策略王COM元件使用說明

-海選報價-

# 函式範例

# SKOOQuoteLib(海選報價)

*函式*

連線

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| **連線** | [SKOOQuoteLib\_EnterMonitorLONG](#_SKOOQuoteLib_EnterMonitorLONG) | LONG index對應 |
| **斷線** | [SKOOQuoteLib\_LeaveMonitor](#_SKOOQuoteLib_LeaveMonitor) |  |
| **連線狀態** | [SKOOQuoteLib\_IsConnected](#_SKOOQuoteLib_IsConnected) |  |

商品清單

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| **要求海期商品** | [SKOOQuoteLib\_RequestProducts](#_SKOOQuoteLib_RequestProducts) |  |

海選資訊

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| **取得商品物件** | [SKOOQuoteLib\_GetStockByNoLONG](#_SKOOQuoteLib_GetStockByNoLONG) | LONG index對應 |

即時報價

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| **訂閱報價** | [SKOOQuoteLib\_RequestStocks](#_SKOOQuoteLib_RequestStocks) |  |
| **取得商品物件** | [SKOOQuoteLib\_GetStockByIndexLONG](#_SKOOQuoteLib_GetStockByIndexLONG) | LONG index對應 |

十檔&五檔&成交明細

|  |  |  |
| --- | --- | --- |
| **功能** | **函式名稱** | **備註** |
| **訂閱Ticks&十檔&五檔** | [SKOOQuoteLib\_RequestTicks](#_SKOOQuoteLib_RequestTicks) | [OnNotifyTicksLONG](#_OnNotifyTicksLONG)、[OnNotifyHistoryTicksLONG](#_OnNotifyHistoryTicksLONG)、[OnNotifyBest5LONG](#_OnNotifyBest5LONG)  [OnNotifyBest10LONG](#_OnNotifyBest10LONG) |
| **訂閱即時成交明細** | [SKOOQuoteLib\_RequestLiveTick](#_SKOOQuoteLib_RequestLiveTick) | 包含訂閱，僅成交明細 |
| **取得Tick物件** | [SKOOQuoteLib\_GetTickLONG](#_SKOOQuoteLib_GetTickLONG) | LONG index對應 |
| **取得五檔** | [SKOOQuoteLib\_GetBest5LONG](#_SKOOQuoteLib_GetBest5LONG) | LONG index對應 |
| **訂閱十檔** | [SKOOQuoteLib\_RequestMarketDepth](#_SKOOQuoteLib_RequestMarketDepth) | 包含訂閱，僅十檔、五檔 |

*事件*

|  |  |  |
| --- | --- | --- |
| **功能** | **事件名稱** | **備註** |
| **連線狀態** | [OnConnect](#_OnConnect) | 避免在此處直接進行[RequestStocks](#_SKOOQuoteLib_RequestStocks) and [RequestTicks](#_SKOOQuoteLib_RequestTicks) 等報價訂閱，因各交易所商品未下載完成，造成無法正常訂閱。 |
| **海選商品資料** | [OnProducts](#_OnProducts) |  |
| **報價更新通知** | [OnNotifyQuoteLONG](#_OnNotifyQuoteLONG) | LONG index對應 |
| **即時Tick** | [OnNotifyTicksLONG](#_OnNotifyTicksLONG) | LONG index對應 |
| **回補Tick** | [OnNotifyHistoryTicksLONG](#_OnNotifyHistoryTicksLONG) | LONG index對應 |
| **即時五檔** | [OnNotifyBest5LONG](#_OnNotifyBest5LONG) | LONG index對應 |
| **即時十檔** | [OnNotifyBest10LONG](#_OnNotifyBest10LONG) | LONG index對應 |

## 連線

### SKOOQuoteLib\_EnterMonitorLONG

|  |  |  |
| --- | --- | --- |
| (LONG index)與報價伺服器建立連線。 | | |
| **宣告** | Long SKOOQuoteLib\_EnterMonitorLONG(); | |
| **參數** | 無 |  |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **備註** | 連線狀態由 [OnConnect](#_OnConnect)事件回傳。  需先簽署期貨API下單聲明書，方可使用 | |

|  |
| --- |
| C#範例程式碼\_OOQuoteForm.cs |
| UI示意圖 |
| |  |  | | --- | --- | |  | 拉出 1個Button，命名(Name)為「buttonSKOOQuoteLib\_EnterMonitorLONG」。 | |
| 全域變數 |
| // 宣告物件  SKCenterLib m\_pSKCenter = new SKCenterLib(); // 登入&環境設定物件  SKOOQuoteLib m\_pSKOOQuote = new SKOOQuoteLib(); // 海選報價物件 |
| Current Block (可放在buttonSKOOQuoteLib\_EnterMonitorLONG\_Click的Click事件裡) |
| // 與報價伺服器連線  int nCode = m\_pSKOOQuote.SKOOQuoteLib\_EnterMonitorLONG();  // 取得回傳訊息  string msg = "【SKOOQuoteLib\_EnterMonitorLONG】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode);  richTextBoxMethodMessage.AppendText(msg + "\n"); |
| 執行成功畫面 |
|  |

### SKOOQuoteLib\_LeaveMonitor

|  |  |  |
| --- | --- | --- |
| 中斷報價伺服器連線。 | | |
| **宣告** | Long SKOOQuoteLib\_LeaveMonitor(); | |
| **參數** | 無 |  |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **備註** | 連線狀態由[OnConnect](#_OnConnect)事件回傳。 | |

|  |
| --- |
| C#範例程式碼\_OOQuoteForm.cs |
| UI示意圖 |
| |  |  | | --- | --- | |  | 拉出 1個Button，命名(Name)為「buttonSKOOQuoteLib\_LeaveMonitor」。 | |
| 全域變數 |
| // 宣告物件  SKCenterLib m\_pSKCenter = new SKCenterLib(); // 登入&環境設定物件  SKOOQuoteLib m\_pSKOOQuote = new SKOOQuoteLib(); // 海選報價物件 |
| Current Block (可放在buttonSKOOQuoteLib\_LeaveMonitor\_Click的Click事件裡) |
| // 中斷報價伺服器連線  int nCode = m\_pSKOOQuote.SKOOQuoteLib\_LeaveMonitor();  // 取得回傳訊息  string msg = "【SKOOQuoteLib\_LeaveMonitor】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode);  richTextBoxMethodMessage.AppendText(msg + "\n"); |
| 執行成功畫面 |
|  |

### SKOOQuoteLib\_IsConnected

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

|  |
| --- |
| C#範例程式碼\_OOQuoteForm.cs |
| UI示意圖 |
| |  |  | | --- | --- | |  | 拉出 1個Button，命名(Name)為「buttonSKOOQuoteLib\_IsConnected」。 | |
| 全域變數 |
| // 宣告物件  SKCenterLib m\_pSKCenter = new SKCenterLib(); // 登入&環境設定物件  SKOOQuoteLib m\_pSKOOQuote = new SKOOQuoteLib(); // 海選報價物件 |
| Current Block (可放在buttonSKOOQuoteLib\_IsConnected\_Click的Click事件裡) |
| // 檢查目前連線狀態  int nCode = m\_pSKOOQuote.SKOOQuoteLib\_IsConnected();  // 取得回傳訊息  string msg = "";  if (nCode == 1) msg = "連線中";  else msg = "失敗";  msg = "【SKOOQuoteLib\_IsConnected】" + msg;  richTextBoxMessage.AppendText(msg + "\n"); |
| 執行成功畫面 |
|  |

## 商品清單

### SKOOQuoteLib\_RequestProducts

|  |  |  |
| --- | --- | --- |
| 取得海外商品檔。 | | |
| **宣告** | Long SKOOQuoteLib\_RequestProducts(); | |
| **參數** | 無 |  |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **備註** | 商品資訊由[OnProducts](#_OnProducts)事件取得。 | |

|  |
| --- |
| C#範例程式碼\_OOQuoteForm.cs |
| UI示意圖 |
| |  |  | | --- | --- | |  | 拉出 1個Button，命名(Name)為「buttonSKOOQuoteLib\_RequestProducts」。 | |
| 全域變數 |
| // 宣告物件  SKCenterLib m\_pSKCenter = new SKCenterLib(); // 登入&環境設定物件  SKOOQuoteLib m\_pSKOOQuote = new SKOOQuoteLib(); // 海選報價物件 |
| Current Block (可放在buttonSKOOQuoteLib\_RequestProducts\_Click的Click事件裡) |
| richTextBoxOnProducts.Clear();  // 取得海外商品檔  int nCode = m\_pSKOOQuote.SKOOQuoteLib\_RequestProducts();  // 取得回傳訊息  string msg = "【SKOOQuoteLib\_RequestProducts】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode);  richTextBoxMethodMessage.AppendText(msg + "\n"); |
| 執行成功畫面 |
|  |

## 海選資訊

### SKOOQuoteLib\_GetStockByNoLONG

|  |  |  |
| --- | --- | --- |
| (LONG index)根據商品代號，取回海選報價的相關資訊。 | | |
| **宣告** | Long SKOOQuoteLib\_GetStockByNoLONG([in] BSTR bstrStockNo, [in,**out**] struct **SKFOREIGNLONG\*** pSKStock); | |
| **參數** | bstrStockNo | 海選商品代號，例如 CBOT,YM35000I1。 |
| pSKStock | SKCOM元件中的**[SKFOREIGNLONG](#_5-22_SKFOREIGNLONG_()**物件，將物件帶入此欄位中。 |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **備註** | ＊須使用[SKOOQuoteLib\_EnterMonitorLONG](#_SKOOQuoteLib_EnterMonitorLONG)登入，方可執行此函式。 | |

|  |
| --- |
| C#範例程式碼\_OOQuoteForm.cs |
| UI示意圖 |
| |  |  | | --- | --- | |  | 拉出 1個TextBox，命名(Name)為「textBoxSKOOQuoteLib\_GetStockByNoLONG」。  拉出 1個DataGridView，命名(Name)為「dataGridViewSKOOQuoteLib\_GetStockByNoLONG」。  拉出 1個Button，命名(Name)為「buttonSKOOQuoteLib\_GetStockByNoLONG」。 | |
| 全域變數 |
| // 宣告物件  SKCenterLib m\_pSKCenter = new SKCenterLib(); // 登入&環境設定物件  SKOOQuoteLib m\_pSKOOQuote = new SKOOQuoteLib(); // 海選報價物件 |

|  |
| --- |
| Constructor |
| // dataGridViewSKOOQuoteLib\_GetStockByNoLONG  {  dataGridViewSKOOQuoteLib\_GetStockByNoLONG.Columns.Add("Column1", "商品代號");  dataGridViewSKOOQuoteLib\_GetStockByNoLONG.Columns.Add("Column2", "商品名稱");  dataGridViewSKOOQuoteLib\_GetStockByNoLONG.Columns.Add("Column3", "報價小數位數");  dataGridViewSKOOQuoteLib\_GetStockByNoLONG.Columns.Add("Column4", "分母");  dataGridViewSKOOQuoteLib\_GetStockByNoLONG.Columns.Add("Column5", "市場代碼");  dataGridViewSKOOQuoteLib\_GetStockByNoLONG.Columns.Add("Column6", "交易所代號");  dataGridViewSKOOQuoteLib\_GetStockByNoLONG.Columns.Add("Column7", "交易所名稱");  dataGridViewSKOOQuoteLib\_GetStockByNoLONG.Columns.Add("Column8", "CallPut");  dataGridViewSKOOQuoteLib\_GetStockByNoLONG.Columns.Add("Column9", "開盤價");  dataGridViewSKOOQuoteLib\_GetStockByNoLONG.Columns.Add("Column10", "最高");  dataGridViewSKOOQuoteLib\_GetStockByNoLONG.Columns.Add("Column11", "最低");  dataGridViewSKOOQuoteLib\_GetStockByNoLONG.Columns.Add("Column12", "成交價");  dataGridViewSKOOQuoteLib\_GetStockByNoLONG.Columns.Add("Column13", "結算價");  dataGridViewSKOOQuoteLib\_GetStockByNoLONG.Columns.Add("Column14", "單量");  dataGridViewSKOOQuoteLib\_GetStockByNoLONG.Columns.Add("Column15", "昨收(參考價)");  dataGridViewSKOOQuoteLib\_GetStockByNoLONG.Columns.Add("Column16", "買價");  dataGridViewSKOOQuoteLib\_GetStockByNoLONG.Columns.Add("Column17", "買量");  dataGridViewSKOOQuoteLib\_GetStockByNoLONG.Columns.Add("Column18", "賣價");  dataGridViewSKOOQuoteLib\_GetStockByNoLONG.Columns.Add("Column19", "賣量");  dataGridViewSKOOQuoteLib\_GetStockByNoLONG.Columns.Add("Column20", "交易日");  } |

|  |
| --- |
| Current Block (可放在buttonSKOOQuoteLib\_RequestProducts\_Click的Click事件裡) |
| SKFOREIGNLONG pSKStock = new SKFOREIGNLONG();  string bstrStockNo = textBoxSKOOQuoteLib\_GetStockByNoLONG.Text;  // (LONG index)根據商品代號，取回海選報價的相關資訊。  int nCode = m\_pSKOOQuote.SKOOQuoteLib\_GetStockByNoLONG(bstrStockNo, ref pSKStock);  // 取得回傳訊息  string msg = "【SKOOQuoteLib\_GetStockByNoLONG】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode);  richTextBoxMethodMessage.AppendText(msg + "\n");  // SKFOREIGNLONG  bool found = false; // 找到就不新增資料  string[] values = bstrStockNo.Split(','); // 取出商品代號  foreach (DataGridViewRow row in dataGridViewSKOOQuoteLib\_GetStockByNoLONG.Rows)  {  if (row.IsNewRow)  continue;  if (row.Cells[0].Value != null)  {  if (values[1] == row.Cells[0].Value.ToString())// 搜尋過的商品代號  {  found = true;  break;  }  }  }  if (found == false)  dataGridViewSKOOQuoteLib\_GetStockByNoLONG.Rows.Add(pSKStock.bstrStockNo, pSKStock.bstrStockName, pSKStock.sDecimal, pSKStock.nDenominator, pSKStock.bstrMarketNo, pSKStock.bstrExchangeNo, pSKStock.bstrExchangeName, pSKStock.bstrCallPut, pSKStock.nOpen, pSKStock.nHigh, pSKStock.nLow, pSKStock.nClose, pSKStock.nSettlePrice, pSKStock.nTickQty, pSKStock.nRef, pSKStock.nBid, pSKStock.nBc, pSKStock.nAsk, pSKStock.nAc, pSKStock.nTradingDay); |

|  |
| --- |
| 執行成功畫面 |
|  |

## 即時報價

### SKOOQuoteLib\_RequestStocks

|  |  |  |
| --- | --- | --- |
| 訂閱指定商品即時報價  要求伺服器針對 bstrStockNos 內的商品代號做報價通知動作。  報價更新由[OnNotifyQuoteLONG](#_OnNotifyQuoteLONG)事件取得更通知。 | | |
| **宣告** | Long SKOOQuoteLib\_RequestStocks([in,out] SHORT\* psPageNo, [in] BSTR bstrStockNos); | |
| **參數** | psPageNo | psPageNo 每一次送出要求報價的動作都需要向報價伺服器指定一個特定的 Page 編號，伺服器會以編號當作Key值維護每一次送出不同的商品代號。當要變更某一 Page 索取報價的內容，即重複使用相同的 Page 編號，即可取消之前索取的內容，以新的內容取代。當psPageNo=-1時帶入，函式庫會指定一個新的編號，並回傳給呼叫端。 |
| bstrStockNos | 欲訂閱的海選商品代號，以{「交易所代碼」,「商品報價代碼」}為單位，多筆以”#”做區隔。 |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **備註** | 舉例說明：  psPageNo = -1 時，pStockNos = “CBOT,ZN12400X7#HKEx,HSI28000L7#CBOT,YM23500X7”，表示向伺服器索取這三檔股票的報價。如果是第一次執行，則 psPageNo 將會是回傳0，表示這次的索取是用Page 編號為 0 的 Page。  當想要取消這三檔的股票，改換其他股票時，則：  psPageNo = 0， pStockNos = “ bla bla…What you want” 帶入即可。  \*避免在[OnConnect](#_OnConnect) Event直接進行RequestStocks等報價訂閱，若各交易所商品未下載完成，即無法進行商品訂閱 | |

|  |
| --- |
| C#範例程式碼\_OOQuoteForm.cs |
| UI示意圖 |
| |  | | --- | |  | | 拉出 2個TextBox，命名(Name)為「textBoxpsPageNo2」、「textBoxStockNos」。  拉出 1個Button，命名(Name)為「buttonSKOOQuoteLib\_RequestStocks」。 | |
| 全域變數 |
| // 宣告物件  SKCenterLib m\_pSKCenter = new SKCenterLib(); // 登入&環境設定物件  SKOOQuoteLib m\_pSKOOQuote = new SKOOQuoteLib(); // 海選報價物件 |
| Current Block (可放在buttonSKOOQuoteLib\_RequestStocks\_Click的Click事件裡) |
| dataGridViewStocks.Rows.Clear();  short psPageNo = short.Parse(textBoxpsPageNo2.Text);  // 訂閱指定商品即時報價，要求伺服器針對 bstrStockNos 內的商品代號做報價通知動作。報價更新由OnNotifyQuote事件取得更通知。  int nCode = m\_pSKOOQuote.SKOOQuoteLib\_RequestStocks(psPageNo, textBoxStockNos.Text);  // 取得回傳訊息  string msg = "【SKOOQuoteLib\_RequestStocks】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode);  richTextBoxMethodMessage.AppendText(msg + "\n"); |
| 執行成功畫面 |
|  |

#### SKOOQuoteLib\_GetStockByIndexLONG

|  |  |  |
| --- | --- | --- |
| (LONG index)根據系統所編的索引代碼，取回海選報價的相關資訊。 | | |
| **宣告** | Long SKOOQuoteLib\_GetStockByIndexLONG([in] LONG nIndex, [in,**out**] struct **SKFOREIGNLONG\*** pSKStock); | |
|  | nIndex | 系統所編的海選商品索引代碼。 |
| pSKStock | SKCOM元件中的 [**SKFOREIGNLONG**](#_5-22_SKFOREIGNLONG_()物件，將物件帶入此欄位中。 |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **備註** | ＊商品物件價格未經小數處理。  ＊避免在[OnConnect](#_OnConnect) Event直接進行，若各交易所商品未下載完成，將無法取得商品基本資料。  ＊須使用[SKOOQuoteLib\_EnterMonitorLONG](#_SKOOQuoteLib_EnterMonitorLONG)登入，方可執行此函式。  一張含有 文字, 螢幕擷取畫面, 字型, 電子藍 的圖片  自動產生的描述 | |

## 十檔&五檔&成交明細

### SKOOQuoteLib\_RequestTicks

|  |  |  |
| --- | --- | --- |
| 訂閱與要求傳送成交明細、五檔、十檔。 | | |
| **宣告** | Long SKOOQuoteLib\_RequestTicks([in,out] SHORT\* psPageNo, [in] BSTR bstrStockNo); | |
| **參數** | psPageNo | 每一次送出要求成交明細的動作都需要向報價伺服器指定一個特定的 Page 編號。(請參考RequestStocks觀念說明)。 |
| bstrStockNo | 指定的商品，**一個Page僅能索取一檔**。格式「交易所代碼」,「商品報價代碼」。Ex：”CBOT,YM35000U1” |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **備註** | 即時Tick由[OnNotifyTicksLONG](#_OnNotifyTicksLONG)事件通知。  Tick回補由 [OnNotifyHistoryTicksLONG](#_OnNotifyHistoryTicksLONG)事件通知。  最佳五檔由[OnNotifyBest5LONG](#_OnNotifyBest5LONG)事件通知。  最佳十檔由[OnNotifyBest10](#_OnNotifyBest10LONG)LONG事件通知。  \*避免在[OnConnect](#_OnConnect) Event直接進行RequestTicks 等報價訂閱，若各交易所商品未下載完成，即無法進行商品訂閱 | |

|  |
| --- |
| C#範例程式碼\_OOQuoteForm.cs |
| UI示意圖 |
| |  | | --- | |  | | 拉出 2個TextBox，命名(Name)為「textBoxpsPageNo」、「textBoxpsPageNo」。  拉出 1個Button，命名(Name)為「buttonSKOOQuoteLib\_RequestTicks」。 | |
| 全域變數 |
| // 宣告物件  SKCenterLib m\_pSKCenter = new SKCenterLib(); // 登入&環境設定物件  SKOOQuoteLib m\_pSKOOQuote = new SKOOQuoteLib(); // 海選報價物件 |
| Current Block (可放在buttonSKOOQuoteLib\_RequestTicks\_Click的Click事件裡) |
| // 訂閱與要求傳送成交明細以及五檔  short psPageNo = short.Parse(textBoxpsPageNo.Text);  int nCode = m\_pSKOOQuote.SKOOQuoteLib\_RequestTicks(psPageNo, textBoxTicks.Text);  // 訂閱與要求傳送即時成交明細(本功能不會訂閱最佳五檔與最佳十檔、亦不包含歷史Ticks)  {  //int nCode = m\_pSKOOQuote.SKOOQuoteLib\_RequestLiveTick(psPageNo, textBoxTicks.Text);  }  // 取得回傳訊息  string msg = "【SKOOQuoteLib\_RequestTicks】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode);  richTextBoxMethodMessage.AppendText(msg + "\n"); |
| 執行成功畫面 |
|  |

#### SKOOQuoteLib\_RequestLiveTick

|  |  |  |
| --- | --- | --- |
| 訂閱與要求傳送**即時**成交明細。  (本功能不會訂閱最佳五檔與最佳十檔、亦不包含歷史Ticks) | | |
| **宣告** | Long SKOOQuoteLib\_RequestLiveTick([in,out] SHORT\* psPageNo, [in] BSTR bstrStockNo); | |
| **參數** | psPageNo | 每一次送出要求即時成交明細的動作都需要向報價伺服器指定一個特定的 Page 編號。(請參考[RequestStocks](#_SKOOQuoteLib_RequestStocks)觀念說明)。 |
| bstrStockNo | 索取的商品股票代號，**一個Page僅能索取一檔**。 |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **備註** | 即時Tick由[OnNotifyTicks](#_OnNotifyTicksLONG)LONG事件通知。  **不會**回補歷史成交明細資料，與[SKOOQuoteLib\_RequestTicks](#_SKOOQuoteLib_RequestTicks)不同，請擇一使用;因無回補歷史成交明細資料，[SKOOQuoteLib\_GetTickLONG](#_4-6-12_SKOOQuoteLib_RequestMarketDe)僅查閱，**訂閱後**之即時成交明細資料。 | |

#### SKOOQuoteLib\_GetTickLONG

|  |  |  |
| --- | --- | --- |
| (LONG index)取得成交明細資訊。(需先訂閱即時成交明細RequestTicks) | | |
| **宣告** | Long SKOOQuoteLib\_GetTickLONG([in] LONG nIndex, [in] LONG nPtr, [in,**out**] struct **SKFOREIGNTICK\*** pSKTick); | |
| **參數** | nIndex | 系統所編的海選商品索引代碼。 |
| nPtr | 第幾筆成交明細，又稱成交明細順序。可以根據此 Index 取得指定第幾筆成交明細資訊。 |
| pSKTick | SKCOM元件中的 [**SKFOREIGNTICK**](#_SKFOREIGNTICK(海外報價TICK物件))物件，將物件帶入此欄位中。 |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **備註** | 避免在[SKOOQuoteLib\_OnNotifyHistoryTicksLONG](#_OnNotifyHistoryTicksLONG)、[SKOOQuoteLib\_OnNotifyTicksLONG](#_OnNotifyTicksLONG)通知事件裡進行  ＊須使用[SKOOQuoteLib\_EnterMonitorLONG](#_SKOOQuoteLib_EnterMonitorLONG)登入，方可執行此函式。  一張含有 文字, 螢幕擷取畫面, 字型, 電子藍 的圖片  自動產生的描述 | |

#### SKOOQuoteLib\_GetBest5LONG

|  |  |  |
| --- | --- | --- |
| (LONG index)取得最佳五檔價格資訊。(需先訂閱最佳五檔RequestTicks) | | |
| **宣告** | Long SKOOQuoteLib\_GetBest5LONG([in] LONG nIndex, [in,**out**] struct **SKBEST5\*** pSKBest5); | |
| **參數** | nIndex | 系統所編的海選商品索引代碼。 |
| pSKBest5 | SKCOM元件中的 [**SKBEST5**](#_SKBEST5(五檔價格物件))物件，將物件帶入此欄位中。 |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **備註** | 五檔更新由[OnNotifyBest5LONG](#_OnNotifyBest5LONG)事件通知取得。  ＊須使用[SKOOQuoteLib\_EnterMonitorLONG](#_SKOOQuoteLib_EnterMonitorLONG)登入，方可執行此函式。 | |

### SKOOQuoteLib\_RequestMarketDepth

|  |  |  |
| --- | --- | --- |
| 要求傳送最佳十檔(包含最佳五檔，未訂閱歷史與即時成交明細)。 | | |
| **宣告** | Long SKOOQuoteLib\_RequestMarketDepth([in,out] SHORT\* psPageNo, [in] BSTR bstrStockNo); | |
| **參數** | psPageNo | 每一次送出要求成交明細的動作都需要向報價伺服器指定一個特定的 Page 編號。(請參考[RequestStocks](#_SKOOQuoteLib_RequestStocks)觀念說明)。 |
| bstrStockNo | 指定的商品，**一個Page僅能索取一檔**。格式「交易所代碼」,「商品報價代碼」。Ex：”CBOT,YM35000I1” |
| **回傳值** | 0表示成功，其餘非0數值都表示失敗。錯誤代碼可參考對照表。 | |
| **備註** | 最佳五檔由[OnNotifyBest5](#_OnNotifyBest5LONG)LONG事件通知。  最佳十檔由[OnNotifyBest10](#_OnNotifyBest10LONG)LONG事件通知。  \*避免在[OnConnect](#_OnConnect) Event直接進行RequestTicks等報價訂閱，若各交易所商品未下載完成，即無法進行商品訂閱 | |

|  |
| --- |
| C#範例程式碼\_OOQuoteForm.cs |
| UI示意圖 |
| |  | | --- | |  | | 拉出 1個Button，命名(Name)為「buttonSKOOQuoteLib\_RequestMarketDepth」。 | |
| 全域變數 |
| // 宣告物件  SKCenterLib m\_pSKCenter = new SKCenterLib(); // 登入&環境設定物件  SKOOQuoteLib m\_pSKOOQuote = new SKOOQuoteLib(); // 海選報價物件 |
| Current Block (可放在buttonSKOOQuoteLib\_RequestMarketDepth\_Click的Click事件裡) |
| // 訂閱與要求傳送最佳十檔(僅包含最佳十、五檔；不包含訂閱歷史與即時成交明細)  short psPageNo = short.Parse(textBoxpsPageNo.Text);  int nCode = m\_pSKOOQuote.SKOOQuoteLib\_RequestMarketDepth(psPageNo, textBoxTicks.Text);  // 取得回傳訊息  string msg = "【SKOOQuoteLib\_RequestMarketDepth】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode);  richTextBoxMethodMessage.AppendText(msg + "\n"); |
| 執行成功畫面 |
|  |

# 事件

## 連線

### OnConnect

|  |  |  |
| --- | --- | --- |
| 接收連線狀態。 | | |
| **宣告** | void OnConnect([in] LONG nCode, [in] LONG nSocketCode); | |
| **參數** | nCode | SK\_SUBJECT\_CONNECTION\_CONNECTED 表示為連線事件。  SK\_SUBJECT\_CONNECTION\_DISCONNECT 表示為斷線事件。  其他錯誤代碼，可參考6代碼定義表 |
| nSocketCode | 表示事件所獲得的代碼，代碼為0值即表示執行正確，  非0表示例外事件。 |
| **備註** | 避免在此處直接進行[SKOOQuoteLib\_EnterMonitorLONG](#_SKOOQuoteLib_EnterMonitorLONG)、[SKOOQuoteLib\_LeaveMonitor](#_SKOOQuoteLib_LeaveMonitor)、[RequestStocks](#_SKOOQuoteLib_RequestStocks) and [RequestTicks](#_SKOOQuoteLib_RequestTicks) 等報價訂閱，若各交易所商品未下載完成，即無法進行商品訂閱。 | |

|  |
| --- |
| C#範例程式碼\_OOQuoteForm.cs |
| UI示意圖 |
| |  |  | | --- | --- | | 一張含有 Rectangle, 白色, 螢幕擷取畫面, 文字 的圖片  自動產生的描述 | 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  SKOOQuoteLib m\_pSKOOQuote = new SKOOQuoteLib(); // 海選報價物件 |
| Current Block (可放在Form的Load事件裡) |
| //接收連線狀態  m\_pSKOOQuote.OnConnect += new \_ISKOOQuoteLibEvents\_OnConnectEventHandler(OnConnect);  void OnConnect(int nCode, int nSocketCode)  {  // 取得回傳訊息  string msg = "【OnConnect】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode);  richTextBoxMessage.AppendText(msg + "\n");  if (nSocketCode != 0)  {  // 取得回傳訊息  msg = "【OnConnect】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nSocketCode);  richTextBoxMessage.AppendText(msg + "\n");  }  } |
| 執行成功畫面 |
|  |

## 商品清單

### OnProducts

|  |  |  |
| --- | --- | --- |
| 海**選**報價商品清單資訊。 | | |
| **宣告** | void OnProducts([in] BSTR bstrValue); | |
| **參數** | bstrValue | 字串格式：[交易所代碼],[交易所名稱],[商品報價代碼],[商品名稱],[最後交易日] |
| **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。 | |

|  |
| --- |
| C#範例程式碼\_OOQuoteForm.cs |
| UI示意圖 |
| |  |  | | --- | --- | | 一張含有 Rectangle, 白色, 螢幕擷取畫面, 文字 的圖片  自動產生的描述 | 拉出1個RichTextBox，命名(Name)為「richTextBoxOnProducts」。 | |
| 全域變數 |
| // 宣告物件  SKCenterLib m\_pSKCenter = new SKCenterLib(); // 登入&環境設定物件  SKOOQuoteLib m\_pSKOOQuote = new SKOOQuoteLib(); // 海選報價物件 |
| Current Block (可放在Form的Load事件裡) |
| //海期報價商品清單資訊  m\_pSKOOQuote.OnProducts += new \_ISKOOQuoteLibEvents\_OnProductsEventHandler(OnProducts);  void OnProducts(string bstrValue)  {  //取得回傳訊息  string msg = bstrValue;  richTextBoxOnProducts.AppendText(msg + "\n");  } |

|  |
| --- |
| 執行成功畫面 |
|  |

## 即時報價

### OnNotifyQuoteLONG

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

|  |
| --- |
| C#範例程式碼\_OOQuoteForm.cs |
| UI示意圖 |
| |  |  | | --- | --- | |  | 拉出1個DataGridView，命名(Name)為「dataGridViewStocks」。 | |
| 全域變數 |
| // 宣告物件  SKCenterLib m\_pSKCenter = new SKCenterLib(); // 登入&環境設定物件  SKOOQuoteLib m\_pSKOOQuote = new SKOOQuoteLib(); // 海選報價物件 |

|  |
| --- |
| Constructor |
| // dataGridViewStocks  {  dataGridViewStocks.Columns.Add("Column1", "商品代號");  dataGridViewStocks.Columns.Add("Column2", "商品名稱");  dataGridViewStocks.Columns.Add("Column3", "報價小數位數");  dataGridViewStocks.Columns.Add("Column4", "分母");  dataGridViewStocks.Columns.Add("Column5", "市場代碼");  dataGridViewStocks.Columns.Add("Column6", "交易所代號");  dataGridViewStocks.Columns.Add("Column7", "交易所名稱");  dataGridViewStocks.Columns.Add("Column8", "CallPut");  dataGridViewStocks.Columns.Add("Column9", "開盤價");  dataGridViewStocks.Columns.Add("Column10", "最高");  dataGridViewStocks.Columns.Add("Column11", "最低");  dataGridViewStocks.Columns.Add("Column12", "成交價");  dataGridViewStocks.Columns.Add("Column13", "結算價");  dataGridViewStocks.Columns.Add("Column14", "單量");  dataGridViewStocks.Columns.Add("Column15", "昨收(參考價)");  dataGridViewStocks.Columns.Add("Column16", "買價");  dataGridViewStocks.Columns.Add("Column17", "買量");  dataGridViewStocks.Columns.Add("Column18", "賣價");  dataGridViewStocks.Columns.Add("Column19", "賣量");  dataGridViewStocks.Columns.Add("Column20", "交易日");  } |

|  |
| --- |
| Current Block (可放在Form的Load事件裡) |
| //(LONG index)當有索取的海選商品報價異動時，將透過此事件通知應用程式處理  m\_pSKOOQuote.OnNotifyQuoteLONG += new \_ISKOOQuoteLibEvents\_OnNotifyQuoteLONGEventHandler(OnNotifyQuoteLONG);  void OnNotifyQuoteLONG(int nIndex)  {  bool stockFound = false; // dataGridView找不到，就新增一筆  SKFOREIGNLONG pSKStock = new SKFOREIGNLONG();  // (LONG index)根據系統所編的索引代碼，取回海選報價的相關資訊  int nCode = m\_pSKOOQuote.SKOOQuoteLib\_GetStockByIndexLONG(nIndex, ref pSKStock);  // 取得回傳訊息  string msg = "【SKOOQuoteLib\_GetStockByIndexLONG】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode);  richTextBoxMethodMessage.AppendText(msg + "\n");  foreach (DataGridViewRow row in dataGridViewStocks.Rows)  {  if (row.Cells[0].Value.ToString() == pSKStock.bstrStockNo)  {  row.Cells[2].Value = pSKStock.sDecimal;  row.Cells[3].Value = pSKStock.nDenominator;  row.Cells[4].Value = pSKStock.bstrMarketNo;  row.Cells[5].Value = pSKStock.bstrExchangeNo;  row.Cells[6].Value = pSKStock.bstrExchangeName;  row.Cells[7].Value = pSKStock.bstrCallPut;  row.Cells[8].Value = pSKStock.nOpen;  row.Cells[9].Value = pSKStock.nHigh;  row.Cells[10].Value = pSKStock.nLow;  row.Cells[11].Value = pSKStock.nClose;  row.Cells[12].Value = pSKStock.nSettlePrice;  row.Cells[13].Value = pSKStock.nTickQty;  row.Cells[14].Value = pSKStock.nRef;  row.Cells[15].Value = pSKStock.nBid;  row.Cells[16].Value = pSKStock.nBc;  row.Cells[17].Value = pSKStock.nAsk;  row.Cells[18].Value = pSKStock.nAc;  row.Cells[19].Value = pSKStock.nTradingDay;  stockFound = true;  break;  }  }  if (stockFound == false)  {  dataGridViewStocks.Rows.Add(pSKStock.bstrStockNo, pSKStock.bstrStockName, pSKStock.sDecimal, pSKStock.nDenominator, pSKStock.bstrMarketNo, pSKStock.bstrExchangeNo, pSKStock.bstrExchangeName, pSKStock.bstrCallPut, pSKStock.nOpen, pSKStock.nHigh, pSKStock.nLow, pSKStock.nClose, pSKStock.nSettlePrice, pSKStock.nTickQty, pSKStock.nRef, pSKStock.nBid, pSKStock.nBc, pSKStock.nAsk, pSKStock.nAc, pSKStock.nTradingDay);  }  } |
| 執行成功畫面 |
|  |

## 十檔&五檔&成交明細

### OnNotifyTicksLONG

|  |  |  |
| --- | --- | --- |
| (LONG index)當有索取的個別海選商品成交明細有所異動，即透過向此註冊事件回傳所異動的個別海選商品成交明細。 | | |
| **宣告** | void OnNotifyTicksLONG([in] LONG nIndex, [in] LONG nPtr, [in] LONG nDate, [in] LONG nTime, [in] LONG nClose, [in] LONG nQty); | |
| **參數** | nIndex | 系統所編的海選商品索引代碼。 |
| nPtr | 表示資料的位址(Key) |
| nDate | 交易日期。(YYYYMMDD) |
| nTime | 時間。 |
| nClose | 成交價。 |
| nQty | 成交量。 |
| **備註** | 價格未進行小數點處理，開發人員需自行接手處理(FOREIGN物件內有小數點位數)。  避免在[SKOOQuoteLib\_OnNotifyHistoryTicksLONG](#_OnNotifyHistoryTicksLONG)、[SKOOQuoteLib\_OnNotifyTicksLONG](#_OnNotifyTicksLONG)通知事件裡進行[SKOOQuoteLib\_GetTickLONG](#_4-6-12_SKOOQuoteLib_RequestMarketDe)  ＊須使用[SKOOQuoteLib\_EnterMonitorLONG](#_SKOOQuoteLib_EnterMonitorLONG)登入，該事件才會被觸發。 | |

|  |
| --- |
| C#範例程式碼\_OOQuoteForm.cs |
| UI示意圖 |
| |  |  | | --- | --- | |  | 拉出1個DataGridView，命名(Name)為「dataGridViewOnNotifyTicksLONG」。 | |
| 全域變數 |
| // 宣告物件  SKCenterLib m\_pSKCenter = new SKCenterLib(); // 登入&環境設定物件  SKOOQuoteLib m\_pSKOOQuote = new SKOOQuoteLib(); // 海選報價物件 |
| Constructor |
| // OnNotifyTicksLONG  {  dataGridViewOnNotifyTicksLONG.Columns.Add("Column1", "交易日期");  dataGridViewOnNotifyTicksLONG.Columns.Add("Column2", "時：分：秒");  dataGridViewOnNotifyTicksLONG.Columns.Add("Column3", "成交價");  dataGridViewOnNotifyTicksLONG.Columns.Add("Column4", "成交量");  for (int i = 0; i < 1; i++)  dataGridViewOnNotifyTicksLONG.Rows.Add();  } |
| Current Block (可放在Form的Load事件裡) |
| //(LONG index)當有索取的個股成交明細有所異動，即透過向此註冊事件回傳所異動的個股成交明細  m\_pSKOOQuote.OnNotifyTicksLONG += new \_ISKOOQuoteLibEvents\_OnNotifyTicksLONGEventHandler(OnNotifyTicksLONG);  void OnNotifyTicksLONG(int nIndex, int nPtr, int nDate, int nTime, int nClose, int nQty)  {  SKFOREIGNTICK pSKTick = new SKFOREIGNTICK();  int nCode = m\_pSKOOQuote.SKOOQuoteLib\_GetTickLONG(nIndex, nPtr, ref pSKTick);  // 取得回傳訊息  string msg = "【SKOOQuoteLib\_GetTickLONG】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode);  richTextBoxMethodMessage.AppendText(msg + "\n");  if (dataGridViewOnNotifyTicksLONG.Rows.Count > 0)  {  DataGridViewRow Row1 = dataGridViewOnNotifyTicksLONG.Rows[0];  Row1.Cells[0].Value = pSKTick.nDate; // 交易日期。(YYYYMMDD)  Row1.Cells[1].Value = pSKTick.nTime; // 時間1。(時：分：秒)  Row1.Cells[2].Value = pSKTick.nClose / 100.0; // 成交價  Row1.Cells[3].Value = pSKTick.nQty / 100.0; // 成交量  }  } |
| 執行成功畫面 |
|  |

#### OnNotifyHistoryTicksLONG

|  |  |  |
| --- | --- | --- |
| (LONG index)當**首次**索取個別海選商品成交明細，此事件會**回補當天Tick**。 | | |
| **宣告** | void OnNotifyHistoryTicksLONG([in] LONG nIndex, [in] LONG nPtr, [in] LONG nDate, [in] LONG nTime, [in] LONG nClose, [in] LONG nQty); | |
| **參數** | nIndex | 系統所編的海選商品索引代碼。 |
| nPtr | 表示資料的位址(Key) |
| nDate | 交易日期。(YYYYMMDD) |
| nTime | 時間。 |
| nClose | 成交價。 |
| nQty | 成交量。 |
| **備註** | 價格未進行小數點處理，開發人員需自行接手處理(FOREIGNLONG物件內有小數點位數)。  避免在[SKOOQuoteLib\_OnNotifyHistoryTicksLONG](#_OnNotifyHistoryTicksLONG)、[SKOOQuoteLib\_OnNotifyTicksLONG](#_OnNotifyTicksLONG)通知事件裡進行[SKOOQuoteLib\_GetTickLONG](#_4-6-12_SKOOQuoteLib_RequestMarketDe)()  ＊須使用[SKOOQuoteLib\_EnterMonitorLONG](#_SKOOQuoteLib_EnterMonitorLONG)登入，該事件才會被觸發。 | |

### OnNotifyBest5LONG

|  |  |  |
| --- | --- | --- |
| (LONG index)當有索取的個別海選商品五檔價格有所異動，即透過向此註冊的事件進行處理。此函式會回傳所異動的個別海選五檔價格。 | | |
| **事件宣告** | void OnNotifyBest5LONG ([in] LONG nStockidx, [in] LONG nBestBid1, [in] LONG nBestBidQty1, [in] LONG nBestBid2, [in] LONG nBestBidQty2, [in] LONG nBestBid3, [in] LONG nBestBidQty3, [in] LONG nBestBid4, [in] LONG nBestBidQty4, [in] LONG nBestBid5, [in] LONG nBestBidQty5, [in] LONG nBestAsk1, [in] LONG nBestAskQty1 , [in] LONG nBestAsk2, [in] LONG nBestAskQty2, [in] LONG nBestAsk3, [in] LONG nBestAskQty3, [in] LONG nBestAsk4, [in] LONG nBestAskQty4, [in] LONG nBestAsk5, [in] LONG nBestAskQty5); | |
| **參數** | nStockidx | 系統所編的商品索引代碼。 |
| nBestBid1 | 第一檔買價。 |
| nBestBidQty1 | 第一檔買量。 |
| nBestBid2 | 第二檔買價。 |
| nBestBidQty2 | 第二檔買量。 |
| nBestBid3 | 第三檔買價。 |
| nBestBidQty3 | 第三檔買量。 |
| nBestBid4 | 第四檔買價。 |
| nBestBidQty4 | 第四檔買量。 |
| nBestBid5 | 第五檔買價。 |
| nBestBidQty5 | 第五檔買量。 |
| nBestAsk1 | 第一檔賣價。 |
| nBestAskQty1 | 第一檔賣量。 |
| nBestAsk2 | 第二檔賣價。 |
| nBestAskQty2 | 第二檔賣量。 |
| nBestAsk3 | 第三檔賣價。 |
| nBestAskQty3 | 第三檔賣量。 |
| nBestAsk4 | 第四檔賣價。 |
| nBestAskQty4 | 第四檔賣量。 |
| nBestAsk5 | 第五檔賣價。 |
| nBestAskQty5 | 第五檔賣量。 |
| **備註** | 價格未進行小數點處理，開發人員需自行接手處理(FOREIGNLONG物件內有小數點位數與分母)。[SKOOQuote\_RequestTicks](#_SKOOQuoteLib_RequestTicks) 相關通知事件。  ＊須使用[SKOOQuoteLib\_EnterMonitorLONG](#_SKOOQuoteLib_EnterMonitorLONG)登入，該事件才會被觸發。 | |

|  |
| --- |
| C#範例程式碼\_OOQuoteForm.cs |
| UI示意圖 |
| |  |  | | --- | --- | | 一張含有 螢幕擷取畫面, Rectangle, 收據, 文字 的圖片  自動產生的描述 | 拉出1個DataGridView，命名(Name)為「dataGridViewTicks」。 | |
| 全域變數 |
| // 宣告物件  SKCenterLib m\_pSKCenter = new SKCenterLib(); // 登入&環境設定物件  SKOOQuoteLib m\_pSKOOQuote = new SKOOQuoteLib(); // 海選報價物件 |
| Constructor |
| // Ticks  {  dataGridViewTicks.Columns.Add("Column1", "買量");  dataGridViewTicks.Columns.Add("Column2", "買價");  dataGridViewTicks.Columns.Add("Column3", "賣價");  dataGridViewTicks.Columns.Add("Column4", "賣量");  for (int i = 0; i < 5; i++)  dataGridViewTicks.Rows.Add();  } |
| Current Block (可放在Form的Load事件裡) |
| //事件最佳五檔(海選報價)  m\_pSKOOQuote.OnNotifyBest5LONG += new \_ISKOOQuoteLibEvents\_OnNotifyBest5LONGEventHandler(OnNotifyBest5LONG);  void OnNotifyBest5LONG(int nStockIdx, int nBestBid1, int nBestBidQty1, int nBestBid2, int nBestBidQty2, int nBestBid3, int nBestBidQty3, int nBestBid4, int nBestBidQty4, int nBestBid5, int nBestBidQty5, int nBestAsk1, int nBestAskQty1, int nBestAsk2, int nBestAskQty2, int nBestAsk3, int nBestAskQty3, int nBestAsk4, int nBestAskQty4, int nBestAsk5, int nBestAskQty5)  {  SKBEST5 pSKBest5 = new SKBEST5();  int nCode = m\_pSKOOQuote.SKOOQuoteLib\_GetBest5LONG(nStockIdx, ref pSKBest5);  // 取得回傳訊息  string msg = "【SKOOQuoteLib\_GetBest5LONG】" + m\_pSKCenter.SKCenterLib\_GetReturnCodeMessage(nCode);  richTextBoxMethodMessage.AppendText(msg + "\n");  if (dataGridViewTicks.Rows.Count > 0)  {  DataGridViewRow Row1 = dataGridViewTicks.Rows[0];  Row1.Cells[0].Value = pSKBest5.nBidQty1; //買量  Row1.Cells[1].Value = pSKBest5.nBid1 / 100.0; //買價  Row1.Cells[2].Value = pSKBest5.nAsk1 / 100.0; //賣價  Row1.Cells[3].Value = pSKBest5.nAskQty1; //賣量  DataGridViewRow Row2 = dataGridViewTicks.Rows[1];  Row2.Cells[0].Value = pSKBest5.nBidQty2; //買量  Row2.Cells[1].Value = pSKBest5.nBid2 / 100.0; //買價  Row2.Cells[2].Value = pSKBest5.nAsk2 / 100.0; //賣價  Row2.Cells[3].Value = pSKBest5.nAskQty2; //賣量  DataGridViewRow Row3 = dataGridViewTicks.Rows[2];  Row3.Cells[0].Value = pSKBest5.nBidQty3; //買量  Row3.Cells[1].Value = pSKBest5.nBid3 / 100.0; //買價  Row3.Cells[2].Value = pSKBest5.nAsk3 / 100.0; //賣價  Row3.Cells[3].Value = pSKBest5.nAskQty3; //賣量  DataGridViewRow Row4 = dataGridViewTicks.Rows[3];  Row4.Cells[0].Value = pSKBest5.nBidQty4; //買量  Row4.Cells[1].Value = pSKBest5.nBid4 / 100.0; //買價  Row4.Cells[2].Value = pSKBest5.nAsk4 / 100.0; //賣價  Row4.Cells[3].Value = pSKBest5.nAskQty4; //賣量  DataGridViewRow Row5 = dataGridViewTicks.Rows[4];  Row5.Cells[0].Value = pSKBest5.nBidQty5; //買量  Row5.Cells[1].Value = pSKBest5.nBid5 / 100.0; //買價  Row5.Cells[2].Value = pSKBest5.nAsk5 / 100.0; //賣價  Row5.Cells[3].Value = pSKBest5.nAskQty5; //賣量  }  } |

|  |
| --- |
| 執行成功畫面 |
|  |

### OnNotifyBest10LONG

|  |  |  |
| --- | --- | --- |
| (LONG index)當有索取的個別海選商品十檔價格有所異動，即透過向此註冊的事件進行處理。此函式會回傳所異動的個別海選十檔價格。 | | |
| **事件宣告** | void OnNotifyBest10LONG([in] LONG nStockidx, [in] LONG nBestBid1, [in] LONG nBestBidQty1, [in] LONG nBestBid2, [in] LONG nBestBidQty2, [in] LONG nBestBid3, [in] LONG nBestBidQty3, [in] LONG nBestBid4, [in] LONG nBestBidQty4, [in] LONG nBestBid5, [in] LONG nBestBidQty5, [in] LONG nBestBid6, [in] LONG nBestBidQty6, [in] LONG nBestBid7, [in] LONG nBestBidQty7, [in] LONG nBestBid8, [in] LONG nBestBidQty8, [in] LONG nBestBid9, [in] LONG nBestBidQty9, [in] LONG nBestBid10, [in] LONG nBestBidQty10, [in] LONG nBestAsk1, [in] LONG nBestAskQty1, [in] LONG nBestAsk2, [in] LONG nBestAskQty2, [in] LONG nBestAsk3, [in] LONG nBestAskQty3, [in] LONG nBestAsk4, [in] LONG nBestAskQty4, [in] LONG nBestAsk5, [in] LONG nBestAskQty5, [in] LONG nBestAsk6, [in] LONG nBestAskQty6, [in] LONG nBestAsk7, [in] LONG nBestAskQty7, [in] LONG nBestAsk8, [in] LONG nBestAskQty8, [in] LONG nBestAsk9, [in] LONG nBestAskQty9, [in] LONG nBestAsk10, [in] LONG nBestAskQty10); | |
| **參數** | nStockidx | 系統所編的商品索引代碼。 |
| nBestBid1 | 第一檔買價。 |
| nBestBidQty1 | 第一檔買量。 |
| nBestBid2 | 第二檔買價。 |
| nBestBidQty2 | 第二檔買量。 |
| nBestBid3 | 第三檔買價。 |
| nBestBidQty3 | 第三檔買量。 |
| nBestBid4 | 第四檔買價。 |
| nBestBidQty4 | 第四檔買量。 |
| nBestBid5 | 第五檔買價。 |
| nBestBidQty5 | 第五檔買量。 |
| nBestBid6 | 第六檔買價。 |
| nBestBidQty6 | 第六檔買量。 |
| nBestBid7 | 第七檔買價。 |
| nBestBidQty7 | 第七檔買量。 |
| nBestBid8 | 第八檔買價。 |
| nBestBidQty8 | 第八檔買量。 |
| nBestBid9 | 第九檔買價。 |
| nBestBidQty9 | 第九檔買量。 |
| nBestBid10 | 第十檔買價。 |
| nBestBidQty10 | 第十檔買量。 |
| nBestAsk1 | 第一檔賣價。 |
| nBestAskQty1 | 第一檔賣量。 |
| nBestAsk2 | 第二檔賣價。 |
| nBestAskQty2 | 第二檔賣量。 |
| nBestAsk3 | 第三檔賣價。 |
| nBestAskQty3 | 第三檔賣量。 |
| nBestAsk4 | 第四檔賣價。 |
| nBestAskQty4 | 第四檔賣量。 |
| nBestAsk5 | 第五檔賣價。 |
| nBestAskQty5 | 第五檔賣量。 |
| nBestAsk6 | 第六檔賣價。 |
| nBestAskQty6 | 第六檔賣量。 |
| nBestAsk7 | 第七檔賣價。 |
| nBestAskQty7 | 第七檔賣量。 |
| nBestAsk8 | 第八檔賣價。 |
| nBestAskQty8 | 第八檔賣量。 |
| nBestAsk9 | 第九檔賣價。 |
| nBestAskQty9 | 第九檔賣量。 |
| nBestAsk10 | 第十檔賣價。 |
| nBestAskQty10 | 第十檔賣量。 |
| **備註** | 價格未進行小數點處理，開發人員需自行接手處理(FOREIGNLONG物件內有小數點位數與分母)。[SKOOQuote\_RequestTicks](#_SKOOQuoteLib_RequestTicks) 相關通知事件。  ＊須使用[SKOOQuoteLib\_EnterMonitorLONG](#_SKOOQuoteLib_EnterMonitorLONG)登入，該事件才會被觸發。 | |

|  |
| --- |
| C#範例程式碼\_OOQuoteForm.cs |
| UI示意圖 |
| |  |  | | --- | --- | | 一張含有 螢幕擷取畫面, Rectangle, 收據, 文字 的圖片  自動產生的描述 | 拉出1個DataGridView，命名(Name)為「dataGridViewOnNotifyBest10NineDigitLONG」。 | |
| 全域變數 |
| // 宣告物件  SKCenterLib m\_pSKCenter = new SKCenterLib(); // 登入&環境設定物件  SKOOQuoteLib m\_pSKOOQuote = new SKOOQuoteLib(); // 海選報價物件 |
| Constructor |
| // dataGridViewOnNotifyBest10NineDigitLONG  {  dataGridViewOnNotifyBest10NineDigitLONG.Columns.Add("Column1", "買量");  dataGridViewOnNotifyBest10NineDigitLONG.Columns.Add("Column2", "買價");  dataGridViewOnNotifyBest10NineDigitLONG.Columns.Add("Column3", "賣價");  dataGridViewOnNotifyBest10NineDigitLONG.Columns.Add("Column4", "賣量");  for (int i = 0; i < 10; i++)  dataGridViewOnNotifyBest10NineDigitLONG.Rows.Add();  } |
| Current Block (可放在Form的Load事件裡) |
| //事件最佳十檔(海選報價)  m\_pSKOOQuote.OnNotifyBest10LONG += new \_ISKOOQuoteLibEvents\_OnNotifyBest10LONGEventHandler(OnNotifyBest10LONG);  void OnNotifyBest10LONG(int nStockIdx, int nBestBid1, int nBestBidQty1, int nBestBid2, int nBestBidQty2, int nBestBid3, int nBestBidQty3, int nBestBid4, int nBestBidQty4, int nBestBid5, int nBestBidQty5, int nBestBid6, int nBestBidQty6, int nBestBid7, int nBestBidQty7, int nBestBid8, int nBestBidQty8, int nBestBid9, int nBestBidQty9, int nBestBid10, int nBestBidQty10, int nBestAsk1, int nBestAskQty1, int nBestAsk2, int nBestAskQty2, int nBestAsk3, int nBestAskQty3, int nBestAsk4, int nBestAskQty4, int nBestAsk5, int nBestAskQty5, int nBestAsk6, int nBestAskQty6, int nBestAsk7, int nBestAskQty7, int nBestAsk8, int nBestAskQty8, int nBestAsk9, int nBestAskQty9, int nBestAsk10, int nBestAskQty10)  {  if (dataGridViewOnNotifyBest10NineDigitLONG.Rows.Count > 0)  {  DataGridViewRow Row1 = dataGridViewOnNotifyBest10NineDigitLONG.Rows[0];  Row1.Cells[0].Value = nBestBidQty1; //買量  Row1.Cells[1].Value = nBestBid1 / 100.0; //買價  Row1.Cells[2].Value = nBestAsk1 / 100.0; //賣價  Row1.Cells[3].Value = nBestAskQty1; //賣量  DataGridViewRow Row2 = dataGridViewOnNotifyBest10NineDigitLONG.Rows[1];  Row2.Cells[0].Value = nBestBidQty2; //買量  Row2.Cells[1].Value = nBestBid2 / 100.0; //買價  Row2.Cells[2].Value = nBestAsk2 / 100.0; //賣價  Row2.Cells[3].Value = nBestAskQty2; //賣量  DataGridViewRow Row3 = dataGridViewOnNotifyBest10NineDigitLONG.Rows[2];  Row3.Cells[0].Value = nBestBidQty3; //買量  Row3.Cells[1].Value = nBestBid3 / 100.0; //買價  Row3.Cells[2].Value = nBestAsk3 / 100.0; //賣價  Row3.Cells[3].Value = nBestAskQty3; //賣量  DataGridViewRow Row4 = dataGridViewOnNotifyBest10NineDigitLONG.Rows[3];  Row4.Cells[0].Value = nBestBidQty4; //買量  Row4.Cells[1].Value = nBestBid4 / 100.0; //買價  Row4.Cells[2].Value = nBestAsk4 / 100.0; //賣價  Row4.Cells[3].Value = nBestAskQty4; //賣量  DataGridViewRow Row5 = dataGridViewOnNotifyBest10NineDigitLONG.Rows[4];  Row5.Cells[0].Value = nBestBidQty5; //買量  Row5.Cells[1].Value = nBestBid5 / 100.0; //買價  Row5.Cells[2].Value = nBestAsk5 / 100.0; //賣價  Row5.Cells[3].Value = nBestAskQty5; //賣量  DataGridViewRow Row6 = dataGridViewOnNotifyBest10NineDigitLONG.Rows[5];  Row6.Cells[0].Value = nBestBidQty6; //買量  Row6.Cells[1].Value = nBestBid6 / 100.0; //買價  Row6.Cells[2].Value = nBestAsk6 / 100.0; //賣價  Row6.Cells[3].Value = nBestAskQty6; //賣量  DataGridViewRow Row7 = dataGridViewOnNotifyBest10NineDigitLONG.Rows[6];  Row7.Cells[0].Value = nBestBidQty7; //買量  Row7.Cells[1].Value = nBestBid7 / 100.0; //買價  Row7.Cells[2].Value = nBestAsk7 / 100.0; //賣價  Row7.Cells[3].Value = nBestAskQty7; //賣量  DataGridViewRow Row8 = dataGridViewOnNotifyBest10NineDigitLONG.Rows[7];  Row8.Cells[0].Value = nBestBidQty8; //買量  Row8.Cells[1].Value = nBestBid8 / 100.0; //買價  Row8.Cells[2].Value = nBestAsk8 / 100.0; //賣價  Row8.Cells[3].Value = nBestAskQty8; //賣量  DataGridViewRow Row9 = dataGridViewOnNotifyBest10NineDigitLONG.Rows[8];  Row9.Cells[0].Value = nBestBidQty9; //買量  Row9.Cells[1].Value = nBestBid9 / 100.0; //買價  Row9.Cells[2].Value = nBestAsk9 / 100.0; //賣價  Row9.Cells[3].Value = nBestAskQty9; //賣量  DataGridViewRow Row10 = dataGridViewOnNotifyBest10NineDigitLONG.Rows[9];  Row10.Cells[0].Value = nBestBidQty10; //買量  Row10.Cells[1].Value = nBestBid10 / 100.0; //買價  Row10.Cells[2].Value = nBestAsk10 / 100.0; //賣價  Row10.Cells[3].Value = nBestAskQty10; //賣量  }  } |
| 執行成功畫面 |
|  |

# Struct結構物件

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

struct SKFOREIGNLONG

{

LONG nStockidx; // 系統所編的海外商品索引代碼

SHORT sDecimal; // 報價小數位數

LONG nDenominator; // 分母

BSTR bstrMarketNo; // 市場代碼

BSTR bstrExchangeNo; // 交易所代號

BSTR bstrExchangeName; // 交易所名稱

BSTR bstrStockNo; // 商品代號

BSTR bstrStockName; // 商品名稱

BSTR bstrCallPut; // CallPut

LONG nOpen; // 開盤價

LONG nHigh; // 最高價

LONG nLow; // 最低價

LONG nClose; // 成交價

LONG nSettlePrice; // 結算價

LONG nTickQty; // 單量

LONG nRef; // 昨收、參考價

LONG nBid; // 買價

LONG nBc; // 買量

LONG nAsk; // 賣價

LONG nAc; // 賣量

LONG nTQty; // 成交量( EXCEL x86版本)

LONG nStrikePrice; // 履約價

LONG nTradingDay // 交易日(YYYYMMDD)

備註 :當日非交易日時, 資料為前一交易日

};

//以上價格欄，需自行處理小數，參考『報價小數位數 sDecimal』

## SKFOREIGNTICK(海外報價TICK物件)

struct SKFOREIGNTICK

{

LONG nPtr; //Index

LONG nTime; //時間

LONG nClose; //成交價,

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

LONG nQty; //成交量

LONG nDate; //成交日期YYYYMMDD，依交易所時區為主

};

## SKBEST5(五檔價格物件)

typedef struct BEST5

{

LONG nBid1; // 第一檔買價

LONG nBidQty1; // 第一檔買量

LONG nBid2; // 第二檔買價

LONG nBidQty2; // 第二檔買量

LONG nBid3; // 第三檔買價

LONG nBidQty3; // 第三檔買量

LONG nBid4; // 第四檔買價

LONG nBidQty4; // 第四檔買量

LONG nBid5; // 第五檔買價

LONG nBidQty5; // 第五檔買量

LONG nExtendBid; // 衍生一檔買價{海外無衍生一檔}

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

LONG nAsk1; // 第一檔賣價

LONG nAskQty1; // 第一檔賣量

LONG nAsk2; // 第二檔賣價

LONG nAskQty2; // 第二檔賣量

LONG nAsk3; // 第三檔賣價

LONG nAskQty3; // 第三檔賣量

LONG nAsk4; // 第四檔賣價

LONG nAskQty4; // 第四檔買量

LONG nAsk5; // 第五檔賣價

LONG nAskQty5; // 第五檔賣量

LONG nExtendAsk; // 衍生一檔賣價{海外無衍生一檔}

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

LONG nSimulate; // 揭示 0:一般 1:試算(試撮\*)

}

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