策略王COM元件使用說明

-下單-海外期選-

# 海期價差單買賣別判斷更動說明

海期商品會根據不同種類的商品，其買賣方向判斷方式將有所不同，主要以**近月或遠月為基準進行判定**。

* 海期商品種類區分如下：

1. 指數商品價差EQ (商品種類6)

2. 一般商品價差SP (商品種類7)

3. 外匯商品價差FX (商品種類8)

* 商品種類取得方式:

2.13.54起，事件[OnOverseaFuture](#_4-2-g_OnOverseaFuture_1)、[OnOverseaProductsDetail](#_4-5-g_OnNotifyServerTime)將提供「商品種類」欄位。

* 一張含有 文字, 螢幕擷取畫面, 數字, 字型 的圖片

  AI 產生的內容可能不正確。OnOverseaFuture查詢畫面：
* 一張含有 文字, 電子產品, 螢幕擷取畫面, 字型 的圖片

  AI 產生的內容可能不正確。OnOverseaProductsDetail查詢畫面:

舉例說明：

1. 當「商品種類」為 6，代表此商品為「指數商品價差（EQ）」。

以商品CME,ES-2506/+2509為例，該商品屬於以**遠月**為主的價差商品，因此pOrder.sBuySell為遠月買賣別，舉例來說:

若委託為「賣」單，價差委託為買進近月，賣出遠月。

若委託為「買」單，價差委託為賣出近月，買進遠月。

買賣別以遠月份商品為主。

一張含有 文字, 螢幕擷取畫面, 字型, 收據 的圖片

AI 產生的內容可能不正確。

1. 當「商品種類」為 7，代表此商品為「一般商品價差（SP）」。

以商品CBOT,TY+2506/-2509為例，該商品屬於以**近月**為主的價差商品，因此pOrder.sBuySell為近月買賣別，舉例來說:

若委託為「賣」單，價差委託為賣出近月，買進遠月。

若委託為「買」單，價差委託為買進近月，賣出遠月。

買賣別以近月份商品為主

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

AI 產生的內容可能不正確。

1. 當「商品種類」為 8，代表此商品為「外匯商品價差（FX）」。

以商品CME,JY+2509/-2506為例，該商品屬於以**遠月**為主的價差商品，因此pOrder.sBuySell為遠月買賣別，舉例來說:

若委託為「賣」單，價差委託為買進近月，賣出遠月。

若委託為「買」單，價差委託為賣出近月，買進遠月。

買賣別以遠月份商品為主。

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

AI 產生的內容可能不正確。

# 函式範例

## SKOrderLib(海外期選)

*函式*

1. **查詢**

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| **海期下單商品** | [GetOverseaFutures](#_GetOverseaFutures) |  |
| **海選下單商品** | [GetOverseaOptions](#_GetOverseaOptions) |  |
| **海外期貨未平倉(GW)查詢** | |  | | --- | | [GetOverseaFutureOpenInterestGW](#_GetOverseaFutureOpenInterestGW) | | [GetOverseaFutureOpenInterest](#_GetOverseaFutureOpenInterest) | | GW版 (查詢格式:彙總或明細) |
| **查詢海外期貨權益數** | [GetRequestOverSeaFutureRight](#_GetRequestOverSeaFutureRight) |  |

1. **下載商品檔**

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| **下載海期商品檔** | SKOrderLib\_[LoadOSCommodity](#_SKOrderLib_LoadOSCommodity) | 取得海期帳才可下載商品檔，進行海期委託下單。 |
| **下載海選商品檔** | SKOrderLib\_[LoadOOCommodity](#_SKOrderLib_LoadOOCommodity) | 取得海期帳才可下載商品檔，進行海選委託下單。 |

1. **國內外出入金互轉**

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| **國內外出入金互轉** | [WithDraw](#_WithDraw) |  |

1. **一般下單/刪改單**

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| 1. **海期委託/(自訂)海期委託ByOLID** | |  | | --- | | [SendOverseaFutureOrder](#_SendOverseaFutureOrder) | | [SendOverseaFutureOrderOLID](#_SendOverseaFutureOrderOLID) | |  |
| 1. **海期價差委託/(自訂)海期價差委託ByOLID** | |  | | --- | | [SendOverseaFutureSpreadOrder](#_SendOverseaFutureSpreadOrder) | | [SendOverseaFutureSpreadOrderOLID](#_SendOverseaFutureSpreadOrderOLID) | | V2.13.54後，不再提供此函式  請改用   |  |  |  | | --- | --- | --- | | **海外期貨價差委託** | [SendOverseaFutureSpreadOrder2](#_4-2-114_SendOverseaFutureSpreadOrde) |  | | **送出海外期貨價差委託。ByOLID** | [SendOverseaFutureSpreadOrder2OLID](#_4-2-115_SendOverseaFutureSpreadOrde) |  | |
| 1. **海選委託** | [SendOverseaOptionOrder](#_SendOverseaOptionOrder) |  |
| 1. **海期改價/(自訂)海期改價-書號ByOLID** | [OverSeaCorrectPriceByBookNo](#_OverSeaCorrectPriceByBookNo)  [OverSeaCorrectPriceByBookNoOLID](#_OverSeaCorrectPriceByBookNoOLID) | 僅原限價單改限價(書號) |
| 1. **海期價差改價/(自訂)海期價差改價-書號ByOLID** | [OverSeaCorrectPriceSpreadByBookNo](#_OverSeaCorrectPriceSpreadByBookNo)  [OverSeaCorrectPriceSpreadByBookNoOLID](#_OverSeaCorrectPriceSpreadByBookNoOL) | 僅原限價單改限價(書號) |
| 1. **海選改價** | [OverSeaOptionCorrectPriceByBookNo](#_OverSeaOptionCorrectPriceByBookNo) | 僅原限價單改限價 |
| 1. **海期選委託減量/(自訂)海期委託減量ByOLID** | [OverSeaDecreaseOrderBySeqNo](#_OverSeaDecreaseOrderBySeqNo)  [OverSeaDecreaseOrderBySeqNoOLID](#_OverSeaDecreaseOrderBySeqNoOLID) | 依序號減量 |
| 1. **海期選刪單/(自訂)海期刪單ByOLID** | [OverSeaCancelOrderBySeqNo](#_OverSeaCancelOrderBySeqNo)  [OverSeaCancelOrderBySeqNoOLID](#_OverSeaCancelOrderBySeqNoOLID) | 依序號刪單 |
| 1. **海期選刪單** | [OverSeaCancelOrderByBookNo](#_OverSeaCancelOrderByBookNo) | 依書號刪單 |

1. **Proxy下單/刪改單**

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| **海期委託** | [SendOverseaFutureProxyOrder](#_SendOverseaFutureProxyOrder) |  |
| **海期價差委託** | [SendOverseaFutureSpreadProxyOrder](#_SendOverseaFutureSpreadProxyOrder) |  |
| **海選委託** | [SendOverseaOptionProxyOrder](#_SendOverseaOptionProxyOrder) |  |
| **海期刪改單** | [SendOverseaFutureProxyAlter](#_SendOverseaFutureProxyAlter) |  |

*事件*

|  |  |  |
| --- | --- | --- |
| **功能** | **事件名稱** | **備註** |
| 1. **海期商品資料** | [OnOverseaFuture](#_OnOverseaFuture) |  |
| 1. **海選商品資料** | [OnOverseaOption](#_OnOverseaOption) |  |
| 1. **海期未平倉(GW)資料** | |  | | --- | | [OnOFOpenInterestGWReport](#_OnOFOpenInterestGWReport) | | [OnOverseaFutureOpenInterest](#_OnOverseaFutureOpenInterest) | | GW: (依查詢格式回傳彙總或明細) |
| 1. **海外期貨權益數** | [OnOverSeaFutureRight](#_OnOverSeaFutureRight) |  |
| 1. **非同步委託結果/(自訂)非同步委託結果ForOLID** | |  | | --- | | [OnAsyncOrder](#_OnAsyncOrder) | | [OnAsyncOrderOLID](#_OnAsyncOrderOLID) | |  |
| 1. **Proxy委託結果** | [OnProxyOrder](#_OnProxyOrder) |  |

#### **查詢**

##### GetOverseaFutures

|  |  |  |
| --- | --- | --- |
| 查詢海外期貨下單商品。 | | |
| **宣告** | Long GetOverseaFutures(); | |
| **參數** | 無 |  |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。 | |
| **備註** | 注意! 請先執行函式[SKOrderLib\_LoadOSCommodity](#_4-2-33_SKOrderLib_LoadOSCommodity)下載商品檔  結果由 [OnOverseaFuture](#_4-2-b_OnAsyncOrder) 事件回傳。 | |

|  |
| --- |
| Python範例程式碼\_OFOrder.py類別ReadForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # buttonGetOverseaFutures          self.buttonGetOverseaFutures = tk.Button(self)          self.buttonGetOverseaFutures["text"] = "海期商品檔查詢"          self.buttonGetOverseaFutures["command"] = self.buttonGetOverseaFutures\_Click          self.buttonGetOverseaFutures.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\_pSKOrder = comtypes.client.CreateObject(sk.SKOrderLib,interface=sk.ISKOrderLib) |
| Current Block(放置於def createWidgets(self): 的同一層) |
| def buttonGetOverseaFutures\_Click(self):          nCode = m\_pSKOrder.GetOverseaFutures()          msg = "【GetOverseaFutures】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

##### GetOverseaOptions

|  |  |  |
| --- | --- | --- |
| 查詢海外選擇權下單商品。 | | |
| **宣告** | Long GetOverseaOptions(); | |
| **參數** | 無 |  |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。 | |
| **備註** | 注意! 請先執行函式[SKOrderLib\_LoadOOCommodity](#_SKOrderLib_LoadOOCommodity) 下載商品檔  結果由 [OnOverseaOption](#_OnOverseaOption_1) 事件回傳。 | |

|  |
| --- |
| Python範例程式碼\_OFOrder.py類別ReadForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # buttonGetOverseaOptions          self.buttonGetOverseaOptions = tk.Button(self)          self.buttonGetOverseaOptions["text"] = "海選商品檔查詢"          self.buttonGetOverseaOptions["command"] = self.buttonGetOverseaOptions\_Click          self.buttonGetOverseaOptions.grid(row=0, column=2) |

|  |
| --- |
| 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 buttonGetOverseaOptions\_Click(self):          nCode = m\_pSKOrder.GetOverseaOptions()          msg = "【GetOverseaOptions】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

##### GetOverseaFutureOpenInterestGW

|  |  |  |
| --- | --- | --- |
| 查詢海外期貨未平倉GW。(含格式彙總及明細) | | |
| **宣告** | Long GetOverseaFutureOpenInterestGW([in] BSTR bstrLogInID, [in] BSTR bstrAccount, [in] LONG nFormat); | |
| **參數** | bstrLogInID | 登入ID。 |
| bstrAccount | 委託帳號 ( IB＋帳號) 。 |
|  | nFormat | 查詢格式(1: 彙總;2: 明細) |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。 | |
| **備註** | 結果由 [OnOFOpenInterestGWReport](#_OnOFOpenInterestGWReport) 事件回傳。 | |

|  |
| --- |
| Python範例程式碼\_OFOrder.py類別ReadForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # buttonGetOverseaFutureOpenInterestGW          self.buttonGetOverseaFutureOpenInterestGW = tk.Button(self)          self.buttonGetOverseaFutureOpenInterestGW["text"] = "海期未平倉"          self.buttonGetOverseaFutureOpenInterestGW["command"] = self.buttonGetOverseaFutureOpenInterestGW\_Click          self.buttonGetOverseaFutureOpenInterestGW.grid(row=1, column=1)          # comboBoxnFormatRead          tk.Label(self, text = "格式").grid(row = 1,column = 2)          self.comboBoxnFormatRead = ttk.Combobox(self, state='readonly')          self.comboBoxnFormatRead['values'] = Config.comboBoxnFormatRead          self.comboBoxnFormatRead.grid(row=1, column=3)          global comboBoxnFormatRead          comboBoxnFormatRead = self.comboBoxnFormatRead |

|  |
| --- |
| Config.py |
| # ReadForm  comboBoxnFormatRead =(      "1.彙總", "2.明細"  ) |
| 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 buttonGetOverseaFutureOpenInterestGW\_Click(self):          if (comboBoxnFormatRead.get() == "1.彙總"):              nFormat = 1          elif (comboBoxnFormatRead.get() == "2.明細"):              nFormat = 2          nCode = m\_pSKOrder.GetOverseaFutureOpenInterestGW(comboBoxUserID.get(), comboBoxAccount.get(), nFormat)          msg = "【GetOverseaFutureOpenInterestGW】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

###### **GetOverseaFutureOpenInterest**

|  |  |  |
| --- | --- | --- |
| 查詢海外期貨未平倉。 | | |
| **宣告** | Long GetOverseaFutureOpenInterest([in] BSTR bstrLogInID, [in] BSTR bstrAccount); | |
| **參數** | bstrLogInID | 登入ID。 |
| bstrAccount | 委託帳號 ( IB＋帳號) 。 |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。 | |
| **備註** | 結果由 [OnOverseaFutureOpenInterest](#_OnOverseaFutureOpenInterest_1) 事件回傳。 | |

##### GetRequestOverSeaFutureRight

|  |  |  |
| --- | --- | --- |
| 查詢海外期貨權益數。 | | |
| **宣告** | Long GetRequestOverSeaFutureRight ([in] BSTR bstrLogInID, [in] BSTR bstrAccount) | |
| **參數** | bstrLogInID | 登入ID。 |
| bstrAccount | 委託帳號 ( IB＋帳號)。 |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。 | |
| **備註** | 結果由 [OnOverSeaFutureRight](#_OnOverSeaFutureRight) 事件回傳。 | |

|  |
| --- |
| Python範例程式碼\_OFOrder.py類別ReadForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # buttonGetRequestOverSeaFutureRight          self.buttonGetRequestOverSeaFutureRight = tk.Button(self)          self.buttonGetRequestOverSeaFutureRight["text"] = "海外權益數"          self.buttonGetRequestOverSeaFutureRight["command"] = self.buttonGetRequestOverSeaFutureRight\_Click          self.buttonGetRequestOverSeaFutureRight.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\_pSKOrder = comtypes.client.CreateObject(sk.SKOrderLib,interface=sk.ISKOrderLib) |
| Current Block(放置於def createWidgets(self): 的同一層) |
| def buttonGetRequestOverSeaFutureRight\_Click(self):            nCode = m\_pSKOrder.GetRequestOverSeaFutureRight(comboBoxUserID.get(), comboBoxAccount.get())          msg = "【GetRequestOverSeaFutureRight】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

#### **下載商品檔**

##### SKOrderLib\_LoadOSCommodity

|  |  |  |
| --- | --- | --- |
| SKOrderLib 下載海期商品檔。取得海期帳號才可讀取商品檔，進行海期委託下單。 | | |
| **宣告** | Long SKOrderLib\_LoadOSCommodity(); | |
| **參數** | 無。 |  |
| **回傳值** | 0表示初始化成功，其餘非0數值都表示初始失敗。錯誤代碼可參考對照表。 | |
| **備註** | 具海期帳號，海期委託下單前須先下載  與SKOSQuoteLib\_EnterMonitorLONG相關，可以先進行海期連線備妥商品檔  \*出現錯誤代碼2015,請重新連海期行情主機或重新下載.  ▼以下LOG（日期\_OSQuote.log）為確認是否下載海期商品檔LoadOSCommdity，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\_OFOrder.py類別ReadForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # buttonSKOrderLib\_LoadOSCommodity          self.buttonSKOrderLib\_LoadOSCommodity = tk.Button(self)          self.buttonSKOrderLib\_LoadOSCommodity["text"] = "下載海期商品檔"          self.buttonSKOrderLib\_LoadOSCommodity["command"] = self.buttonSKOrderLib\_LoadOSCommodity\_Click          self.buttonSKOrderLib\_LoadOSCommodity.grid(row=0, column=1)          global buttonSKOrderLib\_LoadOSCommodity          buttonSKOrderLib\_LoadOSCommodity = self.buttonSKOrderLib\_LoadOSCommodity |

|  |
| --- |
| 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 buttonSKOrderLib\_LoadOSCommodity\_Click(self):          nCode = m\_pSKOrder.SKOrderLib\_LoadOSCommodity()          msg = "【SKOrderLib\_LoadOSCommodity】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

##### SKOrderLib\_LoadOOCommodity

|  |  |  |
| --- | --- | --- |
| SKOrderLib 下載海選商品檔、可交易商品檔。取得海期帳號才可讀取海選商品檔、海選可交易商品資訊，進行海選委託下單。 | | |
| **宣告** | Long SKOrderLib\_LoadOOCommodity(); | |
| **參數** | 無。 |  |
| **回傳值** | 0表示初始化成功，其餘非0數值都表示初始失敗。錯誤代碼可參考對照表。 | |
| **備註** | 具海期帳號，海選委託下單前須先下載  與SKOOQuoteLib\_EnterMonitorLONG相關，可以先進行海選連線備妥商品檔  \*出現錯誤代碼2015,請重新連海期行情主機或重新下載. | |

|  |
| --- |
| Python範例程式碼\_OFOrder.py類別ReadForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # buttonSKOrderLib\_LoadOOCommodity          self.buttonSKOrderLib\_LoadOOCommodity = tk.Button(self)          self.buttonSKOrderLib\_LoadOOCommodity["text"] = "下載海選商品檔"          self.buttonSKOrderLib\_LoadOOCommodity["command"] = self.buttonSKOrderLib\_LoadOOCommodity\_Click          self.buttonSKOrderLib\_LoadOOCommodity.grid(row=1, column=1)          global buttonSKOrderLib\_LoadOOCommodity          buttonSKOrderLib\_LoadOOCommodity = self.buttonSKOrderLib\_LoadOOCommodity |

|  |
| --- |
| 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 buttonSKOrderLib\_LoadOOCommodity\_Click(self):          nCode = m\_pSKOrder.SKOrderLib\_LoadOOCommodity()          msg = "【SKOrderLib\_LoadOOCommodity】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode)          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

#### **國內外出入金互轉**

##### WithDraw

|  |  |  |
| --- | --- | --- |
| 國內外出入金互轉。 | | |
| **宣告** | Long WithDraw ([in] BSTR bstrLogInID, [in] BSTR bstrFullAccountOut, [in] LONG nTypeOut, [in] BSTR bstrFullAccountIn, [in] LONG nTypeIn, [in] LONG nCurrency, [in] BSTR bstrDollars, [in] BSTR bstrPassword, [**out**] BSTR\* bstrMessage); | |
| **參數** | bstrLogInID | 登入ID。 |
| bstrFullAccountOut | 轉出期貨帳號(分公司代碼＋帳號7碼) |
| nTypeOut | 轉出類別(0:國內；1:國外) |
| bstrFullAccountIn | 轉入期貨帳號(分公司代碼＋帳號7碼) |
| nTypeIn | 轉入類別(0:國內；1:國外) |
| nCurrency | 幣別(0~9；詳閱－備註) |
| bstrDollars | 金額 |
| bstrPassword | 出入金密碼 |
| bstrMessage | **非同步：**參照OnAsyncOrder。  如果回傳值為 0表示互轉成功，訊息內容則為送單之Thread ID，表示互轉已送出。  回傳值非0表示互轉失敗，訊息內容為失敗原因。 |
| **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。**  **其他錯誤由「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。  幣別：   |  |  |  | | --- | --- | --- | | 代碼 | 中文 | 英文 | | 0 | 澳幣 | AUD | | 1 | 歐元 | EUR | | 2 | 英鎊 | GBP | | 3 | 港幣 | HKD | | 4 | 日元 | JPY | | 5 | 台幣 | NTD | | 6 | 紐幣 | NZD | | 7 | 人民幣 | RMB | | 8 | 美元 | USD | | 9 | 南非幣 | ZAR | | |

|  |
| --- |
| Python範例程式碼\_OFOrder.py類別ReadForm(tk.Frame)裡面 |
| UI程式碼(放置於def createWidgets(self): 的下一層)/UI示意圖 |
| # comboBoxWithDrawnTypeOut          tk.Label(self, text = "轉出類別").grid(row = 0,column = 1)          self.comboBoxWithDrawnTypeOut = ttk.Combobox(self, state='readonly')          self.comboBoxWithDrawnTypeOut['values'] = Config.comboBoxWithDrawnTypeOut          self.comboBoxWithDrawnTypeOut.grid(row=0, column=2)          global comboBoxWithDrawnTypeOut          comboBoxWithDrawnTypeOut = self.comboBoxWithDrawnTypeOut          # textBoxWithDrawbstrFullAccountOut          tk.Label(self, text = "轉出期貨帳號").grid(row=1, column=1)          #輸入框          self.textBoxWithDrawbstrFullAccountOut = tk.Entry(self)          self.textBoxWithDrawbstrFullAccountOut.grid(row=1, column=2)          global textBoxWithDrawbstrFullAccountOut          textBoxWithDrawbstrFullAccountOut = self.textBoxWithDrawbstrFullAccountOut            # comboBoxWithDrawnTypeIn          tk.Label(self, text = "轉入類別").grid(row = 2,column = 1)          self.comboBoxWithDrawnTypeIn = ttk.Combobox(self, state='readonly')          self.comboBoxWithDrawnTypeIn['values'] = Config.comboBoxWithDrawnTypeIn          self.comboBoxWithDrawnTypeIn.grid(row=2, column=2)          global comboBoxWithDrawnTypeIn          comboBoxWithDrawnTypeIn = self.comboBoxWithDrawnTypeIn          # textBoxWithDrawbstrFullAccountIn          tk.Label(self, text = "轉入期貨帳號").grid(row=3, column=1)          #輸入框          self.textBoxWithDrawbstrFullAccountIn = tk.Entry(self)          self.textBoxWithDrawbstrFullAccountIn.grid(row=3, column=2)          global textBoxWithDrawbstrFullAccountIn          textBoxWithDrawbstrFullAccountIn = self.textBoxWithDrawbstrFullAccountIn            # comboBoxWithDrawnCurrency          tk.Label(self, text = "幣別").grid(row = 4,column = 1)          self.comboBoxWithDrawnCurrency = ttk.Combobox(self, state='readonly')          self.comboBoxWithDrawnCurrency['values'] = Config.comboBoxWithDrawnCurrency          self.comboBoxWithDrawnCurrency.grid(row=4, column=2)          global comboBoxWithDrawnCurrency          comboBoxWithDrawnCurrency = self.comboBoxWithDrawnCurrency          # textBoxWithDrawbstrDollars          tk.Label(self, text = "金額").grid(row=4, column=3)          #輸入框          self.textBoxWithDrawbstrDollars = tk.Entry(self)          self.textBoxWithDrawbstrDollars.grid(row=4, column=4)          global textBoxWithDrawbstrDollars          textBoxWithDrawbstrDollars = self.textBoxWithDrawbstrDollars            # textBoxWithDrawbstrPassword          tk.Label(self, text = "出入金密碼").grid(row=5, column=1)          #輸入框          self.textBoxWithDrawbstrPassword = tk.Entry(self)          self.textBoxWithDrawbstrPassword.grid(row=5, column=2)          global textBoxWithDrawbstrPassword          textBoxWithDrawbstrPassword = self.textBoxWithDrawbstrPassword          # buttonWithDraw          self.buttonWithDraw = tk.Button(self)          self.buttonWithDraw["text"] = "出入金互轉"          self.buttonWithDraw["command"] = self.buttonWithDraw\_Click          self.buttonWithDraw.grid(row=6, column=1)          global buttonWithDraw          buttonWithDraw = self.buttonWithDraw |

|  |
| --- |
| Config.py |
| # WithDrawForm  comboBoxWithDrawnTypeOut =(      "國內", "國外"  )  comboBoxWithDrawnTypeIn =(      "國內", "國外"  )  comboBoxWithDrawnCurrency =(      "澳幣", "歐元","英鎊", "港幣","日元", "台幣","紐幣", "人民幣","美元", "國外","南非幣"  ) |
| 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 buttonWithDraw\_Click(self):          bstrFullAccountOut = textBoxWithDrawbstrFullAccountOut.get()          selectedValue = comboBoxWithDrawnTypeOut.get()          if (selectedValue == "國內"):              nTypeOut = 0          elif (selectedValue == "國外"):              nTypeOut = 1          bstrFullAccountIn = textBoxWithDrawbstrFullAccountIn.get()            selectedValue = comboBoxWithDrawnTypeIn.get()          if (selectedValue == "國內"):              nTypeIn = 0          elif (selectedValue == "國外"):              nTypeIn = 1          selectedValue = comboBoxWithDrawnCurrency.get()          if (selectedValue == "澳幣"):              nCurrency = 0          elif (selectedValue == "歐元"):              nCurrency = 1          elif (selectedValue == "英鎊"):              nCurrency = 2          elif (selectedValue == "港幣"):              nCurrency = 3          elif (selectedValue == "日元"):              nCurrency = 4          elif (selectedValue == "台幣"):              nCurrency = 5          elif (selectedValue == "紐幣"):              nCurrency = 6          elif (selectedValue == "人民幣"):              nCurrency = 7          elif (selectedValue == "美元"):              nCurrency = 8          elif (selectedValue == "南非幣"):              nCurrency = 9          bstrDollars = textBoxWithDrawbstrDollars.get()          bstrPassword = textBoxWithDrawbstrPassword.get()          # 國內外出入金互轉          bstrMessage, nCode = m\_pSKOrder.WithDraw(comboBoxUserID.get(), bstrFullAccountOut, nTypeOut, bstrFullAccountIn, nTypeIn, nCurrency, bstrDollars, bstrPassword)          msg = "【WithDraw】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode) + bstrMessage          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

#### **一般下單/刪改單**

##### 海期委託/海期價差委託

###### **SendOverseaFutureOrder**

|  |  |  |
| --- | --- | --- |
| <請先執行商品檔下載 [SKOrderLib\_LoadOSCommodity()](#_4-2-33_SKOrderLib_LoadOSCommodity)>  送出海外期貨委託。 | | |
| **宣告** | Long SendOverseaFutureOrder([in] BSTR bstrLogInID, [in] VARIANT\_BOOL bAsyncOrder, [in] struct **OVERSEAFUTUREORDER**\* pOrder,[**out**] BSTR\* bstrMessage); | |
| **參數** | bstrLogInID | 登入ID。 |
| bAsyncOrder | 是否為非同步委託。 |
| pOrder | SKCOM元件中的 [**OVERSEAFUTUREORDER**](#_海期委託)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。** | |
| **備註** | 使用非同步委託，委託結果請由[OnAsyncOrder](#_OnAsyncOrder)取得。 | |

|  |
| --- |
| Python範例程式碼\_OFOrder.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)          # checkBoxSpread          # 是否為價差交易          self.checkBoxSpread = tk.Checkbutton(self)          self.var1 = tk.IntVar()          self.checkBoxSpread["variable"] = self.var1          self.checkBoxSpread["onvalue"] = True          self.checkBoxSpread["offvalue"] = False          self.checkBoxSpread["text"] = "是否為價差交易"          self.checkBoxSpread["command"] = self.checkBoxSpread\_CheckedChanged          self.checkBoxSpread.grid( row = 0,column = 1)          # textBoxOFExchangeNo          tk.Label(self, text = "交易所代號").grid(row=1, column=1)              #輸入框          self.textBoxOFExchangeNo = tk.Entry(self)          self.textBoxOFExchangeNo.grid(row=1, column=2)          global textBoxOFExchangeNo          textBoxOFExchangeNo = self.textBoxOFExchangeNo          # textBoxOFStockNo          tk.Label(self, text = "海外期貨代號").grid(row=2, column=1)              #輸入框          self.textBoxOFStockNo = tk.Entry(self)          self.textBoxOFStockNo.grid(row=2, column=2)          global textBoxOFStockNo          textBoxOFStockNo = self.textBoxOFStockNo            # textBoxOFYearMonth          tk.Label(self, text = "近月商品年月(YYYYMM)").grid(row=3, column=1)              #輸入框          self.textBoxOFYearMonth = tk.Entry(self)          self.textBoxOFYearMonth.grid(row=3, column=2)          global textBoxOFYearMonth          textBoxOFYearMonth = self.textBoxOFYearMonth            # textBoxOFYearMonth2          tk.Label(self, text = "遠月商品年月(YYYYMM)").grid(row=4, column=1)              #輸入框          self.textBoxOFYearMonth2 = tk.Entry(self)          self.textBoxOFYearMonth2.grid(row=4, column=2)          global textBoxOFYearMonth2          textBoxOFYearMonth2 = self.textBoxOFYearMonth2            # textBoxOFOrder          tk.Label(self, text = "委託價").grid(row=5, column=1)              #輸入框          self.textBoxOFOrder = tk.Entry(self)          self.textBoxOFOrder.grid(row=5, column=2)          global textBoxOFOrder          textBoxOFOrder = self.textBoxOFOrder            # textBoxOFOrderNumerator          tk.Label(self, text = "委託價分子").grid(row=6, column=1)              #輸入框          self.textBoxOFOrderNumerator = tk.Entry(self)          self.textBoxOFOrderNumerator.grid(row=6, column=2)          global textBoxOFOrderNumerator          textBoxOFOrderNumerator = self.textBoxOFOrderNumerator            # textBoxOFTrigger          tk.Label(self, text = "觸發價").grid(row=7, column=1)              #輸入框          self.textBoxOFTrigger = tk.Entry(self)          self.textBoxOFTrigger.grid(row=7, column=2)          global textBoxOFTrigger          textBoxOFTrigger = self.textBoxOFTrigger            # textBoxOFTriggerNumerator          tk.Label(self, text = "觸發價分子").grid(row=8, column=1)              #輸入框          self.textBoxOFTriggerNumerator = tk.Entry(self)          self.textBoxOFTriggerNumerator.grid(row=8, column=2)          global textBoxOFTriggerNumerator          textBoxOFTriggerNumerator = self.textBoxOFTriggerNumerator            # comboBoxOFBuySell          tk.Label(self, text = "買進/賣出").grid(row=9, column=1)              #輸入框          self.comboBoxOFBuySell = ttk.Combobox(self, state='readonly')          self.comboBoxOFBuySell['values'] = Config.comboBoxOFBuySell          self.comboBoxOFBuySell.grid(row=9, column=2)          global comboBoxOFBuySell          comboBoxOFBuySell = self.comboBoxOFBuySell            # comboBoxOFDayTrade          tk.Label(self, text = "當沖").grid(row=10, column=1)              #輸入框          self.comboBoxOFDayTrade = ttk.Combobox(self, state='readonly')          self.comboBoxOFDayTrade['values'] = Config.comboBoxOFDayTrade          self.comboBoxOFDayTrade.grid(row=10, column=2)          global comboBoxOFDayTrade          comboBoxOFDayTrade = self.comboBoxOFDayTrade            # comboBoxOFTradeType          tk.Label(self, text = "ROD/FOK/IOC").grid(row=11, column=1)              #輸入框          self.comboBoxOFTradeType = ttk.Combobox(self, state='readonly')          self.comboBoxOFTradeType['values'] = Config.comboBoxOFTradeType          self.comboBoxOFTradeType.grid(row=11, column=2)          global comboBoxOFTradeType          comboBoxOFTradeType = self.comboBoxOFTradeType            # comboBoxOFSpecialTradeType          tk.Label(self, text = "LMT/MKT/STL/STP").grid(row=12, column=1)              #輸入框          self.comboBoxOFSpecialTradeType = ttk.Combobox(self, state='readonly')          self.comboBoxOFSpecialTradeType['values'] = Config.comboBoxOFSpecialTradeType          self.comboBoxOFSpecialTradeType.grid(row=12, column=2)          global comboBoxOFSpecialTradeType          comboBoxOFSpecialTradeType = self.comboBoxOFSpecialTradeType            # textBoxOFQty          tk.Label(self, text = "交易口數").grid(row=13, column=1)              #輸入框          self.textBoxOFQty = tk.Entry(self)          self.textBoxOFQty.grid(row=13, column=2)          global textBoxOFQty          textBoxOFQty = self.textBoxOFQty          # buttonSendOverSeaFutureOrder          self.buttonSendOverSeaFutureOrder = tk.Button(self)          self.buttonSendOverSeaFutureOrder["text"] = "海外期貨送出"          self.buttonSendOverSeaFutureOrder["command"] = self.buttonSendOverSeaFutureOrder\_Click          self.buttonSendOverSeaFutureOrder.grid(row=14, column=1) |

|  |
| --- |
| Config.py |
| # SendForm  comboBoxOFBuySell =(      "買進", "賣出"  )  comboBoxOFDayTrade =(      "否", "是"  )  comboBoxOFTradeType =(      "ROD", "FOK", "IOC"  )  comboBoxOFSpecialTradeType =(      "LMT限價單", "MKT市價單", "STL停損限價", "STP停損市價"  ) |
| 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  # 是否為價差交易  Spread = 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      # checkBoxSpread      def checkBoxSpread\_CheckedChanged(self):          global Spread          if self.var1.get() == True:              Spread = True          else:              Spread = False      def buttonSendOverSeaFutureOrder\_Click(self):          pOrder = sk.OVERSEAFUTUREORDER()          pOrder.bstrFullAccount = comboBoxAccount.get()          pOrder.bstrStockNo = textBoxOFStockNo.get()          pOrder.bstrExchangeNo = textBoxOFExchangeNo.get()          pOrder.bstrYearMonth = textBoxOFYearMonth.get()          pOrder.bstrYearMonth2 = textBoxOFYearMonth2.get()          pOrder.bstrOrder = textBoxOFOrder.get()          pOrder.bstrOrderNumerator = textBoxOFOrderNumerator.get()          pOrder.bstrTrigger = textBoxOFTrigger.get()          pOrder.bstrTriggerNumerator = textBoxOFTriggerNumerator.get()          pOrder.sNewClose = 0          if (comboBoxOFTradeType.get() == "ROD"):              pOrder.sTradeType = 0          elif (comboBoxOFTradeType.get() == "FOK"):              pOrder.sTradeType = 1          elif (comboBoxOFTradeType.get() == "IOC"):              pOrder.sTradeType = 2          if (comboBoxOFBuySell.get() == "買進"):              pOrder.sBuySell = 0          elif (comboBoxOFBuySell.get() == "賣出"):              pOrder.sBuySell = 1          if (comboBoxOFDayTrade.get() == "否"):              pOrder.sDayTrade = 0          elif (comboBoxOFDayTrade.get() == "是"):              pOrder.sDayTrade = 1          if (comboBoxOFSpecialTradeType.get() == "LMT限價單"):              pOrder.sSpecialTradeType = 0          elif (comboBoxOFSpecialTradeType.get() == "MKT市價單"):              pOrder.sSpecialTradeType = 1          elif (comboBoxOFSpecialTradeType.get() == "STL停損限價"):              pOrder.sSpecialTradeType = 2          elif (comboBoxOFSpecialTradeType.get() == "STP停損市價"):              pOrder.sSpecialTradeType = 3          pOrder.nQty = int(textBoxOFQty.get())          if (Spread == False):              bstrMessage,nCode= m\_pSKOrder.SendOverseaFutureOrder(comboBoxUserID.get(), bAsyncOrder, pOrder)              msg = "【SendOverseaFutureOrder】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode) + bstrMessage              richTextBoxMethodMessage.insert('end',  msg + "\n")              richTextBoxMethodMessage.see('end')          else:              # 送出海期委託              bstrMessage,nCode= m\_pSKOrder.SendOverseaFutureSpreadOrder(comboBoxUserID.get(), bAsyncOrder, pOrder)              msg = "【SendOverseaFutureSpreadOrder】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode) + bstrMessage              richTextBoxMethodMessage.insert('end',  msg + "\n")              richTextBoxMethodMessage.see('end') |

SendOverseaFutureOrderOLID

|  |  |  |
| --- | --- | --- |
| 送出海外期貨委託。(含單獨自訂資料欄) | | |
| **宣告** | Long SendOverseaFutureOrderOLID([in] BSTR bstrLogInID, [in] VARIANT\_BOOL bAsyncOrder, [in] struct **OVERSEAFUTUREORDER**\* pOrder, [in] BSTR bstrOrderLinkedID,[**out**] BSTR\* bstrMessage); | |
| **參數** | bstrLogInID | 登入ID。 |
| bAsyncOrder | 是否為非同步委託。 |
| pOrder | SKCOM元件中的 [**OVERSEAFUTUREORDER**](#_海期委託)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。**  **其他錯誤由「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](#_OnAsyncOrderOLID)取得。 | |

###### **SendOverseaFutureSpreadOrder**

|  |  |  |
| --- | --- | --- |
| <請先執行商品檔下載 [SKOrderLib\_LoadOSCommodity()](#_4-2-33_SKOrderLib_LoadOSCommodity)>  送出海外期貨價差委託。 | | |
| **宣告** | Long SendOverseaFutureSpreadOrder([in] BSTR bstrLogInID, [in] VARIANT\_BOOL bAsyncOrder, [in] struct **OVERSEAFUTUREORDER**\* pOrder,[**out**] BSTR\* bstrMessage); | |
| **參數** | bstrLogInID | 登入ID。 |
| bAsyncOrder | 是否為非同步委託。 |
| pOrder | SKCOM元件中的 [**OVERSEAFUTUREORDER**](#_海期委託)物件，將下單條件填入該物件後，再帶入此欄位中(價差不提供當沖)。 |
| 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。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](#_OnAsyncOrder)取得。 | |

SendOverseaFutureSpreadOrderOLID

|  |  |  |
| --- | --- | --- |
| 送出海外期貨價差委託。(含單獨自訂資料欄) | | |
| **宣告** | Long SendOverseaFutureSpreadOrderOLID([in] BSTR bstrLogInID, [in] VARIANT\_BOOL bAsyncOrder, [in] struct **OVERSEAFUTUREORDER**\* pOrder, [in] BSTR bstrOrderLinkedID,[**out**] BSTR\* bstrMessage); | |
| **參數** | bstrLogInID | 登入ID。 |
| bAsyncOrder | 是否為非同步委託。 |
| pOrder | SKCOM元件中的 [**OVERSEAFUTUREORDER**](#_海期委託)物件，將下單條件填入該物件後，再帶入此欄位中(價差不提供當沖)。 |
| 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。**  **其他錯誤由「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](#_OnAsyncOrderOLID)取得。 | |

###### **SendOverseaFutureSpreadOrder2**

|  |  |  |
| --- | --- | --- |
| <請先執行 商品檔下載 SKOrderLib\_LoadOSCommodity()>  送出海外期貨價差委託。  使用此函式時，帶入0:買，如果是以近月買賣別為主之商品-買進賣遠；以遠月買賣別為主之商品-買遠賣近 | | |
| **宣告** | Long SendOverseaFutureSpreadOrder2([in] BSTR bstrLogInID, [in] VARIANT\_BOOL bAsyncOrder, [in] struct OVERSEAFUTUREORDER\* pOrder,[out] BSTR\* bstrMessage); | |
| **參數** | bstrLogInID | 登入ID。 |
| bAsyncOrder | 是否為非同步委託。 |
| pOrder | SKCOM元件中的 **OVERSEAFUTUREORDER**物件，將下單條件填入該物件後，再帶入此欄位中(價差不提供當沖)。 |
| 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。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。 | |

SendOverseaFutureSpreadOrder2OLID

|  |  |  |
| --- | --- | --- |
| 送出海外期貨價差委託。(含單獨自訂資料欄)  使用此函式時，帶入0:買，如果是以近月買賣別為主之商品-買進賣遠；以遠月買賣別為主之商品-買遠賣近 | | |
| **宣告** | Long SendOverseaFutureSpreadOrder2OLID([in] BSTR bstrLogInID, [in] VARIANT\_BOOL bAsyncOrder, [in] struct **OVERSEAFUTUREORDER**\* pOrder, [in] BSTR bstrOrderLinkedID,[**out**] BSTR\* bstrMessage); | |
| 參數 | bstrLogInID | 登入ID。 |
| bAsyncOrder | 是否為非同步委託。 |
| pOrder | SKCOM元件中的 **OVERSEAFUTUREORDER**物件，將下單條件填入該物件後，再帶入此欄位中(價差不提供當沖)。 |
| 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。** | |
| **備註** | 使用非同步委託，委託結果請由[OnAsyncOrderOLID](#_4-2-o_OnAsyncOrderOLID)取得。  SGX專線模式：SGX交易所商品均經由專線委託 | |

##### SendOverseaOptionOrder

|  |  |  |
| --- | --- | --- |
| <請先執行商品檔下載 [SKOrderLib\_LoadOOCommodity()](#_SKOrderLib_LoadOOCommodity)>  送出海外選擇權委託。 | | |
| **宣告** | Long SendOverseaOptionOrder([in] BSTR bstrLogInID, [in] VARIANT\_BOOL bAsyncOrder, [in] struct **OVERSEAFUTUREORDER**\* pOrder,[**out**] BSTR\* bstrMessage); | |
| **參數** | bstrLogInID | 登入ID。 |
| bAsyncOrder | 是否為非同步委託。 |
| pOrder | SKCOM元件中的 [**OVERSEAFUTUREORDER**](#_海選委託)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。** | |
| **備註** | 使用非同步委託，委託結果請由[OnAsyncOrder](#_OnAsyncOrder)取得。 | |

|  |
| --- |
| Python範例程式碼\_OFOrder.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 |
| # textBoxOOExchangeNo          tk.Label(self, text = "交易所代號").grid(row=15, column=1)              #輸入框          self.textBoxOOExchangeNo = tk.Entry(self)          self.textBoxOOExchangeNo.grid(row=15, column=2)          global textBoxOOExchangeNo          textBoxOOExchangeNo = self.textBoxOOExchangeNo          # textBoxOOStockNo          tk.Label(self, text = "海外選擇權代號").grid(row=16, column=1)              #輸入框          self.textBoxOOStockNo = tk.Entry(self)          self.textBoxOOStockNo.grid(row=16, column=2)          global textBoxOOStockNo          textBoxOOStockNo = self.textBoxOOStockNo            # textBoxOOYearMonth          tk.Label(self, text = "近月商品年月(YYYYMM)").grid(row=17, column=1)              #輸入框          self.textBoxOOYearMonth = tk.Entry(self)          self.textBoxOOYearMonth.grid(row=17, column=2)          global textBoxOOYearMonth          textBoxOOYearMonth = self.textBoxOOYearMonth            # textBoxOOOrder          tk.Label(self, text = "委託價").grid(row=18, column=1)              #輸入框          self.textBoxOOOrder = tk.Entry(self)          self.textBoxOOOrder.grid(row=18, column=2)          global textBoxOOOrder          textBoxOOOrder = self.textBoxOOOrder            # textBoxOOOrderNumerator          tk.Label(self, text = "委託價分子").grid(row=19, column=1)              #輸入框          self.textBoxOOOrderNumerator = tk.Entry(self)          self.textBoxOOOrderNumerator.grid(row=19, column=2)          global textBoxOOOrderNumerator          textBoxOOOrderNumerator = self.textBoxOOOrderNumerator            # textBoxbstrOrderDenominator          tk.Label(self, text = "分母").grid(row=19, column=3)              #輸入框          self.textBoxbstrOrderDenominator = tk.Entry(self)          self.textBoxbstrOrderDenominator.grid(row=19, column=4)          global textBoxbstrOrderDenominator          textBoxbstrOrderDenominator = self.textBoxbstrOrderDenominator            # textBoxOOTrigger          tk.Label(self, text = "觸發價").grid(row=20, column=1)              #輸入框          self.textBoxOOTrigger = tk.Entry(self)          self.textBoxOOTrigger.grid(row=20, column=2)          global textBoxOOTrigger          textBoxOOTrigger = self.textBoxOOTrigger            # textBoxOOTriggerNumerator          tk.Label(self, text = "觸發價分子").grid(row=21, column=1)              #輸入框          self.textBoxOOTriggerNumerator = tk.Entry(self)          self.textBoxOOTriggerNumerator.grid(row=21, column=2)          global textBoxOOTriggerNumerator          textBoxOOTriggerNumerator = self.textBoxOOTriggerNumerator          # comboBoxOOBuySell          tk.Label(self, text = "買進/賣出").grid(row=22, column=1)              #輸入框          self.comboBoxOOBuySell = ttk.Combobox(self, state='readonly')          self.comboBoxOOBuySell['values'] = Config.comboBoxOOBuySell          self.comboBoxOOBuySell.grid(row=22, column=2)          global comboBoxOOBuySell          comboBoxOOBuySell = self.comboBoxOOBuySell            # comboBoxOONewClose          tk.Label(self, text = "新平倉").grid(row=23, column=1)              #輸入框          self.comboBoxOONewClose = ttk.Combobox(self, state='readonly')          self.comboBoxOONewClose['values'] = Config.comboBoxOONewClose          self.comboBoxOONewClose.grid(row=23, column=2)          global comboBoxOONewClose          comboBoxOONewClose = self.comboBoxOONewClose            # comboBoxOODayTrade          tk.Label(self, text = "當沖").grid(row=24, column=1)              #輸入框          self.comboBoxOODayTrade = ttk.Combobox(self, state='readonly')          self.comboBoxOODayTrade['values'] = Config.comboBoxOODayTrade          self.comboBoxOODayTrade.grid(row=24, column=2)          global comboBoxOODayTrade          comboBoxOODayTrade = self.comboBoxOODayTrade            # comboBoxOOSpecialTradeType          tk.Label(self, text = "LMT/MKT/STL/STP").grid(row=25, column=1)              #輸入框          self.comboBoxOOSpecialTradeType = ttk.Combobox(self, state='readonly')          self.comboBoxOOSpecialTradeType['values'] = Config.comboBoxOOSpecialTradeType          self.comboBoxOOSpecialTradeType.grid(row=25, column=2)          global comboBoxOOSpecialTradeType          comboBoxOOSpecialTradeType = self.comboBoxOOSpecialTradeType            # textBoxOOStrikePrice          tk.Label(self, text = "履約價").grid(row=26, column=1)              #輸入框          self.textBoxOOStrikePrice = tk.Entry(self)          self.textBoxOOStrikePrice.grid(row=26, column=2)          global textBoxOOStrikePrice          textBoxOOStrikePrice = self.textBoxOOStrikePrice            # comboBoxOOCallPut          tk.Label(self, text = "CALL/PUT").grid(row=27, column=1)              #輸入框          self.comboBoxOOCallPut = ttk.Combobox(self, state='readonly')          self.comboBoxOOCallPut['values'] = Config.comboBoxOOCallPut          self.comboBoxOOCallPut.grid(row=27, column=2)          global comboBoxOOCallPut          comboBoxOOCallPut = self.comboBoxOOCallPut            # textBoxOOQty          tk.Label(self, text = "交易口數").grid(row=28, column=1)              #輸入框          self.textBoxOOQty = tk.Entry(self)          self.textBoxOOQty.grid(row=28, column=2)          global textBoxOOQty          textBoxOOQty = self.textBoxOOQty          # buttonSendOverseaOptionOrder          self.buttonSendOverseaOptionOrder = tk.Button(self)          self.buttonSendOverseaOptionOrder["text"] = "海外選擇權送出"          self.buttonSendOverseaOptionOrder["command"] = self.buttonSendOverseaOptionOrder\_Click          self.buttonSendOverseaOptionOrder.grid(row=29, column=1) |

|  |
| --- |
| Config.py |
| comboBoxOOBuySell =(      "買進", "賣出"  )  comboBoxOONewClose =(      "新倉", "平倉"  )  comboBoxOODayTrade =(      "否", "是"  )  comboBoxOOSpecialTradeType =(      "LMT限價單", "MKT市價單", "STL停損限價", "STP停損市價"  )  comboBoxOOCallPut =(      "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\_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) |
| def buttonSendOverseaOptionOrder\_Click(self):          pOrder = sk.OVERSEAFUTUREORDER()          pOrder.bstrFullAccount = comboBoxAccount.get()          pOrder.bstrStockNo = textBoxOOStockNo.get()          pOrder.bstrExchangeNo = textBoxOOExchangeNo.get()          pOrder.bstrYearMonth = textBoxOOYearMonth.get()          pOrder.bstrOrder = textBoxOOOrder.get()          pOrder.bstrOrderNumerator = textBoxOOOrderNumerator.get()          pOrder.bstrOrderDenominator = textBoxbstrOrderDenominator.get()          pOrder.bstrTrigger = textBoxOOTrigger.get()          pOrder.bstrTriggerNumerator = textBoxOOTriggerNumerator.get()          pOrder.sTradeType = 0          if (comboBoxOOBuySell.get() == "買進"):              pOrder.sBuySell = 0          elif (comboBoxOOBuySell.get() == "賣出"):              pOrder.sBuySell = 1          if (comboBoxOODayTrade.get() == "否"):              pOrder.sDayTrade = 0          elif (comboBoxOODayTrade.get() == "是"):              pOrder.sDayTrade = 1          if (comboBoxOONewClose.get() == "新倉"):              pOrder.sNewClose = 0          elif (comboBoxOONewClose.get() == "平倉"):              pOrder.sNewClose = 1          if (comboBoxOOSpecialTradeType.get() == "LMT限價單"):              pOrder.sSpecialTradeType = 0          elif (comboBoxOOSpecialTradeType.get() == "MKT市價單"):              pOrder.sSpecialTradeType = 1          elif (comboBoxOOSpecialTradeType.get() == "STL停損限價"):              pOrder.sSpecialTradeType = 2          elif (comboBoxOOSpecialTradeType.get() == "STP停損市價"):              pOrder.sSpecialTradeType = 3          pOrder.bstrStrikePrice = textBoxOOStrikePrice.get()          pOrder.nQty = int(textBoxOOQty.get())          if (comboBoxOOCallPut.get() == "CALL"):              pOrder.sCallPut = 0          elif (comboBoxOOCallPut.get() == "PUT"):              pOrder.sCallPut = 1          # 送出選擇權委託          bstrMessage,nCode= m\_pSKOrder.SendOverseaOptionOrder(comboBoxUserID.get(), bAsyncOrder, pOrder)          msg = "【SendOverseaOptionOrder】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode) + bstrMessage          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

##### 改價

###### **OverSeaCorrectPriceByBookNo**

|  |  |  |
| --- | --- | --- |
| 海期改價 (By 委託書號)。 | | |
| **宣告** | Long OverSeaCorrectPriceByBookNo([in] BSTR bstrLogInID, [in] VARIANT\_BOOL bAsyncOrder, [in] struct **OVERSEAFUTUREORDERFORGW\*** pOrder , [**out**] BSTR\* bstrMessage); | |
| **參數** | bstrLogInID | 登入ID。 |
| bAsyncOrder | 是否為非同步委託。 |
| pOrder | SKCOM元件中的 [**OVERSEAFUTUREORDERFORGW**](#_OVERSEAFUTUREORDERFORGW(海期委託GW格式物件))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，訊息內容則為修改訊息。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。**  **其他錯誤由「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** | 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](#_OnAsyncOrder)取得。  \*原委託為限價單方可改價(ROD) | |

|  |
| --- |
| Python範例程式碼\_OFOrder.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)          # checkBoxSpread          # 是否為價差交易          self.checkBoxSpread = tk.Checkbutton(self)          self.var1 = tk.IntVar()          self.checkBoxSpread["variable"] = self.var1          self.checkBoxSpread["onvalue"] = True          self.checkBoxSpread["offvalue"] = False          self.checkBoxSpread["text"] = "是否為價差交易"          self.checkBoxSpread["command"] = self.checkBoxSpread\_CheckedChanged          self.checkBoxSpread.grid( row = 1,column = 0)          # textBoxOverSeaCancelOrderBySeqNo3          tk.Label(self, text = "請輸入委託序號").grid(row=0, column=1)              #輸入框          self.textBoxOverSeaCancelOrderBySeqNo3 = tk.Entry(self)          self.textBoxOverSeaCancelOrderBySeqNo3.grid(row=0, column=2)          global textBoxOverSeaCancelOrderBySeqNo3          textBoxOverSeaCancelOrderBySeqNo3 = self.textBoxOverSeaCancelOrderBySeqNo3          # textBoxOverSeaCancelOrderByBookNo3          tk.Label(self, text = "請輸入委託書號").grid(row=1, column=1)              #輸入框          self.textBoxOverSeaCancelOrderByBookNo3 = tk.Entry(self)          self.textBoxOverSeaCancelOrderByBookNo3.grid(row=1, column=2)          global textBoxOverSeaCancelOrderByBookNo3          textBoxOverSeaCancelOrderByBookNo3 = self.textBoxOverSeaCancelOrderByBookNo3          # buttonOverSeaCancelOrderBySeqNo          self.buttonOverSeaCancelOrderBySeqNo = tk.Button(self)          self.buttonOverSeaCancelOrderBySeqNo["text"] = "刪單(序號)"          self.buttonOverSeaCancelOrderBySeqNo["command"] = self.buttonOverSeaCancelOrderBySeqNo\_Click          self.buttonOverSeaCancelOrderBySeqNo.grid(row=2, column=1)          # buttonOverSeaCancelOrderByBookNo          self.buttonOverSeaCancelOrderByBookNo = tk.Button(self)          self.buttonOverSeaCancelOrderByBookNo["text"] = "刪單(書號)"          self.buttonOverSeaCancelOrderByBookNo["command"] = self.buttonOverSeaCancelOrderByBookNo\_Click          self.buttonOverSeaCancelOrderByBookNo.grid(row=2, column=2)          # textBoxOverseaFutureDecreaseQty3          tk.Label(self, text = "請輸入減少數量").grid(row=3, column=1)              #輸入框          self.textBoxOverseaFutureDecreaseQty3 = tk.Entry(self)          self.textBoxOverseaFutureDecreaseQty3.grid(row=3, column=2)          global textBoxOverseaFutureDecreaseQty3          textBoxOverseaFutureDecreaseQty3 = self.textBoxOverseaFutureDecreaseQty3          # buttonOverSeaDecreaseOrderBySeqNo          self.buttonOverSeaDecreaseOrderBySeqNo = tk.Button(self)          self.buttonOverSeaDecreaseOrderBySeqNo["text"] = "減量(序號)"          self.buttonOverSeaDecreaseOrderBySeqNo["command"] = self.buttonOverSeaDecreaseOrderBySeqNo\_Click          self.buttonOverSeaDecreaseOrderBySeqNo.grid(row=4, column=1)          # textBoxOFExchangeNo3          tk.Label(self, text = "交易所代號").grid(row=5, column=1)              #輸入框          self.textBoxOFExchangeNo3 = tk.Entry(self)          self.textBoxOFExchangeNo3.grid(row=5, column=2)          global textBoxOFExchangeNo3          textBoxOFExchangeNo3 = self.textBoxOFExchangeNo3            # textBoxOFStockNo3          tk.Label(self, text = "海外期權代號").grid(row=6, column=1)              #輸入框          self.textBoxOFStockNo3 = tk.Entry(self)          self.textBoxOFStockNo3.grid(row=6, column=2)          global textBoxOFStockNo3          textBoxOFStockNo3 = self.textBoxOFStockNo3            # textBoxOFStockNo23          tk.Label(self, text = "海外期價差代號").grid(row=7, column=1)              #輸入框          self.textBoxOFStockNo23 = tk.Entry(self)          self.textBoxOFStockNo23.grid(row=7, column=2)          global textBoxOFStockNo23          textBoxOFStockNo23 = self.textBoxOFStockNo23            # textBoxOFYearMonth3          tk.Label(self, text = "近月商品年月(YYYYMM)").grid(row=8, column=1)              #輸入框          self.textBoxOFYearMonth3 = tk.Entry(self)          self.textBoxOFYearMonth3.grid(row=8, column=2)          global textBoxOFYearMonth3          textBoxOFYearMonth3 = self.textBoxOFYearMonth3            # textBoxOFYearMonth23          tk.Label(self, text = "遠月商品年月(YYYYMM)").grid(row=9, column=1)              #輸入框          self.textBoxOFYearMonth23 = tk.Entry(self)          self.textBoxOFYearMonth23.grid(row=9, column=2)          global textBoxOFYearMonth23          textBoxOFYearMonth23 = self.textBoxOFYearMonth23            # textBoxOFOrder3          tk.Label(self, text = "新委託價").grid(row=10, column=1)              #輸入框          self.textBoxOFOrder3 = tk.Entry(self)          self.textBoxOFOrder3.grid(row=10, column=2)          global textBoxOFOrder3          textBoxOFOrder3 = self.textBoxOFOrder3            # textBoxOFOrderNumerator3          tk.Label(self, text = "新委託價分子").grid(row=11, column=1)              #輸入框          self.textBoxOFOrderNumerator3 = tk.Entry(self)          self.textBoxOFOrderNumerator3.grid(row=11, column=2)          global textBoxOFOrderNumerator3          textBoxOFOrderNumerator3 = self.textBoxOFOrderNumerator3            # textBoxOFOrderDenominator3          tk.Label(self, text = "新委託價分母").grid(row=12, column=1)              #輸入框          self.textBoxOFOrderDenominator3 = tk.Entry(self)          self.textBoxOFOrderDenominator3.grid(row=12, column=2)          global textBoxOFOrderDenominator3          textBoxOFOrderDenominator3 = self.textBoxOFOrderDenominator3            # textBoxOOStrikePrice3          tk.Label(self, text = "履約價(改期貨帶0)").grid(row=13, column=1)              #輸入框          self.textBoxOOStrikePrice3 = tk.Entry(self)          self.textBoxOOStrikePrice3.grid(row=13, column=2)          global textBoxOOStrikePrice3          textBoxOOStrikePrice3 = self.textBoxOOStrikePrice3            # comboBoxOOCallPut3          tk.Label(self, text = "CALL/PUT").grid(row=14, column=1)              #輸入框          self.comboBoxOOCallPut3 = ttk.Combobox(self, state='readonly')          self.comboBoxOOCallPut3['values'] = Config.comboBoxOOCallPut3          self.comboBoxOOCallPut3.grid(row=14, column=2)          global comboBoxOOCallPut3          comboBoxOOCallPut3 = self.comboBoxOOCallPut3          # buttonOverSeaCorrectPriceByBookNo          self.buttonOverSeaCorrectPriceByBookNo = tk.Button(self)          self.buttonOverSeaCorrectPriceByBookNo["text"] = "改價(書號)"          self.buttonOverSeaCorrectPriceByBookNo["command"] = self.buttonOverSeaCorrectPriceByBookNo\_Click          self.buttonOverSeaCorrectPriceByBookNo.grid(row=15, column=1) |

|  |
| --- |
| Config.py |
| # UpdateForm  comboBoxOOCallPut3 =(      "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\_pSKOrder = comtypes.client.CreateObject(sk.SKOrderLib,interface=sk.ISKOrderLib)  # 是否為非同步委託  bAsyncOrder = False  # 是否為價差交易  Spread = 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        # checkBoxSpread      def checkBoxSpread\_CheckedChanged(self):          global Spread          if self.var1.get() == True:              Spread = True          else:              Spread = False        def buttonOverSeaCancelOrderBySeqNo\_Click(self):          # 海外期貨委託删單(By委託序號)          bstrMessage,nCode= m\_pSKOrder.OverSeaCancelOrderBySeqNo(comboBoxUserID.get(), bAsyncOrder, comboBoxAccount.get(), textBoxOverSeaCancelOrderBySeqNo3.get())          msg = "【OverSeaCancelOrderBySeqNo】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode) + bstrMessage          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end')      def buttonOverSeaCancelOrderByBookNo\_Click(self):          # 海外期貨委託删單(By委託書號)          bstrMessage,nCode= m\_pSKOrder.OverSeaCancelOrderByBookNo(comboBoxUserID.get(), bAsyncOrder, comboBoxAccount.get(), textBoxOverSeaCancelOrderByBookNo3.get())          msg = "【OverSeaCancelOrderByBookNo】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode) + bstrMessage          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end')      def buttonOverSeaDecreaseOrderBySeqNo\_Click(self):          # 海期委託減量(By委託序號)          bstrMessage,nCode= m\_pSKOrder.OverSeaDecreaseOrderBySeqNo(comboBoxUserID.get(), bAsyncOrder, comboBoxAccount.get(), textBoxOverSeaCancelOrderBySeqNo3.get(), int(textBoxOverseaFutureDecreaseQty3.get()))          msg = "【OverSeaDecreaseOrderBySeqNo】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode) + bstrMessage          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end')        def buttonOverSeaCorrectPriceByBookNo\_Click(self):          pOrder = sk.OVERSEAFUTUREORDERFORGW()          pOrder.bstrBookNo = textBoxOverSeaCancelOrderByBookNo3.get()          pOrder.bstrFullAccount = comboBoxAccount.get()          pOrder.bstrExchangeNo = textBoxOFExchangeNo3.get()          pOrder.bstrStockNo = textBoxOFStockNo3.get()          pOrder.bstrStockNo2 = textBoxOFStockNo23.get()          pOrder.bstrYearMonth = textBoxOFYearMonth3.get()          pOrder.bstrYearMonth2 = textBoxOFYearMonth23.get()          pOrder.bstrOrderPrice = textBoxOFOrder3.get()          pOrder.bstrOrderNumerator = textBoxOFOrderNumerator3.get()          pOrder.bstrOrderDenominator = textBoxOFOrderDenominator3.get()          pOrder.bstrStrikePrice = textBoxOOStrikePrice3.get()          pOrder.nTradeType = 0          pOrder.nSpecialTradeType = 0          if (comboBoxOOCallPut3.get() == "CALL"):              pOrder.nCallPut = 0          elif (comboBoxOOCallPut3.get() == "PUT"):              pOrder.nCallPut = 1          if(textBoxOOStrikePrice3.get() != "0"): # 履約價不為0 => 選擇權              # 海選改價 (By 委託書號)              bstrMessage, nCode = m\_pSKOrder.OverSeaOptionCorrectPriceByBookNo(comboBoxUserID.get(), bAsyncOrder, pOrder)                msg = "【OverSeaOptionCorrectPriceByBookNo】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode) + bstrMessage              richTextBoxMethodMessage.insert('end',  msg + "\n")              richTextBoxMethodMessage.see('end')          else:              if (Spread == True): # 為價差改價                  # 海選價差改價 (By 委託書號)                  bstrMessage, nCode = m\_pSKOrder.OverSeaCorrectPriceSpreadByBookNo(comboBoxUserID.get(), bAsyncOrder, pOrder)                    msg = "【OverSeaCorrectPriceSpreadByBookNo】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode) + bstrMessage                  richTextBoxMethodMessage.insert('end',  msg + "\n")                  richTextBoxMethodMessage.see('end')              else:                  # 海選改價 (By 委託書號)                  bstrMessage, nCode = m\_pSKOrder.OverSeaCorrectPriceByBookNo(comboBoxUserID.get(), bAsyncOrder, pOrder)                    msg = "【OverSeaCorrectPriceByBookNo】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode) + bstrMessage                  richTextBoxMethodMessage.insert('end',  msg + "\n")                  richTextBoxMethodMessage.see('end') |

###### **OverSeaCorrectPriceByBookNoOLID**

|  |  |  |
| --- | --- | --- |
| 海期改價 OLID (By 委託書號)。 | | |
| **宣告** | Long OverSeaCorrectPriceByBookNoOLID([in] BSTR bstrLogInID, [in] VARIANT\_BOOL bAsyncOrder, [in] struct **OVERSEAFUTUREORDERFORGW\*** pOrder, [in] BSTR bstrOrderLinkedID, [**out**] BSTR\* bstrMessage); | |
| **參數** | bstrLogInID | 登入ID。 |
| bAsyncOrder | 是否為非同步委託。 |
| pOrder | SKCOM元件中的 **OVRESEAFUTUREORDERFORGW**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，訊息內容則為修改訊息。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。**  **其他錯誤由「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** | 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**OnAsyncOrderOLID**取得。  \*原委託為限價單方可改價(ROD) | |

###### **OverSeaCorrectPriceSpreadByBookNo**

|  |  |  |
| --- | --- | --- |
| 海期價差改價 (By 委託書號)。 | | |
| **宣告** | Long OverSeaCorrectPriceSpreadByBookNo([in] BSTR bstrLogInID, [in] VARIANT\_BOOL bAsyncOrder, [in] struct **OVERSEAFUTUREORDERFORGW\*** pOrder , [**out**] BSTR\* bstrMessage); | |
| **參數** | bstrLogInID | 登入ID。 |
| bAsyncOrder | 是否為非同步委託。 |
| pOrder | SKCOM元件中的 [**OVERSEAFUTUREORDERFORGW**](#_OVERSEAFUTUREORDERFORGW(海期委託GW格式物件))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，訊息內容則為修改訊息。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。**  **其他錯誤由「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** | 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](#_OnAsyncOrder)取得。  \*原委託為限價單方可改價(ROD) | |

程式碼可參考 [OverSeaCorrectPriceByBookNo](#_OverSeaCorrectPriceByBookNo) 作法

###### **OverSeaCorrectPriceSpreadByBookNoOLID**

|  |  |  |
| --- | --- | --- |
| 海期價差改價 OLID (By 委託書號)。 | | |
| **宣告** | Long OverSeaCorrectPriceSpreadByBookNoOLID([in] BSTR bstrLogInID, [in] VARIANT\_BOOL bAsyncOrder, [in] struct **OVERSEAFUTUREORDERFORGW\*** pOrder, [in] BSTR bstrOrderLinkedID, [**out**] BSTR\* bstrMessage); | |
| **參數** | bstrLogInID | 登入ID。 |
| bAsyncOrder | 是否為非同步委託。 |
| pOrder | SKCOM元件中的 **OVRESEAFUTUREORDERFORGW**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，訊息內容則為修改訊息。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。**  **其他錯誤由「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** | 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**OnAsyncOrderOLID**取得。  \*原委託為限價單方可改價(ROD) | |

###### **OverSeaOptionCorrectPriceByBookNo**

|  |  |  |
| --- | --- | --- |
| 海選改價 (By 委託書號)。 | | |
| **宣告** | Long OverSeaOptionCorrectPriceByBookNo([in] BSTR bstrLogInID, [in] VARIANT\_BOOL bAsyncOrder, [in] struct **OVERSEAFUTUREORDERFORGW\*** pOrder , [**out**] BSTR\* bstrMessage); | |
| **參數** | bstrLogInID | 登入ID。 |
| bAsyncOrder | 是否為非同步委託。 |
| pOrder | SKCOM元件中的 [**OVERSEAFUTUREORDERFORGW**](#_OVERSEAFUTUREORDERFORGW(海期委託GW格式物件))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，訊息內容則為修改訊息。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。**  **其他錯誤由「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** | 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](#_OnAsyncOrder)取得。  \*原委託為限價單方可改價(ROD)  \*請填海選商品Call/Put 、履約價欄位。 | |

|  |  |  |
| --- | --- | --- |
| 異動前 | 欄位 | 異動後V2.13.45 |
| \*僅支援自然人身份，尚未支援特殊身份(群組及子帳) | **備註**  刪除身份限制描述 | 如上方表格 |

程式碼可參考 [OverSeaCorrectPriceByBookNo](#_OverSeaCorrectPriceByBookNo) 作法

##### 減量

###### **OverSeaDecreaseOrderBySeqNo**

|  |  |  |
| --- | --- | --- |
| 海期委託減量。 | | |
| **宣告** | Long OverSeaDecreaseOrderBySeqNo([in] BSTR bstrLogInID, [in] VARIANT\_BOOL bAsyncOrder, [in] BSTR bstrAccount, [in] BSTR bstrSeqNo, [in] LONG nDecreaseQty,[**out**] BSTR\* bstrMessage); | |
| **參數** | bstrLogInID | 登入ID。 |
| bAsyncOrder | 是否為非同步委託。 |
| bstrAccount | 委託帳號 ( IB＋帳號) 。 |
| bstrSeqNo | 欲改量的委託序號。 |
| nDecreaseQty | 欲減少的數量。 |
| bstrMessage | **同步委託：**如果回傳值為 0表示委託成功，訊息內容則為修改訊息。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** | 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**減量**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](#_OnAsyncOrder)取得。 | |

程式碼可參考 [OverSeaCorrectPriceByBookNo](#_OverSeaCorrectPriceByBookNo) 作法

###### **OverSeaDecreaseOrderBySeqNoOLID**

|  |  |  |
| --- | --- | --- |
| 海期委託減量OLID (By委託序號)。 | | |
| **宣告** | Long OverSeaDecreaseOrderBySeqNoOLID ([in] BSTR bstrLogInID, [in] VARIANT\_BOOL bAsyncOrder, [in] BSTR bstrAccount, [in] BSTR bstrSeqNo, [in] LONG nDecreaseQty, [in] BSTR bstrOrderLinkedID, [**out**] BSTR\* bstrMessage); | |
| **參數** | bstrLogInID | 登入ID。 |
| bAsyncOrder | 是否為非同步委託。 |
| bstrAccount | 委託帳號 ( IB＋帳號) 。 |
| bstrSeqNo | 欲改量的委託序號。 |
| nDecreaseQty | 欲減少的數量。 |
| 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，訊息內容則為修改訊息。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** | 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**減量**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**OnAsyncOrderOLID**取得。 | |

##### 刪單

###### **OverSeaCancelOrderBySeqNo**

|  |  |  |
| --- | --- | --- |
| 海外期貨委託删單(By委託序號)。 | | |
| **宣告** | Long OverSeaCancelOrderBySeqNo([in] BSTR bstrLogInID, [in] VARIANT\_BOOL bAsyncOrder, [in] BSTR bstrAccount, [in] BSTR bstrSeqNo,[**out**] BSTR\* bstrMessage); | |
| **參數** | bstrLogInID | 登入ID。 |
| bAsyncOrder | 是否為非同步委託。 |
| bstrAccount | 委託帳號 ( IB＋帳號) 。 |
| bstrSeqNo | 欲刪除的委託序號。 |
| bstrMessage | **同步刪單：**如果回傳值為 0表示刪單成功，訊息內容則為原始委託13碼。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](#_OnAsyncOrder)取得。 | |

程式碼可參考 [OverSeaCorrectPriceByBookNo](#_OverSeaCorrectPriceByBookNo) 作法

比較表

|  |  |  |
| --- | --- | --- |
| 異動前 | 欄位 | 異動後V2.13.45 |
| **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。 | **參數**  bstrMessage | **同步刪單：**如果回傳值為 0表示刪單成功，訊息內容則為原始委託  13碼。 |

###### **OverSeaCancelOrderBySeqNoOLID**

|  |  |  |
| --- | --- | --- |
| 海外期貨委託删單OLID (By委託序號)。 | | |
| **宣告** | Long OverSeaCancelOrderBySeqNoOLID ([in] BSTR bstrLogInID, [in] VARIANT\_BOOL bAsyncOrder, [in] BSTR bstrAccount, [in] BSTR bstrSeqNo, [in] BSTR bstrOrderLinkedID, [**out**] BSTR\* bstrMessage); | |
| **參數** | bstrLogInID | 登入ID。 |
| bAsyncOrder | 是否為非同步委託。 |
| bstrAccount | 委託帳號 ( IB＋帳號) 。 |
| bstrSeqNo | 欲刪除的委託序號。 |
| 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，訊息內容則為修改訊息。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。 | |
| **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**OnAsyncOrderOLID**取得。 | |

###### **OverSeaCancelOrderByBookNo**

|  |  |  |
| --- | --- | --- |
| 海外期貨委託删單(By委託書號)。 | | |
| **宣告** | Long OverSeaCancelOrderByBookNo([in] BSTR bstrLogInID, [in] VARIANT\_BOOL bAsyncOrder, [in] BSTR bstrAccount, [in] BSTR bstrBookNo,[**out**] BSTR\* bstrMessage); | |
| **參數** | bstrLogInID | 登入ID。 |
| bAsyncOrder | 是否為非同步委託。 |
| bstrAccount | 委託帳號 ( IB＋帳號) 。 |
| bstrBookNo | 欲刪除的書號。 |
| bstrMessage | **同步刪單：**如果回傳值為 0表示刪單成功，訊息內容則為原始委託13碼。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](#_OnAsyncOrder)取得。 | |

程式碼可參考 [OverSeaCorrectPriceByBookNo](#_OverSeaCorrectPriceByBookNo) 作法

比較表

|  |  |  |
| --- | --- | --- |
| 異動前 | 欄位 | 異動後V2.13.45 |
| **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。 | **參數**  bstrMessage | **同步刪單：**如果回傳值為 0表示刪單成功，訊息內容則為原始  13碼。 |

#### **Proxy下單/刪改單**

##### Proxy Server海期委託/海期價差委託

###### **SendOverseaFutureProxyOrder**

|  |  |  |
| --- | --- | --- |
| 經由proxy server送出海期下單。 | | |
| **宣告** | Long SendOverseaFutureProxyOrder([in] BSTR bstrLogInID, [in] struct OVERSEAFUTUREORDER\* pSKProxyOrder, [out] BSTR\* bstrMessage); | |
| **參數** | bstrLogInID | 登入ID。 |
| pSKProxyOrder | SKCOM元件中的[OVERSEAFUTUREORDER](#_Proxy_Server海期下單)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\_OFOrder.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 |
| # checkBoxSpread          # 是否為價差交易          self.checkBoxSpread = tk.Checkbutton(self)          self.var1 = tk.IntVar()          self.checkBoxSpread["variable"] = self.var1          self.checkBoxSpread["onvalue"] = True          self.checkBoxSpread["offvalue"] = False          self.checkBoxSpread["text"] = "是否為價差交易"          self.checkBoxSpread["command"] = self.checkBoxSpread\_CheckedChanged          self.checkBoxSpread.grid( row = 0,column = 1)          # textBoxOFExchangeNo3          tk.Label(self, text = "交易所代號").grid(row=1, column=1)              #輸入框          self.textBoxOFExchangeNo3 = tk.Entry(self)          self.textBoxOFExchangeNo3.grid(row=1, column=2)          global textBoxOFExchangeNo3          textBoxOFExchangeNo3 = self.textBoxOFExchangeNo3          # textBoxOFStockNo3          tk.Label(self, text = "海外期貨代號").grid(row=2, column=1)              #輸入框          self.textBoxOFStockNo3 = tk.Entry(self)          self.textBoxOFStockNo3.grid(row=2, column=2)          global textBoxOFStockNo3          textBoxOFStockNo3 = self.textBoxOFStockNo3            # textBoxOFYearMonth3          tk.Label(self, text = "近月商品年月(YYYYMM)").grid(row=3, column=1)              #輸入框          self.textBoxOFYearMonth3 = tk.Entry(self)          self.textBoxOFYearMonth3.grid(row=3, column=2)          global textBoxOFYearMonth3          textBoxOFYearMonth3 = self.textBoxOFYearMonth3            # textBoxOFYearMonth23          tk.Label(self, text = "遠月商品年月(YYYYMM)").grid(row=4, column=1)              #輸入框          self.textBoxOFYearMonth23 = tk.Entry(self)          self.textBoxOFYearMonth23.grid(row=4, column=2)          global textBoxOFYearMonth23          textBoxOFYearMonth23 = self.textBoxOFYearMonth23            # textBoxOFOrder3          tk.Label(self, text = "委託價").grid(row=5, column=1)              #輸入框          self.textBoxOFOrder3 = tk.Entry(self)          self.textBoxOFOrder3.grid(row=5, column=2)          global textBoxOFOrder3          textBoxOFOrder3 = self.textBoxOFOrder3            # textBoxOFOrderNumerator3          tk.Label(self, text = "委託價分子").grid(row=6, column=1)              #輸入框          self.textBoxOFOrderNumerator3 = tk.Entry(self)          self.textBoxOFOrderNumerator3.grid(row=6, column=2)          global textBoxOFOrderNumerator3          textBoxOFOrderNumerator3 = self.textBoxOFOrderNumerator3            # textBoxOFTrigger3          tk.Label(self, text = "觸發價").grid(row=7, column=1)              #輸入框          self.textBoxOFTrigger3 = tk.Entry(self)          self.textBoxOFTrigger3.grid(row=7, column=2)          global textBoxOFTrigger3          textBoxOFTrigger3 = self.textBoxOFTrigger3            # textBoxOFTriggerNumerator3          tk.Label(self, text = "觸發價分子").grid(row=8, column=1)              #輸入框          self.textBoxOFTriggerNumerator3 = tk.Entry(self)          self.textBoxOFTriggerNumerator3.grid(row=8, column=2)          global textBoxOFTriggerNumerator3          textBoxOFTriggerNumerator3 = self.textBoxOFTriggerNumerator3            # comboBoxOFBuySell3          tk.Label(self, text = "買進/賣出").grid(row=9, column=1)              #輸入框          self.comboBoxOFBuySell3 = ttk.Combobox(self, state='readonly')          self.comboBoxOFBuySell3['values'] = Config.comboBoxOFBuySell3          self.comboBoxOFBuySell3.grid(row=9, column=2)          global comboBoxOFBuySell3          comboBoxOFBuySell3 = self.comboBoxOFBuySell3            # comboBoxOFDayTrade3          tk.Label(self, text = "當沖").grid(row=10, column=1)              #輸入框          self.comboBoxOFDayTrade3 = ttk.Combobox(self, state='readonly')          self.comboBoxOFDayTrade3['values'] = Config.comboBoxOFDayTrade3          self.comboBoxOFDayTrade3.grid(row=10, column=2)          global comboBoxOFDayTrade3          comboBoxOFDayTrade3 = self.comboBoxOFDayTrade3            # comboBoxOFTradeType3          tk.Label(self, text = "ROD/FOK/IOC").grid(row=11, column=1)              #輸入框          self.comboBoxOFTradeType3 = ttk.Combobox(self, state='readonly')          self.comboBoxOFTradeType3['values'] = Config.comboBoxOFTradeType3          self.comboBoxOFTradeType3.grid(row=11, column=2)          global comboBoxOFTradeType3          comboBoxOFTradeType3 = self.comboBoxOFTradeType3            # comboBoxOFSpecialTradeType3          tk.Label(self, text = "LMT/MKT/STL/STP").grid(row=12, column=1)              #輸入框          self.comboBoxOFSpecialTradeType3 = ttk.Combobox(self, state='readonly')          self.comboBoxOFSpecialTradeType3['values'] = Config.comboBoxOFSpecialTradeType3          self.comboBoxOFSpecialTradeType3.grid(row=12, column=2)          global comboBoxOFSpecialTradeType3          comboBoxOFSpecialTradeType3 = self.comboBoxOFSpecialTradeType3            # textBoxOFQty3          tk.Label(self, text = "交易口數").grid(row=13, column=1)              #輸入框          self.textBoxOFQty3 = tk.Entry(self)          self.textBoxOFQty3.grid(row=13, column=2)          global textBoxOFQty3          textBoxOFQty3 = self.textBoxOFQty3          # buttonSendOverseaFutureSpreadProxyOrder          self.buttonSendOverseaFutureSpreadProxyOrder = tk.Button(self)          self.buttonSendOverseaFutureSpreadProxyOrder["text"] = "海外期貨送出"          self.buttonSendOverseaFutureSpreadProxyOrder["command"] = self.buttonSendOverseaFutureSpreadProxyOrder\_Click          self.buttonSendOverseaFutureSpreadProxyOrder.grid(row=14, column=1) |

|  |
| --- |
| Config.py |
| # SendProxyForm  comboBoxOFBuySell3 =(      "買進", "賣出"  )  comboBoxOFDayTrade3 =(      "否", "是"  )  comboBoxOFTradeType3 =(      "ROD", "FOK", "IOC"  )  comboBoxOFSpecialTradeType3 =(      "LMT限價單", "MKT市價單", "STL停損限價", "STP停損市價"  ) |
| 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)  # 是否為價差交易  Spread = False |

|  |
| --- |
| Current Block(放置於def createWidgets(self): 的同一層) |
| # checkBoxSpread      def checkBoxSpread\_CheckedChanged(self):          global Spread          if self.var1.get() == True:              Spread = True          else:              Spread = False      def buttonSendOverseaFutureSpreadProxyOrder\_Click(self):          pOrder = sk.OVERSEAFUTUREORDER()          pOrder.bstrFullAccount = comboBoxAccount.get()          pOrder.bstrStockNo = textBoxOFStockNo3.get()          pOrder.bstrExchangeNo = textBoxOFExchangeNo3.get()          pOrder.bstrYearMonth = textBoxOFYearMonth3.get()          pOrder.bstrYearMonth2 = textBoxOFYearMonth23.get()          pOrder.bstrOrder = textBoxOFOrder3.get()          pOrder.bstrOrderNumerator = textBoxOFOrderNumerator3.get()          pOrder.bstrTrigger = textBoxOFTrigger3.get()          pOrder.bstrTriggerNumerator = textBoxOFTriggerNumerator3.get()          pOrder.sNewClose = 0          if (comboBoxOFTradeType3.get() == "ROD"):              pOrder.sTradeType = 0          elif (comboBoxOFTradeType3.get() == "IOC"):              pOrder.sTradeType = 1          elif (comboBoxOFTradeType3.get() == "FOK"):              pOrder.sTradeType = 2          if (comboBoxOFBuySell3.get() == "買進"):              pOrder.sBuySell = 0          elif (comboBoxOFBuySell3.get() == "賣出"):              pOrder.sBuySell = 1          if (comboBoxOFDayTrade3.get() == "否"):              pOrder.sDayTrade = 0          elif (comboBoxOFDayTrade3.get() == "是"):              pOrder.sDayTrade = 1          if (comboBoxOFSpecialTradeType3.get() == "LMT限價單"):              pOrder.sSpecialTradeType = 0          elif (comboBoxOFSpecialTradeType3.get() == "MKT市價單"):              pOrder.sSpecialTradeType = 1          elif (comboBoxOFSpecialTradeType3.get() == "STL停損限價"):              pOrder.sSpecialTradeType = 2          elif (comboBoxOFSpecialTradeType3.get() == "STP停損市價"):              pOrder.sSpecialTradeType = 3          pOrder.nQty = int(textBoxOFQty3.get())          if (Spread == False):              bstrMessage,nCode= m\_pSKOrder.SendOverseaFutureProxyOrder(comboBoxUserID.get(), pOrder)              if bstrMessage is not None:                  msg = "【SendOverseaFutureProxyOrder】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode) + bstrMessage              else:                  msg = "【SendOverseaFutureProxyOrder】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode) + " No message"              richTextBoxMethodMessage.insert('end',  msg + "\n")              richTextBoxMethodMessage.see('end')          else:              # 送出海期委託              bstrMessage,nCode= m\_pSKOrder.SendOverseaFutureSpreadProxyOrder(comboBoxUserID.get(), pOrder)              if bstrMessage is not None:                  msg = "【SendOverseaFutureSpreadProxyOrder】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode) + bstrMessage              else:                  msg = "【SendOverseaFutureSpreadProxyOrder】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode) + " No message"              richTextBoxMethodMessage.insert('end',  msg + "\n")              richTextBoxMethodMessage.see('end') |

###### **SendOverseaFutureSpreadProxyOrder**

|  |  |  |
| --- | --- | --- |
| 經由proxy server送出海期價差單下單。 | | |
| **宣告** | Long SendOverseaFutureSpreadProxyOrder ([in] BSTR bstrLogInID, [in] struct OVERSEAFUTUREORDER\* pSKProxyOrder, [out] BSTR\* bstrMessage); | |
| **參數** | bstrLogInID | 登入ID。 |
| pSKProxyOrder | SKCOM元件中的[OVERSEAFUTUREORDER](#_Proxy_Server海期價差下單_1)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。 | |

##### SendOverseaOptionProxyOrder

|  |  |  |
| --- | --- | --- |
| 經由proxy server送出海選下單。 | | |
| **宣告** | Long SendOverseaOptionProxyOrder ([in] BSTR bstrLogInID, [in] struct OVERSEAFUTUREORDER\* pSKProxyOrder, [out] BSTR\* bstrMessage); | |
| **參數** | bstrLogInID | 登入ID。 |
| pSKProxyOrder | SKCOM元件中的[OVERSEAFUTUREORDER](#_Proxy_Server海選下單_1)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\_OFOrder.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  一張含有 文字, 螢幕擷取畫面, 數字, 平行 的圖片  自動產生的描述 |
| # textBoxOOExchangeNo3          tk.Label(self, text = "交易所代號").grid(row=15, column=1)              #輸入框          self.textBoxOOExchangeNo3 = tk.Entry(self)          self.textBoxOOExchangeNo3.grid(row=15, column=2)          global textBoxOOExchangeNo3          textBoxOOExchangeNo3 = self.textBoxOOExchangeNo3          # textBoxOOStockNo3          tk.Label(self, text = "海外選擇權代號").grid(row=16, column=1)              #輸入框          self.textBoxOOStockNo3 = tk.Entry(self)          self.textBoxOOStockNo3.grid(row=16, column=2)          global textBoxOOStockNo3          textBoxOOStockNo3 = self.textBoxOOStockNo3            # textBoxOOYearMonth3          tk.Label(self, text = "近月商品年月(YYYYMM)").grid(row=17, column=1)              #輸入框          self.textBoxOOYearMonth3 = tk.Entry(self)          self.textBoxOOYearMonth3.grid(row=17, column=2)          global textBoxOOYearMonth3          textBoxOOYearMonth3 = self.textBoxOOYearMonth3            # textBoxOOOrder3          tk.Label(self, text = "委託價").grid(row=18, column=1)              #輸入框          self.textBoxOOOrder3 = tk.Entry(self)          self.textBoxOOOrder3.grid(row=18, column=2)          global textBoxOOOrder3          textBoxOOOrder3 = self.textBoxOOOrder3            # textBoxOOOrderNumerator3          tk.Label(self, text = "委託價分子").grid(row=19, column=1)              #輸入框          self.textBoxOOOrderNumerator3 = tk.Entry(self)          self.textBoxOOOrderNumerator3.grid(row=19, column=2)          global textBoxOOOrderNumerator3          textBoxOOOrderNumerator3 = self.textBoxOOOrderNumerator3            # textBoxbstrOrderDenominator3          tk.Label(self, text = "分母").grid(row=19, column=3)              #輸入框          self.textBoxbstrOrderDenominator3 = tk.Entry(self)          self.textBoxbstrOrderDenominator3.grid(row=19, column=4)          global textBoxbstrOrderDenominator3          textBoxbstrOrderDenominator3 = self.textBoxbstrOrderDenominator3            # textBoxOOTrigger3          tk.Label(self, text = "觸發價").grid(row=20, column=1)              #輸入框          self.textBoxOOTrigger3 = tk.Entry(self)          self.textBoxOOTrigger3.grid(row=20, column=2)          global textBoxOOTrigger3          textBoxOOTrigger3 = self.textBoxOOTrigger3            # textBoxOOTriggerNumerator3          tk.Label(self, text = "觸發價分子").grid(row=21, column=1)              #輸入框          self.textBoxOOTriggerNumerator3 = tk.Entry(self)          self.textBoxOOTriggerNumerator3.grid(row=21, column=2)          global textBoxOOTriggerNumerator3          textBoxOOTriggerNumerator3 = self.textBoxOOTriggerNumerator3          # comboBoxOOBuySell3          tk.Label(self, text = "買進/賣出").grid(row=22, column=1)              #輸入框          self.comboBoxOOBuySell3 = ttk.Combobox(self, state='readonly')          self.comboBoxOOBuySell3['values'] = Config.comboBoxOOBuySell3          self.comboBoxOOBuySell3.grid(row=22, column=2)          global comboBoxOOBuySell3          comboBoxOOBuySell3 = self.comboBoxOOBuySell3            # comboBoxOONewClose3          tk.Label(self, text = "新平倉").grid(row=23, column=1)              #輸入框          self.comboBoxOONewClose3 = ttk.Combobox(self, state='readonly')          self.comboBoxOONewClose3['values'] = Config.comboBoxOONewClose3          self.comboBoxOONewClose3.grid(row=23, column=2)          global comboBoxOONewClose3          comboBoxOONewClose3 = self.comboBoxOONewClose3            # comboBoxOODayTrade3          tk.Label(self, text = "當沖").grid(row=24, column=1)              #輸入框          self.comboBoxOODayTrade3 = ttk.Combobox(self, state='readonly')          self.comboBoxOODayTrade3['values'] = Config.comboBoxOODayTrade3          self.comboBoxOODayTrade3.grid(row=24, column=2)          global comboBoxOODayTrade3          comboBoxOODayTrade3 = self.comboBoxOODayTrade3            # comboBoxOOSpecialTradeType3          tk.Label(self, text = "LMT/MKT/STL/STP").grid(row=25, column=1)              #輸入框          self.comboBoxOOSpecialTradeType3 = ttk.Combobox(self, state='readonly')          self.comboBoxOOSpecialTradeType3['values'] = Config.comboBoxOOSpecialTradeType3          self.comboBoxOOSpecialTradeType3.grid(row=25, column=2)          global comboBoxOOSpecialTradeType3          comboBoxOOSpecialTradeType3 = self.comboBoxOOSpecialTradeType3            # textBoxOOStrikePrice3          tk.Label(self, text = "履約價").grid(row=26, column=1)              #輸入框          self.textBoxOOStrikePrice3 = tk.Entry(self)          self.textBoxOOStrikePrice3.grid(row=26, column=2)          global textBoxOOStrikePrice3          textBoxOOStrikePrice3 = self.textBoxOOStrikePrice3            # comboBoxOOCallPut34          tk.Label(self, text = "CALL/PUT").grid(row=27, column=1)              #輸入框          self.comboBoxOOCallPut34 = ttk.Combobox(self, state='readonly')          self.comboBoxOOCallPut34['values'] = Config.comboBoxOOCallPut34          self.comboBoxOOCallPut34.grid(row=27, column=2)          global comboBoxOOCallPut34          comboBoxOOCallPut34 = self.comboBoxOOCallPut34            # textBoxOOQty3          tk.Label(self, text = "交易口數").grid(row=28, column=1)              #輸入框          self.textBoxOOQty3 = tk.Entry(self)          self.textBoxOOQty3.grid(row=28, column=2)          global textBoxOOQty3          textBoxOOQty3 = self.textBoxOOQty3          # buttonSendOverseaOptionProxyOrder          self.buttonSendOverseaOptionProxyOrder = tk.Button(self)          self.buttonSendOverseaOptionProxyOrder["text"] = "海外選擇權送出"          self.buttonSendOverseaOptionProxyOrder["command"] = self.buttonSendOverseaOptionProxyOrder\_Click          self.buttonSendOverseaOptionProxyOrder.grid(row=29, column=1) |

|  |
| --- |
| Config.py |
| comboBoxOOBuySell3 =(      "買進", "賣出"  )  comboBoxOONewClose3 =(      "新倉", "平倉"  )  comboBoxOODayTrade3 =(      "否", "是"  )  comboBoxOOSpecialTradeType3 =(      "LMT限價單", "MKT市價單", "STL停損限價", "STP停損市價"  )  comboBoxOOCallPut34 =(      "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\_pSKOrder = comtypes.client.CreateObject(sk.SKOrderLib,interface=sk.ISKOrderLib) |

|  |
| --- |
| Current Block(放置於def createWidgets(self): 的同一層) |
| def buttonSendOverseaOptionProxyOrder\_Click(self):          pOrder = sk.OVERSEAFUTUREORDER()          pOrder.bstrFullAccount = comboBoxAccount.get()          pOrder.bstrStockNo = textBoxOOStockNo3.get()          pOrder.bstrExchangeNo = textBoxOOExchangeNo3.get()          pOrder.bstrYearMonth = textBoxOOYearMonth3.get()          pOrder.bstrOrder = textBoxOOOrder3.get()          pOrder.bstrOrderNumerator = textBoxOOOrderNumerator3.get()          pOrder.bstrOrderDenominator = textBoxbstrOrderDenominator3.get()          pOrder.bstrTrigger = textBoxOOTrigger3.get()          pOrder.bstrTriggerNumerator = textBoxOOTriggerNumerator3.get()          pOrder.sTradeType = 0          if (comboBoxOOBuySell3.get() == "買進"):              pOrder.sBuySell = 0          elif (comboBoxOOBuySell3.get() == "賣出"):              pOrder.sBuySell = 1          if (comboBoxOODayTrade3.get() == "否"):              pOrder.sDayTrade = 0          elif (comboBoxOODayTrade3.get() == "是"):              pOrder.sDayTrade = 1          if (comboBoxOONewClose3.get() == "新倉"):              pOrder.sNewClose = 0          elif (comboBoxOONewClose3.get() == "平倉"):              pOrder.sNewClose = 1          if (comboBoxOOSpecialTradeType3.get() == "LMT限價單"):              pOrder.sSpecialTradeType = 0          elif (comboBoxOOSpecialTradeType3.get() == "MKT市價單"):              pOrder.sSpecialTradeType = 1          elif (comboBoxOOSpecialTradeType3.get() == "STL停損限價"):              pOrder.sSpecialTradeType = 2          elif (comboBoxOOSpecialTradeType3.get() == "STP停損市價"):              pOrder.sSpecialTradeType = 3          pOrder.bstrStrikePrice = textBoxOOStrikePrice3.get()          pOrder.nQty = int(textBoxOOQty3.get())          if (comboBoxOOCallPut34.get() == "CALL"):              pOrder.sCallPut = 0          elif (comboBoxOOCallPut34.get() == "PUT"):              pOrder.sCallPut = 1          # 送出選擇權委託          bstrMessage,nCode= m\_pSKOrder.SendOverseaOptionProxyOrder(comboBoxUserID.get(), pOrder)            if bstrMessage is not None:              msg = "【SendOverseaOptionProxyOrder】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode) + bstrMessage          else:              msg = "【SendOverseaOptionProxyOrder】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode) + " No message"          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

##### SendOverseaFutureProxyAlter

|  |  |  |
| --- | --- | --- |
| 經由proxy server送出海期選刪改單。 | | |
| **宣告** | Long SendOverseaFutureProxyAlter([in] BSTR bstrLogInID, [in] struct OVERSEAFUTUREORDER\* pAsyncOrder, [out] BSTR\* bstrMessage); | |
| **參數** | bstrLogInID | 登入ID。 |
| pSKProxyOrder | SKCOM元件中的[OVERSEAFUTUREORDER](#_Proxy_Server海期選刪改單_1)物件，將刪單或改單條件填入該物件後，再帶入此欄位中。  Proxy 改單功能物件說明 |
| 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\_OFOrder.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 |
| # textBoxOverSeaCancelOrderBySeqNo5          tk.Label(self, text = "請輸入委託序號").grid(row=0, column=1)              #輸入框          self.textBoxOverSeaCancelOrderBySeqNo5 = tk.Entry(self)          self.textBoxOverSeaCancelOrderBySeqNo5.grid(row=0, column=2)          global textBoxOverSeaCancelOrderBySeqNo5          textBoxOverSeaCancelOrderBySeqNo5 = self.textBoxOverSeaCancelOrderBySeqNo5          # textBoxOverSeaCancelOrderByBookNo5          tk.Label(self, text = "請輸入委託書號").grid(row=1, column=1)              #輸入框          self.textBoxOverSeaCancelOrderByBookNo5 = tk.Entry(self)          self.textBoxOverSeaCancelOrderByBookNo5.grid(row=1, column=2)          global textBoxOverSeaCancelOrderByBookNo5          textBoxOverSeaCancelOrderByBookNo5 = self.textBoxOverSeaCancelOrderByBookNo5          # textBoxOverseaFutureDecreaseQty5          tk.Label(self, text = "請輸入減少數量").grid(row=2, column=1)              #輸入框          self.textBoxOverseaFutureDecreaseQty5 = tk.Entry(self)          self.textBoxOverseaFutureDecreaseQty5.grid(row=2, column=2)          global textBoxOverseaFutureDecreaseQty5          textBoxOverseaFutureDecreaseQty5 = self.textBoxOverseaFutureDecreaseQty5          # textBoxOFExchangeNo5          tk.Label(self, text = "交易所代號").grid(row=3, column=1)              #輸入框          self.textBoxOFExchangeNo5 = tk.Entry(self)          self.textBoxOFExchangeNo5.grid(row=3, column=2)          global textBoxOFExchangeNo5          textBoxOFExchangeNo5 = self.textBoxOFExchangeNo5          # textBoxOFStockNo5          tk.Label(self, text = "海外期權代號").grid(row=4, column=1)              #輸入框          self.textBoxOFStockNo5 = tk.Entry(self)          self.textBoxOFStockNo5.grid(row=4, column=2)          global textBoxOFStockNo5          textBoxOFStockNo5 = self.textBoxOFStockNo5            # textBoxOFYearMonth5          tk.Label(self, text = "近月商品年月(YYYYMM)").grid(row=5, column=1)              #輸入框          self.textBoxOFYearMonth5 = tk.Entry(self)          self.textBoxOFYearMonth5.grid(row=5, column=2)          global textBoxOFYearMonth5          textBoxOFYearMonth5 = self.textBoxOFYearMonth5            # textBoxOFYearMonth25          tk.Label(self, text = "遠月商品年月(YYYYMM)").grid(row=6, column=1)              #輸入框          self.textBoxOFYearMonth25 = tk.Entry(self)          self.textBoxOFYearMonth25.grid(row=6, column=2)          global textBoxOFYearMonth25          textBoxOFYearMonth25 = self.textBoxOFYearMonth25            # textBoxOFOrder5          tk.Label(self, text = "新委託價").grid(row=7, column=1)              #輸入框          self.textBoxOFOrder5 = tk.Entry(self)          self.textBoxOFOrder5.grid(row=7, column=2)          global textBoxOFOrder5          textBoxOFOrder5 = self.textBoxOFOrder5            # textBoxOFOrderNumerator5          tk.Label(self, text = "新委託價分子").grid(row=8, column=1)              #輸入框          self.textBoxOFOrderNumerator5 = tk.Entry(self)          self.textBoxOFOrderNumerator5.grid(row=8, column=2)          global textBoxOFOrderNumerator5          textBoxOFOrderNumerator5 = self.textBoxOFOrderNumerator5            # textBoxOFOrderDenominator5          tk.Label(self, text = "新委託價分母").grid(row=9, column=1)              #輸入框          self.textBoxOFOrderDenominator5 = tk.Entry(self)          self.textBoxOFOrderDenominator5.grid(row=9, column=2)          global textBoxOFOrderDenominator5          textBoxOFOrderDenominator5 = self.textBoxOFOrderDenominator5          # comboBoxOONewClose5          tk.Label(self, text = "新平倉").grid(row=10, column=1)              #輸入框          self.comboBoxOONewClose5 = ttk.Combobox(self, state='readonly')          self.comboBoxOONewClose5['values'] = Config.comboBoxOONewClose5          self.comboBoxOONewClose5.grid(row=10, column=2)          global comboBoxOONewClose5          comboBoxOONewClose5 = self.comboBoxOONewClose5            # comboBoxOOSpecialTradeType5          tk.Label(self, text = "LMT/MKT/STL/STP").grid(row=11, column=1)              #輸入框          self.comboBoxOOSpecialTradeType5 = ttk.Combobox(self, state='readonly')          self.comboBoxOOSpecialTradeType5['values'] = Config.comboBoxOOSpecialTradeType5          self.comboBoxOOSpecialTradeType5.grid(row=11, column=2)          global comboBoxOOSpecialTradeType5          comboBoxOOSpecialTradeType5 = self.comboBoxOOSpecialTradeType5            # textBoxOOStrikePrice5          tk.Label(self, text = "履約價(改期貨帶0)").grid(row=12, column=1)              #輸入框          self.textBoxOOStrikePrice5 = tk.Entry(self)          self.textBoxOOStrikePrice5.grid(row=12, column=2)          global textBoxOOStrikePrice5          textBoxOOStrikePrice5 = self.textBoxOOStrikePrice5            # comboBoxOOCallPut5          tk.Label(self, text = "CALL/PUT").grid(row=13, column=1)              #輸入框          self.comboBoxOOCallPut5 = ttk.Combobox(self, state='readonly')          self.comboBoxOOCallPut5['values'] = Config.comboBoxOOCallPut5          self.comboBoxOOCallPut5.grid(row=13, column=2)          global comboBoxOOCallPut5          comboBoxOOCallPut5 = self.comboBoxOOCallPut5            # comboBoxnSpreadFlag5          tk.Label(self, text = "市場別").grid(row=14, column=1)              #輸入框          self.comboBoxnSpreadFlag5 = ttk.Combobox(self, state='readonly')          self.comboBoxnSpreadFlag5['values'] = Config.comboBoxnSpreadFlag5          self.comboBoxnSpreadFlag5.grid(row=14, column=2)          global comboBoxnSpreadFlag5          comboBoxnSpreadFlag5 = self.comboBoxnSpreadFlag5            # comboBoxnAlterType5          tk.Label(self, text = "異動項目").grid(row=15, column=1)              #輸入框          self.comboBoxnAlterType5 = ttk.Combobox(self, state='readonly')          self.comboBoxnAlterType5['values'] = Config.comboBoxnAlterType5          self.comboBoxnAlterType5.grid(row=15, column=2)          global comboBoxnAlterType5          comboBoxnAlterType5 = self.comboBoxnAlterType5          # buttonSendOverseaFutureProxyAlter          self.buttonSendOverseaFutureProxyAlter = tk.Button(self)          self.buttonSendOverseaFutureProxyAlter["text"] = "海外期貨刪改單送出"          self.buttonSendOverseaFutureProxyAlter["command"] = self.buttonSendOverseaFutureProxyAlter\_Click          self.buttonSendOverseaFutureProxyAlter.grid(row=16, column=1) |

|  |
| --- |
| Config.py |
| # UpdateProxyForm  comboBoxOONewClose5 =(      "新倉", "平倉"  )  comboBoxOOSpecialTradeType5 =(      "LMT限價單", "MKT市價單", "STL停損限價", "STP停損市價"  )  comboBoxOOCallPut5 =(      "CALL", "PUT"  )  comboBoxnSpreadFlag5 =(      "0 :OF海期", "1: OF-spread 海期價差", "2: OO 海選"  )  comboBoxnAlterType5 =(      "0: Cancel 刪單", "1: Decrease 減量", "2: Correct 改價"  ) |
| 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 buttonSendOverseaFutureProxyAlter\_Click(self):          pSKProxyOrder = sk.OVERSEAFUTUREORDER()          pSKProxyOrder.bstrFullAccount = comboBoxAccount.get()          pSKProxyOrder.bstrExchangeNo = textBoxOFExchangeNo5.get()          pSKProxyOrder.bstrStockNo = textBoxOFStockNo5.get()          pSKProxyOrder.bstrYearMonth = textBoxOFYearMonth5.get()          pSKProxyOrder.bstrYearMonth2 = textBoxOFYearMonth25.get()          pSKProxyOrder.bstrOrder = textBoxOFOrder5.get()          pSKProxyOrder.bstrOrderNumerator = textBoxOFOrderNumerator5.get()          pSKProxyOrder.bstrOrderDenominator = textBoxOFOrderDenominator5.get()          pSKProxyOrder.bstrStrikePrice = textBoxOOStrikePrice5.get()          if (comboBoxOOCallPut5.get() == "CALL"):              pSKProxyOrder.sCallPut = 0          elif (comboBoxOOCallPut5.get() == "PUT"):              pSKProxyOrder.sCallPut = 1          if (comboBoxOONewClose5.get() == "新倉"):              pSKProxyOrder.sNewClose = 0          elif (comboBoxOONewClose5.get() == "平倉"):              pSKProxyOrder.sNewClose = 1          pSKProxyOrder.sTradeType = 0          if (comboBoxOOSpecialTradeType5.get() == "LMT限價單"):              pSKProxyOrder.sSpecialTradeType = 0          elif (comboBoxOOSpecialTradeType5.get() == "MKT市價單"):              pSKProxyOrder.sSpecialTradeType = 1          elif (comboBoxOOSpecialTradeType5.get() == "STL停損限價"):              pSKProxyOrder.sSpecialTradeType = 2          elif (comboBoxOOSpecialTradeType5.get() == "STP停損市價"):              pSKProxyOrder.sSpecialTradeType = 3          pSKProxyOrder.nQty = int(textBoxOverseaFutureDecreaseQty5.get())          pSKProxyOrder.bstrBookNo = textBoxOverSeaCancelOrderByBookNo5.get()          pSKProxyOrder.bstrSeqNo = textBoxOverSeaCancelOrderBySeqNo5.get()          if (comboBoxnSpreadFlag5.get() == "0 :OF海期"):              pSKProxyOrder.nSpreadFlag = 0          elif (comboBoxnSpreadFlag5.get() == "1: OF-spread 海期價差"):              pSKProxyOrder.nSpreadFlag = 1          elif (comboBoxnSpreadFlag5.get() == "2: OO 海選"):              pSKProxyOrder.nSpreadFlag = 2          if (comboBoxnAlterType5.get() == "0: Cancel 刪單"):              pSKProxyOrder.nAlterType = 0          elif (comboBoxnAlterType5.get() == "1: Decrease 減量"):              pSKProxyOrder.nAlterType = 1          elif (comboBoxnAlterType5.get() == "2: Correct 改價"):              pSKProxyOrder.nAlterType = 2          # 經由proxy server送出海期選刪改單          bstrMessage,nCode= m\_pSKOrder.SendOverseaFutureProxyAlter(comboBoxUserID.get(), pSKProxyOrder)          if bstrMessage is not None:               msg = "【SendOverseaFutureProxyAlter】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode) + bstrMessage          else:               msg = "【SendOverseaFutureProxyAlter】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode) + " No message"          richTextBoxMethodMessage.insert('end',  msg + "\n")          richTextBoxMethodMessage.see('end') |

# 事件

## OnOverseaFuture

|  |  |  |
| --- | --- | --- |
| 海外期貨下單商品。透過呼叫 [GetOverseaFutures](#_GetOverseaFutures) 後，資訊由該事件回傳。 | | |
| **宣告** | void OnOverseaFuture([in] BSTR bstrData); | |
| **參數** | bstrData | 回傳得字串內容，以分號(;)分隔每一個欄位，欄位依序為：  交易所代碼;  交易所名稱;  商品代碼;  商品名稱;  年月;  跳動點;  分母;  可接受交易種類;  可當沖  委託時效(ROD;FOK;IOC) |
| **備註** | 2020/11 新增委託時效欄位 | |

|  |
| --- |
| Python範例程式碼\_OFOrder.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) |
| # 海外期貨下單商品。透過呼叫 GetOverseaFutures 後，資訊由該事件回傳。      def OnOverseaFuture(self, bstrData):          msg = "【OnOverseaFuture】" + bstrData          richTextBoxMessage.insert('end', msg + "\n")          richTextBoxMessage.see('end') |

## OnOverseaOption

|  |  |  |
| --- | --- | --- |
| 海外選擇權下單商品。透過呼叫 [GetOverseaOptions](#_GetOverseaOptions) 後，資訊由該事件回傳。 | | |
| **宣告** | void OnOverseaOption([in] BSTR bstrData); | |
| **參數** | bstrData | 回傳得字串內容，以逗號(,)分隔每一個欄位，欄位依序為：  下單交易所代碼,  交易所名稱,  下單商品代碼,  商品名稱,  商品年月,  價格跳動點,  履約價最小跳動點,  基準履約價,  最低履約價,  最高履約價,  履約價除數,  分母,  可委託類型,  當沖減收保證金,  標的年月 |
| **備註** | * 價格跳動點顯示範例   例： 5|0.05 5元(不含)以下,0.05以“|“管線隔開 以“**/**“隔開每筆資料  0|0.0025|N/ 100000000|0.005|N  ＜＝代表0~100000000 之間最小價格跳動點為0.0025, 100000000以上為0.005；N代表不顯示成分數  99999999|0.1|N  <=代表０～99999999之間，最小價格跳動點為0.1；N代表不顯示成分數  99999999|1|Y  <=代表０～99999999之間，最小價格跳動點為1；Ｙ代表價格顯示成分數  10|0.1|N/ 50|0.5|N/ 500|1|N/ 1000|5|N/ 100000000|10|N  <=0~10~50~500~1000~100000000000（跳動點由左至右區間分別為0.1－0.5－1－5－10 ；N代表不顯示成分數   * 履約價範例   例：最高履約價 1300，最低履約價 1050 履約跳動點5 履約價除數10  原先5點一跳 1050、1055、1060…….1300，但需除以履約價除數10  最終履約價價格105、105.5、106、106.5……125.5、130 | |

|  |
| --- |
| Python範例程式碼\_OFOrder.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) |
| # 海外選擇權下單商品。透過呼叫 GetOverseaOptions 後，資訊由該事件回傳。      def OnOverseaOption(self, bstrData):          msg = "【OnOverseaOption】" + bstrData;          richTextBoxMessage.insert('end', msg + "\n")          richTextBoxMessage.see('end') |

## OnOFOpenInterestGWReport

|  |  |  |
| --- | --- | --- |
| 海外期貨未平倉**彙總**資料。透過呼叫[GetOverseaFutureOpenInterestGW](#_GetOverseaFutureOpenInterestGW) 後，資訊由該事件回傳。 | | |
| **宣告** | void OnOFOpenInterestGWReport([in] BSTR bstrData); | |
| **參數** | bstrData | 第一筆回傳內容為 「@@」開頭，表示查詢狀態，每一筆資料以「,」分隔每一個欄位，欄位依序為「@@,訊息代碼,訊息內容」  其中，當訊息代碼為  0：表示查詢成功，訊息內容為空字串  其他非「0」皆表示查詢錯誤，訊息內容將描述錯誤原因  如有未平倉資訊將從第二筆開始回傳，每一筆資料以「,」分隔每一個欄位，欄位依序為：  海期：   1. 帳號 2. 交易所代號 3. 交易所中文名稱 4. 含年月-商品代號，以空白間隔   （EX: NG 202012）   1. 不含年月-商品代碼（EX: NG） 2. 商品年月 3. 商品中文名稱(含年月，以空白間隔)   EX: 日經225指 202012   1. 買賣別 2. 未平口數 3. 平均價(又稱成交均價) 4. (非即時)現價又稱市價 5. 未平損益 6. 委託買 7. 買方委託口數 8. 委託賣 9. 賣方委託口數 10. 賣方委託新倉口數 11. 昨日結算價（EX:379 02/4）   若價格含分子分母，中間以空白間隔整數、分子分母。   1. 當沖未平口數 2. 原始保證金 3. 是否為選擇權(0->期貨 1->選擇權)   海選:   1. 帳號 2. 交易所代號 3. 交易所中文名稱 4. 商品代號含年月、履約價、C/P(EX: C00380L0) 5. 不含年月-商品代碼（EX:C　為玉米 ） 6. 商品年月 7. 商品中文名稱（含CALL/PUT、年月、履約價、C/P，以空白間隔）   EX: 玉米 CALL 202012 380 C   1. 買賣別 2. 未平口數 3. 平均價(又稱成交均價) 4. (非即時)現價又稱市價 5. 未平損益 6. 委託買 7. 買方委託口數 8. 委託賣 9. 賣方委託口數 10. 賣方委託新倉口數 11. 昨日結算價   若價格含分子分母，中間以空白間隔整數與分子分母。  EX:379 02/4   1. 當沖未平口數 2. 原始保證金 3. 是否為選擇權(0->期貨 1->選擇權) 4. 履約價 5. C/P (C:Call/P:Put) |
| **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。 | |

|  |  |  |
| --- | --- | --- |
| 海外期貨未平倉**明細**資料。透過呼叫[GetOverseaFutureOpenInterest](#_4-2-25_GetOverSeaFutureOpenInterest)GW 後，資訊由該事件回傳。 | | |
| **宣告** | void OnOFOpenInterestGWReport( [in] BSTR bstrData ); | |
| **參數** | bstrData | 每一筆資料以「,」分隔每一個欄位，欄位依序為：  海期：   1. 帳號 2. 成交日期 3. 交易所代碼 4. 交易所中文名稱 5. 商品代碼不含年月(EX: SSI) 6. 商品年月 7. 商品中文名稱(含年月，以空白間隔)   EX: 日經225指 202012   1. 買賣別 2. 成交口數 3. 未平倉口數 4. 成交價 5. 現價 6. 未平損益 7. 委託書序號(前2碼櫃號) 8. 當沖註記 9. 原始保證金 10. 是否為選擇權(0->期貨 1->選擇權)   海選：   1. 帳號 2. 成交日期 3. 交易所代碼 4. 交易所中文名稱 5. 商品代碼含C/P (以空白間隔C、P)   EX: C C //玉米買權Call  EX: C P //玉米賣權Put   1. 商品年月 2. 商品中文名稱   EX:玉米 202012 370 P   1. 買賣別 2. 成交口數 3. 未平倉口數 4. 成交價 5. 現價 6. 未平損益 7. 委託書序號(前2碼櫃號) 8. 當沖註記 9. 原始保證金 10. 是否為選擇權(0->期貨 1->選擇權) 11. 履約價 |
| **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\_OFOrder.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) |
| # 海外期貨未平倉彙總資料。透過呼叫GetOverseaFutureOpenInterestGW 後，資訊由該事件回傳。      def OnOFOpenInterestGWReport(self, bstrData):          msg = "【OnOFOpenInterestGWReport】" + bstrData          richTextBoxMessage.insert('end', msg + "\n")          richTextBoxMessage.see('end') |

### OnOverseaFutureOpenInterest

|  |  |  |
| --- | --- | --- |
| 海外期貨未平倉。透過呼叫[GetOverSeaFutureOpenInterest](#_GetOverseaFutureOpenInterest) 後，資訊由該事件回傳。 | | |
| **宣告** | void OnOverseaFutureOpenInterest( [in] BSTR bstrData ); | |
| **參數** | bstrData | 第一筆回傳內容為 「@@」開頭，表示查詢狀態，每一筆資料以「,」分隔每一個欄位，欄位依序為「@@,訊息代碼,訊息內容」  其中，當訊息代碼為  0：表示查詢成功，訊息內容為空字串  其他非「0」皆表示查詢錯誤，訊息內容將描述錯誤原因  如有未平倉資訊將從第二筆開始回傳，每一筆資料以「,」分隔每一個欄位，欄位依序為：   1. 海期交易所代碼 2. 海期交易所中文名稱 3. 帳號 4. 海期商品代碼＋年月，代碼與年月間以空白區隔 (年月是一起，未另外以空白區隔) 5. 海期選商品中文名稱   海選商品中文名稱EX:  迷你S&PPUT(空格)202203(空格)3940(空格) P   1. 買賣別 B:買進 S：賣出 2. 數量 3. 市價 4. 平均成交價 5. 昨日結算價 6. 損益 |
| **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。 | |

## OnOverSeaFutureRight

|  |  |  |
| --- | --- | --- |
| 海外期貨權益數。透過呼叫 [GetRequestOverSeaFutureRight](#_GetRequestOverSeaFutureRight)後，資訊由該事件回傳。 | | |
| **宣告** | void OnOverSeaFutureRight ([in] BSTR bstrData); | |
| **參數** | bstrData | 回傳的字串內容，第一筆為查詢結果，成功時以「,」分隔，為「000」開頭與資料筆數；失敗時以「;」分隔，為錯誤代碼與錯誤訊息。  成功時第二筆資料開始就是訊息內容，各筆資料以「,」分隔每一個欄位，欄位依序為：   |  |  | | --- | --- | | **參數** | **說明** | | 期貨商代碼 |  | | IB代號 |  | | 帳號 |  | | 幣別 |  | | 風險指標/維持率 | (風險權益+未沖銷買方選擇權風險市值-未沖銷賣方選擇權風險市值)／(未沖銷部位所需風險原始保證金+未沖銷買方選擇權風險市值-未沖銷賣方選擇權風險市值+加收保證金) | | 足額維持率 |  | | 可動用(出金)保證金 | 帳戶之可委託下單或可出金金額 | | 委託保證金 |  | | 維持保證金 |  | | 原始保證金 |  | | 淨值權益數、權益總值 | 帳戶之清算值，含未沖銷選擇權之市值 | | 預扣權利金 |  | | 未沖銷買方選擇權市值 |  | | 未沖銷賣方選擇權市值 |  | | 帳戶權益、權益數 | 帳戶之淨值，含期貨部位損益及有價證券抵繳總額 | | 盤中未實現 |  | | 未沖銷期貨浮動損益 |  | | 今日權利金支付 |  | | 本日期貨平倉損益淨額 |  | | 已實現費用、手續費 | 交易、交割、履約等手續費 | | 今日存提款 | 入金、出金及手續費調整金額合計 | | 前日帳戶餘額 |  | | 參考匯率 | 依據世華前一日外幣收盤買進匯率做為各幣別當日之匯率，每日約上午10點左右更新 | | 選擇權到期差益 |  | | 選擇權到期差損 |  | | 期貨交割損益 |  | | 超額/追繳保證金 |  | | 到期履約損益 | (期貨交割損益-選擇權到期差損)+選擇權到期差益 | | 帳戶餘額 |  | | 固定保證金 |  | | 變動保證金 |  | |
| **備註** | 當資料已全部傳送完畢時，將回傳一筆「##」表示查詢結束 | |

|  |
| --- |
| Python範例程式碼\_OFOrder.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) |
| # 海外期貨權益數。透過呼叫 GetRequestOverSeaFutureRight後，資訊由該事件回傳。      def OnOverSeaFutureRight(self, bstrData):          msg = "【OnOverSeaFutureRight】" + bstrData;          richTextBoxMessage.insert('end', msg + "\n")          richTextBoxMessage.see('end') |

## 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\_OFOrder.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\_OFOrder.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

## OVERSEAFUTUREORDER(海外期權下單物件)

### 海期委託

struct OVERSEAFUTUREORDER

{

BSTR bstrFullAccount; //海期帳號，分公司代碼＋帳號7碼

BSTR bstrExchangeNo; //交易所代碼。

BSTR bstrStockNo; //海外期權代號。(ex: ES)

BSTR bstrYearMonth; //近月商品年月( YYYYMM) 6碼

BSTR bstrYearMonth2; //遠月商品年月( YYYYMM) 6碼 {價差下單使用}

BSTR bstrOrder; //委託價。

BSTR bstrOrderNumerator; //委託價分子。

BSTR bstrTrigger; //觸發價。

BSTR bstrTriggerNumerator; //觸發價分子。

SHORT sBuySell; //0:買進 1:賣出

　　　　　　　　　　　{價差商品，需留意是否為特殊商品－近遠月前的「+、-」符號}

SHORT sNewClose; //新平倉，0:新倉 {目前海期僅新倉可選}

SHORT sDayTrade; //當沖0:否 1:是；{海期價差單不提供當沖}  
　　　　　　　　　　　　　　 //可當沖商品請參考交易所規定。

SHORT sTradeType; //0:ROD 1:FOK 2:IOC

//{限價單LMT可選ROD/IOC/FOK，市價單依交易所實際提供為主，其餘單別則固定ROD}

SHORT sSpecialTradeType;

//0:LMT 限價單 1:MKT市價單 2:STL停損限價 3.STP 停損市價

LONG nQty; //交易口數。

};

### 海選委託

struct OVERSEAFUTUREORDER

{

BSTR bstrFullAccount; //海期帳號，分公司代碼＋帳號7碼

BSTR bstrExchangeNo; //交易所代碼。

BSTR bstrStockNo; //海外期權代號。

BSTR bstrYearMonth; //近月商品年月( YYYYMM) 6碼

BSTR bstrYearMonth2; //遠月商品年月( YYYYMM) 6碼 {價差下單使用}

BSTR bstrOrder; //委託價。

BSTR bstrOrderNumerator; //委託價分子。

BSTR bstrTrigger; //觸發價。

BSTR bstrTriggerNumerator; //觸發價分子。

SHORT sBuySell; //0:買進 1:賣出{價差商品，需留意是否為特殊商品－近遠月前的「+、-」符號}

SHORT sNewClose; //新平倉，0:新倉 1:平倉 {目前海選可使用新、平倉}

SHORT sDayTrade; //當沖0:否 1:是  
　　　　　　　　　　　　　　 //可當沖商品請參考交易所規定。

SHORT sTradeType; //0:ROD {目前海選均固定ROD}

SHORT sSpecialTradeType; //0:LMT 限價單 1:MKT 2:STL 3.STP

BSTR bstrStrikePrice; //履約價。{選擇權使用}

SHORT sCallPut; //0:CALL 1:PUT {選擇權使用}

LONG nQty; //交易口數。

};

### Proxy Server海期

#### **Proxy Server海期下單**

struct OVERSEAFUTUREORDER

{

BSTR bstrFullAccount; //海期帳號，分公司代碼＋帳號7碼

BSTR bstrExchangeNo; //交易所代碼。

BSTR bstrStockNo; //海外期權代號。

BSTR bstrYearMonth; //近月商品年月( YYYYMM) 6碼

BSTR bstrOrder; //委託價。

BSTR bstrOrderNumerator; //委託價分子。

BSTR bstrTrigger; //觸發價。

BSTR bstrTriggerNumerator; //觸發價分子。

SHORT sBuySell; //0:買進 1:賣出

　　　　　　　　　　　{價差商品，需留意是否為特殊商品－近遠月前的「+、-」符號}

SHORT sNewClose; //新平倉，0:新倉 {目前海期僅新倉可選}

SHORT sDayTrade; //當沖0:否 1:是；{海期價差單不提供當沖}  
　　　　　　　　　　　　　　 //可當沖商品請參考交易所規定。

SHORT sTradeType; //0:ROD 1:IOC 2:FOK

//{限價單LMT可選ROD/IOC/FOK，其餘單別固定ROD}

SHORT sSpecialTradeType; //0:LMT 限價單 1:MKT 2:STL 3.STP

LONG nQty; //交易口數。

};

#### **Proxy Server海期價差下單**

struct OVERSEAFUTUREORDER

{

BSTR bstrFullAccount; //海期帳號，分公司代碼＋帳號7碼

BSTR bstrExchangeNo; //交易所代碼。

BSTR bstrStockNo; //海外期權代號。

BSTR bstrYearMonth; //近月商品年月( YYYYMM) 6碼

BSTR bstrYearMonth2; //遠月商品年月( YYYYMM) 6碼 {價差下單使用}

BSTR bstrOrder; //委託價。

BSTR bstrOrderNumerator; //委託價分子。

BSTR bstrTrigger; //觸發價。

BSTR bstrTriggerNumerator; //觸發價分子。

SHORT sBuySell; //0:買進 1:賣出

　　　　　　　　　　　{價差商品，需留意是否為特殊商品－近遠月前的「+、-」符號}

SHORT sNewClose; //新平倉，0:新倉 {目前海期僅新倉可選}

SHORT sDayTrade; //當沖0:否 1:是；{海期價差單不提供當沖}  
　　　　　　　　　　　　　　 //可當沖商品請參考交易所規定。

SHORT sTradeType; //0:ROD 1:IOC 2:FOK

//{限價單LMT可選ROD/IOC/FOK，其餘單別固定ROD}

SHORT sSpecialTradeType; //0:LMT 限價單 1:MKT 2:STL 3.STP

LONG nQty; //交易口數。

};

#### **Proxy Server海選下單**

struct OVERSEAFUTUREORDER

{

BSTR bstrFullAccount; //海期帳號，分公司代碼＋帳號7碼

BSTR bstrExchangeNo; //交易所代碼。

BSTR bstrStockNo; //海外期權代號。

BSTR bstrYearMonth; //近月商品年月( YYYYMM) 6碼

BSTR bstrOrder; //委託價。

BSTR bstrOrderNumerator; //委託價分子。

BSTR bstrOrderDenominator; //委託價分母。

BSTR bstrTrigger; //觸發價。

BSTR bstrTriggerNumerator; //觸發價分子。

SHORT sBuySell; //0:買進 1:賣出{價差商品，需留意是否為特殊商品－近遠月前的「+、-」符號}

SHORT sNewClose; //新平倉，0:新倉 1:平倉 {目前海選可使用新、平倉}

SHORT sDayTrade; //當沖0:否 1:是  
　　　　　　　　　　　　　　 //可當沖商品請參考交易所規定。

SHORT sTradeType; //0:ROD {目前海選均固定ROD}

SHORT sSpecialTradeType; //0:LMT 限價單 1:MKT 2:STL 3.STP

BSTR bstrStrikePrice; //履約價。{選擇權使用}

SHORT sCallPut; //0:CALL 1:PUT {選擇權使用}

LONG nQty; //交易口數。

};

#### **Proxy Server海期選刪改單**

struct OVERSEAFUTUREORDER

{

BSTR bstrFullAccount; //海期帳號，分公司代碼＋帳號7碼

BSTR bstrExchangeNo; //交易所代碼。

BSTR bstrStockNo; //海外期權代號。

BSTR bstrYearMonth; //近月商品年月( YYYYMM) 6碼

BSTR bstrYearMonth2; //遠月商品年月( YYYYMM) 6碼 {價差刪改單使用}

BSTR bstrOrder; //委託價。

BSTR bstrOrderNumerator; //委託價分子。

BSTR bstrOrderDenominator; //委託價分母。

SHORT sNewClose; //新平倉，0:新倉 1:平倉 {目前海選可使用新、平倉}

SHORT sTradeType; //0:ROD {目前海選均固定ROD}

SHORT sSpecialTradeType; //0:LMT 限價單 1:MKT 2:STL 3.STP

LONG nQty; //交易口數。

BSTR bstrBookNo; //書號。

BSTR bstrSeqNo; //原始13碼序號。

LONG nSpreadFlag; // 0 :OF海期 ; 1: OF-spread 海期價差; 2: OO 海選

LONG nAlterType; //0: Cancel 刪單;1: Decrease 減量; 2: Correct 改價

BSTR bstrStrikePrice; //選擇權履約價 {海選使用}

SHORT sCallPut; //0:CALL ; 1:PUT {海選使用}

};

## OVERSEAFUTUREORDERFORGW(海期委託GW格式物件)

struct OVERSEAFUTUREORDERFORGW

{

BSTR bstrFullAccount; //海期帳號，分公司四碼＋帳號7碼{改價使用}

BSTR bstrExchangeNo; //交易所代碼。{改價使用}

BSTR bstrStockNo; //期權代號。{改價使用}

BSTR bstrStockNo2; //期價差商品代號{價差改價使用}

BSTR bstrYearMonth; //近月商品年月( YYYYMM) 6碼{改價使用}

BSTR bstrYearMonth2; //遠月商品年月( YYYYMM) 6碼 {價差改價使用}

BSTR bstrOrderPrice; //新委託價。{改價使用}

BSTR bstrOrderNumerator; //新委託價分子。{改價使用}

BSTR bstrOrderDenominator; //新委託價分母。{改價使用}

BSTR bstrTriggerPrice; //觸發價。{目前未使用可忽略}

BSTR bstrTriggerNumerator; //觸發價分子。{目前未使用可忽略}

BSTR bstrTriggerDenominator; //觸發價分母。{目前未使用可忽略}

BSTR bstrStrikePrice; //履約價。{選擇權改價使用}

BSTR bstrBookNo; //委託書號{改價使用}

BSTR bstrSeqNo; //委託序號{改價使用}

LONG nBuySell; //0:買進 1:賣出－第一腳{改價可忽略}

LONG nBuySell2; //0:買進 1:賣出－第二腳{改價可忽略}

LONG nNewClose; //新平倉，0:新倉 1:平倉 2:自動

{目前海期僅有新倉，海選有新倉或平倉}

LONG nDayTrade; //當沖0:否 1:是，可當沖商品請參考交易所規定。

{改價可忽略}

LONG nTradeType; //0:ROD 3:IOC 4:FOK

{{改價使用}目前海期、海選均固定ROD}

LONG nSpecialTradeType; // 0:LMT {{改價使用}目前僅提供限價單改限價}

LONG nCallPut; //0:CALL 1:PUT {選擇權改價使用}

LONG nQty; //交易口數。{改價可忽略}

LONG nQty2; //交易口數，複式單使用欄位{改價可忽略}

};