bstrUBT = bstrUBT          pBoolTunel.bstrLBT = bstrLBT          pBoolTunel, nCode = m\_pSKQuote.SKQuoteLib\_GetBoolTunelLONG(sMarketNo, nStockidx, pBoolTunel)          msg = ("【OnNotifyBoolTunelLONG】" + "市場別代號:" + str(sMarketNo) +          "系統所編的索引代碼" + str(nStockidx) +          "均線:" + str(pBoolTunel.bstrAVG) +          "通道上端:" + str(pBoolTunel.bstrUBT) +          "通道下端:" + str(pBoolTunel.bstrLBT))          richTextBoxMessage.insert('end', msg + "\n")          richTextBoxMessage.see('end') |

### OnNotifyKLineData

|  |  |  |
| --- | --- | --- |
| 事件回傳技術分析資訊。 | | |
| **宣告** | void OnNotifyKLineData([in] BSTR bstrStockNo, [in] BSTR bstrData); | |
| **參數** | bstrStockNo | 商品代號。 |
| bstrData | 回傳字串，技術分析資料。傳回字串格式分成四種:  (1.) 舊版輸出格式，1分鐘線。以逗號分開資料。  ( 月/日/年, 時:分, 開盤價, 最高價, 最低價, 收盤價, 成交量 )  例如，06/18/2009, 09:05, 1365, 1385, 1365, 1380, 656  (2.) 舊版輸出格式，日線288天，完整日線，週線，月線。以逗號分開資料。  ( 月/日/年, 開盤價, 最高價, 最低價, 收盤價, 成交量 )  例如，06/18/2009, 1365, 1385, 1365, 1380, 656  舊版輸出函式中所取回的價格都未經過小數點處理，例如1101 價格為「36.50」，則函式所傳回的價格為「3650」，因此使用者所開發的程式則必須接手處理。  特別注意舊版輸出格式中的期匯率商品(TypeNo=209)小位數有四位須特別再處理，例如RH1704價格為「6.8712」，則函式所傳回的價格為「68712」  (3.) 新版輸出格式，1分鐘線。以逗號分開資料。  (年/月/日, 時:分, 開盤價, 最高價, 最低價, 收盤價, 成交量)  例如，商品代號(1108)  2021/8/2 09:01 12.95,12.95,12.95.12.95,5  (4.) 新版輸出格式，日線288天，完整日線，週線，月線。以逗號分開資料。  (年/月/日, 開盤價, 最高價, 最低價, 收盤價, 成交量 )  例如，商品代號(1108)  2021/8/2, 12.95,13.00,12.80,12.95,291  新版輸出函式中所取回的價格已進行過小數點處理，且由Solace K線主機提供． |
| **備註** | 證券-日K:成交量為整股交易，不含【鉅額交易、盤中零股、盤後零股】 | |

比較表:

|  |  |  |
| --- | --- | --- |
| 修改前 | 說明 | V2.13.45 |
| 上市股票(商品)代號。 | bstrStockNo | 商品代號 |
| N/A | 備註 | 日K:成交量為整股交易，不含【鉅額交易、盤中零股、盤後零股】 |

|  |
| --- |
| Python範例程式碼\_Quote.py |
| UI程式碼(放置於 def createWidgets(self): 的下一層)/UI示意圖 |
| # richTextBoxMessage          self.richTextBoxMessage = tk.Listbox(self, height=5)          self.richTextBoxMessage.grid(column = 0, row = 10, columnspan=5, sticky = tk.E + tk.W)          global richTextBoxMessage          richTextBoxMessage = self.richTextBoxMessage |

|  |
| --- |
| 事件(可放在事件SKQuoteLibEvent註冊物件底下一層) |
| def OnNotifyKLineData(self, bstrStockNo, bstrData):          msg = "【OnNotifyKLineData】" + bstrStockNo + "\_" + bstrData          richTextBoxMessage.insert('end', msg + "\n")          richTextBoxMessage.see('end') |

### OnKLineComplete

|  |  |  |
| --- | --- | --- |
| 收完歷史KLine，等收到此事件通知後表示回補完成。 | | |
| **宣告** | void OnKLineComplete([in] BSTR bstrEndString); | |
| **參數** | bstrEndString | 結尾字串 “##” |
| **備註** | 當全部資料已經全部回傳完畢，將回傳一筆以「##」開頭的內容，表示查詢結束。 | |

## 交易資訊(期選)

### OnNotifyFutureTradeInfoLONG

|  |  |  |
| --- | --- | --- |
| (LONG index)事件回傳接收**期貨**商品的交易資訊。 | | |
| **宣告** | void OnNotifyFutureTradeInfoLONG([in]BSTR bstrStockNo, [in] SHORT sMarketNo, [in] LONG nStockidx, [in] LONG nBuyTotalCount, [in] LONG nSellTotalCount, [in] LONG nBuyTotalQty, [in] LONG nSellTotalQty, [in] LONG nBuyDealTotalCount, [in] LONG nSellDealTotalCount  ); | |
| **參數** | bstrStockNo | 商品代號 |
| sMarketNo | 市場別代號 |
| nStockidx | 系統所編的索引代碼。 |
| nBuyTotalCount | 總委託買進筆數 |
| nSellTotalCount | 總委託賣出筆數 |
| nBuyTotalQty | 總委託買進口數 |
| nSellTotalQty | 總委託賣出口數 |
| nBuyDealTotalCount | 總成交買進筆數 |
| nSellDealTotalCount | 總成交賣出筆數 |
| **備註** | nStockidx系統所編的索引代碼，透過MarketNo與 Stockidx ，可以使用 [SKQuoteLib\_GetStockByIndexLONG](#_SKQuoteLib_GetStockByIndexLONG)取得商品基本資訊  ＊須使用[SKQuoteLib\_EnterMonitorLONG](#_SKQuoteLib_EnterMonitorLONG)登入，該事件才會被觸發。 | |

|  |
| --- |
| Python範例程式碼\_Quote.py |
| UI程式碼(放置於 def createWidgets(self): 的下一層)/UI示意圖 |
| # richTextBoxMessage          self.richTextBoxMessage = tk.Listbox(self, height=5)          self.richTextBoxMessage.grid(column = 0, row = 10, columnspan=5, sticky = tk.E + tk.W)          global richTextBoxMessage          richTextBoxMessage = self.richTextBoxMessage |

|  |
| --- |
| 事件(可放在事件SKQuoteLibEvent註冊物件底下一層) |
| def OnNotifyFutureTradeInfoLONG(self, bstrStockNo, sMarketNo, nStockidx, nBuyTotalCount, nSellTotalCount, nBuyTotalQty, nSellTotalQty, nBuyDealTotalCount, nSellDealTotalCount):          msg = ("【OnNotifyFutureTradeInfoLONG】" + " 總委託買進筆數" + str(nBuyTotalCount)+          " 總委託賣出筆數" + str(nSellTotalCount)+          " 總委託買進口數" + str(nBuyTotalQty)+          " 總委託賣出口數" + str(nSellTotalQty)+          " 總成交買進筆數" + str(nBuyDealTotalCount) +          " 總成交賣出筆數" + str(nSellDealTotalCount))          richTextBoxMessage.insert('end', msg + "\n")          richTextBoxMessage.see('end') |

### OnNotifyStrikePrices

|  |  |  |
| --- | --- | --- |
| 選擇權資訊。透過呼叫[SKQuoteLib\_GetStrikePrices](#_SKQuoteLib_GetStrikePrices)後，資訊由該事件回傳。 | | |
| **宣告** | void OnNotifyStrikePrices([in] BSTR bstrOptionData); | |
| **參數** | bstrData | 回傳的字串內容，以「,」分隔每一個欄位，欄位依序為：  [選擇權商品代碼],[選擇權中文名],[Call商品買賣權代碼],[Put商品買賣權代碼],**[履約價]**,[年+月],[最後交易日]  ex: TXO,台選,TXO11000H8,TXO11000T8,**1100000**,201808,20180815   |  |  | | --- | --- | | 0 | Product商品代碼。  例如，台選商品代碼為 TXO，也就是選擇權代號前三碼。 | | 1 | Name中文名稱。例如，台選。 | | 2 | Call Call 商品買賣權代碼。例如，TXO08100C7 | | 3 | Put Put 商品買賣權代碼。例如，TXO08100O7 | | 4 | StrikePrice 履約價。  例如，StrikePrice資料為810000，使用時要除一百，即StrikePrice/100。 | | 5 | YearMonth 年＋月。  例如，2017年3月，YearMonth資料為201703 | | 6 | LastTradeDay 最後交易日  例如，2018年7月19日，資料為20180719  當為0時代表沒有最後交易日或為不可交易商品。 | |
| **備註** | 當全部資料已經全部回傳完畢，將回傳一筆以「##」開頭的內容，表示查詢結束。 | |

|  |
| --- |
| Python範例程式碼\_Quote.py |
| UI程式碼(放置於 def createWidgets(self): 的下一層)/UI示意圖 |
| # richTextBoxMessage          self.richTextBoxMessage = tk.Listbox(self, height=5)          self.richTextBoxMessage.grid(column = 0, row = 10, columnspan=5, sticky = tk.E + tk.W)          global richTextBoxMessage          richTextBoxMessage = self.richTextBoxMessage |

|  |
| --- |
| 事件(可放在事件SKQuoteLibEvent註冊物件底下一層) |
| def OnNotifyStrikePrices(self, bstrOptionData):          msg = "【OnNotifyStrikePrices】" + bstrOptionData          richTextBoxMessage.insert('end', msg + "\n")          richTextBoxMessage.see('end') |

# Struct結構物件

## SKSTOCKLONG (國內報價商品物件 LONG index)

struct SKSTOCKLONG

{

LONG nStockidx; // 商品自定索引代號

SHORT sDecimal; // 小數位數

SHORT sTypeNo; // EX: (證券)類股別 1 水泥 , 2 食品…etc.

BSTR bstrMarketNo; // 市埸代碼

BSTR bstrStockNo; // 商品代碼EX: 1101 台泥, TX12 台指期12月…etc.

BSTR bstrStockName; // 商品名稱

LONG nHigh; // 最高價

LONG nOpen; // 開盤價

LONG nLow; // 最低價

LONG nClose; // 成交價

LONG nTickQty; // 單量

LONG nRef; // 昨收、參考價

LONG nBid; // 買價

LONG nBc; // 買量

LONG nAsk; // 賣價

LONG nAc; // 賣量

LONG nTBc; // 買盤量(即外盤量)

LONG nTAc; // 賣盤量(即內盤量)

LONG nFutureOI; // 期貨未平倉 OI

LONG nTQty; // 總量

LONG nYQty; // 昨量

LONG nUp; // 漲停價

LONG nDown; // 跌停價

LONG nSimulate; // 揭示 0:一般 1:試算　＊

　　　　　　　　// [**證券逐筆**]盤中出現『1:試算』代表行情劇動，

　　　　　　　　　　　　　觸發價格穩定措施狀態

LONG nDayTrade // [**限證券整股商品**]可否當沖 0:一般 1:可先買後賣現股當沖

　　　　　　　　　　　　　2:可先買後賣和先賣後買現股當沖

LONG nTradingDay // 交易日(YYYYMMDD)

備註 :當日非交易日時, 資料為前一交易日

LONG nDealTime //成交時間 (hhmmss)

};

### 盤中零股

＊因盤中仍會進入試撮，請留意是否收取『試撮資料』，即試算揭示.

struct SKSTOCKLONG

{

LONG nStockidx; // 商品自定索引代號

SHORT sDecimal; // 小數位數

SHORT sTypeNo; // EX: (證券)類股別 1 水泥 , 2 食品…etc.

BSTR bstrMarketNo; // 市埸代碼(0:上市, 1:上櫃, 2:期貨, 3:選擇權, 4:盤中零股-上市, 6:盤中零股-上櫃)

BSTR bstrStockNo; // 商品代碼EX: 1101 台泥, TX12 台指期12月…etc.

BSTR bstrStockName; // 商品名稱

LONG nHigh; // 最高價

LONG nOpen; // 開盤價

LONG nLow; // 最低價

LONG nClose; // 成交價

LONG nTickQty; // 單量

LONG nRef; // 昨收、參考價

LONG nBid; // 買價

LONG nBc; // 買量

LONG nAsk; // 賣價

LONG nAc; // 賣量

LONG nTBc; // 買盤量(即外盤量)

LONG nTAc; // 賣盤量(即內盤量)

LONG nTQty; // 總量

LONG nYQty; // 昨量

LONG nUp; // 漲停價

LONG nDown; // 跌停價

LONG nSimulate; // 揭示 0:一般 1:試算(試撮)　＊

　　　　　　　　// [**證券逐筆**]盤中出現『1:試算』代表行情劇動，

　　　　　　　　　　　　　觸發價格穩定措施狀態;

//[證券盤中零股] 盤中會進入試撮，請留意是否收取『試撮資料』

LONG nDayTrade // [**限證券商品**]可否當沖 0:一般

LONG nTradingDay // 交易日(YYYYMMDD)

備註 :當日非交易日時, 資料為前一交易日

LONG nTradingLotFlag //[**證券]** 整股、盤中零股揭示註記 0:現股 ; 1:盤中零股

};

## SKTICK(國內報價TICK物件)

struct SKTICK

{

LONG nPtr; //第幾筆成交明細, 由0開始

LONG nTimehms; //時間(時分秒hh:mm:ss)

LONG nTimemillismicros; //時間(毫秒微秒“ms”μs)

LONG nBid; //買價

LONG nAsk; //賣價

LONG nClose; //成交價

LONG nQty; //量

LONG nSimulate; //揭示 0:一般 1:試算（試撮）＊

LONG nDate; //交易日(YYYYMMDD)

};

＊盤中零股: 因盤中仍會進入試撮，請留意是否收取『試撮資料』，即試算揭示。

＊時間單位：毫秒（ms）：10-3秒 ；微秒（µs）：10-6秒，當nTimemillismicros資料是996886，前三位數字為（ms）10-3秒，後三位數為　（µs）10-6秒。

## SKBEST5(五檔價格物件)

typedef struct BEST5

{

LONG nBid1; // 第一檔買價

LONG nBidQty1; // 第一檔買量

LONG nBid2; // 第二檔買價

LONG nBidQty2; // 第二檔買量

LONG nBid3; // 第三檔買價

LONG nBidQty3; // 第三檔買量

LONG nBid4; // 第四檔買價

LONG nBidQty4; // 第四檔買量

LONG nBid5; // 第五檔買價

LONG nBidQty5; // 第五檔買量

LONG nExtendBid; // 衍生一檔買價{海外無衍生一檔}

LONG nExtendBidQty; // 衍生一檔買量{海外無衍生一檔}

LONG nAsk1; // 第一檔賣價

LONG nAskQty1; // 第一檔賣量

LONG nAsk2; // 第二檔賣價

LONG nAskQty2; // 第二檔賣量

LONG nAsk3; // 第三檔賣價

LONG nAskQty3; // 第三檔賣量

LONG nAsk4; // 第四檔賣價

LONG nAskQty4; // 第四檔買量

LONG nAsk5; // 第五檔賣價

LONG nAskQty5; // 第五檔賣量

LONG nExtendAsk; // 衍生一檔賣價{海外無衍生一檔}

LONG nExtendAskQty; // 衍生一檔賣量{海外無衍生一檔}

LONG nSimulate; // 揭示 0:一般 1:試算(試撮\*)

}

**\*國內行情—盤中零股：盤中仍有試撮行情，請自行判斷及揭示。**

## SKMACD(國內報價SKMACD物件)

typedef struct SKMACD

{

BSTR bstrStockNo; // 商品代碼

BSTR bstrMACD; // 技術指標MACD平滑異同平均線

BSTR bstrDIF; // 技術指標DIF

BSTR bstrOSC // 技術指標OSC

}

## SKBoolTunel (國內報價SKBoolTunel物件)

typedef struct SKBoolTunel

{

BSTR bstrStockNo; // 商品代碼

BSTR bstrAVG; // 技術指標布林通道 AVG均線

BSTR bstrUBT; // 技術指標 UBT通道上端

BSTR bstr LBT; // 技術指標 LBT通道下端

}