策略王COM元件使用說明

-海期報價-

# 函式範例

## SKOSQuoteLib (海期報價)

*函式*

1. 連線

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| 1. **連線** | [SKOSQuoteLib\_EnterMonitorLONG](#_SKOSQuoteLib_EnterMonitorLONG) | LONG index對應 |
| 1. **斷線** | [SKOSQuoteLib\_LeaveMonitor](#_SKOSQuoteLib_LeaveMonitor) |  |
| 1. **連線狀態** | [SKOSQuoteLib\_IsConnected](#_SKOSQuoteLib_IsConnected) |  |
| 1. **取得連線狀態** | [SKOSQuoteLib\_GetQuoteStatus](#_SKOSQuoteLib_GetQuoteStatus) | (僅限連線成功後使用) |
| 1. **連線主機設定** | [SKOSQuoteLib\_SetOSQuoteServer](#_SKOSQuoteLib_SetOSQuoteServer) | 可選擇連正式或備援資訊源 |
| 1. **物件初始** | [SKOSQuoteLib\_Initialize](#_SKOSQuoteLib_Initialize) |  |

1. 商品清單

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| **要求海期商品** | [SKOSQuoteLib\_RequestOverseaProducts](#_SKOSQuoteLib_RequestOverseaProducts) |  |
| **取得海期商品檔**  **(含下單代碼)** | [SKOSQuoteLib\_GetOverseaProductDetail](#_SKOSQuoteLib_GetOverseaProductDetai) |  |

1. 個期資訊

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| **取得商品物件** | [SKOSQuoteLib\_GetStockByNoLONG](#_SKOSQuoteLib_GetStockByNoLONG) | LONG index對應 |
| **取得商品物件** | [SKOSQuoteLib\_GetStockByNoNineDigitLONG](#_SKOSQuoteLib_GetStockByNoNineDigitL) | LONG index +  CME九位小數擴充對應 |

1. 即時報價

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| **訂閱報價** | [SKOSQuoteLib\_RequestStocks](#_SKOSQuoteLib_RequestStocks) |  |
| **取得商品物件** | [SKOSQuoteLib\_GetStockByIndexLONG](#_SKOSQuoteLib_GetStockByIndexLONG) | LONG index對應 |
| **取得商品物件** | [SKOSQuoteLib\_GetStockByIndexNineDigitLONG](#_SKOSQuoteLib_GetStockByIndexNineDig) | LONG index +  CME九位小數擴充對應 |

1. 十檔&五檔&成交明細

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| **訂閱Ticks&十檔** | [SKOSQuoteLib\_RequestTicks](#_SKOSQuoteLib_RequestTicks) | [OnNotifyTicksNineDigitLONG](#_OnNotifyTicksNineDigitLONG)、[OnNotifyHistoryTicksNineDigitLONG](#_OnNotifyHistoryTicksNineDigitLONG)、[OnNotifyBest5NineDigitLONG](#_OnNotifyBest5NineDigitLONG)、[OnNotifyBest10NineDigitLONG](#_OnNotifyBest10NineDigitLONG) |
| **取得Tick物件** | [SKOSQuoteLib\_GetTickNineDigitLONG](#_SKOSQuoteLib_GetTickNineDigitLONG) | LONG index +  CME九位小數擴充對應 |
| **取得五檔** | [SKOSQuoteLib\_GetBest5NineDigitLONG](#_SKOSQuoteLib_GetBest5NineDigitLONG) | LONG index +  CME九位小數擴充對應 |
| **訂閱十檔** | [SKOSQuoteLib\_RequestMarketDepth](#_SKOSQuoteLib_RequestMarketDepth) | 包含訂閱，僅十檔、五檔 |
| **訂閱即時成交明細** | [SKOSQuoteLib\_RequestLiveTick](#_SKOSQuoteLib_RequestLiveTick) | 包含訂閱，僅成交明細 |

1. 技術分析

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| **K線查詢** | |  | | --- | | **指定區間**  [SKOSQuoteLib\_RequestKLineByDate](#_SKOSQuoteLib_RequestKLineByDate) | | [SKOSQuoteLib\_RequestKLine](#_SKOSQuoteLib_RequestKLine) | |  |

*事件*

1. 連線

|  |  |  |
| --- | --- | --- |
| **功能** | **事件名稱** | **備註** |
| **連線狀態** | [OnConnect](#_OnConnect) |  |

1. 商品清單

|  |  |  |
| --- | --- | --- |
| **功能** | **事件名稱** | **備註** |
| **海期商品資料** | [OnOverseaProducts](#_OnOverseaProducts) |  |
| **海期商品資料**  **(含下單代碼)** | [OnOverseaProductsDetail](#_OnOverseaProductsDetail) |  |

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 |

1. 十檔&五檔&成交明細

|  |  |  |
| --- | --- | --- |
| **功能** | **事件名稱** | **備註** |
| **即時Tick** | [OnNotifyTicksNineDigitLONG](#_OnNotifyTicksNineDigitLONG) | LONG index +  CME九位小數擴充對應 |
| **回補Tick** | [OnNotifyHistoryTicksNineDigitLONG](#_OnNotifyHistoryTicksNineDigitLONG) | LONG index +  CME九位小數擴充對應 |
| **即時五檔** | [OnNotifyBest5NineDigitLONG](#_OnNotifyBest5NineDigitLONG) | LONG index +  CME九位小數擴充對應 |
| **即時十檔** | [OnNotifyBest10NineDigitLONG](#_OnNotifyBest10NineDigitLONG) | LONG index +  CME九位小數擴充對應 |

1. 技術分析

|  |  |  |
| --- | --- | --- |
| **功能** | **事件名稱** | **備註** |
| **K線資料** | [OnKLineData](#_OnKLineData) |  |

### 連線

#### SKOSQuoteLib\_EnterMonitorLONG

|  |  |  |
| --- | --- | --- |
| (LONG index)與報價伺服器建立連線。 | | |
| **宣告** | Long SKOSQuoteLib\_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 [OnConnect](#_OnConnect) 事件回傳。  需先簽署期貨API下單聲明書，方可使用 | |

|  |
| --- |
| Python範例程式碼\_OSQuote.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 |
| # buttonSKOSQuoteLib\_EnterMonitorLONG          self.buttonSKOSQuoteLib\_EnterMonitorLONG = tk.Button(self)          self.buttonSKOSQuoteLib\_EnterMonitorLONG["text"] = "連線報價主機"          self.buttonSKOSQuoteLib\_EnterMonitorLONG["command"] = self.buttonSKOSQuoteLib\_EnterMonitorLONG\_Click          self.buttonSKOSQuoteLib\_EnterMonitorLONG.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\_pSKOSQuote = comtypes.client.CreateObject(sk.SKOSQuoteLib,interface=sk.ISKOSQuoteLib) |

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

#### SKOSQuoteLib\_LeaveMonitor

|  |  |  |
| --- | --- | --- |
| 中斷報價伺服器連線。 | | |
| **宣告** | Long SKOSQuoteLib\_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** | 連線狀態由[OnConnect](#_OnConnect)事件回傳。 | |

|  |
| --- |
| Python範例程式碼\_OSQuote.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 |
| # buttonSKOSQuoteLib\_LeaveMonitor          self.buttonSKOSQuoteLib\_LeaveMonitor = tk.Button(self)          self.buttonSKOSQuoteLib\_LeaveMonitor["text"] = "斷線報價主機"          self.buttonSKOSQuoteLib\_LeaveMonitor["command"] = self.buttonSKOSQuoteLib\_LeaveMonitor\_Click          self.buttonSKOSQuoteLib\_LeaveMonitor.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\_pSKOSQuote = comtypes.client.CreateObject(sk.SKOSQuoteLib,interface=sk.ISKOSQuoteLib) |

|  |
| --- |
| Current Block(放置於def createWidgets(self): 的同一層) |
| def buttonSKOSQuoteLib\_LeaveMonitor\_Click(self):            # 中斷報價伺服器連線          nCode= m\_pSKOSQuote.SKOSQuoteLib\_LeaveMonitor()          msg = "【SKOSQuoteLib\_LeaveMonitor】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

#### SKOSQuoteLib\_IsConnected

|  |  |  |
| --- | --- | --- |
| 檢查目前連線狀態。 | | |
| **宣告** | Long SKOSQuoteLib\_IsConnected(); | |
| **參數** | 無 |  |
| **回傳值** | 1表示連線中，其餘非1數值都表示失敗。 | |
| **備註** | 請同時接收通知事件[OnConnect](#_OnConnect) | |

|  |
| --- |
| Python範例程式碼\_OSQuote.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 |
| # buttonSKOSQuoteLib\_IsConnected          self.buttonSKOSQuoteLib\_IsConnected = tk.Button(self)          self.buttonSKOSQuoteLib\_IsConnected["text"] = "檢查連線狀態"          self.buttonSKOSQuoteLib\_IsConnected["command"] = self.buttonSKOSQuoteLib\_IsConnected\_Click          self.buttonSKOSQuoteLib\_IsConnected.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\_pSKOSQuote = comtypes.client.CreateObject(sk.SKOSQuoteLib,interface=sk.ISKOSQuoteLib) |

|  |
| --- |
| Current Block(放置於def createWidgets(self): 的同一層) |
| def buttonSKOSQuoteLib\_IsConnected\_Click(self):            # 檢查目前報價的連線狀態          nCode= m\_pSKOSQuote.SKOSQuoteLib\_IsConnected()            if nCode == 1:              msg = "連線中"          else:              msg = "失敗"          msg = "【SKOSQuoteLib\_IsConnected】" + msg          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

#### SKOSQuoteLib\_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 SKOSQuoteLib\_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  ▼以下LOG（日期\_OSQuote.log）為確認是否下載海期商品檔LoadOSCommdity (可至文件「**9.下單-海外期選**」查看)，下圖代表有海期帳號，預設Login會占用一條報價連線（海期）一張含有 文字, 螢幕擷取畫面, 字型 的圖片  自動產生的描述 | |

|  |
| --- |
| Python範例程式碼\_OSQuote.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 |
| # buttonSKOSQuoteLib\_GetQuoteStatus          self.buttonSKOSQuoteLib\_GetQuoteStatus = tk.Button(self)          self.buttonSKOSQuoteLib\_GetQuoteStatus["text"] = "連線數資訊/限制"          self.buttonSKOSQuoteLib\_GetQuoteStatus["command"] = self.buttonSKOSQuoteLib\_GetQuoteStatus\_Click          self.buttonSKOSQuoteLib\_GetQuoteStatus.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\_pSKOSQuote = comtypes.client.CreateObject(sk.SKOSQuoteLib,interface=sk.ISKOSQuoteLib) |

|  |
| --- |
| Current Block(放置於def createWidgets(self): 的同一層) |
| def buttonSKOSQuoteLib\_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)          nCode= m\_pSKOSQuote.SKOSQuoteLib\_GetQuoteStatus(pnConnectionCount, pbIsOutLimit)          msg = "【SKOSQuoteLib\_GetQuoteStatus】"  + "連線數:" + str(pnConnectionCount) + "超過限制:" + str(pbIsOutLimit)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

#### SKOSQuoteLib\_SetOSQuoteServer

|  |  |  |
| --- | --- | --- |
| 切換海期、海選報價資訊源。 | | |
| **宣告** | Long SKOSQuoteLib\_SetOSQuoteServer([in] SHORT sServer); | |
| **參數** | sServer | 0：預設 1：備援 |
| **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** | ※這不是切換Server，Server是自動分配的。  切換資訊源後，需要斷線後再重新連線，才會將報價連至另一個資訊源上。  目前僅以下交易所有資訊源切換的功能：  海期：CME、CBOT、NYM、SGX、HKEx  海選：HKEx | |

|  |
| --- |
| Python範例程式碼\_OSQuote.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 |
| # buttonSKOSQuoteLib\_SetOSQuoteServer          self.buttonSKOSQuoteLib\_SetOSQuoteServer = tk.Button(self)          self.buttonSKOSQuoteLib\_SetOSQuoteServer["text"] = "切換資訊源"          self.buttonSKOSQuoteLib\_SetOSQuoteServer["command"] = self.buttonSKOSQuoteLib\_SetOSQuoteServer\_Click          self.buttonSKOSQuoteLib\_SetOSQuoteServer.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\_pSKOSQuote = comtypes.client.CreateObject(sk.SKOSQuoteLib,interface=sk.ISKOSQuoteLib)  # 海期報價伺服器 0：預設  1：備援  sServer = 0 |

|  |
| --- |
| Current Block(放置於def createWidgets(self): 的同一層) |
| def buttonSKOSQuoteLib\_SetOSQuoteServer\_Click(self):          global sServer          if sServer == 0:              sServer = 1          else:              sServer = 0            # 切換海期、海選報價資訊源          nCode= m\_pSKOSQuote.SKOSQuoteLib\_SetOSQuoteServer(sServer)          msg = "【SKOSQuoteLib\_SetOSQuoteServer】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

#### SKOSQuoteLib\_Initialize

|  |  |  |
| --- | --- | --- |
| 重新初始海期物件。 | | |
| **宣告** | Long SKOSQuoteLib\_Initialize(); | |
| **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** | 需先簽署期貨API下單聲明書，方可使用 | |

|  |
| --- |
| Python範例程式碼\_OSQuote.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 |
| # buttonSKOSQuoteLib\_Initialize          self.buttonSKOSQuoteLib\_Initialize = tk.Button(self)          self.buttonSKOSQuoteLib\_Initialize["text"] = "重新初始海期物件"          self.buttonSKOSQuoteLib\_Initialize["command"] = self.buttonSKOSQuoteLib\_Initialize\_Click          self.buttonSKOSQuoteLib\_Initialize.grid(row=5, 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\_pSKOSQuote = comtypes.client.CreateObject(sk.SKOSQuoteLib,interface=sk.ISKOSQuoteLib) |

|  |
| --- |
| Current Block(放置於def createWidgets(self): 的同一層) |
| def buttonSKOSQuoteLib\_Initialize\_Click(self):          # 重新初始海期物件          nCode= m\_pSKOSQuote.SKOSQuoteLib\_Initialize()          msg = "【SKOSQuoteLib\_Initialize】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

### 商品清單

#### SKOSQuoteLib\_RequestOverseaProducts

|  |  |  |
| --- | --- | --- |
| 取得海外商品檔。 | | |
| **宣告** | Long SKOSQuoteLib\_RequestOverseaProducts(); | |
| **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** | 商品資訊由[OnOverseaProducts](#_OnOverseaProducts)事件取得。  ＊須使用[SKOSQuoteLib\_EnterMonitorLONG](#_SKOSQuoteLib_EnterMonitorLONG)登入，方可執行此函式。 | |

|  |
| --- |
| Python範例程式碼\_OSQuote.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 |
| # buttonSKOSQuoteLib\_RequestOverseaProducts          self.buttonSKOSQuoteLib\_RequestOverseaProducts = tk.Button(self)          self.buttonSKOSQuoteLib\_RequestOverseaProducts["text"] = "取得海外商品檔"          self.buttonSKOSQuoteLib\_RequestOverseaProducts["command"] = self.buttonSKOSQuoteLib\_RequestOverseaProducts\_Click          self.buttonSKOSQuoteLib\_RequestOverseaProducts.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\_pSKOSQuote = comtypes.client.CreateObject(sk.SKOSQuoteLib,interface=sk.ISKOSQuoteLib) |

|  |
| --- |
| Current Block(放置於def createWidgets(self): 的同一層) |
| def buttonSKOSQuoteLib\_RequestOverseaProducts\_Click(self):          # 取得海外商品檔          nCode= m\_pSKOSQuote.SKOSQuoteLib\_RequestOverseaProducts()          msg = "【SKOSQuoteLib\_RequestOverseaProducts】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

#### SKOSQuoteLib\_GetOverseaProductDetail

|  |  |  |
| --- | --- | --- |
| 取得海期商品檔(含下單代碼) | | |
| **宣告** | Long SKOSQuoteLib\_GetOverseaProductDetail(([in] SHORT sType); | |
| **參數** | sType | 1:取得海期商品檔，含下單交易所代碼及下單商品代碼 |
| **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。 | |
| **備註** | 商品資訊由[OnOverseaProductsDetail](#_OnOverseaProductsDetail)事件取得。  ＊須使用[SKOSQuoteLib\_EnterMonitorLONG](#_SKOSQuoteLib_EnterMonitorLONG)登入，方可執行此函式。 | |

|  |
| --- |
| Python範例程式碼\_OSQuote.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 |
| # buttonSKOSQuoteLib\_GetOverseaProductDetail          self.buttonSKOSQuoteLib\_GetOverseaProductDetail = tk.Button(self)          self.buttonSKOSQuoteLib\_GetOverseaProductDetail["text"] = "取得海期商品檔(含下單代碼)"          self.buttonSKOSQuoteLib\_GetOverseaProductDetail["command"] = self.buttonSKOSQuoteLib\_GetOverseaProductDetail\_Click          self.buttonSKOSQuoteLib\_GetOverseaProductDetail.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\_pSKOSQuote = comtypes.client.CreateObject(sk.SKOSQuoteLib,interface=sk.ISKOSQuoteLib) |

|  |
| --- |
| Current Block(放置於def createWidgets(self): 的同一層) |
| def buttonSKOSQuoteLib\_GetOverseaProductDetail\_Click(self):          sType = 1          # 取得海期商品檔(含下單代碼)          nCode= m\_pSKOSQuote.SKOSQuoteLib\_GetOverseaProductDetail(sType)          msg = "【SKOSQuoteLib\_GetOverseaProductDetail】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

### 個期資訊

#### SKOSQuoteLib\_GetStockByNoLONG

|  |  |  |
| --- | --- | --- |
| (LONG index)  根據商品代號，取回海期報價的相關資訊。 | | |
| **宣告** | Long SKOSQuoteLib\_GetStockByNoLONG([in] BSTR bstrStockNo, [in,**out**] struct **SKFOREIGNLONG\*** pSKStock); | |
| **參數** | bstrStockNo | 海期商品代號，例如 “CME,ES2109” (小SP指數)。 |
| pSKStock | SKCOM元件中的 [**SKFOREIGNLONG**](#_SKFOREIGNLONG_(_海外報價商品物件)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** | ＊須使用[SKOSQuoteLib\_EnterMonitorLONG](#_SKOSQuoteLib_EnterMonitorLONG)登入，方可執行此函式。 | |

|  |
| --- |
| Python範例程式碼\_OSQuote.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 |
| # textBoxSKOSQuoteLib\_GetStockByNoLONG          tk.Label(self, text = "商品代碼").grid(row=2, column=1)              #輸入框          self.textBoxSKOSQuoteLib\_GetStockByNoLONG = tk.Entry(self)          self.textBoxSKOSQuoteLib\_GetStockByNoLONG.grid(row=2, column=2)          global textBoxSKOSQuoteLib\_GetStockByNoLONG          textBoxSKOSQuoteLib\_GetStockByNoLONG = self.textBoxSKOSQuoteLib\_GetStockByNoLONG            # buttonSKOSQuoteLib\_GetStockByNoLONG          self.buttonSKOSQuoteLib\_GetStockByNoLONG = tk.Button(self)          self.buttonSKOSQuoteLib\_GetStockByNoLONG["text"] = "個期資訊"          self.buttonSKOSQuoteLib\_GetStockByNoLONG["command"] = self.buttonSKOSQuoteLib\_GetStockByNoLONG\_Click          self.buttonSKOSQuoteLib\_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\_pSKOSQuote = comtypes.client.CreateObject(sk.SKOSQuoteLib,interface=sk.ISKOSQuoteLib) |

|  |
| --- |
| Current Block(放置於def createWidgets(self): 的同一層) |
| def buttonSKOSQuoteLib\_GetStockByNoLONG\_Click(self):            pSKStock = sk.SKFOREIGNLONG()          # (LONG index)根據商品代號，取回海期報價的相關資訊。          pSKStock, nCode= m\_pSKOSQuote.SKOSQuoteLib\_GetStockByNoLONG(textBoxSKOSQuoteLib\_GetStockByNoLONG.get(), pSKStock)          msg = "【SKOSQuoteLib\_GetStockByNoLONG】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end')          msg = "【SKOSQuoteLib\_GetStockByNoLONG】" + "商品代號:" + str(pSKStock.bstrStockNo) + " 商品名稱:" + str(pSKStock.bstrStockName) + " 報價小數位數:" + str(pSKStock.sDecimal) + " 分母:" + str(pSKStock.nDenominator) + " 市場代碼:" + str(pSKStock.bstrMarketNo) + " 交易所代號:" + str(pSKStock.bstrExchangeNo) + " 交易所名稱:" + str(pSKStock.bstrExchangeName) + " CallPut:" + str(pSKStock.bstrCallPut) + " 開盤價:" +  str(pSKStock.nOpen) + " 最高:" + str(pSKStock.nHigh) + " 最低:" + str(pSKStock.nLow) + " 成交價:" +  str(pSKStock.nClose) + " 結算價:" + str(pSKStock.nSettlePrice) + " 單量:" + str(pSKStock.nTickQty) + " 昨收(參考價):" +  str(pSKStock.nRef) + " 買價:" +  str(pSKStock.nBid) + " 買量:" + str(pSKStock.nBc) + " 賣價:" + str(pSKStock.nAsk) + " 賣量:" + str(pSKStock.nAc) + " 交易日:" + str(pSKStock.nTradingDay)          richTextBoxMessage.insert('end', msg + "\n")          richTextBoxMessage.see('end') |

#### SKOSQuoteLib\_GetStockByNoNineDigitLONG

|  |  |  |
| --- | --- | --- |
| (LONG index)(CME九位小數擴充)  根據商品代號，取回海期報價的相關資訊。 | | |
| **宣告** | Long SKOSQuoteLib\_GetStockByNoNineDigitLONG([in] BSTR bstrStockNo, [in,**out**] struct **SKFOREIGN\_9LONG\*** pSKStock); | |
| **參數** | bstrStockNo | 海期商品代號，例如 “CME,ES2109” (小SP指數)。 |
| pSKStock | SKCOM元件中的**[SKFOREIGN\_9LONG](#_SKFOREIGN_9LONG_(_海外報價商品物件-九位擴充)**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** | ＊須使用[SKOSQuoteLib\_EnterMonitorLONG](#_SKOSQuoteLib_EnterMonitorLONG)登入，方可執行此函式。 | |

|  |
| --- |
| Python範例程式碼\_OSQuote.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 |
| # buttonSKOSQuoteLib\_GetStockByNoNineDigitLONG          self.buttonSKOSQuoteLib\_GetStockByNoNineDigitLONG = tk.Button(self)          self.buttonSKOSQuoteLib\_GetStockByNoNineDigitLONG["text"] = "個期資訊(CME九位擴充)"          self.buttonSKOSQuoteLib\_GetStockByNoNineDigitLONG["command"] = self.buttonSKOSQuoteLib\_GetStockByNoNineDigitLONG\_Click          self.buttonSKOSQuoteLib\_GetStockByNoNineDigitLONG.grid(row=2, column=4) |

|  |
| --- |
| 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\_pSKOSQuote = comtypes.client.CreateObject(sk.SKOSQuoteLib,interface=sk.ISKOSQuoteLib) |

|  |
| --- |
| Current Block(放置於def createWidgets(self): 的同一層) |
| def buttonSKOSQuoteLib\_GetStockByNoNineDigitLONG\_Click(self):            pSKStock = sk.SKFOREIGN\_9LONG()          # (LONG index)(CME九位小數擴充)根據商品代號，取回海期報價的相關資訊。          pSKStock, nCode= m\_pSKOSQuote.SKOSQuoteLib\_GetStockByNoNineDigitLONG(textBoxSKOSQuoteLib\_GetStockByNoLONG.get(), pSKStock)          msg = "【SKOSQuoteLib\_GetStockByNoNineDigitLONG】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end')          msg = "【SKOSQuoteLib\_GetStockByNoNineDigitLONG】" + "商品代號:" + str(pSKStock.bstrStockNo) + " 商品名稱:" + str(pSKStock.bstrStockName) + " 報價小數位數:" + str(pSKStock.sDecimal) + " 分母:" + str(pSKStock.nDenominator) + " 市場代碼:" + str(pSKStock.bstrMarketNo) + " 交易所代號:" + str(pSKStock.bstrExchangeNo) + " 交易所名稱:" + str(pSKStock.bstrExchangeName) + " CallPut:" + str(pSKStock.bstrCallPut) + " 開盤價:" +  str(pSKStock.nOpen) + " 最高:" + str(pSKStock.nHigh) + " 最低:" + str(pSKStock.nLow) + " 成交價:" +  str(pSKStock.nClose) + " 結算價:" + str(pSKStock.nSettlePrice) + " 單量:" + str(pSKStock.nTickQty) + " 昨收(參考價):" +  str(pSKStock.nRef) + " 買價:" +  str(pSKStock.nBid) + " 買量:" + str(pSKStock.nBc) + " 賣價:" + str(pSKStock.nAsk) + " 賣量:" + str(pSKStock.nAc) + " 交易日:" + str(pSKStock.nTradingDay)          richTextBoxMessage.see('end') |

### 即時報價

#### SKOSQuoteLib\_RequestStocks

|  |  |  |
| --- | --- | --- |
| 訂閱指定商品即時報價  要求伺服器針對 bstrStockNos 內的商品代號做報價通知動作。  報價更新由[[OnNotifyQuote](#_OnNotifyQuoteLONG)LONG](#_4-5-e_OnNotifyQuoteLONG)事件取得更通知。 | | |
| **宣告** | Long SKOSQuoteLib\_RequestStocks([in,out] SHORT\* psPageNo, [in] BSTR bstrStockNos); | |
| **參數** | psPageNo | 請固定帶1。 |
| bstrStockNos | 欲訂閱的海期商品代號，以{「交易所代碼」,「商品報價代碼」}為單位，多筆以”#”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** | \*若bstrStockNos字串中”含無效之商品代碼”，則不會送出此次訂閱即時報價。  舉例說明：  psPageNo = -1 時，pStockNos = “CBOT,ZB1712#HKEx,HSI1712#CBOT,YM1712”，表示向伺服器索取這三檔股票的報價。  \*避免在[OnConnect](#_OnConnect) Event直接進行RequestStocks等報價訂閱，若各交易所商品未下載完成，即無法進行商品訂閱 | |

|  |
| --- |
| Python範例程式碼\_OSQuote.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            # buttonSKOSQuoteLib\_RequestStocks          self.buttonSKOSQuoteLib\_RequestStocks = tk.Button(self)          self.buttonSKOSQuoteLib\_RequestStocks["text"] = "訂閱"          self.buttonSKOSQuoteLib\_RequestStocks["command"] = self.buttonSKOSQuoteLib\_RequestStocks\_Click          self.buttonSKOSQuoteLib\_RequestStocks.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\_pSKOSQuote = comtypes.client.CreateObject(sk.SKOSQuoteLib,interface=sk.ISKOSQuoteLib) |

|  |
| --- |
| Current Block(放置於def createWidgets(self): 的同一層) |
| def buttonSKOSQuoteLib\_RequestStocks\_Click(self):          psPageNo = int(textBoxpsPageNo2.get())          # 訂閱指定商品即時報價，要求伺服器針對 bstrStockNos 內的商品代號做報價通知動作。報價更新由OnNotifyQuote事件取得更通知。          psPageNo, nCode= m\_pSKOSQuote.SKOSQuoteLib\_RequestStocks(psPageNo, textBoxStockNos.get())          msg = "【SKOSQuoteLib\_RequestStocks】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

##### SKOSQuoteLib\_GetStockByIndexLONG

|  |  |  |
| --- | --- | --- |
| (LONG index)根據系統所編的索引代碼，取回海期報價的相關資訊。 | | |
| **宣告** | Long SKOSQuoteLib\_GetStockByIndexLONG([in] LONG nIndex, [in,**out**] struct **SKFOREIGNLONG\*** pSKStock); | |
| **參數** | nIndex | 系統所編的索引代碼。 |
| pSKStock | SKCOM元件中的 [**SKFOREIGNLONG**](#_SKFOREIGNLONG_(_海外報價商品物件)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** | ＊商品物件價格未經小數處理。  ＊避免在[OnConnect](#_OnConnect) Event直接進行，若各交易所商品未下載完成，將無法取得商品基本資料。  ＊須使用[SKOSQuoteLib\_EnterMonitorLONG](#_SKOSQuoteLib_EnterMonitorLONG)登入，方可執行此函式。  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 | |

##### SKOSQuoteLib\_GetStockByIndexNineDigitLONG

|  |  |  |
| --- | --- | --- |
| (LONG index)(CME九位擴充)  根據系統所編的索引代碼，取回海期報價的相關資訊。 | | |
| **宣告** | Long SKOSQuoteLib\_GetStockByIndexNineDigitLONG([in] LONG nIndex, [in,**out**] struct **SKFOREIGN\_9LONG\*** pSKStock); | |
| **參數** | nIndex | 系統所編的索引代碼。 |
| pSKStock | SKCOM元件中的**[SKFOREIGN\_9LONG](#_SKFOREIGN_9LONG_(_海外報價商品物件-九位擴充)**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** | ＊商品物件價格未經小數處理。  ＊避免在[OnConnect](#_OnConnect) Event直接進行，若各交易所商品未下載完成，將無法取得商品基本資料。  ＊須使用[SKOSQuoteLib\_EnterMonitorLONG](#_SKOSQuoteLib_EnterMonitorLONG)登入，方可執行此函式。  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 | |

### 十檔&五檔&成交明細

#### SKOSQuoteLib\_RequestTicks

|  |  |  |
| --- | --- | --- |
| 訂閱與要求傳送成交明細以及五檔、十檔。 | | |
| **宣告** | Long SKOSQuoteLib\_RequestTicks([in,out] SHORT\* psPageNo, [in] BSTR bstrStockNo); | |
| **參數** | psPageNo | 請從1開始。 |
| bstrStockNo | 指定的商品，**一個Page僅能索取一檔**。格式「交易所代碼」,「商品報價代碼」。Ex：”CME,ES2012” |
| **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** | \*若bstrStockNo為無效之商品代碼”，則不會送出此次訂閱要求。  即時Tick由[OnNotifyTicksNineDigitLONG](#_OnNotifyTicksNineDigitLONG)事件通知。  Tick回補由 [OnNotifyHistoryTicksNineDigitLONG](#_OnNotifyHistoryTicksNineDigitLONG)事件通知。  最佳五檔位數擴充版由[[OnNotifyBest5](#_4-5-h_OnNotifyBest5NineDigitLONG)](#_OnNotifyBest5NineDigitLONG)[NineDigitLONG](#_4-5-h_OnNotifyBest5NineDigitLONG)事件通知。  最佳十檔位數擴充版由[OnNotifyBest10NineDigitLONG](#_OnNotifyBest10NineDigitLONG)事件通知。  \*避免在[OnConnect](#_OnConnect) Event直接進行RequestTicks等報價訂閱，若各交易所商品未下載完成，即無法進行商品訂閱 | |

|  |
| --- |
| Python範例程式碼\_OSQuote.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=1, column=2)          global textBoxpsPageNo          textBoxpsPageNo = self.textBoxpsPageNo            # textBoxTicks          tk.Label(self, text = "輸入商品代碼(僅1檔)").grid(row=1, column=3)              #輸入框          self.textBoxTicks = tk.Entry(self)          self.textBoxTicks.grid(row=1, column=4)          global textBoxTicks          textBoxTicks = self.textBoxTicks            # buttonSKOSQuoteLib\_RequestTicks          self.buttonSKOSQuoteLib\_RequestTicks = tk.Button(self)          self.buttonSKOSQuoteLib\_RequestTicks["text"] = "訂閱"          self.buttonSKOSQuoteLib\_RequestTicks["command"] = self.buttonSKOSQuoteLib\_RequestTicks\_Click          self.buttonSKOSQuoteLib\_RequestTicks.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\_pSKOSQuote = comtypes.client.CreateObject(sk.SKOSQuoteLib,interface=sk.ISKOSQuoteLib) |

|  |
| --- |
| Current Block(放置於def createWidgets(self): 的同一層) |
| def buttonSKOSQuoteLib\_RequestTicks\_Click(self):          psPageNo = int(textBoxpsPageNo.get())          # 訂閱與要求傳送成交明細以及五檔          psPageNo, nCode= m\_pSKOSQuote.SKOSQuoteLib\_RequestTicks(psPageNo, textBoxTicks.get())          msg = "【SKOSQuoteLib\_RequestTicks】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

##### SKOSQuoteLib\_GetTickNineDigitLONG

|  |  |  |
| --- | --- | --- |
| (LONG index)(CME九位小數擴充)  取得成交明細資訊。(需先訂閱即時成交明細RequestTicks) | | |
| **宣告** | Long SKOSQuoteLib\_GetTickNineDigitLONG([in] LONG nIndex, [in] LONG nPtr, [in,**out**] struct **SKFOREIGNTICK\_9\*** pSKTick); | |
| **參數** | nIndex | 系統所編的海期商品索引代碼。 |
| nPtr | 第幾筆成交明細，又稱成交明細順序。可以根據此 Index 取得指定第幾筆成交明細資訊。 |
| pSKStock | SKCOM元件中的**[SKFOREIGNTICK\_9](#_SKFOREIGNTICK_9(_海外報價TICK物件-九位擴充_))**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[SKOSQuoteLib\_OnNotifyHistoryTicksNineDigitLONG](#_OnNotifyHistoryTicksNineDigitLONG)、[SKOSQuoteLib\_OnNotifyTicksNineDigitLONG](#_OnNotifyTicksNineDigitLONG)通知事件裡進行  ＊須使用[SKOSQuoteLib\_EnterMonitorLONG](#_SKOSQuoteLib_EnterMonitorLONG)登入，方可執行此函式。  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 | |

##### SKOSQuoteLib\_GetBest5NineDigitLONG

|  |  |  |
| --- | --- | --- |
| (LONG index)(CME九位小數擴充)  取得最佳五檔價格資訊。(需先訂閱最佳五檔RequestTicks) | | |
| **宣告** | Long SKOSQuoteLib\_GetBest5NineDigitLONG([in] LONG nIndex, [in,**out**] struct **SKBEST5\_9\*** pSKBest5); | |
| **參數** | nIndex | 系統所編的海期商品索引代碼。 |
| pSKStock | SKCOM元件中的**[SKBEST5\_9](#_SKBEST5_9(_五檔價格物件-九位擴充_))**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** | 五檔更新由[OnNotifyBest5NineDigitLONG](#_OnNotifyBest5NineDigitLONG)事件通知取得。  避免在[SKOSQuoteLib\_OnNotifyBest5NineDigitLONG](#_OnNotifyBest5NineDigitLONG)通知事件裡進行  ＊須使用[SKOSQuoteLib\_EnterMonitorLONG](#_SKOSQuoteLib_EnterMonitorLONG)登入，方可執行此函式。 | |

#### SKOSQuoteLib\_RequestMarketDepth

|  |  |  |
| --- | --- | --- |
| 訂閱與要求傳送最佳十檔(僅包含最佳十、五檔；不包含訂閱歷史與即時成交明細) | | |
| **宣告** | Long SKOSQuoteLib\_RequestMarketDepth([in,out] SHORT\* psPageNo, [in] BSTR bstrStockNo); | |
| **參數** | psPageNo | 請從1開始。 |
| bstrStockNo | 索取的商品股票代號，**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。 | |
| **備註** | 最佳五檔位數擴充版由[[OnNotifyBest5](#_4-5-h_OnNotifyBest5NineDigitLONG)](#_OnNotifyBest5NineDigitLONG)[NineDigitLONG](#_4-5-h_OnNotifyBest5NineDigitLONG)事件通知。  最佳十檔位數擴充版由[OnNotifyBest10NineDigitLONG](#_OnNotifyBest10NineDigitLONG)事件通知。  **不會**進行傳送成交明細資料，與[SKOSQuoteLib\_RequestTicks](#_SKOSQuoteLib_RequestStocks)不同，請擇一使用;因無回補歷史成交明細資料，[SKOSQuoteLib\_GetTickNineDigitLONG](#_SKOSQuoteLib_GetTickNineDigitLONG)僅查閱，**訂閱後**之即時成交明細資料。 | |

##### SKOSQuoteLib\_RequestLiveTick

|  |  |  |
| --- | --- | --- |
| 訂閱與要求傳送**即時**成交明細。  (本功能**不會訂閱**最佳五檔與最佳十檔、亦不包含歷史Ticks) | | |
| **宣告** | Long SKOSQuoteLib\_RequestLiveTick([in,out] SHORT\* psPageNo, [in] BSTR bstrStockNo); | |
| **參數** | psPageNo | 請從1開始。 |
| bstrStockNo | 索取的商品股票代號，**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。 | |
| **備註** | 即時Tick由[OnNotifyTicksNineDigitLONG](#_OnNotifyTicksNineDigitLONG)事件通知。  **不會**回補歷史成交明細資料，與[SKOSQuoteLib\_RequestTicks](#_SKOSQuoteLib_RequestTicks)不同，請擇一使用;因無回補歷史成交明細資料，[SKOSQuoteLib\_GetTickNineDigitLONG](#_SKOSQuoteLib_GetTickNineDigitLONG)僅查閱，**訂閱後**之即時成交明細資料。 | |

### 技術分析

#### SKOSQuoteLib\_RequestKLineByDate

|  |  |  |
| --- | --- | --- |
| 取得K線資料，可指定日期區間，分K時可指定幾分K。 | | |
| **宣告** | Long SKOSQuoteLib\_RequestKLineByDate([in] BSTR bstrStockNo, [in]SHORT sKLineType, [in] BSTR bstrStartDate, [in] BSTR bstrEndDate, [in] SHORT sMinuteNumber); | |
| **參數** | bstrStockNo | 海期商品代號，例如 CME,ES1609。 |
| sKLineType | 0 = 分線。  1 = 日線。  2 = 週線。  3 = 月線。 |
| 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。 | |
| **備註** | \*若bstrStockNo為無效之商品代碼”，則不會送出此次K線查詢。  K線資料[OnKLineData](#_OnKLineData)事件通知取得。 | |

|  |
| --- |
| Python範例程式碼\_OSQuote.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 |
| # textBoxbstrStockNo          tk.Label(self, text = "商品代碼").grid(row=0, column=1)              #輸入框          self.textBoxbstrStockNo = tk.Entry(self)          self.textBoxbstrStockNo.grid(row=0, column=2)          global textBoxbstrStockNo          textBoxbstrStockNo = self.textBoxbstrStockNo            # comboBoxsKLineType          tk.Label(self, text = "K線種類").grid(row=1, column=1)              #輸入框          self.comboBoxsKLineType = ttk.Combobox(self, state='readonly')          self.comboBoxsKLineType['values'] = Config.comboBoxsKLineType          self.comboBoxsKLineType.grid(row=1, column=2)          global comboBoxsKLineType          comboBoxsKLineType = self.comboBoxsKLineType          # textBoxbstrStartDate          tk.Label(self, text = "起始日期").grid(row=2, column=1)              #輸入框          self.textBoxbstrStartDate = tk.Entry(self)          self.textBoxbstrStartDate.grid(row=2, column=2)          global textBoxbstrStartDate          textBoxbstrStartDate = self.textBoxbstrStartDate            # textBoxbstrEndDate          tk.Label(self, text = "結束日期").grid(row=3, column=1)              #輸入框          self.textBoxbstrEndDate = tk.Entry(self)          self.textBoxbstrEndDate.grid(row=3, column=2)          global textBoxbstrEndDate          textBoxbstrEndDate = self.textBoxbstrEndDate            # textBoxsMinuteNumber          tk.Label(self, text = "幾分K(分線)").grid(row=4, column=1)              #輸入框          self.textBoxsMinuteNumber = tk.Entry(self)          self.textBoxsMinuteNumber.grid(row=4, column=2)          global textBoxsMinuteNumber          textBoxsMinuteNumber = self.textBoxsMinuteNumber            # buttonSKOSQuoteLib\_RequestKLineByDate          self.buttonSKOSQuoteLib\_RequestKLineByDate = tk.Button(self)          self.buttonSKOSQuoteLib\_RequestKLineByDate["text"] = "KLine"          self.buttonSKOSQuoteLib\_RequestKLineByDate["command"] = self.buttonSKOSQuoteLib\_RequestKLineByDate\_Click          self.buttonSKOSQuoteLib\_RequestKLineByDate.grid(row=5, column=2) |

|  |
| --- |
| Config.py |
| # KLineForm  comboBoxsKLineType =(      "分線", "日線", "週線", "月線"  ) |
| 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\_pSKOSQuote = comtypes.client.CreateObject(sk.SKOSQuoteLib,interface=sk.ISKOSQuoteLib) |

|  |
| --- |
| Current Block(放置於def createWidgets(self): 的同一層) |
| def buttonSKOSQuoteLib\_RequestKLineByDate\_Click(self):          if (comboBoxsKLineType.get() == "分線"):              sKLineType = 0          elif (comboBoxsKLineType.get() == "日線"):              sKLineType = 1          elif (comboBoxsKLineType.get() == "週線"):              sKLineType = 2          elif (comboBoxsKLineType.get() == "月線"):              sKLineType = 3          # 取得K線資料，可指定日期區間，分K時可指定幾分K          nCode= m\_pSKOSQuote.SKOSQuoteLib\_RequestKLineByDate(textBoxbstrStockNo.get(), sKLineType, textBoxbstrStartDate.get(), textBoxbstrEndDate.get(), int(textBoxsMinuteNumber.get()))          msg = "【SKOSQuoteLib\_RequestKLineByDate】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

##### SKOSQuoteLib\_RequestKLine

|  |  |  |
| --- | --- | --- |
| 取得K線資料。 | | |
| **宣告** | Long SKOSQuoteLib\_RequestKLine([in] BSTR bstrStockNo, [in]SHORT sKLineType); | |
| **參數** | bstrStockNo | 海期商品代號，例如 CME,ES1609。 |
| sKLineType | 0 = 分鐘線。  1 = 日線。  2 = 週線。  3 = 月線。 |
| **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** | \*若bstrStockNo為無效之商品代碼”，則不會送出此次K線查詢。  K線資料[OnKLineData](#_OnKLineData)事件通知取得。 | |

# 事件

## 連線

### OnConnect

|  |  |  |
| --- | --- | --- |
| 接收連線狀態。 | | |
| **宣告** | void OnConnect([in] LONG nCode, [in] LONG nSocketCode); | |
| **參數** | nCode | SK\_SUBJECT\_CONNECTION\_CONNECTED 表示為連線事件。(3001)  SK\_SUBJECT\_CONNECTION\_DISCONNECT 表示為斷線事件。  其他錯誤代碼，可參考代碼定義表 |
| nSocketCode | 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[SKOSQuoteLib\_EnterMonitorLONG](#_SKOSQuoteLib_EnterMonitorLONG)、[SKOSQuoteLib\_LeaveMonitor](#_SKOSQuoteLib_LeaveMonitor)、[RequestStocks](#_SKOSQuoteLib_RequestStocks) and [RequestTicks](#_SKOSQuoteLib_RequestTicks) 等報價訂閱，若各交易所商品未下載完成，即無法進行商品訂閱。 | |

|  |
| --- |
| Python範例程式碼\_OSQuote.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SKOSQuoteLibEvent註冊物件底下一層) |
| def OnConnect(self, nCode, nSocketCode):          msg = "【OnConnect】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode) + "\_" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nSocketCode)          richTextBoxMessage.insert('end', msg + "\n")          richTextBoxMessage.see('end') |

## 商品清單

### OnOverseaProducts

|  |  |  |
| --- | --- | --- |
| 海期報價商品清單資訊。 | | |
| **宣告** | void OnOverseaProducts([in] BSTR bstrValue); | |
| **參數** | bstrValue | 字串格式：[交易所代碼],[交易所名稱],[商品報價代碼],[商品名稱],[最後交易日] ,[第一通知日] |
| **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。  最後交易日的值為0或99991231時，表示該商品沒有最後交易日或為指數匯率等不可交易商品 | |

|  |
| --- |
| Python範例程式碼\_OSQuote.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SKOSQuoteLibEvent註冊物件底下一層) |
| def OnOverseaProducts(self, bstrValue):          msg = "【OnOverseaProducts】" + bstrValue          richTextBoxMessage.insert('end', msg + "\n")          richTextBoxMessage.see('end') |

### OnOverseaProductsDetail

|  |  |  |
| --- | --- | --- |
| 海期報價商品清單資訊。(含下單代碼) | | |
| **宣告** | void OnOverseaProductsDetail([in] BSTR bstrValue); | |
| **參數** | bstrValue | 字串格式：[交易所代碼],[交易所名稱],[商品報價代碼],[商品名稱], [交易所下單代碼], [商品下單代碼],[最後交易日] ,[第一通知日] |
| **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。  最後交易日的值為0或99991231時，表示該商品沒有最後交易日或為指數匯率等不可交易商品  [商品下單代碼]:  EX:價差商品  由商品代號\_年月1/年月2組成，ex:AP\_201906/201909 or JY\_201912/201909 | |

|  |
| --- |
| Python範例程式碼\_OSQuote.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SKOSQuoteLibEvent註冊物件底下一層) |
| def OnOverseaProductsDetail(self, bstrValue):          msg = "【OnOverseaProductsDetail】" + bstrValue          richTextBoxMessage.insert('end', msg + "\n")          richTextBoxMessage.see('end') |

## 即時報價

### OnNotifyQuoteLONG

|  |  |  |
| --- | --- | --- |
| 當有索取的個別海期商品報價異動時，將透過此事件通知應用程式處理。 | | |
| **宣告** | void OnNotifyQuoteLONG( [in] LONG nIndex); | |
| **參數** | nIndex | 系統所編的海期商品索引代碼。 |
| **備註** | 事件觸發後，可由nIndex 帶入[SKOSQuoteLib\_GetStockByIndexLONG](#_SKOSQuoteLib_GetStockByIndexLONG)來取得商品報價物件。  ＊須使用[SKOSQuoteLib\_EnterMonitorLONG](#_SKOSQuoteLib_EnterMonitorLONG)登入，該事件才會被觸發。 | |

|  |
| --- |
| Python範例程式碼\_OSQuote.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SKOSQuoteLibEvent註冊物件底下一層) |
| def OnNotifyQuoteLONG(self, nIndex):          pSKStock = sk.SKFOREIGNLONG()          pSKStock, nCode = m\_pSKOSQuote.SKOSQuoteLib\_GetStockByIndexLONG(nIndex, pSKStock)          msg = "【OnNotifyQuoteLONG】" + " 報價小數位數" + str(pSKStock.sDecimal) + " 分母" + str(pSKStock.nDenominator) + " 市場代碼" + str(pSKStock.bstrMarketNo) + " 交易所代號" + str(pSKStock.bstrExchangeNo) + " 交易所名稱" + str(pSKStock.bstrExchangeName) + " 商品代號" + str(pSKStock.bstrStockNo) + " 商品名稱" + str(pSKStock.bstrStockName) + " CallPut" + str(pSKStock.bstrCallPut) + " 開盤價" + str(pSKStock.nOpen) + " 最高價" + str(pSKStock.nHigh) + " 最低價" + str(pSKStock.nLow) + " 成交價" + str(pSKStock.nClose) + " 結算價" + str(pSKStock.nSettlePrice) + " 單量" + str(pSKStock.nTickQty) + " 昨收、參考價" + str(pSKStock.nRef) + " 買價" + str(pSKStock.nBid) + " 買量" + str(pSKStock.nBc) + " 賣價" + str(pSKStock.nAsk) + " 賣量" + str(pSKStock.nAc) + " 成交量" + str(pSKStock.nTQty) + " 履約價" + str(pSKStock.nStrikePrice) + " 交易日(YYYYMMDD)" + str(pSKStock.nTradingDay)          richTextBoxMessage.insert('end', msg + "\n")          richTextBoxMessage.see('end') |

## 十檔&五檔&成交明細

### OnNotifyTicksNineDigitLONG

|  |  |  |
| --- | --- | --- |
| 當有索取的個別海期商品成交明細有所異動，即透過向此註冊事件回傳所異動的個別海期成交明細。 | | |
| **宣告** | void OnNotifyTicksNineDigitLONG ([in] LONG nIndex, [in] LONG nPtr, [in] LONG nDate, [in] LONG nTime, [in] LONGLONG nClose, [in] LONG nQty); | |
|  | nIndex | 系統所編的海期商品索引代碼。 |
| nPtr | 表示資料的位址(Key) |
| nDate | 交易日期。(YYYYMMDD) |
| nTime | 時間。 |
| nClose | 成交價。 |
| nQty | 成交量。 |
| **備註** | 價格未進行小數點處理，開發人員需自行接手處理(SKFOREIGN\_9LONG物件內有小數點位數)。  避免在  [SKOSQuoteLib\_OnNotifyHistoryTicksNineDigitLONG](#_OnNotifyHistoryTicksNineDigitLONG)、  [SKOSQuoteLib\_OnNotifyTicksNineDigitLONG](#_OnNotifyTicksNineDigitLONG)  通知事件裡進行[SKOSQuoteLib\_GetTickNineDigitLONG](#_SKOSQuoteLib_GetTickNineDigitLONG)  ＊須使用[SKOSQuoteLib\_EnterMonitorLONG](#_SKOSQuoteLib_EnterMonitorLONG)登入，該事件才會被觸發。 | |

|  |
| --- |
| Python範例程式碼\_OSQuote.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SKOSQuoteLibEvent註冊物件底下一層) |
| def OnNotifyTicksNineDigitLONG(self, nIndex, nPtr, nDate, nTime, nClose, nQty):          pSKTick = sk.SKFOREIGNTICK\_9()          pSKTick, nCode = m\_pSKOSQuote.SKOSQuoteLib\_GetTickNineDigitLONG(nIndex, nPtr, pSKTick)          msg = "【OnNotifyTicksNineDigitLONG】" + " 成交時間" + str(pSKTick.nTime) + " 成交價" + str(pSKTick.nClose) + " 成交量" + str(pSKTick.nQty) + " 成交日期YYYYMMDD" + str(pSKTick.nDate)          richTextBoxMessage.insert('end', msg + "\n")          richTextBoxMessage.see('end') |

#### OnNotifyHistoryTicksNineDigitLONG

|  |  |  |
| --- | --- | --- |
| 當**首次**索取個別海期商品成交明細，此事件會**回補當天Tick**。 | | |
| **宣告** | void OnNotifyHistoryTicksNineDigitLONG ([in] LONG nIndex, [in] LONG nPtr, [in] LONG nDate, [in] LONG nTime, [in] LONGLONG nClose, [in] LONG nQty); | |
|  | nIndex | 系統所編的海期商品索引代碼。 |
| nPtr | 表示資料的位址(Key) |
| nDate | 交易日期。(YYYYMMDD) |
| nTime | 時間。 |
| nClose | 成交價。 |
| nQty | 成交量。 |
| **備註** | 價格未進行小數點處理，開發人員需自行接手處理(SKFOREIGN\_9LONG物件內有小數點位數)。  避免在  [SKOSQuoteLib\_OnNotifyHistoryTicksNineDigitLONG](#_OnNotifyHistoryTicksNineDigitLONG)、  [SKOSQuoteLib\_OnNotifyTicksNineDigitLONG](#_OnNotifyTicksNineDigitLONG)  通知事件裡進行[SKOSQuoteLib\_GetTickNineDigitLONG](#_SKOSQuoteLib_GetTickNineDigitLONG)  ＊須使用[SKOSQuoteLib\_EnterMonitorLONG](#_SKOSQuoteLib_EnterMonitorLONG)登入，該事件才會被觸發。 | |

### OnNotifyBest5NineDigitLONG

|  |  |  |
| --- | --- | --- |
| 當有索取的個別海期商品五檔價格有所異動，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別海期商品五檔價格。 | | |
| **事件宣告** | void OnNotifyBest5NineDigitLONG ([in] LONG nStockidx, [in] LONGLONG nBestBid1, [in] LONG nBestBidQty1, [in] LONGLONG nBestBid2, [in] LONG nBestBidQty2, [in] LONGLONG nBestBid3, [in] LONG nBestBidQty3, [in] LONGLONG nBestBid4, [in] LONG nBestBidQty4, [in] LONGLONG nBestBid5, [in] LONG nBestBidQty5, [in] LONGLONG nBestAsk1, [in] LONG nBestAskQty1, [in] LONGLONG nBestAsk2, [in] LONG nBestAskQty2, [in] LONGLONG nBestAsk3, [in] LONG nBestAskQty3, [in] LONGLONG nBestAsk4, [in] LONG nBestAskQty4, [in] LONGLONG nBestAsk5, [in] LONG nBestAskQty5); | |
| **參數** | nStockidx | 系統所編的商品索引代碼。 |
| 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。 |
| 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。 |
| **備註** | 價格未進行小數點處理，開發人員需自行接手處理  (FOREIGN物件內有小數點位數與分母)。[SKOSQuote\_RequestTicks](#_SKOSQuoteLib_RequestTicks) 相關通知事件。  ＊須使用[SKOSQuoteLib\_EnterMonitorLONG](#_SKOSQuoteLib_EnterMonitorLONG)登入，該事件才會被觸發。 | |

|  |
| --- |
| Python範例程式碼\_OSQuote.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SKOSQuoteLibEvent註冊物件底下一層) |
| def OnNotifyBest5NineDigitLONG(self, nStockidx, nBestBid1, nBestBidQty1, nBestBid2, nBestBidQty2, nBestBid3, nBestBidQty3, nBestBid4, nBestBidQty4, nBestBid5, nBestBidQty5, nBestAsk1, nBestAskQty1, nBestAsk2, nBestAskQty2, nBestAsk3, nBestAskQty3, nBestAsk4, nBestAskQty4, nBestAsk5, nBestAskQty5):          pSKBest5 = sk.SKBEST5\_9()          pSKBest5, nCode = m\_pSKOSQuote.SKOSQuoteLib\_GetBest5NineDigitLONG(nStockidx, pSKBest5)          msg = "【OnNotifyBest5NineDigitLONG】" + " 1買量" + str(pSKBest5.nBidQty1) + " 1買價" + str(pSKBest5.nBid1 / 100) + " 1賣價" + str(pSKBest5.nAsk1 / 100) + " 1賣量" + str(pSKBest5.nAskQty1) + " 2買量" + str(pSKBest5.nBidQty2) + " 2買價" + str(pSKBest5.nBid2 / 100) + " 2賣價" + str(pSKBest5.nAsk2 / 100) + " 2賣量" + str(pSKBest5.nAskQty2)  + " 3買量" + str(pSKBest5.nBidQty3) + " 3買價" + str(pSKBest5.nBid3 / 100) + " 3賣價" + str(pSKBest5.nAsk3 / 100) + " 3賣量" + str(pSKBest5.nAskQty3)  + " 4買量" + str(pSKBest5.nBidQty4) + " 4買價" + str(pSKBest5.nBid4 / 100) + " 4賣價" + str(pSKBest5.nAsk4 / 100) + " 4賣量" + str(pSKBest5.nAskQty4)  + " 5買量" + str(pSKBest5.nBidQty5) + " 5買價" + str(pSKBest5.nBid5 / 100) + " 5賣價" + str(pSKBest5.nAsk5 / 100) + " 5賣量" + str(pSKBest5.nAskQty5)          richTextBoxMessage.insert('end', msg + "\n")          richTextBoxMessage.see('end') |

### OnNotifyBest10NineDigitLONG

|  |  |  |
| --- | --- | --- |
| 當有索取的個別海期商品十檔價格有所異動，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別海期十檔價格。 | | |
| **事件宣告** | void OnNotifyBest10NineDigitLONG ([in] LONG nStockidx, [in] LONGLONG nBestBid1, [in] LONG nBestBidQty1, [in] LONGLONG nBestBid2, [in] LONG nBestBidQty2, [in] LONGLONG nBestBid3, [in] LONG nBestBidQty3, [in] LONGLONG nBestBid4, [in] LONG nBestBidQty4, [in] LONGLONG nBestBid5, [in] LONG nBestBidQty5, [in] LONGLONG nBestBid6, [in] LONG nBestBidQty6, [in] LONGLONG nBestBid7, [in] LONG nBestBidQty7, [in] LONGLONG nBestBid8, [in] LONG nBestBidQty8, [in] LONGLONG nBestBid9, [in] LONG nBestBidQty9, [in] LONGLONG nBestBid10, [in] LONG nBestBidQty10, [in] LONGLONG nBestAsk1, [in] LONG nBestAskQty1, [in] LONGLONG nBestAsk2, [in] LONG nBestAskQty2, [in] LONGLONG nBestAsk3, [in] LONG nBestAskQty3, [in] LONGLONG nBestAsk4, [in] LONG nBestAskQty4, [in] LONGLONG nBestAsk5, [in] LONG nBestAskQty5, [in] LONGLONG nBestAsk6, [in] LONG nBestAskQty6, [in] LONGLONG nBestAsk7, [in] LONG nBestAskQty7, [in] LONGLONG nBestAsk8, [in] LONG nBestAskQty8, [in] LONGLONG nBestAsk9, [in] LONG nBestAskQty9, [in] LONGLONG nBestAsk10, [in] LONG nBestAskQty10); | |
| **參數** | nStockidx | 系統所編的商品索引代碼。 |
| 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。 |
| nBestBid6 | 第六檔買價。 |
| nBestBidQty6 | 第六檔買量。 |
| nBestBid7 | 第七檔買價。 |
| nBestBidQty7 | 第七檔買量。 |
| nBestBid8 | 第八檔買價。 |
| nBestBidQty8 | 第八檔買量。 |
| nBestBid9 | 第九檔買價。 |
| nBestBidQty9 | 第九檔買量。 |
| nBestBid10 | 第十檔買價。 |
| nBestBidQty10 | 第十檔買量。 |
| 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。 |
| nBestAsk6 | 第六檔賣價。 |
| nBestAskQty6 | 第六檔賣量。 |
| nBestAsk7 | 第七檔賣價。 |
| nBestAskQty7 | 第七檔賣量。 |
| nBestAsk8 | 第八檔賣價。 |
| nBestAskQty8 | 第八檔賣量。 |
| nBestAsk9 | 第九檔賣價。 |
| nBestAskQty9 | 第九檔賣量。 |
| nBestAsk10 | 第十檔賣價。 |
| nBestAskQty10 | 第十檔賣量。 |
| **備註** | 價格未進行小數點處理，開發人員需自行接手處理(FOREIGN物件內有小數點位數與分母)。[SKOSQuote\_RequestTicks](#_SKOSQuoteLib_RequestTicks) 相關通知事件。  ＊須使用[SKOSQuoteLib\_EnterMonitorLONG](#_SKOSQuoteLib_EnterMonitorLONG)登入，該事件才會被觸發。 | |

|  |
| --- |
| Python範例程式碼\_OSQuote.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SKOSQuoteLibEvent註冊物件底下一層) |
| def OnNotifyBest10NineDigitLONG(self, nStockidx, nBestBid1, nBestBidQty1, nBestBid2, nBestBidQty2, nBestBid3, nBestBidQty3, nBestBid4, nBestBidQty4, nBestBid5, nBestBidQty5, nBestBid6, nBestBidQty6, nBestBid7, nBestBidQty7, nBestBid8, nBestBidQty8, nBestBid9, nBestBidQty9, nBestBid10, nBestBidQty10, nBestAsk1, nBestAskQty1, nBestAsk2, nBestAskQty2, nBestAsk3, nBestAskQty3, nBestAsk4, nBestAskQty4, nBestAsk5, nBestAskQty5, nBestAsk6, nBestAskQty6, nBestAsk7, nBestAskQty7, nBestAsk8, nBestAskQty8, nBestAsk9, nBestAskQty9, nBestAsk10, nBestAskQty10):          msg = "【OnNotifyBest10NineDigitLONG】" + " 1買量" + str(nBestBidQty1) + " 1買價" + str(nBestBid1 / 100) + " 1賣價" + str(nBestAsk1 / 100) + " 1賣量" + str(nBestAskQty1) + " 2買量" + str(nBestBidQty2) + " 2買價" + str(nBestBid2 / 100) + " 2賣價" + str(nBestAsk2 / 100) + " 2賣量" + str(nBestAskQty2) + " 3買量" + str(nBestBidQty3) + " 3買價" + str(nBestBid3 / 100) + " 3賣價" + str(nBestAsk3 / 100) + " 3賣量" + str(nBestAskQty3) + " 4買量" + str(nBestBidQty4) + " 4買價" + str(nBestBid4 / 100) + " 4賣價" + str(nBestAsk4 / 100) + " 4賣量" + str(nBestAskQty4) + " 5買量" + str(nBestBidQty5) + " 5買價" + str(nBestBid5 / 100) + " 5賣價" + str(nBestAsk5 / 100) + " 5賣量" + str(nBestAskQty5)  + " 6買量" + str(nBestBidQty6) + " 6買價" + str(nBestBid6 / 100) + " 6賣價" + str(nBestAsk6 / 100) + " 6賣量" + str(nBestAskQty6) + " 7買量" + str(nBestBidQty7) + " 7買價" + str(nBestBid7 / 100) + " 7賣價" + str(nBestAsk7 / 100) + " 7賣量" + str(nBestAskQty7) + " 8買量" + str(nBestBidQty8) + " 8買價" + str(nBestBid8 / 100) + " 8賣價" + str(nBestAsk8 / 100) + " 8賣量" + str(nBestAskQty8) + " 9買量" + str(nBestBidQty9) + " 9買價" + str(nBestBid9 / 100) + " 9賣價" + str(nBestAsk9 / 100) + " 9賣量" + str(nBestAskQty9) + " 10買量" + str(nBestBidQty10) + " 10買價" + str(nBestBid10 / 100) + " 10賣價" + str(nBestAsk10 / 100) + " 10賣量" + str(nBestAskQty10)          richTextBoxMessage.insert('end', msg + "\n")          richTextBoxMessage.see('end') |

## 技術分析

### OnKLineData

|  |  |  |
| --- | --- | --- |
| 歴史K線資料。 | | |
| **宣告** | void OnKLineData([in] BSTR bstrStockNo, [in] BSTR bstrData); | |
| **參數** | bstrStockNo | 商品代碼 |
| bstrData | [日期(時間)], [開盤價],[最高價],[最低價],[收盤價],[ 成交量(張) 或 成交金額 ] |
| **備註** | 不提供開盤參考價 | |

|  |
| --- |
| Python範例程式碼\_OSQuote.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SKOSQuoteLibEvent註冊物件底下一層) |
| def OnKLineData(self, bstrStockNo, bstrData):          msg = "【OnKLineData】" + " 商品代號" + bstrStockNo + "\_" + bstrData          richTextBoxMessage.insert('end', msg + "\n")          richTextBoxMessage.see('end') |

# Struct結構物件

## SKFOREIGNLONG ( 海外報價商品物件 LONG index )

struct SKFOREIGNLONG

{

LONG nStockidx; // 系統所編的海外商品索引代碼

SHORT sDecimal; // 報價小數位數

LONG nDenominator; // 分母

BSTR bstrMarketNo; // 市場代碼

BSTR bstrExchangeNo; // 交易所代號

BSTR bstrExchangeName; // 交易所名稱

BSTR bstrStockNo; // 商品代號

BSTR bstrStockName; // 商品名稱

BSTR bstrCallPut; // CallPut

LONG nOpen; // 開盤價

LONG nHigh; // 最高價

LONG nLow; // 最低價

LONG nClose; // 成交價

LONG nSettlePrice; // 結算價

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 nTQty; // 成交量( EXCEL x86版本)

LONG nStrikePrice; // 履約價

LONG nTradingDay // 交易日(YYYYMMDD)

備註 :當日非交易日時, 資料為前一交易日

};//以上價格欄，需自行處理小數，參考『報價小數位數 sDecimal』

## SKFOREIGN\_9LONG ( 海外報價商品物件-九位擴充 LONG index )

struct SKFOREIGN\_9LONG

{

LONG nStockidx; // 系統所編的海外商品索引代碼

SHORT sDecimal; // 報價小數位數

LONG nDenominator; // 分母

BSTR bstrMarketNo; // 市場代碼

BSTR bstrExchangeNo; // 交易所代號

BSTR bstrExchangeName; // 交易所名稱

BSTR bstrStockNo; // 商品代號

BSTR bstrStockName; // 商品名稱

BSTR bstrCallPut; // CallPut

LONGLONG nOpen; // 開盤價

LONGLONG nHigh; // 最高價

LONGLONG nLow; // 最低價

LONGLONG nClose; // 成交價

LONGLONG nSettlePrice; // 結算價

LONG nTickQty; // 單量

LONGLONG nRef; // 昨收、參考價

LONGLONG nBid; // 買價

LONG nBc; // 買量

LONGLONG nAsk; // 賣價

LONG nAc; // 賣量

LONGLONG nTQty; // 成交量

LONG nStrikePrice; // 履約價

LONG nTradingDay // 交易日(YYYYMMDD)

備註 :當日非交易日時, 資料為前一交易日

};

//以上價格欄，需自行處理小數，參考『報價小數位數 sDecimal』

## SKFOREIGNTICK\_9( 海外報價TICK物件-九位擴充 )

struct SKFOREIGNTICK\_9

{

LONG nPtr; //Index

LONG nTime; //成交時間

LONGLONG nClose; //成交價

//以上價格欄，需自行處理小數，參考SKFOREIGN『報價小數位數 sDecimal』

LONG nQty; //成交量

LONG nDate; //成交日期YYYYMMDD，依交易所時區為主

};

## SKBEST5\_9( 五檔價格物件-九位擴充 )

typedef struct BEST5\_9

{

LONG LONG nBid1; // 第一檔買價

LONG nBidQty1; // 第一檔買量

LONG LONG nBid2; // 第二檔買價

LONG nBidQty2; // 第二檔買量

LONG LONG nBid3; // 第三檔買價

LONG nBidQty3; // 第三檔買量

LONG LONG nBid4; // 第四檔買價

LONG nBidQty4; // 第四檔買量

LONG LONG nBid5; // 第五檔買價

LONG nBidQty5; // 第五檔買量

LONG 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 LONG nAsk1; // 第一檔賣價

LONG nAskQty1; // 第一檔賣量

LONG LONG nAsk2; // 第二檔賣價

LONG nAskQty2; // 第二檔賣量

LONG LONG nAsk3; // 第三檔賣價

LONG nAskQty3; // 第三檔賣量

LONG LONG nAsk4; // 第四檔賣價

LONG nAskQty4; // 第四檔買量

LONG LONG nAsk5; // 第五檔賣價

LONG nAskQty5; // 第五檔賣量

LONG 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

}

//以上價格欄，需自行處理小數，參考SKFOREIGN『報價小數位數 sDecimal』