策略王COM元件使用說明

-下單-國內期選-

# 函式範例

## SKOrderLib(國內期選)

*函式*

1. **查詢**

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| **(新)期貨未平倉查詢-GW** | [GetOpenInterest](#_GetOpenInterestGW)GW |  |
| **查詢國內權益數** | [GetFutureRights](#_GetFutureRights) |  |

1. **期貨商品互抵**

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| **(大小台/電/金)指定期貨商品互抵**  **大小台互抵** | |  | | --- | | [SendTFOffset](#_SendTFOffSet) | | [SendTXOffset](#_SendTXOffset) | |  |

1. **(選擇權)組合/複式單拆解/雙邊了結**

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| **組合部位** | [AssembleOptions](#_AssembleOptions) | 非交易行為，無回報 |
| **複式單拆解** | [DisassembleOptions](#_DisassembleOptions) | 非交易行為，無回報 |
| **雙邊部位了結** | [CoverAllProduct](#_CoverAllProduct) | 非交易行為，無回報 |

1. **一般下單/刪改單**

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| **期貨委託含倉位盤別** | [SendFutureOrderCLR](#_SendFutureOrderCLR) | (新增)可選盤別、倉別 |
| **選擇權委託** | [SendOptionOrder](#_SendOptionOrder) |  |
| **國內選擇權複式單** | [SendDuplexOrder](#_SendDuplexOrder) |  |
| 1. **證期權改價** | [CorrectPriceBySeqNo](#_CorrectPriceBySeqNo) | 依序號改價 |
| 1. **證期權改價** | [CorrectPriceByBookNo](#_CorrectPriceByBookNo_1) | 依書號改價 |
| 1. **委託減量** | [DecreaseOrderBySeqNo](#_DecreaseOrderBySeqNo) |  |
| 1. **國內刪單-序號** | [CancelOrderBySeqNo](#_CancelOrderBySeqNo_1) | 13碼序號 |
| 1. **國內刪單-書號** | [CancelOrderByBookNo](#_CancelOrderByBookNo_1) | 依5碼書號刪單 |
| 1. **國內刪單-商品代號** | [CancelOrderByStockNo](#_CancelOrderByStockNo_1) | 依帳號所屬**登入ID**+商品代號 |

1. **Proxy下單/刪改單**

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| **期貨委託含倉位盤別** | [SendFutureProxyOrderCLR](#_SendFutureProxyOrderCLR) | 須選倉別、盤別 |
| **選擇權委託** | [SendOptionProxyOrder](#_SendOptionProxyOrder) |  |
| **選擇權複式下單** | [SendDuplexProxyOrder](#_SendDuplexProxyOrder) |  |
| **期貨刪改單** | [SendFutureProxyAlter](#_SendFutureProxyAlter) |  |
| **選擇權刪改單** | [SendOptionProxyAlter](#_SendOptionProxyAlter) |  |

*事件*

|  |  |  |
| --- | --- | --- |
| **功能** | **事件名稱** | **備註** |
| **期貨未平倉** | [OnOpenInterest](#_4-2-b_OnAsyncOrder) |  |
| **國內權益數** | [OnFutureRights](#_OnFutureRights) |  |
| **非同步委託結果** | [OnAsyncOrder](#_OnAsyncOrder) |  |
| **Proxy委託結果** | [OnProxyOrder](#_OnProxyOrder) |  |

#### **查詢**

##### 期貨未平倉

###### **GetOpenInterestGW**

|  |  |  |
| --- | --- | --- |
| 查詢期貨未平倉－可指定回傳格式。 | | |
| **宣告** | Long GetOpenInterestGW([in] BSTR bstrLogInID, [in] BSTR bstrAccount,[in] LONG nFormat); | |
| **參數** | bstrLogInID | 登入ID。 |
| bstrAccount | 委託帳號 ( IB＋帳號) 。 |
| nFormat | 回傳格式：1。 |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。 | |
| **備註** | 結果由[OnOpenInterest](#_(新)期貨未平倉GW。透過呼叫_GetOpenInterestGW_後)事件回傳。  新增參數：指定回傳格式（內容參考[OnOpenInterest](#_4-2-b_OnAsyncOrder)） | |

|  |
| --- |
| C#範例程式碼\_TFReadOrderForm.cs |
| UI示意圖 |
| |  |  | | --- | --- | |  | 拉出 1個Button，命名(Name)為「buttonGetOpenInterestGW」。 | |
| 全域變數 |
| // 宣告物件  SKCenterLib m\_pSKCenter = new SKCenterLib(); //登入&環境設定物件  SKOrderLib m\_pSKOrder = new SKOrderLib(); //下單物件 |
| Current Block (可放在buttonGetOpenInterestGW\_Click的Click事件裡) |
| //Method  string bstrLogInID = comboBoxUserID.Text; // 登入ID  string bstrAccount = comboBoxAccount.Text; // 委託帳號 (IB＋帳號)  int nFormat = 1; // 回傳格式：1  dataGridViewOnOpenInterest.Rows.Clear(); //清空  // 查詢期貨未平倉－可指定回傳格式  int nCode = m\_pSKOrder.GetOpenInterestGW(bstrLogInID, bstrAccount, nFormat);  // 取得回傳訊息  string msg = "【GetOpenInterestGW】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode);  richTextBoxMethodMessage.AppendText(msg + "\n"); |
| 執行成功畫面 |
|  |

##### GetFutureRights

|  |  |  |
| --- | --- | --- |
| 查詢國內權益數。 | | |
| **宣告** | Long GetFutureRights([in] BSTR bstrLogInID, [in] BSTR bstrAccount,[in]SHORT sCoinType) | |
| **參數** | bstrLogInID | 登入ID。 |
| bstrAccount | 委託帳號 ( IB＋帳號) 。 |
| sCoinType | 0:全幣別，1:基幣(台幣TWD)，2:人民幣RMB。 |
|  |  |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。 | |
| **備註** | * 結果由[OnFutureRights](#_OnFutureRights)事件回傳。 * 全幣別選項含基幣。 | |

|  |
| --- |
| C#範例程式碼\_TFReadOrderForm.cs |
| UI示意圖 |
| |  |  | | --- | --- | |  | 拉出 1個Button，命名(Name)為「buttonGetFutureRights」。  1個ComboBox，命名(Name)為「comboBoxCoinType」。 | |
| 全域變數 |
| // 宣告物件  SKCenterLib m\_pSKCenter = new SKCenterLib(); //登入&環境設定物件  SKOrderLib m\_pSKOrder = new SKOrderLib(); //下單物件 |
| Constructor |
| // comboBoxCoinType  {  comboBoxCoinType.Items.Add("0:全幣別");  comboBoxCoinType.Items.Add("1:基幣(台幣TWD)");  comboBoxCoinType.Items.Add("2:人民幣RMB");  } |
| Current Block (可放在buttonGetFutureRights\_Click的Click事件裡) |
| //Method  dataGridViewOnFutureRights.Rows.Clear(); // 每次查詢前，清除上一次查詢結果  short sCoinType = 0; //0:全幣別，1:基幣(台幣TWD)，2:人民幣RMB  string bstrAccount = comboBoxAccount.Text;  string selectedValue = comboBoxCoinType.SelectedItem.ToString(); //0:全幣別，1:基幣(台幣TWD)，2:人民幣RMB  if (selectedValue == "0:全幣別") sCoinType = 0;  else if (selectedValue == "1:基幣(台幣TWD)") sCoinType = 1;  else if (selectedValue == "2:人民幣RMB") sCoinType = 2;  // 查詢國內權益數  int nCode = m\_pSKOrder.GetFutureRights(comboBoxUserID.Text, bstrAccount, sCoinType);  // 取得回傳訊息  string msg = "【GetFutureRights】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode);  richTextBoxMethodMessage.AppendText(msg + "\n"); |
| 執行成功畫面 |
|  |

#### **期貨商品互抵**

##### SendTFOffSet

|  |  |  |
| --- | --- | --- |
| 指定大小台/電/金互抵。 | | |
| **宣告** | Long SendTFOffset([in] BSTR bstrLogInID, [in] VARIANT\_BOOL bAsyncOrder, [in] BSTR bstrAccount, [in] LONG nCommodity, [in] BSTR bstrYearMonth, [in] LONG nBuySell, [in] LONG nQty, [**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) 。 |
| nCommodity | 指定商品0:大小台; 1:大小電; 2:大小金 |
| bstrYearMonth | 大小台(電、金)年月(YYYYMM)。 |
| nBuySell | 大台(電、金)買賣別。0:多方(買) 1:空方(賣) |
| nQty | 互抵口數，以大台(電、金)口數為基本單位。 |
| bstrMessage | **同步：**如果回傳值為 0表示委託成功，訊息內容則為互抵訊息。回傳值非0表示互抵失敗，訊息內容為失敗原因。  **非同步：**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，詳細委託內容請致電交易室查詢，  回傳值非0時表示失敗，請參考訊息確認錯誤原因。  使用非同步委託，委託結果請由[OnAsyncOrder](#_OnAsyncOrder)取得。  EX:互抵需要1口大台抵4口小台(金)或一口大電抵8口小電，且大小台(電、金)需互為反向部位、相同月份，才可以互抵成功。 | |

|  |
| --- |
| C#範例程式碼\_MainForm.cs |
| UI示意圖 |
| |  |  | | --- | --- | |  | 拉出 2個ComboBox，命名(Name)為「comboBoxSendTFOffSetnCommodity」、「comboBoxSendTFOffSetnBuySell」。  2個TextBox，命名(Name)為「textBoxSendTFOffSetbstrYearMonth」、「textBoxSendTFOffSetnQty」。  1個Button，命名(Name)為「buttonSendTFOffSet」。 | |
| 全域變數 |
| // 宣告物件  SKCenterLib m\_pSKCenter = new SKCenterLib(); //登入&環境設定物件  SKOrderLib m\_pSKOrder = new SKOrderLib(); //下單物件  // 是否為非同步委託  bool bAsyncOrder = false; |
| Constructor |
| // 大小台電金互抵  {  //comboBoxSendTFOffSetnCommodity  {  comboBoxSendTFOffSetnCommodity.Items.Add("大小台");  comboBoxSendTFOffSetnCommodity.Items.Add("大小電");  comboBoxSendTFOffSetnCommodity.Items.Add("大小金");  }  //comboBoxSendTFOffSetnBuySell  {  comboBoxSendTFOffSetnBuySell.Items.Add("多方(買)");  comboBoxSendTFOffSetnBuySell.Items.Add("空方(賣)");  }  } |

|  |
| --- |
| Current Block (可放在buttonSendTFOffSet\_Click的Click事件裡) |
| //Method  string bstrLogInID = comboBoxUserID.Text; // 登入ID  string bstrAccount = comboBoxAccount.Text; // 委託帳號 ( IB＋帳號)  int nCommodity = 0; // 指定商品0:大小台; 1:大小電; 2:大小金  string selectedValue = comboBoxSendTFOffSetnCommodity.Text;  if (selectedValue == "大小台") nCommodity = 0;  else if (selectedValue == "大小電") nCommodity = 1;  else if (selectedValue == "大小金") nCommodity = 2;  string bstrYearMonth = textBoxSendTFOffSetbstrYearMonth.Text; // 大小台(電、金)年月(YYYYMM)  int nBuySell = 0; // 大台(電、金)買賣別。0:多方(買) 1:空方(賣)  selectedValue = comboBoxSendTFOffSetnBuySell.Text;  if (selectedValue == "多方(買)") nBuySell = 0;  else if (selectedValue == "空方(賣)") nBuySell = 1;  int nQty = int.Parse(textBoxSendTFOffSetnQty.Text); // 互抵口數，以大台(電、金)口數為基本單位  string bstrMessage = ""; // 同步：如果回傳值為 0表示委託成功，訊息內容則為互抵訊息。回傳值非0表示互抵失敗，訊息內容為失敗原因。非同步：參照4 - 2 - b OnAsyncOrder。  // 指定大小台/電/金互抵  int nCode = m\_pSKOrder.SendTFOffset(bstrLogInID, bAsyncOrder, bstrAccount, nCommodity, bstrYearMonth, nBuySell, nQty, out bstrMessage);  // 取得回傳訊息  string msg = "【SendTFOffSet】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode);  richTextBoxMethodMessage.AppendText(msg + "\n");  // 取得bstrMessage  msg = "【SendTFOffSet】" + bstrMessage;  richTextBoxMessage.AppendText(msg + "\n"); |
| 執行成功畫面 |
|  |

###### SendTXOffset

|  |  |  |
| --- | --- | --- |
| 大小台互抵。 | | |
| **宣告** | Long SendTXOffset([in] BSTR bstrLogInID, [in] VARIANT\_BOOL bAsyncOrder, [in] BSTR bstrAccount, [in] BSTR bstrYearMonth, [in] LONG nBuySell, [in] LONG nQty, [**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) 。 |
| bstrYearMonth | 大小台年月(YYYYMM)。 |
| nBuySell | 大台買賣別。0:多方(買) 1:空方(賣) |
| nQty | 互抵口數，以大台口數為基本單位。 |
| bstrMessage | **同步：**如果回傳值為 0表示委託成功，訊息內容則為互抵訊息。回傳值非0表示互抵失敗，訊息內容為失敗原因。  **非同步：**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，詳細委託內容請致電交易室查詢，  回傳值非0時表示失敗，請參考訊息確認錯誤原因。  使用非同步委託，委託結果請由[OnAsyncOrder](#_OnAsyncOrder)取得。  大小台互抵需要1口大台抵4口小台，且大小台需互為反向部位、相同月份，才可以互抵成功。 | |

#### **(選擇權)組合/複式單拆解/雙邊了結**

##### AssembleOptions

|  |  |  |
| --- | --- | --- |
| 國內選擇權組合部位。 | | |
| **宣告** | Long AssebleOptions([in] BSTR bstrLogInID, [in] VARIANT\_BOOL bAsyncOrder, [in] struct **FUTUREORDER**\* pOrder,[**out**] BSTR\* bstrMessage); | |
| **參數** | bstrLogInID | 登入ID。 |
| bAsyncOrder | 可忽略，限同步委託。 |
| pOrder | SKCOM元件中的 [**FUTUREORDER**](#_組合部位、複式單拆解、雙邊部位了結▼)物件，將組合條件填入該物件後，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，表示委託已送出。  回傳值非0表示委託失敗，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** | ◆委託欄位，請參考期權下單物件說明，請留意需有可組合選擇權部位。  ★非交易的行為，無回報，可由未平倉查詢確認部位。  因組合/拆解/了結功能是直送結算所，需等處理結果回傳後，方可再進行下一筆。 | |

|  |
| --- |
| C#範例程式碼\_MainForm.cs |
| UI示意圖 |
| |  |  | | --- | --- | |  | 拉出 3個ComboBox，命名(Name)為「comboBoxAssembleOptions」、「comboBoxsBuySell」、「comboBoxsBuySell2」。  3個TextBox，命名(Name)為「textBoxbstrStockNo」、「textBoxbstrStockNo2」、「textBoxnQty」。  1個Button，命名(Name)為「buttonAssembleOptions」。 | |
| 全域變數 |
| // 宣告物件  SKCenterLib m\_pSKCenter = new SKCenterLib(); //登入&環境設定物件  SKOrderLib m\_pSKOrder = new SKOrderLib(); //下單物件 |
| Constructor |
| // 選擇權部位  {  // comboBoxAssembleOptions  {  comboBoxAssembleOptions.Items.Add("組合");  comboBoxAssembleOptions.Items.Add("複式單拆解");  comboBoxAssembleOptions.Items.Add("雙邊了結");  }  //comboBoxsBuySell  {  comboBoxsBuySell.Items.Add("買進");  comboBoxsBuySell.Items.Add("賣出");  }  //comboBoxsBuySell2  {  comboBoxsBuySell2.Items.Add("買進");  comboBoxsBuySell2.Items.Add("賣出");  }  } |

|  |
| --- |
| Current Block (可放在buttonAssembleOptions\_Click的Click事件裡) |
| //Method  string bstrLogInID = comboBoxUserID.Text; // 登入ID  bool bAsyncOrder = true; // 可忽略，限同步委託  FUTUREORDER pOrder = new FUTUREORDER(); //SKCOM元件中的 FUTUREORDER物件，將組合條件填入該物件後，再帶入此欄位中  pOrder.bstrFullAccount = comboBoxAccount.Text;//期貨帳號，分公司代碼＋帳號7碼  pOrder.bstrStockNo = textBoxbstrStockNo.Text;//委託選擇權代號1{組合部位}  pOrder.bstrStockNo2 = textBoxbstrStockNo2.Text;//委託選擇權代號2{組合部位}  pOrder.sBuySell = 0;//0:買進 1:賣出{組合部位}  string selectedValue = comboBoxsBuySell.Text;  if (selectedValue == "買進") pOrder.sBuySell = 0;  else if (selectedValue == "賣出") pOrder.sBuySell = 1;  pOrder.sBuySell2 = 0;//0:買進 1:賣出{組合部位}  selectedValue = comboBoxsBuySell2.Text;  if (selectedValue == "買進") pOrder.sBuySell2 = 0;  else if (selectedValue == "賣出") pOrder.sBuySell2 = 1;  pOrder.nQty = int.Parse(textBoxnQty.Text);//交易口數{組合部位}  string bstrMessage = ""; // 同步委託：如果回傳值為 0表示委託成功，表示委託已送出。回傳值非0表示委託失敗，訊息內容為失敗原因。  if (comboBoxAssembleOptions.Text == "組合")  {  // 國內選擇權組合部位  int nCode = m\_pSKOrder.AssembleOptions(bstrLogInID, bAsyncOrder, ref pOrder, out bstrMessage);  string msg = "【AssembleOptions】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode);  richTextBoxMethodMessage.AppendText(msg + "\n");  // 取得bstrMessage  msg = "【AssembleOptions】" + bstrMessage;  richTextBoxMessage.AppendText(msg + "\n");  }  else if (comboBoxAssembleOptions.Text == "複式單拆解")  {  // 國內選擇權複式單拆解  int nCode = m\_pSKOrder.DisassembleOptions(bstrLogInID, bAsyncOrder, ref pOrder, out bstrMessage);  // 取得回傳訊息  string msg = "【DisassembleOptions】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode);  richTextBoxMethodMessage.AppendText(msg + "\n");  // 取得bstrMessage  msg = "【DisassembleOptions】" + bstrMessage;  richTextBoxMessage.AppendText(msg + "\n");  }  else if (comboBoxAssembleOptions.Text == "雙邊了結")  {  // 國內選擇權雙邊部位了結  int nCode = m\_pSKOrder.CoverAllProduct(bstrLogInID, bAsyncOrder, ref pOrder, out bstrMessage);  // 取得回傳訊息  string msg = "【CoverAllProduct】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode);  richTextBoxMethodMessage.AppendText(msg + "\n");  // 取得bstrMessage  msg = "【CoverAllProduct】" + bstrMessage;  richTextBoxMessage.AppendText(msg + "\n");  } |
| 執行成功畫面 |
| |  |  | | --- | --- | | 組合 |  | | 拆解 |  | | 了結 |  | |

##### DisassembleOptions

|  |  |  |
| --- | --- | --- |
| 國內選擇權複式單拆解。 | | |
| **宣告** | Long DisassebleOptions([in] BSTR bstrLogInID, [in] VARIANT\_BOOL bAsyncOrder, [in] struct **FUTUREORDER**\* pOrder,[**out**] BSTR\* bstrMessage); | |
| **參數** | bstrLogInID | 登入ID。 |
| bAsyncOrder | 可忽略，限同步委託。 |
| pOrder | SKCOM元件中的 [**FUTUREORDER**](#_組合部位、複式單拆解、雙邊部位了結▼)物件，將拆解條件填入該物件後，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，表示委託已送出。  回傳值非0表示委託失敗，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「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** | ◆委託欄位，請參考期權下單物件說明，請留意需有複式單部位。  ★非交易的行為，無回報，可由未平倉查詢確認部位。  因組合/拆解/了結功能是直送結算所，需等處理結果回傳後，方可再進行下一筆。 | |

##### CoverAllProduct

|  |  |  |
| --- | --- | --- |
| 國內選擇權雙邊部位了結。 | | |
| **宣告** | Long CoverAllProduct([in] BSTR bstrLogInID, [in] VARIANT\_BOOL bAsyncOrder, [in] struct **FUTUREORDER**\* pOrder,[**out**] BSTR\* bstrMessage); | |
| **參數** | bstrLogInID | 登入ID。 |
| bAsyncOrder | 可忽略，限同步委託。 |
| pOrder | SKCOM元件中的 [**FUTUREORDER**](#_組合部位、複式單拆解、雙邊部位了結▼)物件，將了結條件填入該物件後，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，表示委託已送出。  回傳值非0表示委託失敗，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「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** | ◆委託欄位，請參考期權下單物件說明，請留意需具有雙邊部位。  ★非交易的行為，無回報，可由未平倉查詢確認部位。  因組合/拆解/了結功能是直送結算所，需等處理結果回傳後，方可再進行下一筆。 | |

#### **一般下單/刪改單**

##### SendFutureOrderCLR

|  |  |  |
| --- | --- | --- |
| 送出期貨委託，需設倉別與盤別。 | | |
| **宣告** | Long SendFutureOrderCLR([in] BSTR bstrLogInID, [in] VARIANT\_BOOL bAsyncOrder, [in] struct **FUTUREORDER**\* pAsyncOrder,[**out**] BSTR\* bstrMessage); | |
| **參數** | bstrLogInID | 登入ID。 |
| bAsyncOrder | 是否為非同步委託。 |
| pOrder | SKCOM元件中的 [**FUTUREORDER**](#_一般期選委託▼)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。**  **其他錯誤則由「交易主機」回傳錯誤代碼及錯誤原因。若仍有疑問請洽詢您所屬營業員。** | |
| **備註** | 使用非同步委託，委託結果請由[OnAsyncOrder](#_OnAsyncOrder)取得。 | |

|  |
| --- |
| C#範例程式碼\_ TFSendOrderForm.cs |
| UI示意圖 |
| |  |  | | --- | --- | |  | 拉出3個TextBox，命名(Name)為「textBoxFutureID」、「textBoxFutureQty」、「textBoxFuturePrice」。  5個ComboBox，命名(Name)為「comboBoxFutureDayTrade」、  「comboBoxFutureReserved」、  「comboBoxFutureNewClose」、「comboBoxFutureTradeType」、「comboBoxFutureBuySell」。  1個Button，命名(Name)為「buttonSendFutureOrderCLR」。 | |
| 全域變數 |
| // 宣告物件  SKCenterLib m\_pSKCenter = new SKCenterLib(); //登入&環境設定物件  SKOrderLib m\_pSKOrder = new SKOrderLib(); //下單物件  // 是否為非同步委託  bool bAsyncOrder = false; |

|  |
| --- |
| Constructor |
| // TF  {  //comboBoxFutureTradeType  comboBoxFutureTradeType.Items.Add("ROD");  comboBoxFutureTradeType.Items.Add("IOC");  comboBoxFutureTradeType.Items.Add("FOK");  //comboBoxFutureBuySell  comboBoxFutureBuySell.Items.Add("買進");  comboBoxFutureBuySell.Items.Add("賣出");  //comboBoxFutureDayTrade  comboBoxFutureDayTrade.Items.Add("否");  comboBoxFutureDayTrade.Items.Add("是");  //comboBoxFutureNewClose  comboBoxFutureNewClose.Items.Add("新倉");  comboBoxFutureNewClose.Items.Add("平倉");  comboBoxFutureNewClose.Items.Add("自動");  //comboBoxFutureReserved  comboBoxFutureReserved.Items.Add("盤中(T盤及T+1盤)");  comboBoxFutureReserved.Items.Add("T盤預約");  } |
| Current Block (可放在buttonSendFutureOrderCLR\_Click的Click事件裡) |
| //Method  FUTUREORDER pOrder = new FUTUREORDER();  string bstrMessage;  pOrder.bstrFullAccount = comboBoxAccount.Text;  pOrder.bstrStockNo = textBoxFutureID.Text;  string selectedValue = comboBoxFutureTradeType.Text;  if (selectedValue == "ROD") pOrder.sTradeType = 0;  else if (selectedValue == "IOC") pOrder.sTradeType = 1;  else if (selectedValue == "FOK") pOrder.sTradeType = 2;  selectedValue = comboBoxFutureBuySell.Text;  if (selectedValue == "買進") pOrder.sBuySell = 0;  else if (selectedValue == "賣出") pOrder.sBuySell = 1;  selectedValue = comboBoxFutureDayTrade.Text;  if (selectedValue == "否") pOrder.sDayTrade = 0;  else if (selectedValue == "是") pOrder.sDayTrade = 1;  selectedValue = comboBoxFutureNewClose.Text;  if (selectedValue == "新倉") pOrder.sNewClose = 0;  else if (selectedValue == "平倉") pOrder.sNewClose = 1;  else if (selectedValue == "自動") pOrder.sNewClose = 2;  pOrder.bstrPrice = textBoxFuturePrice.Text;  pOrder.nQty = int.Parse(textBoxFutureQty.Text);  selectedValue = comboBoxFutureReserved.Text;  if (selectedValue == "盤中(T盤及T+1盤)") pOrder.sReserved = 0;  else if (selectedValue == "T盤預約") pOrder.sReserved = 1;  // 送出期貨委託  int nCode = m\_pSKOrder.SendFutureOrderCLR(comboBoxUserID.Text, bAsyncOrder, ref pOrder, out bstrMessage);  // 取得回傳訊息  string msg = "【SendFutureOrderCLR】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode);  richTextBoxMethodMessage.AppendText(msg + "\n");  // 取得回傳訊息(bstrMessage)  if (bAsyncOrder == false)  {  msg = "【同步委託結果】" + bstrMessage;  richTextBoxMessage.AppendText(msg + "\n");  } |
| 執行成功畫面 |
|  |

##### SendOptionOrder

|  |  |  |
| --- | --- | --- |
| 送出選擇權委託，需設盤別。 | | |
| **宣告** | Long SendOptionOrder([in] BSTR bstrLogInID, [in] VARIANT\_BOOL bAsyncOrder, [in] struct **FUTUREORDER**\* 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 [**FUTUREORDER**](#_一般期選委託▼)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。**  **其他錯誤則由「交易主機」回傳錯誤代碼及錯誤原因。若仍有疑問請洽詢您所屬營業員。** | |
| **備註** | 使用非同步委託，委託結果請由[OnAsyncOrder](#_OnAsyncOrder)取得。 | |

|  |
| --- |
| C#範例程式碼\_TFSendOrderForm.cs |
| UI示意圖 |
| |  |  | | --- | --- | |  | 拉出1個CheckBox，命名(Name)為「checkBoxSendDuplexOrder」。  4個TextBox，命名(Name)為「textBoxOptionID」、「textBoxOptionID2」、「textBoxOptionQty」、「textBoxOptionPrice」。  6個ComboBox，命名(Name)為「comboBoxOptionDayTrade」、  「comboBoxOptionReserved」、  「comboBoxOptionNewClose」、「comboBoxOptionTradeType」、「comboBoxOptionBuySell」、「comboBoxOptionBuySell2」。  1個Button，命名(Name)為「buttonSendOptionOrder」。 | |
| 全域變數 |
| // 宣告物件  SKCenterLib m\_pSKCenter = new SKCenterLib(); //登入&環境設定物件  SKOrderLib m\_pSKOrder = new SKOrderLib(); //下單物件  // 是否為非同步委託  bool bAsyncOrder = false;  // 是否為複式單  bool isDuplexOrder = false; |

|  |
| --- |
| Constructor |
| // TO  {  //comboBoxOptionTradeType  comboBoxOptionTradeType.Items.Add("ROD");  comboBoxOptionTradeType.Items.Add("IOC");  comboBoxOptionTradeType.Items.Add("FOK");  //comboBoxOptionBuySell  comboBoxOptionBuySell.Items.Add("買進");  comboBoxOptionBuySell.Items.Add("賣出");  //comboBoxOptionBuySell2  comboBoxOptionBuySell2.Items.Add("買進");  comboBoxOptionBuySell2.Items.Add("賣出");  //comboBoxOptionDayTrade  comboBoxOptionDayTrade.Items.Add("否");  comboBoxOptionDayTrade.Items.Add("是");  //comboBoxOptionNewClose  comboBoxOptionNewClose.Items.Add("新倉");  comboBoxOptionNewClose.Items.Add("平倉");  comboBoxOptionNewClose.Items.Add("自動");  //comboBoxOptionReserved  comboBoxOptionReserved.Items.Add("盤中(T盤及T+1盤)");  comboBoxOptionReserved.Items.Add("T盤預約");  } |
| 可放在checkBoxSendDuplexOrder\_CheckedChanged的CheckedChanged事件裡 |
| if (checkBoxSendDuplexOrder.Checked) isDuplexOrder = true;  else isDuplexOrder = false; |

|  |
| --- |
| Current Block (可放在buttonSendOptionOrder\_Click的Click事件裡) |
| //Method  FUTUREORDER pOrder = new FUTUREORDER();  string bstrMessage;  pOrder.bstrFullAccount = comboBoxAccount.Text;  pOrder.bstrStockNo = textBoxOptionID.Text;  string selectedValue = comboBoxOptionTradeType.Text;  if (selectedValue == "ROD") pOrder.sTradeType = 0;  else if (selectedValue == "IOC") pOrder.sTradeType = 1;  else if (selectedValue == "FOK") pOrder.sTradeType = 2;  selectedValue = comboBoxOptionBuySell.Text;  if (selectedValue == "買進") pOrder.sBuySell = 0;  else if (selectedValue == "賣出") pOrder.sBuySell = 1;  selectedValue = comboBoxOptionDayTrade.Text;  if (selectedValue == "否") pOrder.sDayTrade = 0;  else if (selectedValue == "是") pOrder.sDayTrade = 1;  selectedValue = comboBoxOptionNewClose.Text;  if (selectedValue == "新倉") pOrder.sNewClose = 0;  else if (selectedValue == "平倉") pOrder.sNewClose = 1;  else if (selectedValue == "自動") pOrder.sNewClose = 2;  pOrder.bstrPrice = textBoxOptionPrice.Text;  pOrder.nQty = int.Parse(textBoxOptionQty.Text);  selectedValue = comboBoxOptionReserved.Text;  if (selectedValue == "盤中(T盤及T+1盤)") pOrder.sReserved = 0;  else if (selectedValue == "T盤預約") pOrder.sReserved = 1;  if (isDuplexOrder != true)  {  // 送出選擇權委託  int nCode = m\_pSKOrder.SendOptionOrder(comboBoxUserID.Text, bAsyncOrder, ref pOrder, out bstrMessage);  // 取得回傳訊息  string msg = "【SendOptionOrder】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode);  richTextBoxMethodMessage.AppendText(msg + "\n");  }  else  {  pOrder.bstrStockNo2 = textBoxOptionID2.Text; //委託選擇權代號2{複式單}  // 0:買進 1:賣出{ 複式單}  if (comboBoxOptionBuySell2.Text == "買進") pOrder.sBuySell2 = 0;  else if (comboBoxOptionBuySell2.Text == "賣出") pOrder.sBuySell2 = 1;  // 送出國內選擇權複式單委託  int nCode = m\_pSKOrder.SendDuplexOrder(comboBoxUserID.Text, bAsyncOrder, ref pOrder, out bstrMessage);  // 取得回傳訊息  string msg = "【SendDuplexOrder】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode);  richTextBoxMethodMessage.AppendText(msg + "\n");  }  // 取得回傳訊息(bstrMessage)  if (bAsyncOrder == false)  {  string msg = "【同步委託結果】" + bstrMessage;  richTextBoxMessage.AppendText(msg + "\n");  } |
| 執行成功畫面 |
| |  |  | | --- | --- | | 選擇權委託 |  | | 選擇權複式單委託 |  | |

###### SendDuplexOrder

|  |  |  |
| --- | --- | --- |
| 送出國內選擇權複式單委託。 | | |
| **宣告** | Long SendDuplexOrder([in] BSTR bstrLogInID, [in] VARIANT\_BOOL bAsyncOrder, [in] struct **FUTUREORDER**\* 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 [**FUTUREORDER**](#_選擇權複式單▼)物件，將下單條件填入該物件後，再帶入此欄位中。 |
| bstrMessage | **同步委託：**如果回傳值為 0表示委託成功，訊息內容則為13碼的委託序號。回傳值非0表示委託失敗，訊息內容為失敗原因。  **非同步委託：**參照OnAsyncOrder。 |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。  **當錯誤代碼為4碼時，可參考文件「4.下單準備介紹」裡面的【錯誤代碼定義表】，所屬委託物件亦有欄位值說明供參。**  **其他錯誤則由「交易主機」回傳錯誤代碼及錯誤原因。若仍有疑問請洽詢您所屬營業員。** | |
| **備註** | 使用非同步委託，委託結果請由[OnAsyncOrder](#_OnAsyncOrder)取得。    **複式單下法** (FUTUREORDER 物件填法，以上圖為例):  Call多頭價差(策略王上溫漲)，買低履約價+賣高履約價  第一隻腳: bstrStockNo 填:高履約價(ex:11000)，sBuySell填:賣  第二隻腳: bstrStockNo2 填:低履約價(ex:10900)，sBuySell2填:買  bstrPrice填法基本上為:bstrStockNo2賣價(99)–bstrStockNo買價(46.5)= 52.5  如果bstrPrice價格填的越大，越能立即成交，假設目前價差52,5，超過52.5那就能幾乎能立即成交(快速直覺:Call作多價格越高買就相當於市價買進，越容易成交)。  Call空頭價差(策略王上溫跌作莊)  第一隻腳: bstrStockNo 填:高履約價(ex:11000)，sBuySell填:買  第二隻腳: bstrStockNo2 填:低履約價(ex:10900)，sBuySell2填:賣  bstrPrice填法基本上為:bstrStockNo2買價(97)–bstrStockNo賣價(47)= 50  如果bstrPrice價格填的越小，越能立即成交，假設目前價差50，小於50那就能幾乎能立即成交(快速直覺:Call作空價格越低賣就相當於賤價賣，越容易成交)。  Put多頭價差(策略王上溫漲作莊)  第一隻腳: bstrStockNo 填:低履約價(ex:10900)，sBuySell填:買  第二隻腳: bstrStockNo2 填:高履約價(ex:11000)，sBuySell2填:賣  bstrPrice填法基本上為:bstrStockNo2買價(109)–bstrStockNo賣價(64)= 45  如果bstrPrice價格填的越小，越能立即成交，假設目前價差45，小於45那就能幾乎能立即成交(快速直覺:Put作多價格越低作莊就相當於賤價賣，越容易成交)。  Put空頭價差(策略王上溫跌)  第一隻腳: bstrStockNo 填:低履約價(ex:10900)，sBuySell填:賣  第二隻腳: bstrStockNo2 填:高履約價(ex:11000)，sBuySell2填:買  bstrPrice填法基本上為:bstrStockNo2賣價(113)–bstrStockNo買價(62)= 51  如果bstrPrice價格填的越大，越能立即成交，假設目前價差51，大於51那就能幾乎能立即成交(快速直覺:Put作多價格越高買就相當於市價買，越容易成交)。  突破  第一隻腳: bstrStockNo填:call/put階可(不同履約價也可)，sBuySell填:買  第二隻腳: bstrStockNo2填: call/put階可(不同履約價也可)，sBuySell2填:買  bstrPrice填法基本上為:bstrStockNo2賣價(99)+bstrStockNo賣價(113)= 212  如果bstrPrice價格填的越大，越能立即成交，假設目前價差212，大於212那就能幾乎能立即成交。  盤整  第一隻腳: bstrStockNo填: call/put階可(不同履約價也可)，sBuySell填:賣  第二隻腳: bstrStockNo2填: call/put階可(不同履約價也可)，sBuySell2填:賣  bstrPrice填法基本上為:bstrStockNo2買價(97)+bstrStockNo買價(109)= 206  如果bstrPrice價格填的越小，越能立即成交，假設目前價差206，小於206那就能幾乎能立即成交。 | |

##### 刪改單

###### CorrectPriceBySeqNo

|  |  |  |
| --- | --- | --- |
| 證期權依序號改價。 | | |
| **宣告** | Long CorrectPriceBySeqNo([in] BSTR bstrLogInID, [in] VARIANT\_BOOL bAsyncOrder, [in] BSTR bstrAccount, [in] BSTR bstrSeqNo, [in] BSTR bstrPrice, [in] LONG nTradeType,[**out**] BSTR\* bstrMessage); | |
| **參數** | bstrLogInID | 登入ID。 |
| bAsyncOrder | 是否為非同步委託。 |
| bstrAccount | 證券：委託帳號 (BrokerID＋帳號)。  期貨：委託帳號 ( IB＋帳號) 。 |
| bstrSeqNo | 欲改量的委託序號。 |
| bstrPrice | 修改價格。 |
| nTradeType | **證券0:ROD**  期選0:ROD 1:IOC 2:FOK |
| 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](#_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**改價**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。  ＊證券逐筆nTradeType:請設0(ROD).  ＊不適用盤中零股 | |

|  |
| --- |
| C#範例程式碼\_ TSTFUpdateOrderForm.cs |
| UI示意圖 |
| |  |  | | --- | --- | | 一張含有 文字, 螢幕擷取畫面, 數字, 字型 的圖片  自動產生的描述 | 拉出2個TextBox，命名(Name)為「textBoxSeqNo」、「textBoxPrice」。 拉出3個ComboBox，命名(Name)為「comboBoxUpdateTFOrder」、「comboBoxTradeType」、「comboBoxMarketSymbol」。  拉出 1個Button，命名(Name)為「buttonCorrectPriceBySeqNo」。 | |
| 全域變數 |
| // 宣告物件  SKCenterLib m\_pSKCenter = new SKCenterLib(); //登入&環境設定物件  SKOrderLib m\_pSKOrder = new SKOrderLib(); //下單物件  // 是否為非同步委託  bool bAsyncOrder = false; |

|  |
| --- |
| Constructor |
| // comboBoxTradeType  {  comboBoxTradeType.Items.Add("0:ROD");  comboBoxTradeType.Items.Add("1:IOC");  comboBoxTradeType.Items.Add("2:FOK");  }  // comboBoxCorrectPriceByBookNoMarketSymbol  {  comboBoxMarketSymbol.Items.Add("TS:證券");  comboBoxMarketSymbol.Items.Add("TF:期貨");  comboBoxMarketSymbol.Items.Add("TO:選擇權");  }  // comboBoxUpdateTFOrder  {  comboBoxUpdateTFOrder.Items.Add("刪單");  comboBoxUpdateTFOrder.Items.Add("減量");  comboBoxUpdateTFOrder.Items.Add("改價");  } |

|  |
| --- |
| Current Block (可放在buttonCorrectPriceBySeqNo\_Click的Click事件裡) |
| string bstrAccount = comboBoxAccount.Text;  string bstrSeqNo = textBoxSeqNo.Text; // 欲改量的委託序號  string bstrPrice = textBoxPrice.Text; // 修改價格  int nTradeType = 0; // 證券0:ROD 期選0: ROD 1:IOC 2:FOK  string selectedValue2 = comboBoxTradeType.Text;  if (selectedValue2 == "0:ROD") nTradeType = 0;  else if (selectedValue2 == "1:IOC") nTradeType = 1;  else if (selectedValue2 == "2:FOK") nTradeType = 2;  string bstrMessage; //同步刪單：如果回傳值為 0表示刪單成功，訊息內容則為修改訊息。回傳值非0表示刪單失敗，訊息內容為失敗原因。非同步刪單：參照4 - 2 - b OnAsyncOrder。  // 證期權依序號改價(By委託序號)  int nCode = m\_pSKOrder.CorrectPriceBySeqNo(comboBoxUserID.Text, bAsyncOrder, bstrAccount, bstrSeqNo, bstrPrice, nTradeType, out bstrMessage);  // 取得回傳訊息  string msg = "【CorrectPriceBySeqNo】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode) + bstrMessage;  richTextBoxMethodMessage.AppendText(msg + "\n"); |
| 執行成功畫面 |
|  |

###### CorrectPriceByBookNo

|  |  |  |
| --- | --- | --- |
| 證期權依書號改價。 | | |
| **宣告** | Long CorrectPriceByBookNo([in] BSTR bstrLogInID, [in] VARIANT\_BOOL bAsyncOrder, [in] BSTR bstrAccount, [in] BSTR bstrMarketSymbol, [in] BSTR bstrBookNo, [in] BSTR bstrPrice, [in] LONG nTradeType,[**out**] BSTR\* bstrMessage); | |
| **參數** | bstrLogInID | 登入ID。 |
| bAsyncOrder | 是否為非同步委託。 |
| bstrAccount | 證券：委託帳號 (BrokerID＋帳號)。  期貨：委託帳號 ( IB＋帳號) 。 |
| bstrMarketSymbol | 市場類別：TS: 證券TF:期貨TO:選擇權 |
| bstrBookNo | 欲改量的委託書號。 |
| bstrPrice | 修改價格。 |
| nTradeType | **證券0:ROD**  期選0:ROD 1:IOC 2:FOK |
| 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](#_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**改價**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。  ＊證券逐筆nTradeType:請設0(ROD).  ＊不適用盤中零股 | |

|  |
| --- |
| C#範例程式碼\_ TSTFUpdateOrderForm.cs |
| UI示意圖 |
| |  |  | | --- | --- | | 一張含有 文字, 螢幕擷取畫面, 數字, 字型 的圖片  自動產生的描述 | 【紅色虛線部分控件可參考 [CorrectPriceBySeqNo](#_CorrectPriceBySeqNo_1)做法】  拉出1個TextBox，命名(Name)為「textBoxBookNo」。  拉出 1個Button，命名(Name)為「buttonCorrectPriceByBookNo」。 | |
| 全域變數 |
| // 宣告物件  SKCenterLib m\_pSKCenter = new SKCenterLib(); //登入&環境設定物件  SKOrderLib m\_pSKOrder = new SKOrderLib(); //下單物件  // 是否為非同步委託  bool bAsyncOrder = false; |

|  |
| --- |
| Current Block (可放在buttonCorrectPriceByBookNo\_Click的Click事件裡) |
| |  | | --- | | string bstrAccount = comboBoxAccount.Text;  string bstrMarketSymbol = ""; // 市場類別 TS: 證券TF:期貨 TO:選擇權  string selectedValue3 = comboBoxMarketSymbol.Text;  if (selectedValue3 == "TS:證券") bstrMarketSymbol = "TS";  else if (selectedValue3 == "TF:期貨") bstrMarketSymbol = "TF";  else if (selectedValue3 == "TO:選擇權") bstrMarketSymbol = "TO";  string bstrBookNo = textBoxBookNo.Text; // 欲改量的委託書號  string bstrPrice = textBoxPrice.Text; // 修改價格  int nTradeType = 0; // 證券0:ROD 期選0: ROD 1:IOC 2:FOK  string selectedValue2 = comboBoxTradeType.Text;  if (selectedValue2 == "0:ROD") nTradeType = 0;  else if (selectedValue2 == "1:IOC") nTradeType = 1;  else if (selectedValue2 == "2:FOK") nTradeType = 2;  string bstrMessage; //同步刪單：如果回傳值為 0表示刪單成功，訊息內容則為修改訊息。回傳值非0表示刪單失敗，訊息內容為失敗原因。非同步刪單：參照4 - 2 - b OnAsyncOrder。  // 證期權依書號改價(By委託書號)  int nCode = m\_pSKOrder.CorrectPriceByBookNo(comboBoxUserID.Text, bAsyncOrder, bstrAccount, bstrMarketSymbol, bstrBookNo, bstrPrice, nTradeType, out bstrMessage);  // 取得回傳訊息  string msg = "【CorrectPriceByBookNo】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode) + bstrMessage;  richTextBoxMethodMessage.AppendText(msg + "\n"); | |
| 執行成功畫面 |
|  |

###### DecreaseOrderBySeqNo

|  |  |  |
| --- | --- | --- |
| 委託減量。 | | |
| **宣告** | Long DecreaseOrderBySeqNo([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 | 證券：委託帳號 (BrokerID＋帳號)。  期貨：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](#_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。  ＊亦適用證券逐筆及擬真平台 | |

|  |
| --- |
| C#範例程式碼\_ TSTFUpdateOrderForm.cs |
| UI示意圖 |
| |  |  | | --- | --- | | 一張含有 文字, 螢幕擷取畫面, 數字, 字型 的圖片  自動產生的描述 | 【紅色虛線部分控件可參考 [CorrectPriceBySeqNo](#_CorrectPriceBySeqNo_1)做法】  拉出1個TextBox，命名(Name)為「textBoxStockDecreaseQty」。  拉出 1個Button，命名(Name)為「buttonDecreaseOrderBySeqNo」。 | |
| 全域變數 |
| // 宣告物件  SKCenterLib m\_pSKCenter = new SKCenterLib(); //登入&環境設定物件  SKOrderLib m\_pSKOrder = new SKOrderLib(); //下單物件  // 是否為非同步委託  bool bAsyncOrder = false; |

|  |
| --- |
| Current Block (可放在buttonDecreaseOrderBySeqNo\_Click的Click事件裡) |
| string bstrSeqNo = textBoxSeqNo.Text;  string bstrAccount = comboBoxAccount.Text;  int nDecreaseQty = int.Parse(textBoxStockDecreaseQty.Text);  string bstrMessage; //同步刪單：如果回傳值為 0表示刪單成功，訊息內容則為修改訊息。回傳值非0表示刪單失敗，訊息內容為失敗原因。非同步刪單：參照4 - 2 - b OnAsyncOrder。  // 國內委託減量(By委託序號)  int nCode = m\_pSKOrder.DecreaseOrderBySeqNo(comboBoxUserID.Text, bAsyncOrder, bstrAccount, bstrSeqNo, nDecreaseQty, out bstrMessage);  // 取得回傳訊息  string msg = "【DecreaseOrderBySeqNo】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode) + bstrMessage;  richTextBoxMethodMessage.AppendText(msg + "\n"); |
| 執行成功畫面 |
|  |

###### CancelOrderBySeqNo

|  |  |  |
| --- | --- | --- |
| 國內委託删單(By委託序號)。 | | |
| **宣告** | Long CancelOrderBySeqNo([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 | 證券：委託帳號 (BrokerID＋帳號)。  期貨：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，訊息內容則為刪單訊息。回傳值非0表示刪單失敗，訊息內容為失敗原因。  **非同步刪單：**參照[OnAsyncOrder](#_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**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。  ＊亦適用證券逐筆及擬真平台 | |

|  |
| --- |
| C#範例程式碼\_ TSTFUpdateOrderForm.cs |
| UI示意圖 |
| |  |  | | --- | --- | | 一張含有 文字, 螢幕擷取畫面, 數字, 陳列 的圖片  自動產生的描述 | 【紅色虛線部分控件可參考 [CorrectPriceBySeqNo](#_CorrectPriceBySeqNo_1)做法】  拉出 1個Button，命名(Name)為「buttonCancelOrderBySeqNo」。 | |
| 全域變數 |
| // 宣告物件  SKCenterLib m\_pSKCenter = new SKCenterLib(); //登入&環境設定物件  SKOrderLib m\_pSKOrder = new SKOrderLib(); //下單物件  // 是否為非同步委託  bool bAsyncOrder = false; |

|  |
| --- |
| Current Block (可放在buttonCancelOrderBySeqNo\_Click的Click事件裡) |
| string bstrSeqNo = textBoxSeqNo.Text;  string bstrAccount = comboBoxAccount.Text;  string bstrMessage; //同步刪單：如果回傳值為 0表示刪單成功，訊息內容則為修改訊息。回傳值非0表示刪單失敗，訊息內容為失敗原因。非同步刪單：參照4 - 2 - b OnAsyncOrder。  // 國內委託删單(By委託序號)  int nCode = m\_pSKOrder.CancelOrderBySeqNo(comboBoxUserID.Text, bAsyncOrder, bstrAccount, bstrSeqNo, out bstrMessage);  // 取得回傳訊息  string msg = "【CancelOrderBySeqNo】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode) + bstrMessage;  richTextBoxMethodMessage.AppendText(msg + "\n"); |
| 執行成功畫面 |
|  |

###### CancelOrderByBookNo

|  |  |  |
| --- | --- | --- |
| 國內委託删單(By書號)。 | | |
| **宣告** | Long CancelOrderByBookNo([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 | 證券：委託帳號 (BrokerID＋帳號)。  期貨：委託帳號 ( IB＋帳號) 。 |
| bstrBookNo | 欲刪除的委託書號。 |
| bstrMessage | **同步刪單：**如果回傳值為 0表示刪單成功，訊息內容則為刪單訊息。回傳值非0表示刪單失敗，訊息內容為失敗原因。  **非同步刪單：**參照OnAsyncOrder。 |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。 | |
| **備註** | 回傳值0 表示委託伺服器接收成功，詳細委託狀態仍須以委託回報內容為主。  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。 | |

|  |
| --- |
| C#範例程式碼\_ TSTFUpdateOrderForm.cs |
| UI示意圖 |
| |  |  | | --- | --- | | 一張含有 文字, 螢幕擷取畫面, 軟體, 數字 的圖片  自動產生的描述 | 【紅色虛線部分控件可參考 [CorrectPriceBySeqNo](#_CorrectPriceBySeqNo_1)做法】  拉出 1個Button，命名(Name)為「buttonCancelOrderByBookNo」。 | |
| 全域變數 |
| // 宣告物件  SKCenterLib m\_pSKCenter = new SKCenterLib(); //登入&環境設定物件  SKOrderLib m\_pSKOrder = new SKOrderLib(); //下單物件  // 是否為非同步委託  bool bAsyncOrder = false; |

|  |
| --- |
| Current Block (可放在buttonCancelOrderByBookNo\_Click的Click事件裡) |
| string bstrBookNo = textBoxBookNo.Text;  string bstrAccount = comboBoxAccount.Text;  string bstrMessage; //同步刪單：如果回傳值為 0表示刪單成功，訊息內容則為修改訊息。回傳值非0表示刪單失敗，訊息內容為失敗原因。非同步刪單：參照4 - 2 - b OnAsyncOrder。  // 國內委託删單(By委託書號)  int nCode = m\_pSKOrder.CancelOrderByBookNo(comboBoxUserID.Text, bAsyncOrder, bstrAccount, bstrBookNo, out bstrMessage);  // 取得回傳訊息  string msg = "【CancelOrderByBookNo】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode) + bstrMessage;  richTextBoxMethodMessage.AppendText(msg + "\n"); |
| 執行成功畫面 |
|  |

###### CancelOrderByStockNo

|  |  |  |
| --- | --- | --- |
| 國內委託删單(By帳號所屬**登入ID**+商品代號)。 | | |
| **宣告** | Long CancelOrderByStockNo([in] BSTR bstrLogInID, [in] VARIANT\_BOOL bAsyncOrder, [in] BSTR bstrAccount, [in] BSTR bstrStockNo,[**out**] BSTR\* bstrMessage); | |
| **參數** | bstrLogInID | 登入ID。 |
| bAsyncOrder | 是否為非同步委託。 |
| bstrAccount | 證券：委託帳號 (BrokerID＋帳號)。  期貨：委託帳號 ( IB＋帳號) 。 |
| bstrStockNo | 欲刪除的委託商品代號，請輸入有效商品代碼。 |
| bstrMessage | **同步刪單：**如果回傳值為 0表示刪單成功，訊息內容則為刪單訊息。回傳值非0表示刪單失敗，訊息內容為失敗原因。  **非同步刪單：**參照[OnAsyncOrder](#_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**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。  ＊請留意證券代碼，包含刪除盤中零股委託 | |

|  |  |  |
| --- | --- | --- |
| 異動前 | 欄位 | 異動後V2.13.45 |
| 欲刪除的委託商品代號，空白則刪除所該帳號所屬登入ID之所有委託，請注意。 | bstrStockNo  欄位說明：  移除商品代碼為”空白”情況。  功能部分：  V2.13.45（含）以上，商品代碼不可填空白。 | 欲刪除的委託商品代號，請輸入有效商品代碼。 |

|  |
| --- |
| C#範例程式碼\_ TSTFUpdateOrderForm.cs |
| UI示意圖 |
| |  |  | | --- | --- | | 一張含有 文字, 螢幕擷取畫面, 軟體, 數字 的圖片  自動產生的描述 | 【紅色虛線部分控件可參考 [CorrectPriceBySeqNo](#_CorrectPriceBySeqNo_1)做法】  拉出 1個Button，命名(Name)為「buttonCancelOrderByStockNo」。 | |
| 全域變數 |
| // 宣告物件  SKCenterLib m\_pSKCenter = new SKCenterLib(); //登入&環境設定物件  SKOrderLib m\_pSKOrder = new SKOrderLib(); //下單物件  // 是否為非同步委託  bool bAsyncOrder = false; |

|  |
| --- |
| Current Block (可放在buttonCancelOrderByStockNo\_Click的Click事件裡) |
| string bstrStockNo = textBoxCancelOrderByStockNo.Text;  string bstrAccount = comboBoxAccount.Text;  string bstrMessage; //同步刪單：如果回傳值為 0表示刪單成功，訊息內容則為修改訊息。回傳值非0表示刪單失敗，訊息內容為失敗原因。非同步刪單：參照4 - 2 - b OnAsyncOrder。  // 國內委託删單(By ID+商品代號)  int nCode = m\_pSKOrder.CancelOrderByStockNo(comboBoxUserID.Text, bAsyncOrder, bstrAccount, bstrStockNo, out bstrMessage);  // 取得回傳訊息  string msg = "【CancelOrderByStockNo】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode) + bstrMessage;  richTextBoxMethodMessage.AppendText(msg + "\n"); |
| 執行成功畫面 |
|  |

#### **Proxy下單/刪改單**

##### SendFutureProxyOrderCLR

|  |  |  |
| --- | --- | --- |
| 經由proxy server送出期貨委託，需設倉別與盤別。 | | |
| **宣告** | Long SendFutureProxyOrderCLR ([in]BSTR bstrLogInID, [in]struct FUTUREPROXYORDER\* pFUTUREPROXYORDER, [out] BSTR\* bstrMessage); | |
| **參數** | bstrLogInID | 登入ID。 |
| pFUTUREPROXYORDER | SKCOM元件中的[FUTUREPROXYORDER](#_Proxy_Server期貨委託)物件，將下單條件填入該物件後，再帶入此欄位中。 |
| bstrMessage | **Proxy委託通知：**參照OnProxyOrder。 |
| **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。 | |

|  |
| --- |
| C#範例程式碼\_TFSKProxySendOrderForm.cs |
| UI示意圖 |
| |  |  | | --- | --- | |  | 4個TextBox，命名(Name)為「textBoxFutureID」、「textBoxTFbstrSettleYM」、「textBoxFutureQty」、「textBoxFuturePrice」。  6個ComboBox，命名(Name)為「comboBoxTFnPriceFlag」、  「comboBoxFutureDayTrade」、「comboBoxFutureReserved」、「comboBoxFutureNewClose」、「comboBoxFutureTradeType」、「comboBoxFutureBuySell」。  1個Button，命名(Name)為「buttonSendFutureOrderCLR」。 | |
| 全域變數 |
| // 宣告物件  SKCenterLib m\_pSKCenter = new SKCenterLib(); //登入&環境設定物件  SKOrderLib m\_pSKOrder = new SKOrderLib(); //下單物件 |

|  |
| --- |
| Constructor |
| // TF  {  //sTradeType  comboBoxFutureTradeType.Items.Add("ROD");  comboBoxFutureTradeType.Items.Add("IOC");  comboBoxFutureTradeType.Items.Add("FOK");  //sBuySell  comboBoxFutureBuySell.Items.Add("買進");  comboBoxFutureBuySell.Items.Add("賣出");  //sDayTrade  comboBoxFutureDayTrade.Items.Add("否");  comboBoxFutureDayTrade.Items.Add("是");  //sNewClose  comboBoxFutureNewClose.Items.Add("0:新倉");  comboBoxFutureNewClose.Items.Add("1:平倉");  comboBoxFutureNewClose.Items.Add("2:自動");  //comboBoxFutureReserved  comboBoxFutureReserved.Items.Add("盤中單");  comboBoxFutureReserved.Items.Add("預約單");  //comboBoxTFnPriceFlag  {  comboBoxTFnPriceFlag.Items.Add("市價");  comboBoxTFnPriceFlag.Items.Add("限價");  comboBoxTFnPriceFlag.Items.Add("範圍市價");  }  } |

|  |
| --- |
| Current Block (可放在buttonSendFutureOrderCLR\_Click的Click事件裡) |
| //Method  FUTUREPROXYORDER pFUTUREPROXYORDER = new FUTUREPROXYORDER();  string bstrMessage;  pFUTUREPROXYORDER.bstrFullAccount = comboBoxAccount.Text;  pFUTUREPROXYORDER.bstrStockNo = textBoxFutureID.Text;  pFUTUREPROXYORDER.bstrSettleYM = textBoxTFbstrSettleYM.Text;  string selectedValue = comboBoxTFnPriceFlag.Text;  if (selectedValue == "市價") pFUTUREPROXYORDER.nPriceFlag = 0;  else if (selectedValue == "限價") pFUTUREPROXYORDER.nPriceFlag = 1;  else if (selectedValue == "範圍市價") pFUTUREPROXYORDER.nPriceFlag = 2;  selectedValue = comboBoxFutureTradeType.Text;  if (selectedValue == "ROD") pFUTUREPROXYORDER.nTradeType = 0;  else if (selectedValue == "IOC") pFUTUREPROXYORDER.nTradeType = 1;  else if (selectedValue == "FOK") pFUTUREPROXYORDER.nTradeType = 2;  selectedValue = comboBoxFutureBuySell.Text;  if (selectedValue == "買進") pFUTUREPROXYORDER.nBuySell = 0;  else if (selectedValue == "賣出") pFUTUREPROXYORDER.nBuySell = 1;  selectedValue = comboBoxFutureDayTrade.Text;  if (selectedValue == "否") pFUTUREPROXYORDER.nDayTrade = 0;  else if (selectedValue == "是") pFUTUREPROXYORDER.nDayTrade = 1;  if (comboBoxFutureNewClose.Text == "0:新倉") pFUTUREPROXYORDER.bstrOrderType = "0";  else if (comboBoxFutureNewClose.Text == "1:平倉") pFUTUREPROXYORDER.bstrOrderType = "1";  else if (comboBoxFutureNewClose.Text == "2:自動") pFUTUREPROXYORDER.bstrOrderType = "2";  pFUTUREPROXYORDER.bstrPrice = textBoxFuturePrice.Text;  pFUTUREPROXYORDER.nQty = int.Parse(textBoxFutureQty.Text);  selectedValue = comboBoxFutureReserved.Text;  if (selectedValue == "盤中單") pFUTUREPROXYORDER.nReserved = 0;  else if (selectedValue == "預約單") pFUTUREPROXYORDER.nReserved = 1;  // 經由proxy server送出期貨委託，需設倉別與盤別  int nCode = m\_pSKOrder.SendFutureProxyOrderCLR(comboBoxUserID.Text, ref pFUTUREPROXYORDER, out bstrMessage);  // 取得回傳訊息  string msg = "【SendFutureProxyOrderCLR】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode);  richTextBoxMethodMessage.AppendText(msg + "\n");  msg = "【同步委託結果】" + bstrMessage;  richTextBoxMessage.AppendText(msg + "\n"); |
| 執行成功畫面 |
|  |

##### SendOptionProxyOrder

|  |  |  |
| --- | --- | --- |
| 經由proxy server送出選擇權委託 | | |
| **宣告** | Long SendOptionProxyOrder([in]BSTR bstrLogInID, [in]struct FUTUREPROXYORDER\* pFUTUREPROXYORDER, [out] BSTR\* bstrMessage); | |
| **參數** | bstrLogInID | 登入ID。 |
| pFUTUREPROXYORDER | SKCOM元件中的[FUTUREPROXYORDER](#_Proxy_Server選擇權下單_1)物件，將下單條件填入該物件後，再帶入此欄位中。 |
| bstrMessage | **Proxy委託通知：**參照OnProxyOrder。 |
| **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。 | |

|  |
| --- |
| C#範例程式碼\_TFSKProxySendOrderForm.cs |
| UI示意圖 |
| |  |  | | --- | --- | |  | 拉出1個CheckBox，命名(Name)為「checkBoxSendDuplexOrder」。  8個TextBox，命名(Name)為「textBoxOptionID」、「textBoxOptionID2」、「textBoxTObstrSettleYM」、「textBoxTObstrSettleYM2」、「textBoxbstrStrike」、「textBoxbstrStrike2」、「textBoxOptionQty」、「textBoxOptionPrice」。  9個ComboBox，命名(Name)為「comboBoxOptionDayTrade」、  「comboBoxOptionReserved」、「comboBoxOptionbstrOrderType」、「comboBoxOptionTradeType」、「comboBoxnCP」、「comboBoxnCP2」、  「comboBoxTOnPriceFlag」、「comboBoxOptionBuySell」、「comboBoxOptionBuySell2」。  1個Button，命名(Name)為「buttonSendOptionOrder」。 | |

|  |
| --- |
| 全域變數 |
| // 宣告物件  SKCenterLib m\_pSKCenter = new SKCenterLib(); //登入&環境設定物件  SKOrderLib m\_pSKOrder = new SKOrderLib(); //下單物件  // 是否為複式單  bool isDuplexOrder = false; |
| 可放在checkBoxSendDuplexOrder\_CheckedChanged的CheckedChanged事件裡 |
| if (checkBoxSendDuplexOrder.Checked) isDuplexOrder = true;  else isDuplexOrder = false; |
| Constructor |
| // TO  {  //sTradeType  comboBoxOptionTradeType.Items.Add("ROD");  comboBoxOptionTradeType.Items.Add("IOC");  comboBoxOptionTradeType.Items.Add("FOK");  //sBuySell  comboBoxOptionBuySell.Items.Add("買進");  comboBoxOptionBuySell.Items.Add("賣出");  //sBuySell2  comboBoxOptionBuySell2.Items.Add("買進");  comboBoxOptionBuySell2.Items.Add("賣出");  //sDayTrade  comboBoxOptionDayTrade.Items.Add("否");  comboBoxOptionDayTrade.Items.Add("是");  //comboBoxOptionbstrOrderType  comboBoxOptionbstrOrderType.Items.Add("0:新倉");  comboBoxOptionbstrOrderType.Items.Add("1:平倉");  comboBoxOptionbstrOrderType.Items.Add("2:自動");  //sReserved  comboBoxOptionReserved.Items.Add("盤中單");  comboBoxOptionReserved.Items.Add("預約單");  //comboBoxnCP  comboBoxnCP.Items.Add("CALL");  comboBoxnCP.Items.Add("PUT");  //comboBoxnCP2  comboBoxnCP2.Items.Add("CALL");  comboBoxnCP2.Items.Add("PUT");  // comboBoxTOnPriceFlag  {  comboBoxTOnPriceFlag.Items.Add("市價");  comboBoxTOnPriceFlag.Items.Add("限價");  comboBoxTOnPriceFlag.Items.Add("範圍市價");  }  } |

|  |
| --- |
| Current Block (可放在buttonSendOptionOrder\_Click的Click事件裡) |
| FUTUREPROXYORDER pFUTUREPROXYORDER = new FUTUREPROXYORDER();  string bstrMessage;  pFUTUREPROXYORDER.bstrFullAccount = comboBoxAccount.Text;  pFUTUREPROXYORDER.bstrStockNo = textBoxOptionID.Text;  pFUTUREPROXYORDER.bstrSettleYM = textBoxTObstrSettleYM.Text;  pFUTUREPROXYORDER.bstrStrike = textBoxbstrStrike.Text;  string selectedValue = comboBoxOptionTradeType.Text;  if (selectedValue == "IOC") pFUTUREPROXYORDER.nTradeType = 1;  else if (selectedValue == "FOK") pFUTUREPROXYORDER.nTradeType = 2;  selectedValue = comboBoxOptionBuySell.Text;  if (selectedValue == "買進") pFUTUREPROXYORDER.nBuySell = 0;  else if (selectedValue == "賣出") pFUTUREPROXYORDER.nBuySell = 1;  selectedValue = comboBoxOptionDayTrade.Text;  if (selectedValue == "否") pFUTUREPROXYORDER.nDayTrade = 0;  else if (selectedValue == "是") pFUTUREPROXYORDER.nDayTrade = 1;  if (comboBoxOptionbstrOrderType.Text == "0:新倉") pFUTUREPROXYORDER.bstrOrderType = "0";  else if (comboBoxOptionbstrOrderType.Text == "1:平倉") pFUTUREPROXYORDER.bstrOrderType = "1";  else if (comboBoxOptionbstrOrderType.Text == "2:自動") pFUTUREPROXYORDER.bstrOrderType = "2";  pFUTUREPROXYORDER.bstrPrice = textBoxOptionPrice.Text;  pFUTUREPROXYORDER.nQty = int.Parse(textBoxOptionQty.Text);  selectedValue = comboBoxOptionReserved.Text;  if (selectedValue == "盤中單") pFUTUREPROXYORDER.nReserved = 0;  else if (selectedValue == "預約單") pFUTUREPROXYORDER.nReserved = 1;  selectedValue = comboBoxnCP.Text;  if (selectedValue == "CALL") pFUTUREPROXYORDER.nCP = 0;  else if (selectedValue == "PUT") pFUTUREPROXYORDER.nCP = 1;  selectedValue = comboBoxTOnPriceFlag.Text;  if (selectedValue == "市價") pFUTUREPROXYORDER.nPriceFlag = 0;  else if (selectedValue == "限價") pFUTUREPROXYORDER.nPriceFlag = 1;  else if (selectedValue == "範圍市價") pFUTUREPROXYORDER.nPriceFlag = 2;  if (isDuplexOrder != true)  {  // 送出選擇權委託  int nCode = m\_pSKOrder.SendOptionProxyOrder(comboBoxUserID.Text, ref pFUTUREPROXYORDER, out bstrMessage);  // 取得回傳訊息  string msg1 = "【SendOptionProxyOrder】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode);  richTextBoxMethodMessage.AppendText(msg1 + "\n");  }  else  {  pFUTUREPROXYORDER.bstrStockNo2 = textBoxOptionID2.Text; // 委託選擇權代號2{複式單}  pFUTUREPROXYORDER.bstrSettleYM2 = textBoxTObstrSettleYM2.Text; // 指定月份商品契約年月2，共6碼EX:202301  pFUTUREPROXYORDER.bstrStrike2 = textBoxbstrStrike2.Text; // 履約價2  selectedValue = comboBoxnCP2.Text;  if (selectedValue == "CALL") pFUTUREPROXYORDER.nCP2 = 0;  else if (selectedValue == "PUT") pFUTUREPROXYORDER.nCP2 = 1;  // 0:買進 1:賣出{複式單}  if (comboBoxOptionBuySell2.Text == "買進") pFUTUREPROXYORDER.nBuySell2 = 0;  else if (comboBoxOptionBuySell2.Text == "賣出") pFUTUREPROXYORDER.nBuySell2 = 1;  // 經由proxy server送出選擇權複式下單  int nCode = m\_pSKOrder.SendDuplexProxyOrder(comboBoxUserID.Text, ref pFUTUREPROXYORDER, out bstrMessage);  // 取得回傳訊息  string msg2 = "【SendDuplexProxyOrder】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode);  richTextBoxMethodMessage.AppendText(msg2 + "\n");  }  string msg = "【同步委託結果】" + bstrMessage;  richTextBoxMessage.AppendText(msg + "\n"); |
| 執行成功畫面 |
| |  |  | | --- | --- | | 選擇權委託 |  | | 選擇權複式單委託 |  | |

###### SendDuplexProxyOrder

|  |  |  |
| --- | --- | --- |
| 經由proxy server送出選擇權複式下單。 | | |
| **宣告** | Long SendDuplexProxyOrder ([in]BSTR bstrLogInID, [in]struct FUTUREPROXYORDER\* pFUTUREPROXYORDER, [out] BSTR\* bstrMessage); | |
| **參數** | bstrLogInID | 登入ID。 |
| pFUTUREPROXYORDER | SKCOM元件中的[FUTUREPROXYORDER](#_Proxy_Server選擇權複式單_1)物件，將下單條件填入該物件後，再帶入此欄位中。 |
| bstrMessage | **Proxy委託通知：**參照OnProxyOrder。 |
| **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。 | |

##### 刪改單

###### SendFutureProxyAlter

|  |  |  |
| --- | --- | --- |
| 經由proxy server送出期貨刪改單。 | | |
| **宣告** | Long SendFutureProxyAlter ([in]BSTR bstrLogInID, [in]struct FUTUREPROXYORDER\* pFUTUREPROXYORDER, [out] BSTR\* bstrMessage); | |
| **參數** | bstrLogInID | 登入ID。 |
| pFUTUREPROXYORDER | SKCOM元件中的[FUTUREPROXYORDER](#_Proxy_Server期選刪改單_1)物件，將下單條件填入該物件後，再帶入此欄位中。 |
| bstrMessage | **Proxy委託通知：**參照OnProxyOrder。 |
| **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。 | |

|  |
| --- |
| C#範例程式碼\_TFSKProxyUpdateOrderForm.cs |
| UI示意圖 |
| |  |  | | --- | --- | |  | 4個TextBox，命名(Name)為「textBoxSeqNo」、「textBoxBookNo」、「textBoxStockDecreaseQty」、「textBoxPrice」。  3個ComboBox，命名(Name)為  「comboBoxUpdateTFOrder」、「comboBoxFutureReserved」、「comboBoxTradeType」。  2個Button，命名(Name)為「buttonSendFutureProxyAlter」、「buttonSendOptionProxyAlter」。 | |
| 全域變數 |
| // 宣告物件  SKCenterLib m\_pSKCenter = new SKCenterLib(); //登入&環境設定物件  SKOrderLib m\_pSKOrder = new SKOrderLib(); //下單物件 |

|  |
| --- |
| Constructor |
| // comboBoxTradeType  {  comboBoxTradeType.Items.Add("0:ROD");  comboBoxTradeType.Items.Add("1:IOC");  comboBoxTradeType.Items.Add("2:FOK");  }  // comboBoxUpdateTFOrder  {  comboBoxUpdateTFOrder.Items.Add("刪單");  comboBoxUpdateTFOrder.Items.Add("減量");  comboBoxUpdateTFOrder.Items.Add("改價");  }  //comboBoxFutureReserved  {  comboBoxFutureReserved.Items.Add("盤中單");  comboBoxFutureReserved.Items.Add("預約單");  } |

|  |
| --- |
| Current Block (可放在buttonSendFutureProxyAlter\_Click的Click事件裡) |
| //Method  FUTUREPROXYORDER pFUTUREPROXYORDER = new FUTUREPROXYORDER();  string bstrMessage; //同步刪單：如果回傳值為 0表示刪單成功，訊息內容則為修改訊息。回傳值非0表示刪單失敗，訊息內容為失敗原因  pFUTUREPROXYORDER.bstrFullAccount = comboBoxAccount.Text;  if (comboBoxUpdateTFOrder.Text == "刪單") pFUTUREPROXYORDER.bstrOrderType = "0";  else if (comboBoxUpdateTFOrder.Text == "減量") pFUTUREPROXYORDER.bstrOrderType = "1";  else if (comboBoxUpdateTFOrder.Text == "改價") pFUTUREPROXYORDER.bstrOrderType = "2";  pFUTUREPROXYORDER.bstrPrice = textBoxPrice.Text;  if (comboBoxFutureReserved.Text == "盤中單") pFUTUREPROXYORDER.nReserved = 0;  else if (comboBoxFutureReserved.Text == "預約單") pFUTUREPROXYORDER.nReserved = 1;  pFUTUREPROXYORDER.nQty = int.Parse(textBoxStockDecreaseQty.Text);  if (comboBoxTradeType.Text == "ROD") pFUTUREPROXYORDER.nTradeType = 0;  else if (comboBoxTradeType.Text == "IOC") pFUTUREPROXYORDER.nTradeType = 1;  else if (comboBoxTradeType.Text == "FOK") pFUTUREPROXYORDER.nTradeType = 2;  pFUTUREPROXYORDER.bstrBookNo = textBoxBookNo.Text;  pFUTUREPROXYORDER.bstrSeqNo = textBoxSeqNo.Text;  // 經由proxy server送出期貨刪改單  int nCode = m\_pSKOrder.SendFutureProxyAlter(comboBoxUserID.Text, ref pFUTUREPROXYORDER, out bstrMessage);  // 取得回傳訊息  string msg = "【SendFutureProxyAlter】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode) + bstrMessage;  richTextBoxMethodMessage.AppendText(msg + "\n"); |

|  |
| --- |
| Current Block (可放在buttonSendOptionProxyAlter\_Click的Click事件裡) |
| FUTUREPROXYORDER pFUTUREPROXYORDER = new FUTUREPROXYORDER();  string bstrMessage; //同步刪單：如果回傳值為 0表示刪單成功，訊息內容則為修改訊息。回傳值非0表示刪單失敗，訊息內容為失敗原因  pFUTUREPROXYORDER.bstrFullAccount = comboBoxAccount.Text;  if (comboBoxUpdateTFOrder.Text == "刪單") pFUTUREPROXYORDER.bstrOrderType = "0";  else if (comboBoxUpdateTFOrder.Text == "減量") pFUTUREPROXYORDER.bstrOrderType = "1";  else if (comboBoxUpdateTFOrder.Text == "改價") pFUTUREPROXYORDER.bstrOrderType = "2";  pFUTUREPROXYORDER.bstrPrice = textBoxPrice.Text;  if (comboBoxFutureReserved.Text == "盤中單") pFUTUREPROXYORDER.nReserved = 0;  else if (comboBoxFutureReserved.Text == "預約單") pFUTUREPROXYORDER.nReserved = 1;  pFUTUREPROXYORDER.nQty = int.Parse(textBoxStockDecreaseQty.Text);  if (comboBoxTradeType.Text == "ROD") pFUTUREPROXYORDER.nTradeType = 0;  else if (comboBoxTradeType.Text == "IOC") pFUTUREPROXYORDER.nTradeType = 1;  else if (comboBoxTradeType.Text == "FOK") pFUTUREPROXYORDER.nTradeType = 2;  pFUTUREPROXYORDER.bstrBookNo = textBoxBookNo.Text;  pFUTUREPROXYORDER.bstrSeqNo = textBoxSeqNo.Text;  // 經由proxy server送出選擇權刪改單  int nCode = m\_pSKOrder.SendOptionProxyAlter(comboBoxUserID.Text, ref pFUTUREPROXYORDER, out bstrMessage);  // 取得回傳訊息  string msg = "【SendOptionProxyAlter】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode) + bstrMessage;  richTextBoxMethodMessage.AppendText(msg + "\n"); |
| 執行成功畫面 |
| |  |  | | --- | --- | | 期貨 |  | | 選擇權 |  | |

###### SendOptionProxyAlter

|  |  |  |
| --- | --- | --- |
| 經由proxy server送出選擇權刪改單。 | | |
| **宣告** | Long SendOptionProxyAlter([in]BSTR bstrLogInID, [in]struct FUTUREPROXYORDER\* pFUTUREPROXYORDER, [out] BSTR\* bstrMessage); | |
| **參數** | bstrLogInID | 登入ID。 |
| pFUTUREPROXYORDER | SKCOM元件中的[FUTUREPROXYORDER](#_Proxy_Server期選刪改單_1)物件，將下單條件填入該物件後，再帶入此欄位中。 |
| bstrMessage | **Proxy委託通知：**參照OnProxyOrder。 |
| **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。 | |

# 事件

## OnOpenInterest

|  |  |  |
| --- | --- | --- |
| (新)期貨未平倉GW。透過呼叫 [GetOpenInterest](#_GetOpenInterestGW)GW 後，資訊由該事件回傳。 | | |
| **宣告** | void OnOpenInterest( [in] BSTR bstrData); | |
| **參數** | bstrData | 每一筆資料以「,」分隔每一個欄位，欄位依序為：  格式1:   |  |  | | --- | --- | | 1 | 市場別 | | 2 | 帳號 | | 3 | 商品 | | 4 | 買賣別 | | 5 | 未平倉部位 | | 6 | 當沖未平倉部位 | | 7 | 平均成本(小數部分已處理) | | 8 | 單口手續費 | | 9 | 交易稅(萬分之X) | | 10 | LOGIN\_ID | |
| **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。  若查無資料，則回傳M003 NO DATA#   * 含複式單，市場別：TM | |

|  |
| --- |
| C#範例程式碼\_TFReadOrderForm.cs |
| UI示意圖 |
| 拉出1個 DataGridView，命名(Name)為「dataGridViewOnOpenInterest」。 一張含有 白色, 螢幕擷取畫面, 設計 的圖片  自動產生的描述 |
| 全域變數 |
| // 宣告物件  SKCenterLib m\_pSKCenter = new SKCenterLib(); //登入&環境設定物件  SKOrderLib m\_pSKOrder = new SKOrderLib(); //下單物件 |
| Constructor |
| // dataGridViewOnOpenInterest  {  dataGridViewOnOpenInterest.Columns.Add("Column1", "市場別");  dataGridViewOnOpenInterest.Columns.Add("Column2", "帳號");  dataGridViewOnOpenInterest.Columns.Add("Column3", "商品");  dataGridViewOnOpenInterest.Columns.Add("Column4", "買賣別");  dataGridViewOnOpenInterest.Columns.Add("Column5", "未平倉部位");  dataGridViewOnOpenInterest.Columns.Add("Column6", "當沖未平倉部位");  dataGridViewOnOpenInterest.Columns.Add("Column7", "平均成本(小數部分已處理)");  dataGridViewOnOpenInterest.Columns.Add("Column8", "單口手續費");  dataGridViewOnOpenInterest.Columns.Add("Column9", "交易稅(萬分之X)");  dataGridViewOnOpenInterest.Columns.Add("Column10", "LOGIN\_ID");  } |

|  |
| --- |
| Current Block (可放在Form的Load事件裡) |
| //Event  //查詢期貨未平倉－可指定回傳格式  m\_pSKOrder.OnOpenInterest += new \_ISKOrderLibEvents\_OnOpenInterestEventHandler(OnOpenInterest);  void OnOpenInterest(string bstrData)  {  // 使用 Split 方法將字串拆分成陣列  string[] values = bstrData.Split(',');  if (values[0] == "001" || values[0] == "970") // 查無資料 或 後台的問題  {  dataGridViewOnOpenInterest.Rows.Add(bstrData);  }  else  {  dataGridViewOnOpenInterest.Rows.Add(values[0], values[1], values[2], values[3], values[4], values[5], values[6], values[7], values[8], values[9]);  }  } |

|  |
| --- |
| 查詢示意畫面 |
| 一張含有 文字, 螢幕擷取畫面, 數字, 字型 的圖片  自動產生的描述 |

## OnFutureRights

|  |  |  |
| --- | --- | --- |
| 國內期貨權益數。透過呼叫 [[GetFuture](#_4-2-41_GetFutureRights)Rights](#_GetFutureRights) 後，資訊由該事件回傳。 | | |
| **宣告** | void OnFutureRights([in] BSTR bstrData); | |
| **參數** | bstrData | 回傳的字串內容，以「,」分隔每一個欄位，欄位依序為：  0 帳戶餘額  1 浮動損益  2 已實現費用  3 交易稅  4 預扣權利金  5 權利金收付  6 權益數  7 超額保證金  8 存提款  9 買方市值  10 賣方市值  11 期貨平倉損益  12 盤中未實現  13 原始保證金  14 維持保證金  15 部位原始保證金  16 部位維持保證金  17 委託保證金  18 超額最佳保證金  19 權利總值  20 預扣費用  21 原始保證金  22 昨日餘額  23 選擇權組合單加不加收保證金  24 維持率  25 幣別  26 足額原始保證金  27 足額維持保證金  28 足額可用  29 抵繳金額  30 有價可用  31 可用餘額  32 足額現金可用  33 有價價值  34 風險指標  35 選擇權到期差異  36 選擇權到期差損  37 期貨到期損益  38 加收保證金  **39 LOGIN\_ID**  **40 ACCOUNT\_NO** |
| **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。 * 全幣別（含基幣），回傳第一筆資料為基幣。 | |

|  |
| --- |
| C#範例程式碼\_TFReadOrderForm.cs |
| UI示意圖 |
| 拉出1個 DataGridView，命名(Name)為「dataGridViewOnFutureRights」。 一張含有 白色, 螢幕擷取畫面, 設計 的圖片  自動產生的描述 |
| 全域變數 |
| // 宣告物件  SKCenterLib m\_pSKCenter = new SKCenterLib(); //登入&環境設定物件  SKOrderLib m\_pSKOrder = new SKOrderLib(); //下單物件 |
| Constructor |
| //dataGridViewOnFutureRights  {  dataGridViewOnFutureRights.Columns.Add("Column1", "帳戶餘額");  dataGridViewOnFutureRights.Columns.Add("Column2", "浮動損益");  dataGridViewOnFutureRights.Columns.Add("Column3", "已實現費用");  dataGridViewOnFutureRights.Columns.Add("Column4", "交易稅");  dataGridViewOnFutureRights.Columns.Add("Column5", "預扣權利金");  dataGridViewOnFutureRights.Columns.Add("Column6", "權利金收付");  dataGridViewOnFutureRights.Columns.Add("Column7", "權益數");  dataGridViewOnFutureRights.Columns.Add("Column8", "超額保證金");  dataGridViewOnFutureRights.Columns.Add("Column9", "存提款");  dataGridViewOnFutureRights.Columns.Add("Column10", "買方市值");  dataGridViewOnFutureRights.Columns.Add("Column11", "賣方市值");  dataGridViewOnFutureRights.Columns.Add("Column12", "期貨平倉損益");  dataGridViewOnFutureRights.Columns.Add("Column13", "盤中未實現");  dataGridViewOnFutureRights.Columns.Add("Column14", "原始保證金");  dataGridViewOnFutureRights.Columns.Add("Column15", "維持保證金");  dataGridViewOnFutureRights.Columns.Add("Column16", "部位原始保證金");  dataGridViewOnFutureRights.Columns.Add("Column17", "部位維持保證金");  dataGridViewOnFutureRights.Columns.Add("Column18", "委託保證金");  dataGridViewOnFutureRights.Columns.Add("Column19", "超額最佳保證金");  dataGridViewOnFutureRights.Columns.Add("Column20", "權利總值");  dataGridViewOnFutureRights.Columns.Add("Column21", "預扣費用");  dataGridViewOnFutureRights.Columns.Add("Column22", "原始保證金");  dataGridViewOnFutureRights.Columns.Add("Column23", "昨日餘額");  dataGridViewOnFutureRights.Columns.Add("Column24", "選擇權組合單加不加收保證金");  dataGridViewOnFutureRights.Columns.Add("Column25", "維持率");  dataGridViewOnFutureRights.Columns.Add("Column26", "幣別");  dataGridViewOnFutureRights.Columns.Add("Column27", "足額原始保證金");  dataGridViewOnFutureRights.Columns.Add("Column28", "足額維持保證金");  dataGridViewOnFutureRights.Columns.Add("Column29", "足額可用");  dataGridViewOnFutureRights.Columns.Add("Column30", "抵繳金額");  dataGridViewOnFutureRights.Columns.Add("Column31", "有價可用");  dataGridViewOnFutureRights.Columns.Add("Column32", "可用餘額");  dataGridViewOnFutureRights.Columns.Add("Column33", "足額現金可用");  dataGridViewOnFutureRights.Columns.Add("Column34", "有價價值");  dataGridViewOnFutureRights.Columns.Add("Column35", "風險指標");  dataGridViewOnFutureRights.Columns.Add("Column36", "選擇權到期差異");  dataGridViewOnFutureRights.Columns.Add("Column37", "選擇權到期差損");  dataGridViewOnFutureRights.Columns.Add("Column38", "期貨到期損益");  dataGridViewOnFutureRights.Columns.Add("Column39", "加收保證金");  dataGridViewOnFutureRights.Columns.Add("Column40", "LOGIN\_ID");  dataGridViewOnFutureRights.Columns.Add("Column41", "ACCOUNT\_NO");  } |

|  |
| --- |
| Current Block (可放在Form的Load事件裡) |
| //Event  //查詢國內權益數  m\_pSKOrder.OnFutureRights += new \_ISKOrderLibEvents\_OnFutureRightsEventHandler(OnFutureRights);  void OnFutureRights(string bstrData)  {  // 使用 Split 方法將字串拆分成陣列  string[] values = bstrData.Split(',');  if (values[0] == "980" || values[0] == "970") // 後台的問題  {  dataGridViewOnFutureRights.Rows.Add(bstrData);  }  else  {  dataGridViewOnFutureRights.Rows.Add(values[0], values[1], values[2], values[3], values[4], values[5], values[6], values[7], values[8], values[9], values[10], values[11], values[12], values[13], values[14], values[15], values[16], values[17], values[18], values[19], values[20], values[21], values[22], values[23], values[24], values[25], values[26], values[27], values[28], values[29], values[30], values[31], values[32], values[33], values[34], values[35], values[36], values[37], values[38], values[39], values[40]);  }  } |

|  |
| --- |
| 查詢示意畫面 |
| 一張含有 文字, 數字, 行, 字型 的圖片  自動產生的描述 |

## 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 | |

|  |
| --- |
| C#範例程式碼\_TFSendOrderForm.cs |
| UI示意圖 |
| 拉出1個RichTextBox，命名(Name)為「richTextBoxMessage」。 |
| 全域變數 |
| // 宣告物件  SKCenterLib m\_pSKCenter = new SKCenterLib(); //登入&環境設定物件  SKOrderLib m\_pSKOrder = new SKOrderLib(); //下單物件  // 是否為非同步委託  bool bAsyncOrder = false; |
| Current Block (可放在Form的Load事件裡) |
| //Event  // 非同步委託結果  m\_pSKOrder.OnAsyncOrder += new \_ISKOrderLibEvents\_OnAsyncOrderEventHandler(OnAsyncOrder);  void OnAsyncOrder(int nThreadID, int nCode, string bstrMessage)  {  // 取得回傳訊息  string msg = "TID:" + nThreadID + "收單訊息:" + bstrMessage;  msg = "【非同步委託結果】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode) + msg;  richTextBoxMessage.AppendText(msg + "\n");  } |

## 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

|  |
| --- |
| C#範例程式碼\_TFSKProxySendOrderForm.cs |
| UI示意圖 |
| 拉出1個 RichTextBox，命名(Name)為「richTextBoxMessage」。  一張含有 文字, 字型, 白色, 螢幕擷取畫面 的圖片  自動產生的描述 |
| 全域變數 |
| // 宣告物件  SKCenterLib m\_pSKCenter = new SKCenterLib(); //登入&環境設定物件  SKOrderLib m\_pSKOrder = new SKOrderLib(); //下單物件 |
| Current Block (可放在Form的Load事件裡) |
| //Event  // Proxy委託結果  m\_pSKOrder.OnProxyOrder += new \_ISKOrderLibEvents\_OnProxyOrderEventHandler(OnProxyOrder);  void OnProxyOrder(int nStampID, int nCode, string bstrMessage)  {  // 取得回傳訊息  string msg = "Time Stamp:" + nStampID + "收單訊息:" + bstrMessage;  msg = "【Proxy委託結果】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode) + msg;  richTextBoxMessage.AppendText(msg + "\n");  } |

# Struct結構物件

## FUTUREORDER (期權下單物件)

### 一般期選委託▼

struct FUTUREORDER

{

BSTR bstrFullAccount; //期貨帳號，分公司代碼＋帳號7碼

BSTR bstrStockNo; //委託期權代號

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

SHORT sBuySell; //0:買進 1:賣出

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

SHORT sNewClose; //新平倉，0:新倉 1:平倉 2:自動{新期貨、選擇權使用}

BSTR bstrPrice; //委託價格(IOC and FOK，可用「M」表示市價，「P」表示範圍市價)

LONG nQty; //交易口數

SHORT sReserved; //{期貨委託SendFutureOrderCLR適用}盤別，0:盤中(T盤及T+1盤)；1:T盤預約

};

### 選擇權複式單▼

struct FUTUREORDER

{

BSTR bstrFullAccount; //期貨帳號，分公司代碼＋帳號7碼

BSTR bstrStockNo; //委託選擇權代號1

BSTR bstrStockNo2; //委託選擇權代號2{複式單}

SHORT sTradeType; // 1:IOC 2:FOK

SHORT sBuySell; //0:買進 1:賣出

SHORT sBuySell2; //0:買進 1:賣出{複式單}

SHORT sNewClose; //新平倉，0:新倉 1:平倉

BSTR bstrPrice; //委託價格

LONG nQty; //交易口數

};

### 組合部位、複式單拆解、雙邊部位了結▼

組合部位▼

struct FUTUREORDER

{

BSTR bstrFullAccount; //期貨帳號，分公司代碼＋帳號7碼

BSTR bstrStockNo; //委託選擇權代號1{組合部位}

BSTR bstrStockNo2; //委託選擇權代號2{組合部位}

SHORT sBuySell; //0:買進 1:賣出{組合部位}

SHORT sBuySell2; //0:買進 1:賣出{組合部位}

LONG nQty; //交易口數{組合部位}

};

複式單拆解(單邊或雙邊)▼

struct FUTUREORDER

{

BSTR bstrFullAccount; //期貨帳號，分公司代碼＋帳號7碼

BSTR bstrStockNo; //**複式單商品代碼**{複式單拆解}

SHORT sBuySell; //0:買進 1:賣出{複式單拆解}

LONG nQty; //交易口數{複式單拆解}

};

複式單拆解(單邊或雙邊)

|  |  |  |
| --- | --- | --- |
| 異動前 | 欄位 | 異動後 |
| 複式單拆解(單邊或雙邊)  struct FUTUREORDER  {  BSTR bstrFullAccount;  BSTR bstrStockNo;  BSTR **bstrStockNo2;**  SHORT sBuySell;  LONG nQty;  }; | 刪除  雙邊  **bstrStockNo2;**  **//委託選擇權代號2** | 複式單拆解(單邊)  struct FUTUREORDER  {  BSTR bstrFullAccount;  BSTR bstrStockNo;  SHORT sBuySell;  LONG nQty;  }; |

雙邊部位了結▼

struct FUTUREORDER

{

BSTR bstrFullAccount; //期貨帳號，分公司代碼＋帳號7碼

BSTR bstrStockNo; //委託選擇權代號1{雙邊部位了結}

LONG nQty; //交易口數{雙邊部位了結}

};

## Proxy Server期選

### Proxy Server期貨下單

struct FUTUREPROXYORDER

{

BSTR bstrFullAccount; //期貨帳號，分公司四碼＋帳號7碼

BSTR bstrStockNo; //委託商品代號 ex:FITX

BSTR bstrSettleYM; // 指定月份商品契約年月，共6碼EX:202212

LONG nBuySell; //0:買進 1:賣出

LONG nPriceFlag; //0:市價 1:限價 2:範圍市價

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

BSTR bstrOrderType; //0:新倉 1:平倉 2:自動

LONG nReserved; //0:盤中單 1:預約單

LONG nQty; //交易口數{組合部位}

BSTR bstrPrice; //委託價格

LONG nTradeType; // 0:ROD 1:IOC 2:FOK

};

|  |
| --- |
| 備註 |
| 欲下單近月商品，EX: TX00，請於bstrStockNo輸入FITX、bstrSettleYM輸入(近月月份)202402 |

### Proxy Server選擇權下單

struct FUTUREPROXYORDER

{

BSTR bstrFullAccount; //期貨帳號，分公司四碼＋帳號7碼

BSTR bstrStockNo; //委託商品代號 ex:TXO

BSTR bstrPrice; //委託價格

BSTR bstrSettleYM; // 指定月份商品契約年月，共6碼EX:202212

BSTR bstrStrike; //履約價1

BSTR bstrOrderType; //0:新倉 1:平倉 2:自動

LONG nReserved; //0:盤中單 1:預約單

LONG nQty; //交易口數{組合部位}

LONG nCP; //0:CALL 1:PUT

LONG nBuySell; //0:買進 1:賣出

LONG nPriceFlag; //0:市價 1:限價 2:範圍市價

LONG nTradeType; // 0:ROD 1:IOC 2:FOK

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

};

\*下單範例：2024年3月第4週選擇權商品，請於bstrStockNo輸入TX4、bstrSettleYM輸入202403。2024年4月第1週選擇權商品，請於bstrStockNo輸入TX1、bstrSettleYM輸入202404。

### Proxy Server選擇權複式單

struct FUTUREPROXYORDER

{

BSTR bstrFullAccount; //期貨帳號，分公司四碼＋帳號7碼

BSTR bstrStockNo; //委託商品代號 ex:TXO

BSTR bstrPrice; //委託價格

BSTR bstrSettleYM; //指定月份商品契約年月，共6碼EX:202212

BSTR bstrStrike; //履約價1

BSTR bstrSettleYM2; //指定月份商品契約年月2，共6碼EX:202301

BSTR bstrStrike2; //履約價2

BSTR bstrOrderType; //0:新倉 1:平倉 2:自動

LONG nReserved; //0:盤中單 1:預約單

LONG nQty; //交易口數{組合部位}

LONG nCP; //買賣權1 ( 0:CALL 1:PUT )

LONG nBuySell; //買賣別1 ( 0:買進 1:賣出)

BSTR bstrStockNo2; //委託商品代號2

LONG nCP2; //買賣權2 ( 0:CALL 1:PUT )

LONG nBuySell2; //買賣別2 ( 0:買進 1:賣出)

LONG nPriceFlag; // 0:市價 1:限價 2:範圍市價

LONG nTradeType // 1:IOC 2:FOK

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

};

### Proxy Server期選刪改單

struct FUTUREPROXYORDER

{

BSTR bstrFullAccount; //期貨帳號，分公司四碼＋帳號7碼

BSTR bstrOrderType; //0:刪單 1:減量 2:改價

BSTR bstrPrice; //委託價格

(改量:帶0，也可以帶""。改價:帶需要改的價。刪單:帶原始價，也可帶"")

LONG nReserved; //0:盤中單 1:預約單

LONG nQty; //交易口數

(改量 : 帶要減的量。改價 : 帶0也可帶""。刪單 : 帶原始量)

LONG nTradeType; // 0:ROD 1:IOC 2:FOK

BSTR bstrBookNo; //委託書號

BSTR bstrSeqNo; //委託序號

};