策略王COM元件使用說明

-下單-複委託-

## 函式範例

### SKOrderLib (複委託)

*函式*

1. **一般下單/刪單**

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| **複委託下單** | [SendForeignStockProxyOrder](#_SendForeignStockOrder) |  |
| **複委託下單(含單獨自訂資料欄)** | [SendForeignStockOrderOLID](#_SendForeignStockOrderOLID) |  |
| **複委託刪單** | [SendForeignStockProxyCancel](#_CancelForeignStockOrder) |  |

1. **Proxy下單/刪單**

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| **複委託** | [SendForeignStockOrder](#_SendForeignStockProxyOrder) |  |
| **(新)複委託刪單** | [CancelForeignStockOrder](#_SendForeignStockProxyCancel) |  |

1. **帳務查詢**

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| **複委託外幣圈存查詢** | [GetBankBlock](#_4-2-112_GetBankBlock) |  |
| **複委託台幣圈存查詢** | [GetNTDBlock](#_4-2-113_GetNTDBlock) |  |

*事件*

|  |  |  |
| --- | --- | --- |
| **功能** | **事件名稱** | **備註** |
| **非同步委託結果** | [OnAsyncOrder](#_4-2-b_OnAsyncOrder) |  |
| **非同步委託結果。(含單獨自訂資料欄)** | [OnAsyncOrderOLID](#_OnProxyOrder) |  |
| **Proxy委託結果** | [OnProxyOrder](#_OnProxyOrder) |  |

#### **一般下單/刪單**

##### SendForeignStockOrder

|  |  |  |
| --- | --- | --- |
| 送出複委託。 | | |
| **宣告** | Long SendForeignStockOrder([in] BSTR bstrLogInID, [in] VARIANT\_BOOL bAsyncOrder, [in] struct **FOREIGNORDER**\* pAsyncOrder, [**out**] BSTR\* bstrMessage); | |
| **參數** | bstrLogInID | 登入ID。 |
| bAsyncOrder | 是否為非同步委託。 |
| pAsyncOrder | SKCOM元件中的**[FOREIGNORDER](#_複委託下單)**物件，將下單條件填入該物件後，再帶入此欄位中。 |
| bstrMessage | **同步委託：**如果回傳值為 0表示委託成功，訊息內容則為13碼的委託序號。回傳值非0表示委託失敗，訊息內容為失敗原因。  **非同步委託：**參照OnAsyncOrder。 |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。  **當錯誤代碼為4碼時，可參考文件「4.下單準備介紹」裡面的【錯誤代碼定義表】，所屬委託物件亦有欄位值說明供參。**  **其他錯誤則由「交易主機」回傳錯誤代碼及錯誤原因。若仍有疑問請洽詢您所屬營業員。** | |
| **備註** | 使用非同步委託，委託結果請由[OnAsyncOrder](#_OnAsyncOrder)取得。  Ver 2.13.46版本以上(含) 新增幣別CNY、GBP；下單市場新增滬深股 (C#範例程式另加港股、日股、新加坡下單選項) | |

|  |
| --- |
| Python範例程式碼\_OSOrder.py類別SendForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # checkBoxAsyncOrder          # 是否為非同步委託          self.checkBoxAsyncOrder = tk.Checkbutton(self)          self.var1 = tk.IntVar()          self.checkBoxAsyncOrder["variable"] = self.var1          self.checkBoxAsyncOrder["onvalue"] = True          self.checkBoxAsyncOrder["offvalue"] = False          self.checkBoxAsyncOrder["text"] = "非同步委託"          self.checkBoxAsyncOrder["command"] = self.checkBoxAsyncOrder\_CheckedChanged          self.checkBoxAsyncOrder.grid( row = 0,column = 0)          # textBoxForeignStockID          tk.Label(self, text = "股票代號").grid(row=1, column=1)              #輸入框          self.textBoxForeignStockID = tk.Entry(self)          self.textBoxForeignStockID.grid(row=1, column=2)          global textBoxForeignStockID          textBoxForeignStockID = self.textBoxForeignStockID            # textBoxForeignExchangeNo          tk.Label(self, text = "交易所代碼，美股：US").grid(row=2, column=1)              #輸入框          self.textBoxForeignExchangeNo = tk.Entry(self)          self.textBoxForeignExchangeNo.grid(row=2, column=2)          global textBoxForeignExchangeNo          textBoxForeignExchangeNo = self.textBoxForeignExchangeNo            # comboBoxForeignAccountType          tk.Label(self, text = "專戶別種類").grid(row=3, column=1)              #輸入框          self.comboBoxForeignAccountType = ttk.Combobox(self, state='readonly')          self.comboBoxForeignAccountType['values'] = Config.comboBoxForeignAccountType          self.comboBoxForeignAccountType.grid(row=3, column=2)          global comboBoxForeignAccountType          comboBoxForeignAccountType = self.comboBoxForeignAccountType            # textBoxForeignCurrency1          tk.Label(self, text = "扣款幣別1").grid(row=4, column=1)              #輸入框          self.textBoxForeignCurrency1 = tk.Entry(self)          self.textBoxForeignCurrency1.grid(row=4, column=2)          global textBoxForeignCurrency1          textBoxForeignCurrency1 = self.textBoxForeignCurrency1            # textBoxForeignCurrency2          tk.Label(self, text = "扣款幣別2").grid(row=5, column=1)              #輸入框          self.textBoxForeignCurrency2 = tk.Entry(self)          self.textBoxForeignCurrency2.grid(row=5, column=2)          global textBoxForeignCurrency2          textBoxForeignCurrency2 = self.textBoxForeignCurrency2            # textBoxForeignCurrency3          tk.Label(self, text = "扣款幣別3").grid(row=6, column=1)              #輸入框          self.textBoxForeignCurrency3 = tk.Entry(self)          self.textBoxForeignCurrency3.grid(row=6, column=2)          global textBoxForeignCurrency3          textBoxForeignCurrency3 = self.textBoxForeignCurrency3            # textBoxForeignQty          tk.Label(self, text = "委託量").grid(row=7, column=1)              #輸入框          self.textBoxForeignQty = tk.Entry(self)          self.textBoxForeignQty.grid(row=7, column=2)          global textBoxForeignQty          textBoxForeignQty = self.textBoxForeignQty            # textBoxForeignPrice          tk.Label(self, text = "委託價").grid(row=8, column=1)              #輸入框          self.textBoxForeignPrice = tk.Entry(self)          self.textBoxForeignPrice.grid(row=8, column=2)          global textBoxForeignPrice          textBoxForeignPrice = self.textBoxForeignPrice            # comboBoxForeignOrderType          tk.Label(self, text = "買進/賣出").grid(row=9, column=1)              #輸入框          self.comboBoxForeignOrderType = ttk.Combobox(self, state='readonly')          self.comboBoxForeignOrderType['values'] = Config.comboBoxForeignOrderType          self.comboBoxForeignOrderType.grid(row=9, column=2)          global comboBoxForeignOrderType          comboBoxForeignOrderType = self.comboBoxForeignOrderType            # comboBoxForeignTradeType          tk.Label(self, text = "庫存別").grid(row=10, column=1)              #輸入框          self.comboBoxForeignTradeType = ttk.Combobox(self, state='readonly')          self.comboBoxForeignTradeType['values'] = Config.comboBoxForeignTradeType          self.comboBoxForeignTradeType.grid(row=10, column=2)          global comboBoxForeignTradeType          comboBoxForeignTradeType = self.comboBoxForeignTradeType            # buttonSendForeignStockOrder          self.buttonSendForeignStockOrder = tk.Button(self)          self.buttonSendForeignStockOrder["text"] = "複委託送出"          self.buttonSendForeignStockOrder["command"] = self.buttonSendForeignStockOrder\_Click          self.buttonSendForeignStockOrder.grid(row=11, column=1) |

|  |
| --- |
| Config.py |
| # SendForm  comboBoxForeignAccountType =(      "外幣專戶", "台幣專戶"  )  comboBoxForeignOrderType =(      "買", "賣"  )  comboBoxForeignTradeType =(      "一般/定股(CITI)", "定額(VIEWTRADE)"  ) |
| 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\_pSKOrder = comtypes.client.CreateObject(sk.SKOrderLib,interface=sk.ISKOrderLib)  # 是否為非同步委託  bAsyncOrder = False |

|  |
| --- |
| Current Block(放置於def createWidgets(self): 的同一層) |
| # checkBoxAsyncOrder      def checkBoxAsyncOrder\_CheckedChanged(self):          global bAsyncOrder          if self.var1.get() == True:              bAsyncOrder = True          else:              bAsyncOrder = False      def buttonSendForeignStockOrder\_Click(self):            pOrder = sk.FOREIGNORDER()            pOrder.bstrFullAccount = comboBoxAccount.get()          pOrder.bstrStockNo = textBoxForeignStockID.get()          pOrder.bstrExchangeNo = textBoxForeignExchangeNo.get()          pOrder.bstrPrice = textBoxForeignPrice.get()          pOrder.bstrCurrency1 = textBoxForeignCurrency1.get()          pOrder.bstrCurrency2 = textBoxForeignCurrency2.get()          pOrder.bstrCurrency3 = textBoxForeignCurrency3.get()          if (comboBoxForeignAccountType.get() == "外幣專戶"):              pOrder.nAccountType = 1          elif (comboBoxForeignAccountType.get() == "台幣專戶"):              pOrder.nAccountType = 2          pOrder.nQty = int(textBoxForeignQty.get())          if (comboBoxForeignOrderType.get() == "買"):              pOrder.nOrderType = 1          elif (comboBoxForeignOrderType.get() == "賣"):              pOrder.nOrderType = 2          if (comboBoxForeignTradeType.get() == "一般/定股(CITI)"):              pOrder.nTradeType = 1          elif (comboBoxForeignTradeType.get() == "定額(VIEWTRADE)"):              pOrder.nTradeType = 2            # 送出複委託委託          bstrMessage,nCode= m\_pSKOrder.SendForeignStockOrder(comboBoxUserID.get(), bAsyncOrder, pOrder)          msg = "【SendForeignStockOrder】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode) + bstrMessage          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

##### SendForeignStockOrderOLID

|  |  |  |
| --- | --- | --- |
| 送出複委託。(含單獨自訂資料欄) | | |
| **宣告** | Long SendForeignStockOrderOLID([in] BSTR bstrLogInID, [in] VARIANT\_BOOL bAsyncOrder, [in] struct **FOREIGNORDER**\* pAsyncOrder, [in] BSTR bstrOrderLinkedID, [**out**] BSTR\* bstrMessage); | |
| **參數** | bstrLogInID | 登入ID。 |
| bAsyncOrder | 是否為非同步委託。 |
| pAsyncOrder | SKCOM元件中的**[FOREIGNORDER](#_複委託下單)**物件，將下單條件填入該物件後，再帶入此欄位中。 |
| bstrOrderLinkedID | 僅非同步委託有效。  客戶自訂資料，會在OnAsyncOrderOLID返回。 |
| bstrMessage | **同步委託：**如果回傳值為 0表示委託成功，訊息內容則為13碼的委託序號。回傳值非0表示委託失敗，訊息內容為失敗原因。  **非同步委託：**參照OnAsyncOrderOLID。 |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。  **當錯誤代碼為4碼時，可參考文件「4.下單準備介紹」裡面的【錯誤代碼定義表】，所屬委託物件亦有欄位值說明供參。**  **其他錯誤則由「交易主機」回傳錯誤代碼及錯誤原因。若仍有疑問請洽詢您所屬營業員。** | |
| **備註** | 使用非同步委託，委託結果請由[OnAsyncOrderOLID](#_OnProxyOrder)取得。  Ver 2.13.46版本以上(含) 新增幣別CNY、GBP；下單市場新增滬深股 (C#範例程式另加港股、日股、新加坡下單選項) | |

##### CancelForeignStockOrder

|  |  |  |
| --- | --- | --- |
| 新版-複委託刪單(需同時填序號及委託書號)。 | | |
| **宣告** | Long CancelForeignStockOrder([in] BSTR bstrLogInID, [in] VARIANT\_BOOL bAsyncOrder, [in] struct FOREIGNORDER\* pOrder, [**out**] BSTR\* bstrMessage); | |
| **參數** | bstrLogInID | 登入ID。 |
| bAsyncOrder | 是否為非同步委託。 |
| pAsyncOrder | SKCOM元件中的**[FOREIGNORDER](#_複委託刪單)**物件，將刪單條件填入該物件後，再帶入此欄位中。 |
| bstrMessage | **同步刪單：**如果回傳值為 0表示刪單成功，訊息內容則為修改訊息。回傳值非0表示刪單失敗，訊息內容為失敗原因。  **非同步刪單：**參照OnAsyncOrder。 |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。 | |
| **備註** | 回傳值0 表示委託伺服器接收成功，詳細委託狀態仍須以委託**刪單**回報內容為主。  其他非 0 數值表示異常，詳細原因請參考 Log 內容說明。  使用非同步委託，委託結果請由[OnAsyncOrder](#_4-2-b_OnAsyncOrder)取得。 | |

|  |
| --- |
| Python範例程式碼\_OSOrder.py類別UpdateForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # checkBoxAsyncOrder          # 是否為非同步委託          self.checkBoxAsyncOrder = tk.Checkbutton(self)          self.var1 = tk.IntVar()          self.checkBoxAsyncOrder["variable"] = self.var1          self.checkBoxAsyncOrder["onvalue"] = True          self.checkBoxAsyncOrder["offvalue"] = False          self.checkBoxAsyncOrder["text"] = "非同步委託"          self.checkBoxAsyncOrder["command"] = self.checkBoxAsyncOrder\_CheckedChanged          self.checkBoxAsyncOrder.grid( row = 0,column = 0)          # textBoxCancelForeignStockOrderbstrSeqNoUpdate          tk.Label(self, text = "請輸入委託序號").grid(row=0, column=1)              #輸入框          self.textBoxCancelForeignStockOrderbstrSeqNoUpdate = tk.Entry(self)          self.textBoxCancelForeignStockOrderbstrSeqNoUpdate.grid(row=0, column=2)          global textBoxCancelForeignStockOrderbstrSeqNoUpdate          textBoxCancelForeignStockOrderbstrSeqNoUpdate = self.textBoxCancelForeignStockOrderbstrSeqNoUpdate            # textBoxCancelForeignStockOrderbstrBookNoUpdate          tk.Label(self, text = "請輸入委託書號").grid(row=1, column=1)              #輸入框          self.textBoxCancelForeignStockOrderbstrBookNoUpdate = tk.Entry(self)          self.textBoxCancelForeignStockOrderbstrBookNoUpdate.grid(row=1, column=2)          global textBoxCancelForeignStockOrderbstrBookNoUpdate          textBoxCancelForeignStockOrderbstrBookNoUpdate = self.textBoxCancelForeignStockOrderbstrBookNoUpdate            # textBoxForeignStockIDUpdate          tk.Label(self, text = "股票代號").grid(row=2, column=1)              #輸入框          self.textBoxForeignStockIDUpdate = tk.Entry(self)          self.textBoxForeignStockIDUpdate.grid(row=2, column=2)          global textBoxForeignStockIDUpdate          textBoxForeignStockIDUpdate = self.textBoxForeignStockIDUpdate            # textBoxbstrExchangeNoUpdate          tk.Label(self, text = "交易所代碼，美股：US").grid(row=3, column=1)              #輸入框          self.textBoxbstrExchangeNoUpdate = tk.Entry(self)          self.textBoxbstrExchangeNoUpdate.grid(row=3, column=2)          global textBoxbstrExchangeNoUpdate          textBoxbstrExchangeNoUpdate = self.textBoxbstrExchangeNoUpdate          # buttonCancelForeignStockOrder          self.buttonCancelForeignStockOrder = tk.Button(self)          self.buttonCancelForeignStockOrder["text"] = "刪單(序號+書號)"          self.buttonCancelForeignStockOrder["command"] = self.buttonCancelForeignStockOrder\_Click          self.buttonCancelForeignStockOrder.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\_pSKOrder = comtypes.client.CreateObject(sk.SKOrderLib,interface=sk.ISKOrderLib)  # 是否為非同步委託  bAsyncOrder = False |

|  |
| --- |
| Current Block(放置於def createWidgets(self): 的同一層) |
| # checkBoxAsyncOrder      def checkBoxAsyncOrder\_CheckedChanged(self):          global bAsyncOrder          if self.var1.get() == True:              bAsyncOrder = True          else:              bAsyncOrder = False        def buttonCancelForeignStockOrder\_Click(self):          pOrder = sk.FOREIGNORDER()          pOrder.bstrStockNo = textBoxForeignStockIDUpdate.get()          pOrder.bstrExchangeNo = textBoxbstrExchangeNoUpdate.get()          pOrder.bstrFullAccount = comboBoxAccount.get()          pOrder.bstrSeqNo = textBoxCancelForeignStockOrderbstrSeqNoUpdate.get()          pOrder.bstrBookNo = textBoxCancelForeignStockOrderbstrBookNoUpdate.get()          pOrder.nOrderType = 4            # 新版-複委託刪單(需同時填序號及委託書號)          bstrMessage,nCode= m\_pSKOrder.CancelForeignStockOrder(comboBoxUserID.get(), bAsyncOrder, pOrder)          msg = "【CancelForeignStockOrder】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode) + bstrMessage          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

#### **Proxy下單/刪單**

##### SendForeignStockProxyOrder

|  |  |  |
| --- | --- | --- |
| 經由proxy server送出複委託下單。 | | |
| **宣告** | Long SendForeignStockProxyOrder([in] BSTR bstrLogInID, [in] struct OSSTOCKPROXYORDER\* pAsyncOrder, [out] BSTR\* bstrMessage); | |
| **參數** | bstrLogInID | 登入ID。 |
| pAsyncOrder | SKCOM元件中的[OSSTOCKPROXYORDER](#_Proxy_Server複委託下單)物件，將下單條件填入該物件後，再帶入此欄位中。 |
| bstrMessage | **委託送出成功：ORKEY**  **委託送出失敗：錯誤訊息** |
| **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** | 透過Proxy Server委託，委託結果請由[OnProxyOrder](#_OnProxyOrder)取得。  連線且成功登入，通知OnProxyStatus為5001時，送至proxy server進行下單。 | |

|  |
| --- |
| Python範例程式碼\_OSOrder.py類別SendProxyForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # textBoxForeignStockIDProxy          tk.Label(self, text = "股票代號").grid(row=0, column=1)              #輸入框          self.textBoxForeignStockIDProxy = tk.Entry(self)          self.textBoxForeignStockIDProxy.grid(row=0, column=2)          global textBoxForeignStockIDProxy          textBoxForeignStockIDProxy = self.textBoxForeignStockIDProxy            # comboBoxbstrExchangeNoProxy          tk.Label(self, text = "交易所代碼，美股：US").grid(row=1, column=1)              #輸入框          self.comboBoxbstrExchangeNoProxy = ttk.Combobox(self, state='readonly')          self.comboBoxbstrExchangeNoProxy['values'] = Config.comboBoxbstrExchangeNoProxy          self.comboBoxbstrExchangeNoProxy.grid(row=1, column=2)          global comboBoxbstrExchangeNoProxy          comboBoxbstrExchangeNoProxy = self.comboBoxbstrExchangeNoProxy            # comboBoxForeignAccountTypeProxy          tk.Label(self, text = "專戶別種類").grid(row=2, column=1)              #輸入框          self.comboBoxForeignAccountTypeProxy = ttk.Combobox(self, state='readonly')          self.comboBoxForeignAccountTypeProxy['values'] = Config.comboBoxForeignAccountTypeProxy          self.comboBoxForeignAccountTypeProxy.grid(row=2, column=2)          global comboBoxForeignAccountTypeProxy          comboBoxForeignAccountTypeProxy = self.comboBoxForeignAccountTypeProxy            # textBoxForeignCurrency1Proxy          tk.Label(self, text = "扣款幣別1").grid(row=3, column=1)              #輸入框          self.textBoxForeignCurrency1Proxy = tk.Entry(self)          self.textBoxForeignCurrency1Proxy.grid(row=3, column=2)          global textBoxForeignCurrency1Proxy          textBoxForeignCurrency1Proxy = self.textBoxForeignCurrency1Proxy            # textBoxForeignCurrency2Proxy          tk.Label(self, text = "扣款幣別2").grid(row=4, column=1)              #輸入框          self.textBoxForeignCurrency2Proxy = tk.Entry(self)          self.textBoxForeignCurrency2Proxy.grid(row=4, column=2)          global textBoxForeignCurrency2Proxy          textBoxForeignCurrency2Proxy = self.textBoxForeignCurrency2Proxy            # textBoxForeignCurrency3Proxy          tk.Label(self, text = "扣款幣別3").grid(row=5, column=1)              #輸入框          self.textBoxForeignCurrency3Proxy = tk.Entry(self)          self.textBoxForeignCurrency3Proxy.grid(row=5, column=2)          global textBoxForeignCurrency3Proxy          textBoxForeignCurrency3Proxy = self.textBoxForeignCurrency3Proxy            # textBoxForeignQtyProxy          tk.Label(self, text = "委託量").grid(row=6, column=1)              #輸入框          self.textBoxForeignQtyProxy = tk.Entry(self)          self.textBoxForeignQtyProxy.grid(row=6, column=2)          global textBoxForeignQtyProxy          textBoxForeignQtyProxy = self.textBoxForeignQtyProxy            # textBoxForeignPriceProxy          tk.Label(self, text = "委託價").grid(row=7, column=1)              #輸入框          self.textBoxForeignPriceProxy = tk.Entry(self)          self.textBoxForeignPriceProxy.grid(row=7, column=2)          global textBoxForeignPriceProxy          textBoxForeignPriceProxy = self.textBoxForeignPriceProxy            # comboBoxForeignOrderTypeProxy          tk.Label(self, text = "買進/賣出").grid(row=8, column=1)              #輸入框          self.comboBoxForeignOrderTypeProxy = ttk.Combobox(self, state='readonly')          self.comboBoxForeignOrderTypeProxy['values'] = Config.comboBoxForeignOrderTypeProxy          self.comboBoxForeignOrderTypeProxy.grid(row=8, column=2)          global comboBoxForeignOrderTypeProxy          comboBoxForeignOrderTypeProxy = self.comboBoxForeignOrderTypeProxy            # comboBoxForeignTradeTypeProxy          tk.Label(self, text = "庫存別").grid(row=9, column=1)              #輸入框          self.comboBoxForeignTradeTypeProxy = ttk.Combobox(self, state='readonly')          self.comboBoxForeignTradeTypeProxy['values'] = Config.comboBoxForeignTradeTypeProxy          self.comboBoxForeignTradeTypeProxy.grid(row=9, column=2)          global comboBoxForeignTradeTypeProxy          comboBoxForeignTradeTypeProxy = self.comboBoxForeignTradeTypeProxy          # buttonSendForeignStockProxyOrder          self.buttonSendForeignStockProxyOrder = tk.Button(self)          self.buttonSendForeignStockProxyOrder["text"] = "複委託送出"          self.buttonSendForeignStockProxyOrder["command"] = self.buttonSendForeignStockProxyOrder\_Click          self.buttonSendForeignStockProxyOrder.grid(row=10, column=1) |

|  |
| --- |
| Config.py |
| # SendProxyForm  comboBoxbstrExchangeNoProxy =(      "US：美股", "HK：港股", "JP：日股", "SP：新加坡", "SG：新(幣)坡股", "SA: 滬股", "HA: 深股"  )  comboBoxForeignAccountTypeProxy =(      "外幣專戶", "台幣專戶"  )  comboBoxForeignOrderTypeProxy =(      "買", "賣"  )  comboBoxForeignTradeTypeProxy =(      "一般/定股(CITI)", "定額(VIEWTRADE)"  ) |
| 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\_pSKOrder = comtypes.client.CreateObject(sk.SKOrderLib,interface=sk.ISKOrderLib) |

|  |
| --- |
| Current Block(放置於def createWidgets(self): 的同一層) |
| def buttonSendForeignStockProxyOrder\_Click(self):          pAsyncOrder = sk.OSSTOCKPROXYORDER()          pAsyncOrder.bstrFullAccount = comboBoxAccount.get()          pAsyncOrder.bstrStockNo = textBoxForeignStockIDProxy.get()          if (comboBoxbstrExchangeNoProxy.get() == "US：美股"):              pAsyncOrder.bstrExchangeNo = "US"          elif (comboBoxbstrExchangeNoProxy.get() == "HK：港股"):              pAsyncOrder.bstrExchangeNo = "HK"          elif (comboBoxbstrExchangeNoProxy.get() == "JP：日股"):              pAsyncOrder.bstrExchangeNo = "JP"          elif (comboBoxbstrExchangeNoProxy.get() == "SP：新加坡"):              pAsyncOrder.bstrExchangeNo = "SP"          elif (comboBoxbstrExchangeNoProxy.get() == "SG：新(幣)坡股"):              pAsyncOrder.bstrExchangeNo = "SG"          elif (comboBoxbstrExchangeNoProxy.get() == "SA: 滬股"):              pAsyncOrder.bstrExchangeNo = "SA"          elif (comboBoxbstrExchangeNoProxy.get() == "HA: 深股"):              pAsyncOrder.bstrExchangeNo = "HA"          selectedValue = comboBoxForeignAccountTypeProxy.get()          if (selectedValue == "外幣專戶"):              pAsyncOrder.nAccountType = 1          elif (selectedValue == "台幣專戶"):              pAsyncOrder.nAccountType = 2          pAsyncOrder.bstrCurrency1 = textBoxForeignCurrency1Proxy.get()          pAsyncOrder.bstrCurrency2 = textBoxForeignCurrency2Proxy.get()          pAsyncOrder.bstrCurrency3 = textBoxForeignCurrency3Proxy.get()          pAsyncOrder.bstrProxyQty = textBoxForeignQtyProxy.get()          pAsyncOrder.bstrPrice = textBoxForeignPriceProxy.get()          selectedValue = comboBoxForeignOrderTypeProxy.get()          if (selectedValue == "買"):              pAsyncOrder.nOrderType = 1          elif (selectedValue == "賣"):              pAsyncOrder.nOrderType = 2          selectedValue = comboBoxForeignTradeTypeProxy.get()          if (selectedValue == "一般/定股(CITI)"):              pAsyncOrder.nTradeType = 1          elif (selectedValue == "定額(VIEWTRADE)"):              pAsyncOrder.nTradeType = 2            # 經由proxy server送出複委託下單          bstrMessage,nCode= m\_pSKOrder.SendForeignStockProxyOrder(comboBoxUserID.get(), pAsyncOrder)            if bstrMessage is not None:              msg = "【SendForeignStockProxyOrder】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode) + bstrMessage          else:              msg = "【SendForeignStockProxyOrder】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode) + " No message"          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

##### SendForeignStockProxyCancel

|  |  |  |
| --- | --- | --- |
| 經由proxy server送出複委託刪單。 | | |
| **宣告** | Long SendForeignStockProxyCancel([in] BSTR bstrLogInID, [in] struct OSSTOCKPROXYORDER \* pAsyncOrder, [out] BSTR\* bstrMessage); | |
| **參數** | bstrLogInID | 登入ID。 |
| pAsyncOrder | SKCOM元件中的[OSSTOCKPROXYORDER](#_Proxy_Server複委託刪單)物件，將下單條件填入該物件後，再帶入此欄位中。 |
| bstrMessage | **委託送出成功：ORKEY**  **委託送出失敗：錯誤訊息** |
| **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** | 使用非同步委託，委託結果請由[OnAsyncOrder](#_4-2-b_OnAsyncOrder)取得。  在有連上且成功登入proxy server的狀態下，會由proxy server進行非同步下單。 | |

|  |
| --- |
| Python範例程式碼\_OSOrder.py類別UpdateProxyForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # textBoxCancelForeignStockOrderbstrSeqNo          tk.Label(self, text = "請輸入委託序號").grid(row=0, column=1)              #輸入框          self.textBoxCancelForeignStockOrderbstrSeqNo = tk.Entry(self)          self.textBoxCancelForeignStockOrderbstrSeqNo.grid(row=0, column=2)          global textBoxCancelForeignStockOrderbstrSeqNo          textBoxCancelForeignStockOrderbstrSeqNo = self.textBoxCancelForeignStockOrderbstrSeqNo            # textBoxCancelForeignStockOrderbstrBookNo          tk.Label(self, text = "請輸入委託書號").grid(row=1, column=1)              #輸入框          self.textBoxCancelForeignStockOrderbstrBookNo = tk.Entry(self)          self.textBoxCancelForeignStockOrderbstrBookNo.grid(row=1, column=2)          global textBoxCancelForeignStockOrderbstrBookNo          textBoxCancelForeignStockOrderbstrBookNo = self.textBoxCancelForeignStockOrderbstrBookNo            # textBoxbstrStockNo          tk.Label(self, text = "股票代號").grid(row=2, column=1)              #輸入框          self.textBoxbstrStockNo = tk.Entry(self)          self.textBoxbstrStockNo.grid(row=2, column=2)          global textBoxbstrStockNo          textBoxbstrStockNo = self.textBoxbstrStockNo            # textBoxbstrExchangeNo          tk.Label(self, text = "交易所代碼，美股：US").grid(row=3, column=1)              #輸入框          self.textBoxbstrExchangeNo = tk.Entry(self)          self.textBoxbstrExchangeNo.grid(row=3, column=2)          global textBoxbstrExchangeNo          textBoxbstrExchangeNo = self.textBoxbstrExchangeNo          # buttonSendForeignStockProxyCancel          self.buttonSendForeignStockProxyCancel = tk.Button(self)          self.buttonSendForeignStockProxyCancel["text"] = "刪單送出"          self.buttonSendForeignStockProxyCancel["command"] = self.buttonSendForeignStockProxyCancel\_Click          self.buttonSendForeignStockProxyCancel.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\_pSKOrder = comtypes.client.CreateObject(sk.SKOrderLib,interface=sk.ISKOrderLib) |

|  |
| --- |
| Current Block(放置於def createWidgets(self): 的同一層) |
| def buttonSendForeignStockProxyCancel\_Click(self):          pOrder = sk.OSSTOCKPROXYORDER()          pOrder.bstrStockNo = textBoxbstrStockNo.get()          pOrder.bstrExchangeNo = textBoxbstrExchangeNo.get()          pOrder.bstrFullAccount = comboBoxAccount.get()          pOrder.bstrSeqNo = textBoxCancelForeignStockOrderbstrSeqNo.get()          pOrder.bstrBookNo = textBoxCancelForeignStockOrderbstrBookNo.get()          pOrder.nOrderType = 4            # 經由proxy server送出複委託刪單          bstrMessage,nCode= m\_pSKOrder.SendForeignStockProxyCancel(comboBoxUserID.get(), pOrder)            if bstrMessage is not None:              msg = "【SendForeignStockProxyCancel】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode) + bstrMessage          else:              msg = "【SendForeignStockProxyCancel】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode) + " No message"          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

#### **帳務查詢**

##### GetBankBlock

|  |  |  |
| --- | --- | --- |
| 複委託外幣圈存查詢\_一戶通 | | |
| **宣告** | BSTR GetBankBlock ([in] BSTR bstrLogInID, [in] BSTR bstrAccount,) | |
| **參數** | bstrLogInID | 登入ID。 |
| bstrAccount | 複委託帳號，分公司四碼＋帳號7碼。 |
| **回傳值** | 欄位以「,」分隔，結尾為「#」，欄位依序為：  格式：   |  |  | | --- | --- | | 1 | 幣別 | | 2 | 圈存金額 | | 3 | 買進未扣款 | | 4 | 買賣別 | | 5 | 股市委買 | | 6 | 匯出換匯金額 | | 7 | 可解圈金額 | | 8 | 基金委買 | | 9 | 銀行代碼 | | 10 | 銀行分行代號 | | 11 | 銀行帳號 | | 12 | 銀行名稱 |   EX: CNY,5000000.00,1070.08,0.00,452.45,4998477.47,0.00,011,002,11111111111111,上銀營 | |
| **備註** |  | |

##### GetNTDBlock

|  |  |  |
| --- | --- | --- |
| 複委託台幣圈存查詢\_一戶通 | | |
| **宣告** | BSTR GetBankBlock ([in] BSTR bstrLogInID, [in] BSTR bstrAccount,) | |
| **參數** | bstrLogInID | 登入ID。 |
| bstrAccount | 複委託帳號，分公司四碼＋帳號7碼。 |
| **回傳值** | 欄位以「,」分隔，結尾為「#」，欄位依序為：  格式：   |  |  | | --- | --- | | 1 | 台幣專戶 | | 2 | 銀行分行代號 | | 3 | 銀行帳號 | | 4 | 銀行名稱 | | 5 | 幣別 | | 6 | 圈存金額 | | 7 | 買進未扣款 | | 8 | 股市委買 | | 9 | 匯出換匯金額 | | 10 | 可解圈金額 | | 11 | 基金委買 | | |
| **備註** |  | |

## 事件

### OnAsyncOrder

|  |  |  |
| --- | --- | --- |
| 非同步委託結果。 | | |
| **宣告** | void OnAsyncOrder([in] LONG nThreadID, [in] LONG nCode, [in] BSTR bstrMessage ); | |
| **參數** | nThreadID | Thread ID |
| nCode | 收單結果代碼。 |
| bstrMessage | 收單回傳訊息。 |
| **備註** | 送單函式中使用非同步下單時會取得一個Thread ID，當此事件觸發時可藉由nThreadID對應下單來源。   * 委託成功：收單回傳訊息為 委託序號 * 委託失敗：收單回傳訊息為 委託失敗原因 | |
| 期貨智慧單: 如果回傳值為 0表示委託成功，訊息內容則為委託日期、智慧單序號、委託書號、委託內容(帳號等)，以實際回傳內容為主。  回傳成功訊息範例:  EX1: 000,20220419,智慧條件單已送出 條件單號：568157,無委託書號,568157,1689200000003  EX2: 000,20220420,智慧條件單已送出 條件單號：568353,y0004,568353,1688000000001 | |

|  |
| --- |
| Python範例程式碼\_OSOrder.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SKOrderLibEvent註冊物件底下一層) |
| # 非同步委託結果。      def OnAsyncOrder(self, nThreadID, nCode, bstrMessage):          msg = "【OnAsyncOrder】" + str(nThreadID) + str(nCode) + bstrMessage;          richTextBoxMessage.insert('end', msg + "\n")          richTextBoxMessage.see('end') |

### OnAsyncOrderOLID

|  |  |  |
| --- | --- | --- |
| 非同步委託結果。(含單獨自訂資料欄) | | |
| **宣告** | void OnAsyncOrderOLID([in] LONG nThreadID, [in] LONG nCode, [in] BSTR bstrMessage, [in] BSTR bstrOrderLinkedID); | |
| **參數** | nThreadID | Thread ID |
| nCode | 收單結果代碼。 |
| bstrMessage | 收單回傳訊息。(13碼序號) |
| bstrOrderLinkedID | 客戶自訂資料，下單時帶入。 |
| **備註** | 送單函式中使用非同步下單時會取得一個Thread ID，當此事件觸發時可藉由nThreadID對應下單來源。 | |

### OnProxyOrder

|  |  |  |
| --- | --- | --- |
| Proxy委託結果。 | | |
| **宣告** | void OnProxyOrder([in] LONG nStampID [in] LONG nCode, [in] BSTR bstrMessage); | |
| **參數** | nStampID | StampID |
| nCode | Proxy收單結果代碼。 |
| bstrMessage | Proxy收單回傳訊息。 |
| **備註** | 送出委託後會回傳二筆通知   |  |  | | --- | --- | | 第一筆 | | | nCode為0代表成功;其餘非0 代表失敗. | bstrMessage  Pending Order送出委託  ORKEY:為Time Stamp供比對第二筆通知使用 | | 第二筆 | | | nCode 為0代表成功;其餘非0 代表失敗. | bstrMessage  **Proxy回傳訊息欄位說明**  SeqNo:代表原始委託13碼  OrderNo:代表委託書號  ORKEY:為Time Stamp供比對第一筆通知使用 | | |

\*v2.13.47異動：補回傳參數nStampID文字說明

|  |
| --- |
| Python範例程式碼\_OSOrder.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SKOrderLibEvent註冊物件底下一層) |
| # Proxy委託結果。      def OnProxyOrder(self, nStampID, nCode, bstrMessage):          msg = "【OnProxyOrder】" + str(nStampID) + str(nCode) + bstrMessage;          richTextBoxMessage.insert('end', msg + "\n")          richTextBoxMessage.see('end') |

## Struct 結構物件

### FOREIGNORDER複委託下單/刪單物件

#### 複委託下單

struct FOREIGNORDER

{

BSTR bstrFullAccount; //複委託帳號，分公司代碼＋帳號7碼

BSTR bstrStockNo; //委託股票代號

BSTR bstrExchangeNo; //交易所代碼，US：美股， HK：港股，JP：日股， SP：新加坡，SG：新(幣)加坡股，SA: 滬股，HA: 深股

BSTR bstrPrice; //委託價格

BSTR bstrCurrency1; //扣款幣別，幣別順序1

BSTR bstrCurrency2; //扣款幣別，幣別順序2

BSTR bstrCurrency3; //扣款幣別，幣別順序3

(幣別可輸入 : HKD、NTD、USD、JPY、SGD、EUR、AUD、CNY、GBP)

BSTR bstrSeqNo; //[刪單時需填入]

BSTR bstrBookNo; //[刪單時需填入]

LONG nAccountType; //專戶別種類，1:外幣專戶 2:台幣專戶

LONG nQty; //委託量

LONG nOrderType; //1:買 2:賣 4:刪單

LONG nTradeType; //庫存別 ，賣出美股時必填

//1:[美股]一般/定股(CITI) 2:定額(VIEWTRADE)

};

比較表

|  |
| --- |
| 異動後V2.13.46 |
| **下單市場新增：**HK：港股，JP：日股， SP：新加坡，SG：新(幣)坡股，SA: 滬股，HA: 深股  **下單幣別新增：**CNY、GBP |

#### 複委託刪單

struct FOREIGNORDER

{

BSTR bstrFullAccount; //複委託帳號，分公司代碼＋帳號7碼

BSTR bstrStockNo; //委託股票代號

BSTR bstrExchangeNo; //交易所代碼，美股：US， HK：港股，JP：日股， SP：新加坡，SG：新(幣)加坡股，SA: 滬股，HA: 深股

BSTR bstrSeqNo; //[刪單時需填入]

BSTR bstrBookNo; //[刪單時需填入]

LONG nOrderType; // 4:刪單

};

### Proxy Server複委託物件

#### Proxy Server複委託下單

struct OSSTOCKPROXYORDER

{

BSTR bstrFullAccount; //複委託帳號，分公司代碼＋帳號7碼

BSTR bstrStockNo; //委託股票代號

BSTR bstrExchangeNo; //交易所代碼，US：美股， HK：港股，JP：日股， SP：新加坡，SG：新(幣)坡股，SA: 滬股，HA: 深股

BSTR bstrPrice; //委託價格

BSTR bstrCurrency1; //扣款幣別，幣別順序1

BSTR bstrCurrency2; //扣款幣別，幣別順序2

BSTR bstrCurrency3; //扣款幣別，幣別順序3

(幣別可輸入 : HKD、NTD、USD、JPY、SGD、EUR、AUD、CNY、GBP)

BSTR bstrProxyQty; //委託量 (股數。只有賣出美股且庫存別為定額(VIEWTRADE)時，股數才能有小數位數，其餘都必須為整數)

LONG nAccountType; //專戶別種類，1:外幣專戶 2:台幣專戶

LONG nOrderType; //1:買 2:賣

LONG nTradeType; //庫存別 ，賣出美股時必填

//1:[美股]一般/定股(CITI) 2:定額(VIEWTRADE) 3:其他股市(一般)

非賣出美股可輸入0 :表示填空值

};

#### Proxy Server複委託刪單

struct OSSTOCKPROXYORDER

{

BSTR bstrFullAccount; //複委託帳號，分公司代碼＋帳號7碼

BSTR bstrStockNo; //委託股票代號

BSTR bstrExchangeNo; //交易所代碼，美股：US

BSTR bstrSeqNo; //[刪單時需填入]

BSTR bstrBookNo; //[刪單時需填入]

LONG nOrderType; // 4:刪單 };