

策略王之下單與回報獨立 **API** 使用說明

版本	日期	異動說明	備註
2.08.10	2008/4/16	文件新建立	
2.08.14	2008/9/10	1.增加憑證 2.增加期貨委託 3.增加選擇權	
2.08.15	2008/9/23	函式名稱變動 1. Close => SKReplyLib_Close 2. Connect => SKReplyLib_Connect 3. IsConnected => SKReplyLib_IsConnected 4. 修改 SendStockOrder、SendFutureOrder、SendOptionOrder 中 int&nMessageBufferSize -> int* pnMessageBufferSize	
2.08.20	2008/11/28	修改 Include 說明中，函式呼叫方式修改為 __stdcall 的方式	
2.08.20	2008/12/3	增加非同步傳送委託的方式 1.SendStockOrderAsync 2.SendFutureOrderAsync 3.SendOptionOrderAsync	
2.08.20	2008/12/4	增加海外期貨委託、取消、改量、商品查詢 1. SendOverseaFutureOrder 2. OverseaCancelOrderBySeqNo 3. OverseaDecreaseOrderBySeqNo 4. GetOverseaProducts 4-1. GetOverseaCount	
2.08.28	2009/7/14	新增查詢期貨未平倉函式，此兩函式必須搭配使用。 1. RegisterOnOpenInterestCallBack 2. GetOpenInterest	

2.08.28	2009/7/14	<p>新增查詢可交易帳號，此兩函式必須搭配使用。</p> <ol style="list-style-type: none"> 1. RegisterOnAccountCallBack 2. GetUserAccount 	
2.08.28	2009/7/14	<p>修改以下三個函式，加傳「帳號」欄位</p> <ol style="list-style-type: none"> 1. CancelOrderByStockNo 2. DecreaseOrderBySeqNo 3. OverseaDecreaseOrderBySeqNo 	
		<p>新增以下三個回傳值定義</p> <pre>#define SK_ERROR_QUERY_IN_PROCESSING 18 #define SK_ERROR_LOGIN_INVALID 19 #define SK_ERROR_REGISTER_CALLBACK 20</pre>	
2.08.28	2009/7/14	<p>新增停損單委託與查詢功能</p> <ol style="list-style-type: none"> 1. SendFutureStopLoss 2. CancelFutureStopLoss 3. RegisterOnStopLossReportCallBack 4. GetStopLossReport 	
2.08.36	2009/10/19	<p>回報 API 新增</p> <ol style="list-style-type: none"> 1. SKReplyLib_ConnectByID 2. SKReplyLib_CloseByID 3. SKReplyLib_IsConnectedByID 	
2.08.36	2009/10/19	<p>修改 OnConnect、OnDisconnect Callback 函式定義</p> <pre>void (__stdcall* FOnConnect)(BSTR bstrID, int nErrorCode); void (__stdcall* FOnDisconnect)(BSTR bstrID, int nErrorCode);</pre>	
2.08.37	2009/11/03	<p>為區隔 Server 傳下的代碼部份產生衝突，因此修改常數定義值</p> <pre>#define SK_ERROR_ACCOUNT_NOT_EXIST 1001</pre>	

		#define SK_ERROR_ACCOUNT_MARKET_NOT_MATCH 1002 #define SK_ERROR_PERIOD_OUT_OF_RANGE 1003 #define SK_ERROR_FLAG_OUT_OF_RANGE 1004 #define SK_ERROR_BUYSELL_OUT_OF_RANGE 1005 #define SK_ERROR_ORDER_SERVER_INVALID 1006 #define SK_ERROR_PERMISSION_DENIED 1007 #define SK_ERROR_TRADE_TYPR_OUT_OF_RANGE 1008 #define SK_ERROR_DAY_TRADE_OUT_OF_RANGE 1009 #define SK_ERROR_ORDER_SIGN_INVALID 1010 #define SK_ERROR_NEW_CLOSE_OUT_OF_RANGE 1011 #define SK_ERROR_PRODUCT_INVALID 1012 #define SK_ERROR_QTY_INVALID 1013 #define SK_ERROR_DAYTRADE_DENIED 1014 #define SK_ERROR_SPCIAL_TRADE_TYPE_INVALID 1015 #define SK_ERROR_PRICE_INVALID 1016 #define SK_ERROR_INDEX_OUT_OF_RANGE 1017 #define SK_ERROR_QUERY_IN_PROCESSING 1018 #define SK_ERROR_LOGIN_INVALID 1019 #define SK_ERROR_REGISTER_CALLBACK 1020	
2.1.1	2009/11/23	API 序號重新規劃	
		(1) 新增期貨移動停損下單功能 SendMovingStopLoss (2) 新增選擇權停損單下單功能 SendOptionStopLoss (3) 新增取消移動停損單功能 CancelMovingStopLoss (4) 新增取消選擇權停損單功能 CancelOptionStopLoss	
		新增以下回傳值定義	

		#define SK_ERROR_FUNCTION_PERMISSION_DENIED 1021 #define SK_ERROR_MARKET_OUT_OF_RANGE 1022	
		修改停損單查詢函式以及回傳內容 GetStopLossReport 新增篩選停損單條件，停損單/移動停損/選擇權停損單(STP/MST/OST) RegisterOnStopLossReportCallBack 查詢回傳內容調整	
2.1.2	2009/12/31	新增查詢證券即時庫存函式，此兩函式必須搭配使用。 1. RegisterOnRealBalanceReportCallBack 2. GetRealBalanceReport	
2.1.4	2010/6/29	修正登入問題	
		新增以下回傳定義，當一段時間沒有委託，則會將 API 使用權限暫時關閉，請重新洽詢營業員協助。 #define SK_ERROR_PERMISSION_TIMEOUT 1023	
		修正 SendOptionStopLoss 函式說明，原 usDayTrade 當沖記號，應修正為 usNewClose 新平倉記號	
2.1.5	2010/8/31	當呼叫 SKOrderLib_Initialize 初始時，海期商品資訊下載失敗，回傳以下代碼： #define SK_WARNING_OF_COM_DATA_MISSING 2001 當發生此代碼錯誤，如果不交易海期商品，還是可以交易國內商品。	
2.1.6	2010/9/14	新增美股下單功能 1. 送出美股下單委託 SendForeignStockOrder 2. 送出非同步美股下單委託 SendForeignStockOrderAsync 3. 利用書號取消美股下單委託	

		CancelForeignStockOrderByBookNo 4. 利用委託序號取消美股下單委託 CancelForeignStockOrderBySeqNo 新增以下回傳定義， SK_ERROR_FOREIGNSTOCK_PRICE_OUT_OF_RANGE 1024 SK_ERROR_FOREIGNSTOCK_UNDEFINE_COINTYPE 1025 SK_ERROR_FOREIGNSTOCK_SAME_COINSTYPE 1026 SK_ERROR_FOREIGNSTOCK_SALE_SHOULD_ORIGINAL_COIN 1027 SK_ERROR_FOREIGNSTOCK_TRADE_UNIT_INVALID 1028 SK_ERROR_FOREIGNSTOCK_STOCKNO_INVALID 1029 SK_ERROR_FOREIGNSTOCK_ACCOUNTTYPE_INVALID 1030 SK_ERROR_FOREIGNSTOCK_INITIALIZE_FAIL 1031	
2.1.7	2010/10/20	新增 Trade Station (Omega Research ProSuite 2000i) 訊號接收函式 1. 初始化 TradeStation 串接 SKOrderLib_InitializeTS 2. 註冊訊號 CallBack 函式 RegisterOnTSStrategyCallBack 新增以下回傳定義： #define SK_ERROR_TS_INITIALIZE_FAIL 1032 #define SK_ERROR_OVERSEA_TRADE_PRODUCT_FAIL 1033 #define SK_ERROR_OVERSEA_TRADE_DATA_NOT_COMPLETE 1034	
2.1.8	2010/11/01	修改 Trade Station (Omega Research ProSuite 2000i) 訊號接收函式 原 RegisterOnTSStrategyCallBack 改名為 RegisterOnFilledOrderCallBack 新增 Trade Station (Omega Research ProSuite 2000i) 訊號接收函式	

		1. RegisterOnTSActiveOrderCallBack 2. RegisterOnTSCanceledOrderCallBack	
		新增以下回傳定義 #define SK_WARNING_TS_READY 2002 當重複呼叫一個已經成功的TS 連線時，即會回傳此以代碼之警示，表示已經完成初始。	
2.1.9	2011/03/05	新增以下回傳定義 #define SK_WARNING_LOGIN_ALREADY 2003 當已經完成 SKOrderLib_Initialize 呼叫，身份認證成功後，當再重新呼叫 SKOrderLib_Initialize 即會得到此一回傳值，表示無須重複執行認證。	
		修改 GetOpenInterest 與 GetRealBalanceReport CallBack 內容。當查詢完成將會回傳以「##」開頭的一筆資料，表示查詢結果已經完成。	
		新增海期未平倉查詢功能 1. GetOverseaFutureOpenInterest 2. RegisterOnOverseaFutureOpenInterestCallBall	
2.1.10	2011/05/05	新增 港股下單，刪單功能 1. 送出港股下單委託(同美股下單函式) SendForeignStockOrder 2. 送出非同步港股下單委託(同美股下單函式) SendForeignStockOrderAsync 3. 利用書號取消港股下單委託(同美股刪單函式) CancelForeignStockOrderByBookNo 4. 利用委託序號取消港股下單委託(同美股刪單函式) CancelForeignStockOrderBySeqNo	

1、委託 SKOrderLib.dll 使用説明

1.1 confing.ini 説明

1.2 Export functions

1.1 config.ini

Config.ini 主要設定項目如下：

[version]

id=2.08.28

[Servers]

0=order2.capital.com.tw

1=order3.capital.com.tw

[debug]

enable=1

[order]

file=c:\order.log

[reply]

file=c:\reply.log

[quote]

file=c:\quote.log

[version] 區段已經不使用。

[Servers] 區段主要定義驗證伺服器網路位址，如果有多台可供驗證的伺服器，可以將編號由 0 依序遞增，當其中一台無法連線，將會往下一台繼續做驗證的工作。

[debug] 區段主要定義是否開啓 Log 的功能，當 enable 設定爲 1，則開啓紀錄功能。

[order] file 指定下單所要紀錄檔案的位置，用法爲 file=目錄+檔名(c:\order.log)

[reply] file 指定回報所要紀錄檔案的位置，用法爲 file=目錄+檔名(c:\reply.log)

[quote] file 指定報價所要紀錄檔案的位置，用法爲 file=目錄+檔名(c:\quote.log)

此檔案必須存放在與程式相同的目錄下方可正確解析，如果呼叫初始函式回傳 **SK_ERROR_INITIALIZE_FAIL (-1)** 請確認檔案所在的位置是否正確，或是提供的伺服器無法連線，導致初始化失敗。

1.2 Export functions

int SKOrderLib_Initialize([in]const char* lpszUser, [in]const char* lpszPassword)

說明：

在使用此 Library 前必須先通過使用者的身份認證，方可使用。此外，使用者必須先填寫申請開放使用外部 API 的申請書後才能通過初始程序，否則將會得到 SK_ERROR_PERMISSION_DENIED 的錯誤代碼。

當太久沒有使用，將會得到一個 SK_ERROR_PERMISSION_TIMEOUT 的錯誤代碼。請重新洽詢營業員申請開放。

傳入值

lpszUser 使用者代號

lpszPassword 密碼

回傳值：

0 表示初始化成功

SK_ERROR_INITIALIZE_FAIL，初始化失敗，可能是因為驗證伺服器未指定或無法與驗證伺服器做連線，所發生的錯誤

SK_ERROR_PERMISSION_DENIED，初始化的使用者沒有使用外部 API 的權限。

其他錯誤代碼，1~199 間的代碼為伺服器回傳的錯誤，請檢查 log 檔內將有錯誤原因的說明。

SK_ERROR_PERMISSION_TIMEOUT，太久沒有使用 API 下單且必須有成交時，將會被限制使用，如有這種情況，請重新洽詢營業員申請。

SK_WARNING_OF_COM_DATA_MISSING，海期商品資訊下載失敗，當發生此代碼錯誤，如果只交易國內商品，還是可以視為登入成功。

int SKOrderLib_ReadCertByID([in]char* lpszID)

說明：

讀取憑證資訊。委託下單必須透過憑證，因此當元件初始化成功後即需要做讀取憑證的動作，如果使用群組的帳號做初始，則必須自行將所有的帳號依序做讀取憑證的動作。如果送出委託前未經讀取憑證，送委託會得到 **SK_ERROR_ORDER_SIGN_INVALID** 的錯誤。

憑證需要搭配另一個元件，檔名為 **FCliPKILIB.dll**，將檔案存放在相同目錄下即可。

傳入值：

lpszID 使用者 ID，所交易帳號開戶時所使用的 ID，一般使用者即為身份證字號。

回傳值：

SK_SUCCESS 表示讀取成功。

SK_FAIL 表示失敗。如果有開啓錯誤紀錄檔，可以檢查錯誤內容說明。

```
int SendStockOrder([in]const TCHAR* lpszAccount, [in]const TCHAR* lpszStockNo,  
[in]unsigned short usPeriod, [in]unsigned short usFlag, [in] unsigned short usBuySell, [in]  
const TCHAR* lpszPrice, [in]int nQty, [in,out]TCHAR* lpszMessage, [in,out] int*  
pnMessageBufferSize)
```

說明：

送出證券委託

傳入值：

lpszAccount 證券帳號，分公司四碼＋帳號 7 碼

lpszStockNo 委託股票代號

usPeriod 0:盤中 1:盤後 2:零股

usFlag 0:現股 1:融資 2:融券

usBuySell 0:買進 1:賣出

lpszPrice 委託價格，「M」表示參考價（昨收價）

nQty 整股交易為張數，如果是零股則為股數

lpszMessage 回覆訊息，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

pnMessageBufferSize lpszMessage 的 Buffer 寬度，如果傳入的 Buffer 寬度不足，則訊息內容將會被截斷以符合 Buffer 的大小，回傳會回覆真正填入 Buffer 的訊息長度。

回傳值：

0 表示委託伺服器接收成功，5 表示預約單，請務必查詢委託回報確認真正的委託狀態。
其他數值表示委託異常，詳細原因可以參考回傳的代碼，或是參考 Log 內容說明。

SK_ERROR_INITIALIZE_FAIL 尚未完成 SKOrderLib_Initialize 初始動作。

SK_ERROR_ACCOUNT_NOT_EXIST 帶入的帳號是無效的，請檢查 lpseAccount 帶入的值。

SK_ERROR_ACCOUNT_MARKET_NOT_MATCH 所帶入的帳號與交易的市場別不符合，有可能是委託期貨商品卻帶成股票帳號。

SK_ERROR_PERIOD_OUT_OF_RANGE 所帶入的交易區間有問題，必須是 0 盤中整股、1 盤後、2 零股。

SK_ERROR_BUYSELL_OUT_OF_RANGE 帶入的買賣別錯誤，必須是 0:買進 1:賣出

SK_ERROR_FLAG_OUT_OF_RANGE 帶入的交易種類錯誤，必須是 0:現股 1:融資 2:融券。

SK_ERROR_ORDER_SIGN_INVALID 委託單加簽失敗，請確認憑證安裝是否正確，或已經完成 ReadCert 的動作。

```
int SendFutureOrder( [in]const TCHAR* lpszAccount, [in]const TCHAR* lpszStockNo,  
[in]unsigned short usTradeType, [in]unsigned short usDayTrade, [in]unsigned short usBuySell,  
[in]const TCHAR* lpszPrice, [in]int nQty, [in,out] TCHAR* lpszMessage, [in,out] int*  
pnMessageBufferSize)
```

說明：

送出期貨委託。

傳入值：

lpszAccount 期貨帳號，IB + 帳號

lpszStockNo 委託期貨代號

usTradeType 0:ROD 1:IOC 2:FOK

usDayTrade 當沖 0:否 1:是，可當沖商品請參考交易所規定。

usBuySell 買賣別，0:買進 1:賣出

lpszPrice 委託價格，「M」表示市價

nQty 交易口數

lpszMessage 回覆訊息，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

pnMessageBufferSize lpszMessage 的 Buffer 寬度，如果傳入的 Buffer 寬度不足，則訊息內容將會被截斷以符合 Buffer 的大小，回傳會回覆真正填入 Buffer 的訊息長度。

回傳值：

0 表示委託伺服器接收成功，5 表示預約單，請務必查詢委託回報確認真正的委託狀態。

其他數值表示委託異常，詳細原因可以參考回傳的代碼，或是參考 Log 內容說明。

SK_ERROR_INITIALIZE_FAIL 尚未完成 SKOrderLib_Initialize 初始動作。

SK_ERROR_ACCOUNT_NOT_EXIST 帶入的帳號是無效的，請檢查 lpseAccount 帶入的值。

SK_ERROR_ACCOUNT_MARKET_NOT_MATCH 所帶入的帳號與交易的市場別不符合，有可能是委託期貨商品卻帶成股票帳號。

SK_ERROR_TRADE_TYPR_OUT_OF_RANG 所帶入的委託單有效區間有問題，必須是 0:ROD 1:IOC 2:FOK。

SK_ERROR_BUYSELL_OUT_OF_RANGE 帶入的買賣別錯誤，必須是 0:買進 1:賣出。

SK_ERROR_DAY_TRADE_OUT_OF_RANGE 帶入的當沖錯誤，必須是 0:否 1:是。

SK_ERROR_ORDER_SIGN_INVALID 委託單加簽失敗，請確認憑證安裝是否正確，或已經完成 ReadCert 的動作。

```
int SendOptionOrder( [in]const TCHAR* lpszAccount, [in]const TCHAR* lpszStockNo,  
[in]unsigned short usTradeType,[in]unsigned short usNewClose, [in]unsigned short usBuySell,  
[in]const TCHAR* lpszPrice, [in]int nQty, [in,out]TCHAR* lpszMessage,[in,out] int*  
pnMessageBufferSize)
```

說明：

送出選擇權委託。

傳入值：

lpszAccount 期貨帳號，IB + 帳號

lpszStockNo 委託選擇權代號

usTradeType 0:ROD 1:IOC 2:FOK

usNewClose 新平倉，0:新倉 1:平倉 2:自動

usBuySell 買賣別，0:買進 1:賣出

lpszPrice 委託價格，「M」表示市價

nQty 交易口數

lpszMessage 回覆訊息，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

pnMessageBufferSize lpszMessage 的 Buffer 寬度，如果傳入的 Buffer 寬度不足，則訊息內容將會被截斷以符合 Buffer 的大小，回傳會回覆真正填入 Buffer 的訊息長度。

回傳值：

0 表示委託伺服器接收成功，5 表示預約單，請務必查詢委託回報確認真正的委託狀態。

其他數值表示委託異常，詳細原因可以參考回傳的訊息，或是參考 Log 內容說明。

SK_ERROR_INITIALIZE_FAIL 尚未完成 SKOrderLib_Initialize 初始動作。

SK_ERROR_ACCOUNT_NOT_EXIST 帶入的帳號是無效的，請檢查 lpseAccount 帶入的值。

SK_ERROR_ACCOUNT_MARKET_NOT_MATCH 所帶入的帳號與交易的市場別不符合，有可能是委託期貨商品卻帶成股票帳號。

SK_ERROR_TRADE_TYPR_OUT_OF_RANG 所帶入的委託單有效區間有問題，必須是 0:ROD 1:IOC 2:FOK。

SK_ERROR_NEW_CLOSE_OUT_OF_RANGE 新平倉錯誤，必須是 0:新倉 1:平倉 2:自動。

SK_ERROR_BUYSELL_OUT_OF_RANGE 帶入的買賣別錯誤，必須是 0:買進 1:賣出。

SK_ERROR_ORDER_SIGN_INVALID 委託單加簽失敗，請確認憑證安裝是否正確，或已經完成 ReadCert 的動作。

int CancelOrderBySeqNo([in]const TCHAR* lpzAccount, [in]const TCHAR* lpzSeqNo)

說明：

根據委託序號取消委託。

傳入值：

lpzAccount 委託的帳號

lpzSeqNo 委託序號 (長度為 13 碼)

回傳值：

0 表示委託伺服器接收取消委託成功，詳細委託狀態仍須以委託回報為主，所以務必查詢委託回報確認最後委託狀態。

4 表示取消失敗

其他非 0 數值皆表示異常，請查閱 Log 內容詳細說明

int CancelOrderByStockNo([in]const TCHAR* lpszAccount, [in]const TCHAR* lpszStockNo)

說明：

根據股票代號取消委託，換句話說，就是相同股票代號的委託同時會被取消。如果 lpszStockNo 填入為空值 (NULL)，則表示將取消帳號下的全部委託，包括證券委託、期貨委託、選擇權委託等，目前在市場上所有的有效委託，因此使用上請務必小心。

傳入值：

lpazAccount 委託的帳號

lpszStockNo 欲取消的股票代號

回傳值：

這裡的回傳值為 4 表示委託伺服器已經收到，詳細取消狀態請查詢委託回報內容為準。

```
int DecreaseOrderBySeqNo([in]const TCHAR* lpzAccount , [in]const TCHAR* lpzSeqNo,  
[in]int nDecreaseQty)
```

說明：

委託減量

傳入值：

lpazAccount 委託的帳號

lpzSeqNo 欲改量的委託序號

nDecreaseQty 欲減少的數量

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

海外期貨傳送委託

請注意：由於交易商品眾多與代號熟悉度偏低，請於委託下單前先行獲取商品資訊，以確保委託順利進行。

獲取商品資訊方式：

1. 呼叫 `GetOverseaCount` 取得商品總數
2. 根據商品總數呼叫 `GetOverseaProducts` 傳入 Index 和商品物件，此 function 及自動將商品資訊填入對應項目
3. 呼叫 `GetOverseaFutures` 搭配 `RegisterOnOverseaFuturesCallBack`，以字串的方式接收可交易商品資料。

`int GetOverseaCount ()`

說明：

取得海外可委託商品數量

回傳值：

-4 表示尚未完成 `SKOrderLib_Initialize` 初始動作。

-2 表示無海外期貨帳戶

0~n 表示成功，回傳值為商品數量

```
int GetOverseaProducts([in] int index, [in,out]struct SOfComProduct* product);
```

說明：

取得海外可委託商品內容，搭配 GetOverseaCount 取得 index 界限取得商品

struct SOfComProduct

```
{  
    char m_strExchange[10]; ///< 交易所代號  
    char m_strProductNo[10]; ///< 商品代號  
    char m_strYearMonth[10]; ///< 商品年月  
    unsigned short m_sSpecialTradeType; ///< 特殊條件  
    double m_dMinJump; ///< 跳動點  
    unsigned short m_nDenominator; ///< 分母  
    unsigned short m_sDayTrade; ///< 是否可當沖  
};
```

回傳值：

SK_ERROR_INDEX_OUT_OF_RANGE 表示 index 錯誤

0:正確

int RegisterOnOverseaFuturesCallBack (long lCallbackFunction)

說明：

註冊一個 Callback function，當呼叫 GetOverseaFutures，即會透過此函式回傳海期商品資訊。

函式宣告方式如下：

Void __stdcall OnOverseaFuture(BSTR bstrData);

回傳得字串內容，以分號(;)分隔每一個欄位，欄位依序為：

交易所代碼;交易所名稱;商品代碼;商品名稱;年月;跳動點;分母;可接受交易種類

傳入值：

lCallbackFunction 將 Callback 函式位址轉成 long 的表示方式，傳入 Library 內，完成註冊程序。

回傳值：

SK_SUCCESS 表示函式呼叫成功。

SK_ERROR_INITIALIZE_FAIL 尚未完成 SKOrderLib_Initialize 初始動作。

int GetOverseaFutures()

說明：

取回目前海外期貨可交易商品資訊。此函式需配合 RegisterOnOverseaFuturesCallBack 函式，先註冊一個 Call back 函式，當呼叫 GetOverseaFutures，即會透過此 call back 函式，將商品資訊以字串回傳，字串中以分號(;) 分隔每一個欄位，欄位依序為：

交易所代碼;交易所名稱;商品代碼;商品名稱;年月;跳動點;分母;可接受交易種類

傳入值：

無

回傳值：

SK_SUCCESS 表示函式呼叫成功。

SK_ERROR_INITIALIZE_FAIL 尚未完成 SKOrderLib_Initialize 初始動作。

SK_ERROR_REGISTER_CALLBACK 尚未註冊 Call back 函式。

SK_ERROR_OVERSEA_TRADE_DATA_NOT_COMPLETE 海期商品資訊未成功下載。

```
int SendOverseaFutureOrder( [in] const TCHAR* lpszAccount, [in]const TCHAR*
lpszTradeName, [in] const TCHAR* lpszStockNo, [in] const TCHAR* lpszYearMonth,
[in]unsigned short usBuySell, [in] unsigned short usNewClose, [in] unsigned short usDayTrade,
[in]unsigned short usTradeType, [in] unsigned short usSpecialTradeType, [in] int nQty,
[in]const TCHAR* lpszOrder, [in] const TCHAR* lpszOrderNumerator, [in] const TCHAR*
lpszTrigger, [in] const TCHAR* lpszTriggerNumerator, [in,out] TCHAR* lpszMessage, [in,out]
int *pnMessageBufferSize)
```

說明：

- 1.送出海外期貨委託。
- 2.輸入之價格依據 usSpecialTradeType 自動忽略不必要價格參數・例:LMT 時，不論觸發價格欄位輸入自動代 0(即無須傳入參數，即使傳入亦被設定為 0)
- 3.目前 CBOT 交易所之 STL 限制觸發價格需和委託價格相同，請注意委託時之參數

傳入值：

lpszAccount 海外期貨帳號，IB + 帳號

lpszTradeName 交易所代號

lpszStockNo 委託期貨代號

lpszYearMonth 委託商品年月(YYYYMM)6 碼

usBuySell 買賣別，0:買進 1:賣出

usNewClose 新平倉，0:新倉 1:平倉 2:自動(目前委託固定傳送新倉，自動忽略傳入值)

usDayTrade 是否當沖，0:否 1:是

usTradeType 0:ROD 1:IOC 2:FOK(目前委託固定傳送 ROD，自動忽略傳入值)

usSpecialTradeType 0:LMT 1:MKT 2:STL3:STP

lpszOrder 委託價格

lpszOrderNumerator 委託價格分子

lpszTrigger 觸發價格

lpszTriggerNumerator 觸發價格分子

nQty 交易口數

lpszMessage 回覆訊息，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

pnMessageBufferSize lpszMessage 的 Buffer 寬度，如果傳入的 Buffer 寬度不足，則訊息內容將會被截斷以符合 Buffer 的大小，回傳會回覆真正填入 Buffer 的訊息長度。

回傳值：

0 表示委託伺服器接收成功，請務必查詢委託回報確認真正的委託狀態。

其他非 0 數值表示委託異常，詳細原因可以參考回傳的訊息，或是參考 Log 內容說明。


```
int OverseaCancelOrderBySeqNo([in]const TCHAR* lpzAccount, [in]const TCHAR*  
lpzSeqNo)
```

說明：

根據委託序號取消海外委託。

傳入值：

lpzAccount 委託的帳號

lpzSeqNo 委託序號 (長度為 13 碼)

回傳值：

0 表示委託伺服器接取消委託成功，詳細委託狀態仍須以委託回報為主，所以務必查詢委託回報確認最後委託狀態。

4 表示取消失敗

其他非 0 數值皆表示異常，請查閱 Log 內容詳細說明

int OverseaDecreaseOrderBySeqNo([in]const TCHAR* lpszAccount, [in]const TCHAR* lpszSeqNo, [in]int nDecreaseQty)

說明：

海外委託減量

傳入值：

lpszAccount 帳號

lpszSeqNo 欲改量的委託序號

nDecreaseQty 欲減少的數量

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

非同步傳送委託

請注意：以非同步傳送委託將不會等待收單主機的回應，因此無法從所呼叫函式回應的代碼判斷委託是否成功，所有委託的狀態將會紀錄在設定檔中，`[order]` 區段所指定的路徑與檔案，請參閱 `1.1 config.ini` 的設定說明。

目前有

`SendStockOrderAsync`

`SendFutureOrderAsync`

`SendOptionOrderAsync`

`SendForeignStockOrderAsync`

```
int SendStockOrderAsync([in]const TCHAR* lpszAccount, [in]const TCHAR* lpszStockNo,  
[in]unsigned short usPeriod, [in]unsigned short usFlag, [in] unsigned short usBuySell, [in]  
const TCHAR* lpszPrice, [in]int nQty)
```

說明：

送出證券委託。(非同步傳送)

傳入值：

lpszAccount	證券帳號，分公司四碼＋帳號 7 碼
lpszStockNo	委託股票代號
usPeriod	0:盤中 1:盤後 2:零股
usFlag	0:現股 1:融資 2:融券
usBuySell	0:買進 1:賣出
lpszPrice	委託價格，「M」表示參考價（昨收價）
nQty	整股交易為張數，如果是零股則為股數

回傳值：

SK_SUCCESS 表示函式呼叫成功，詳細委託狀態，請參考 Log 內容說明。

SK_ERROR_INITIALIZE_FAIL 尚未完成 SKOrderLib_Initialize 初始動作。

SK_ERROR_ACCOUNT_NOT_EXIST 帶入的帳號是無效的，請檢查 lpseAccount 帶入的值。

SK_ERROR_ACCOUNT_MARKET_NOT_MATCH 所帶入的帳號與交易的市場別不符合，有可能是委託期貨商品卻帶成股票帳號。

SK_ERROR_PERIOD_OUT_OF_RANGE 所帶入的交易區間有問題，必須是 0 盤中整股、1 盤後、2 零股。

SK_ERROR_BUYSELL_OUT_OF_RANGE 帶入的買賣別錯誤，必須是 0:買進 1:賣出

SK_ERROR_FLAG_OUT_OF_RANGE 帶入的交易種類錯誤，必須是 0:現股 1:融資 2:融券。

SK_ERROR_ORDER_SIGN_INVALID 委託單加簽失敗，請確認憑證安裝是否正確，或已經完成 ReadCert 的動作。

int SendFutureOrderAsync([in]const TCHAR* lpszAccount, [in]const TCHAR* lpszStockNo, [in]unsigned short usTradeType, [in]unsigned short usDayTrade, [in]unsigned short usBuySell, [in]const TCHAR* lpszPrice, [in]int nQty)

說明：

送出期貨委託。(非同步傳送)

傳入值：

lpszAccount 期貨帳號，IB + 帳號

lpszStockNo 委託期貨代號

usTradeType 0:ROD 1:IOC 2:FOK

usDayTrade 當沖 0:否 1:是，可當沖商品請參考交易所規定。

usBuySell 買賣別，0:買進 1:賣出

lpszPrice 委託價格，「M」表示市價

nQty 交易口數

回傳值：

SK_SUCCESS 表示函式呼叫成功，詳細委託狀態，請參考 Log 內容說明。

SK_ERROR_INITIALIZE_FAIL 尚未完成 SKOrderLib_Initialize 初始動作。

SK_ERROR_ACCOUNT_NOT_EXIST 帶入的帳號是無效的，請檢查 lpszAccount 帶入的值。

SK_ERROR_ACCOUNT_MARKET_NOT_MATCH 所帶入的帳號與交易的市場別不符合，有可能是委託期貨商品卻帶成股票帳號。

SK_ERROR_TRADE_TYPR_OUT_OF_RANG 所帶入的委託單有效區間有問題，必須是 0:ROD 1:IOC 2:FOK。

SK_ERROR_BUYSELL_OUT_OF_RANGE 帶入的買賣別錯誤，必須是 0:買進 1:賣出。

SK_ERROR_DAY_TRADE_OUT_OF_RANGE 帶入的當沖錯誤，必須是 0:否 1:是。

SK_ERROR_ORDER_SIGN_INVALID 委託單加簽失敗，請確認憑證安裝是否正確，或已經完成 ReadCert 的動作。

int SendOptionOrderAsync([in]const TCHAR* lpszAccount, [in]const TCHAR* lpszStockNo, [in]unsigned short usTradeType,[in]unsigned short usNewClose, [in]unsigned short usBuySell, [in]const TCHAR* lpszPrice, [in]int nQty)

說明：

送出選擇權委託。(非同步傳送)

傳入值：

lpszAccount 期貨帳號，IB + 帳號

lpszStockNo 委託選擇權代號

usTradeType 0:ROD 1:IOC 2:FOK

usNewClose 新平倉，0:新倉 1:平倉 2:自動

usBuySell 買賣別，0:買進 1:賣出

lpszPrice 委託價格，「M」表示市價

nQty 交易口數

lpszMessage 回覆訊息，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
pnMessageBufferSize lpszMessage 的 Buffer 寬度，如果傳入的 Buffer 寬度不足，則訊息內容將會被截斷以符合 Buffer 的大小，回傳會回覆真正填入 Buffer 的訊息長度。

回傳值：

SK_SUCCESS 表示函式呼叫成功，詳細委託狀態，請參考 Log 內容說明。

SK_ERROR_INITIALIZE_FAIL 尚未完成 SKOrderLib_Initialize 初始動作。

SK_ERROR_ACCOUNT_NOT_EXIST 帶入的帳號是無效的，請檢查 lpszAccount 帶入的值。

SK_ERROR_ACCOUNT_MARKET_NOT_MATCH 所帶入的帳號與交易的市場別不符合，有可能是委託期貨商品卻帶成股票帳號。

SK_ERROR_TRADE_TYPR_OUT_OF_RANG 所帶入的委託單有效區間有問題，必須是 0:ROD 1:IOC 2:FOK。

SK_ERROR_NEW_CLOSE_OUT_OF_RANGE 新平倉錯誤，必須是 0:新倉 1:平倉 2:自動。

SK_ERROR_BUYSELL_OUT_OF_RANGE 帶入的買賣別錯誤，必須是 0:買進 1:賣出。

SK_ERROR_ORDER_SIGN_INVALID 委託單加簽失敗，請確認憑證安裝是否正確，或已經完成 ReadCert 的動作。

帳務查詢相關函式說明

1. GetUserAccount
2. RegisterOnAccountCallBack
3. GetOpenInterest
4. RegisterOnOpenInterestCallBack
5. GetRealBalanceReport
6. RegisterOnRealBalanceReportCallBack
7. GetOverseaFutureOpenInterest
8. RegisterOnOverseaFutureOpenInterestCallBall

9. int GetUserAccount()

說明：

取回目前可交易的所有帳號。此函式需配合 RegisterOnAccountCallBack 函式，先註冊一個 Call back 函式，當呼叫 GetUserAccount，即會透過此 call back 函式，將帳號資訊以字串回傳，字串中以逗點分隔每一個欄位，欄位依序為：

市場,分公司,分公司代號,帳號,身份證字號,姓名

傳入值：

無

回傳值：

SK_SUCCESS 表示函式呼叫成功。

SK_ERROR_INITIALIZE_FAIL 尚未完成 SKOrderLib_Initialize 初始動作。

SK_ERROR_REGISTER_CALLBACK 尚未註冊 Call back 函式

`int RegisterOnAccountCallBack(long lCallbackFunction)`

說明：

註冊一個 Callback function，當有呼叫 GetUserAccount，即會透過此函式回傳帳號資訊。

函式宣告方式如下：

`Void __stdcall OnAccount(BSTR bstrData);`

回傳得字串內容，以逗點分隔每一個欄位，欄位依序為：

市場,分公司,分公司代號,帳號,身份證字號,姓名

傳入值：

lCallbackFunction 將 Callback 函式位址轉成 long 的表示方式，傳入 Library 內，完成註冊程序。

回傳值：

SK_SUCCESS 表示函式呼叫成功。

SK_ERROR_INITIALIZE_FAIL 尚未完成 SKOrderLib_Initialize 初始動作。

int GetOpenInterest([in] TCHAR* lpszAccount)

說明：

查詢傳入帳號目前的未平倉內容。此函式需搭配 RegisterOnOpenInterestCallBack 註冊一個處理未平倉的 Call back 函式。

此查詢以字串回傳查詢結果，字串中以「,」分隔每一個欄位，欄位依序為：

市場別，帳號，商品，買賣別，未平倉部位，當沖未平倉部位，平均成本(三位小數)，一點價值，單口手續費，交易稅(萬分之 X)

如果有任何組合部份，回傳時會拆解為單隻腳的商品內容後回傳。

傳入值：

lpszAccount 帳號，必須為 分公司 + 帳號。

回傳值：

SK_SUCCESS 表示函式呼叫成功。

SK_ERROR_INITIALIZE_FAIL 尚未完成 SKOrderLib_Initialize 初始動作。

SK_ERROR_QUERY_IN_PROCESSING 表示查詢已經正在進行中，另外請注意，每次查詢完成，需要間隔五秒後才能再次進行查詢的動作，否則都會取得此代碼的錯誤。

SK_ERROR_REGISTER_

CALLBACK 尚未註冊 Call back 函式。

int RegisterOnOpenInterestCallBack(long ICallBackFunction)

說明：

註冊一個 Callback function，當有呼叫 GetOpenInterest，即會透過此函式回傳帳號資訊。

函式宣告方式如下：

void __stdcall OnOpenInterest(BSTR bstrData);

回傳的字串內容，以「,」分隔每一個欄位，欄位依序為：

市場別, 帳號, 商品, 買賣別, 未平倉部位, 當沖未平倉部位, 平均成本(三位小數), 一點價值, 單口手續費, 交易稅(萬分之 X)

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。

傳入值：

ICallBackFunction 將 Callback 函式位址轉成 long 的表示方式，傳入 Library 內，完成註冊程序。

回傳值：

SK_SUCCESS 表示函式呼叫成功。

SK_ERROR_INITIALIZE_FAIL 尚未完成 SKOrderLib_Initialize 初始動作。

`int GetRealBalanceReport([in] char* lpszAccount)`

說明：

查詢證券即時庫存內容。

傳入值：

`lpszAccount` 證券帳號，分公司+帳號。

回傳值：

`SK_SUCCESS` 表示函式呼叫成功，回報內容將從 `CallBack` 回傳。

`SK_ERROR_INITIALIZE_FAIL` 尚未完成 `SKOrderLib_Initialize` 初始動作。

`SK_ERROR_QUERY_IN_PROCESSING` 前一次查詢尚未完成，必須等待前一次查詢完成方可繼續下一次的查詢。

`SK_ERROR_ACCOUNT_NOT_EXIST` 帶入的帳號是無效的，請檢查 `lpszAccount` 帶入的值。

`SK_ERROR_ACCOUNT_MARKET_NOT_MATCH` 帶入的帳號非證券帳號。

`SK_ERROR_REGISTER_CALLBACK`; 尚未註冊 `Call Back` 函式。

int RegisterOnRealBalanceReportCallBack(long lCallBackFunction)

說明：

註冊一個 Callback function，當有呼叫 GetRealBalanceReport，即會透過此函式回傳證券庫存資訊。

函式宣告方式如下：

```
void __stdcall OnRealBalanceReport( BSTR bstrData);
```

每一筆資料以「,」分隔每一個欄位，欄位依序為：

股票代號,
庫存種類, (T:集保 C:融資 L:融券)
資額度(原始),
資額度(可用),
券額度(原始),
券額度(可用),
股數:昨日庫存,
 今日委買,
 今日委賣,
 今日買進成交,
 今日賣出成交,
 今日資券可回補/集保可賣出,
 可資沖股數,
 可券沖股數,
 CTLBOND 庫存,
no_use

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。

傳入值：

lCallBackFunction 將 Callback 函式位址轉成 long 的表示方式，傳入 Library 內，完成註冊程序。

回傳值：

SK_SUCCESS 表示函式呼叫成功。

SK_ERROR_INITIALIZE_FAIL 尚未完成 SKOrderLib_Initialize 初始動作。

int GetOverseaFutureOpenInterest ([in] char* lpszAccount)

說明：

查詢海外期貨未平倉內容。

傳入值：

lpszAccount 海外期貨帳號，IB + 帳號。

回傳值：

SK_SUCCESS 表示函式呼叫成功，回報內容將從 Callback 回傳。

SK_ERROR_INITIALIZE_FAIL 尚未完成 SKOrderLib_Initialize 初始動作。

SK_ERROR_QUERY_IN_PROCESSING 前一次查詢尚未完成，必須等待前一次查詢完成方可繼續下一次的查詢。

SK_ERROR_ACCOUNT_NOT_EXIST 帶入的帳號是無效的，請檢查 lpszAccount 帶入的值。

SK_ERROR_ACCOUNT_MARKET_NOT_MATCH 帶入的帳號非海外期貨帳號。

SK_ERROR_REGISTER_CALLBACK; 尚未註冊 Call Back 函式。

int RegisterOnOverseaFutureOpenInterestCallBall

說明：

註冊一個 Callback function，當有呼叫 GetOverseaFutureOpenInterest，即會透過此函式回傳海外期貨未平倉資訊。

函式宣告方式如下：

```
void __stdcall OnOverseaFutureOpenInterest ( BSTR bstrData);
```

第一筆回傳內容為「@@」開頭，表示查詢狀態，每一筆資料以「,」分隔每一個欄位，欄位依序為：

「@@,訊息代碼,訊息內容」

其中，當訊息代碼為

0：表示查詢成功，訊息內容為空字串

其他非「0」皆表示查詢錯誤，訊息內容將描述錯誤原因

如有未平倉資訊將從第二筆開始回傳，每一筆資料以「,」分隔每一個欄位，欄位依序為：

- (1) 海期交易所代碼
- (2) 海期交易所中文名稱
- (3) 帳號
- (4) 海期商品代碼+年月，代碼與年月間以空白區隔
- (5) 海期商品中文名稱
- (6) 買賣別 B:買進 S：賣出
- (7) 數量
- (8) 市價
- (9) 平均成交價
- (10) 昨日結算價
- (11) 損益

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。

傳入值：

lCallbackFunction 將 Callback 函式位址轉成 long 的表示方式，傳入 Library 內，完成註冊程序。

回傳值：

SK_SUCCESS 表示函式呼叫成功。

SK_ERROR_INITIALIZE_FAIL 尚未完成 **SKOrderLib_Initialize** 初始動作。

國內停損單相關函式說明

1. SendFutureStopLoss 送出一筆停損委託
2. CancelFutureStopLoss 取消委託停損單
3. SendMovingStopLoss 送出一筆移動停損停損委託
4. CancelMovingStopLoss 取消移動停損停損單
5. SendOptionStopLos 送出一筆選擇權停損委託
6. CancelOptionStopLoss 取消選擇權停損單委託
7. RegisterOnStopLossReportCallBack 註冊一個 Callback function，當有呼叫 GetStopLossReprot，即會透過此函式回傳帳號資訊。
8. GetStopLossReport 查詢停損委託內容

```
int SendFutureStopLoss( [in] const char* lpszAccount, [in]const char* lpszStockNo,  
[in]unsigned short usTradeType, [in]unsigned short usDayTrade, [in]unsigned short  
usBuySell, [in]const char* lpszPrice, [in]int nQty, [in]const char* lpszTriggerPrice,  
[in,out]char* lpszMessage, [in,out]int* pnMessageBufferSize)
```

說明：

送出一筆停損委託。

傳入值：

lpszAccount 期貨帳號，IB + 帳號。

lpszStockNo 委託期貨代號。

usTradeType 0:ROD 1:IOC 2:FOK。

usDayTrade 當沖 0:否 1:是，可當沖商品請參考交易所規定。

usBuySell 買賣別，0:買進 1:賣出。

lpszPrice 委託價格，「M」表示市價。

nQty 交易口數。

lpszTriggerPrice 觸發價格。

lpszMessage 回覆訊息，訊息內容則為委託停損狀態。

pnMessageBufferSize lpszMessage 的 Buffer 寬度，如果傳入的 Buffer 寬度不足，則訊息內容將會被截斷以符合 Buffer 的大小，回傳會回覆真正填入 Buffer 的訊息長度。

回傳值：

>= 100 表示委託伺服器接收成功，委託狀態請務必確認回傳訊息內說明。

其他非 0 數值表示委託異常，詳細原因可以參考回傳的代碼，或是參考 Log 內容說明。

SK_ERROR_INITIALIZE_FAIL 尚未完成 SKOrderLib_Initialize 初始動作。

SK_ERROR_ACCOUNT_NOT_EXIST 帶入的帳號是無效的，請檢查 lpszAccount 帶入的值。

SK_ERROR_ACCOUNT_MARKET_NOT_MATCH 所帶入的帳號與交易的市場別不符合，有可能是委託期貨商品卻帶成股票帳號。

SK_ERROR_TRADE_TYPR_OUT_OF_RANG 所帶入的委託單有效區間有問題，必須是 0:ROD 1:IOC 2:FOK。

SK_ERROR_BUYSELL_OUT_OF_RANGE 帶入的買賣別錯誤，必須是 0:買進 1:賣出。

SK_ERROR_DAY_TRADE_OUT_OF_RANGE 帶入的當沖錯誤，必須是 0:否 1:是。

SK_ERROR_ORDER_SIGN_INVALID 委託單加簽失敗，請確認憑證安裝是否正確，或已經完成 ReadCert 的動作。

```
int CancelFutureStopLoss( [in] const char* lpszAccount, [in]const char* lpszBookNo,  
    [in]const char* lpszSymbol, [in]const char* lpszBuySell, [in]const char* lpszPrice,  
    [in]const char* lpszQty, [in]const char* lpszTriggerPrice, [in]const char* lpszTradeType,  
    [in]const char* lpszDayTrade, [in,out]char* lpszMessage, [in,out] int*  
    pnMessageBufferSize)
```

說明：

取消委託停損單。

傳入值：

以下欄位請參考 GetStopLossReport 回傳的內容。

lpszAccount 期貨帳號，IB + 帳號。

lpszBookNo, 五碼委託書號。

lpszSymbol 委託期貨代號，與下單的商品代號不同。

lpszBuySell B:買進， S:賣出。

lpszPrice 委託價格。

lpszQty 委託量。

lpszTriggerPrice 觸發價格。

lpszTradeType ROD, IOC, FOK。

lpszDayTrade Y:當沖，N:非當沖。

lpszMessage 回覆訊息，訊息內容則為取消停損單狀態。

pnMessageBufferSize lpszMessage 的 Buffer 寬度，如果傳入的 Buffer 寬度不足，則訊息內容將會被截斷以符合 Buffer 的大小，回傳會回覆真正填入 Buffer 的訊息長度。

回傳值：

0 取消已經送出，詳細請參考回傳訊息說明。

SK_ERROR_INITIALIZE_FAIL 尚未完成 SKOrderLib_Initialize 初始動作。

SK_ERROR_ACCOUNT_NOT_EXIST 帶入的帳號是無效的，請檢查 lpseAccount 帶入的值

SK_ERROR_ORDER_SIGN_INVALID 委託單加簽失敗，請確認憑證安裝是否正確，或已經完成 ReadCert 的動作。

```
int SendMovingStopLoss(char* lpszAccount, char* lpszStockNo, unsigned short  
usTradeType, unsigned short usDayTrade, unsigned short usBuySell, char* lpszPrice, int nQty,  
char* lpszMovingPoint, char* lpszMessage, int* pnMessageBufferSize)
```

說明：

送出一筆移動停損停損委託。

傳入值：

lpszAccount 期貨帳號，IB + 帳號。

lpszStockNo 委託期貨代號。

usTradeType 0:ROD 1:IOC 2:FOK。

usDayTrade 當沖 0:否 1:是，可當沖商品請參考交易所規定。

usBuySell 買賣別，0:買進 1:賣出。

lpszPrice 觸價基準價格，「M」表示市價。

nQty 交易口數。

lpszMovingPoint 移動點數。

lpszMessage 回覆訊息，訊息內容則為委託停損狀態。

pnMessageBufferSize lpszMessage 的 Buffer 寬度，如果傳入的 Buffer 寬度不足，則訊息內容將會被截斷以符合 Buffer 的大小，回傳會回覆真正填入 Buffer 的訊息長度。

回傳值：

>= 100 表示委託伺服器接收成功，委託狀態請務必確認回傳訊息內說明。

其他非 0 數值表示委託異常，詳細原因可以參考回傳的代碼，或是參考 Log 內容說明。

SK_ERROR_INITIALIZE_FAIL 尚未完成 SKOrderLib_Initialize 初始動作。

SK_ERROR_ACCOUNT_NOT_EXIST 帶入的帳號是無效的，請檢查 lpszAccount 帶入的值。

SK_ERROR_ACCOUNT_MARKET_NOT_MATCH 所帶入的帳號與交易的市場別不符合，有可能是委託期貨商品卻帶成股票帳號。

SK_ERROR_TRADE_TYPR_OUT_OF_RANG 所帶入的委託單有效區間有問題，必須是 0:ROD 1:IOC 2:FOK。

SK_ERROR_BUYSELL_OUT_OF_RANGE 帶入的買賣別錯誤，必須是 0:買進 1:賣出。

SK_ERROR_DAY_TRADE_OUT_OF_RANGE 帶入的當沖錯誤，必須是 0:否 1:是。

SK_ERROR_ORDER_SIGN_INVALID 委託單加簽失敗，請確認憑證安裝是否正確，或已經完成 ReadCert 的動作。

```
int CancelMovingStopLoss( [in] const char* lpszAccount, [in]const char* lpszBookNo,  
    [in]const char* lpszSymbol, [in]const char* lpszBuySell, [in]const char* lpszPrice,  
    [in]const char* lpszQty, [in]const char* lpszMovingPoint, [in]const char* lpszTradeType,  
    [in]const char* lpszDayTrade, [in,out]char* lpszMessage, [in,out] int*  
    pnMessageBufferSize)
```

說明：

取消移動停損停損單。

傳入值：

以下欄位請參考 GetStopLossReport 回傳的內容。

lpszAccount 期貨帳號，IB + 帳號。

lpszBookNo, 五碼委託書號。

lpszSymbol 委託期貨代號，與下單的商品代號不同。

lpszBuySell B:買進，S:賣出。

lpszPrice 觸發基準價。

lpszQty 委託量。

lpszMovingPoint 移動點數。

lpszTradeType ROD, IOC, FOK。

lpszDayTrade Y:當沖，N:非當沖。

lpszMessage 回覆訊息，訊息內容則為取消停損單狀態。

pnMessageBufferSize lpszMessage 的 Buffer 寬度，如果傳入的 Buffer 寬度不足，則訊息內容將會被截斷以符合 Buffer 的大小，回傳會回覆真正填入 Buffer 的訊息長度。

回傳值：

0 取消已經送出，詳細請參考回傳訊息說明。

SK_ERROR_INITIALIZE_FAIL 尚未完成 SKOrderLib_Initialize 初始動作。

SK_ERROR_ACCOUNT_NOT_EXIST 帶入的帳號是無效的，請檢查 lpseAccount 帶入的值

SK_ERROR_ORDER_SIGN_INVALID 委託單加簽失敗，請確認憑證安裝是否正確，或已經完成 ReadCert 的動作。

```
int SendOptionStopLoss( const TCHAR* lpszAccount, const TCHAR* lpszStockNo, unsigned
short usTradeType, unsigned short usNewClose, unsigned short usBuySell, const TCHAR*
lpszPrice, int nQty, const TCHAR* lpszTriggerPrice, TCHAR* lpszMessage, int*
pnMessageBufferSize)
```

說明：

送出一筆選擇權停損委託。

傳入值：

lpszAccount 期貨帳號，IB + 帳號。

lpszStockNo 委託期貨代號。

usTradeType 0:ROD 1:IOC 2:FOK。

usNewClose 新平倉：0: 新倉，1: 平倉，2: 自動

usBuySell 買賣別，0:買進 1:賣出。

lpszPrice 觸價基準價格，「M」表示市價。

nQty 交易口數。

lpszMovingPoint 移動點數。

lpszMessage 回覆訊息，訊息內容則為委託停損狀態。

pnMessageBufferSize lpszMessage 的 Buffer 寬度，如果傳入的 Buffer 寬度不足，則訊息內容將會被截斷以符合 Buffer 的大小，回傳會回覆真正填入 Buffer 的訊息長度。

回傳值：

>= 100 表示委託伺服器接收成功，委託狀態請務必確認回傳訊息內說明。

其他非 0 數值表示委託異常，詳細原因可以參考回傳的代碼，或是參考 Log 內容說明。

SK_ERROR_INITIALIZE_FAIL 尚未完成 SKOrderLib_Initialize 初始動作。

SK_ERROR_ACCOUNT_NOT_EXIST 帶入的帳號是無效的，請檢查 lpszAccount 帶入的值。

SK_ERROR_ACCOUNT_MARKET_NOT_MATCH 所帶入的帳號與交易的市場別不符合，有可能是委託期貨商品卻帶成股票帳號。

SK_ERROR_TRADE_TYPR_OUT_OF_RANG 所帶入的委託單有效區間有問題，必須是 0:ROD 1:IOC 2:FOK。

SK_ERROR_BUYSELL_OUT_OF_RANGE 帶入的買賣別錯誤，必須是 0:買進 1:賣出。

SK_ERROR_DAY_TRADE_OUT_OF_RANGE 帶入的當沖錯誤，必須是 0:否 1:是。

SK_ERROR_ORDER_SIGN_INVALID 委託單加簽失敗，請確認憑證安裝是否正確，或已經完成 ReadCert 的動作。

```
int CancelOptionStopLoss( [in] const char* lpszAccount, [in]const char* lpszBookNo,  
    [in]const char* lpszSymbol, [in]const char* lpszBuySell, [in]const char* lpszPrice,  
    [in]const char* lpszQty, [in]const char* lpszTriggerPrice, [in]const char* lpszTradeType,  
    [in]const char* lpszDayTrade, [in,out]char* lpszMessage, [in,out] int*  
    pnMessageBufferSize)
```

說明：

取消選擇權停損單委託（未觸發的狀態下方可透過此函式取消）。

注意，當已經觸發的選擇權停損單，必須透過取消一般委託的方式取消委託。

傳入值：

以下欄位請參考 GetStopLossReport 回傳的內容。

lpszAccount 期貨帳號，IB + 帳號。

lpszBookNo, 委託序號，非正式委託書號

lpszSymbol 委託期貨代號，與下單的商品代號不同。

lpszBuySell B:買進，S:賣出。

lpszPrice 委託價格。

lpszQty 委託量。

lpszTriggerPrice 觸發價格。

lpszTradeType ROD, IOC, FOK。

lpszDayTrade Y:當沖，N:非當沖，目前請固定帶“N”。

lpszMessage 回覆訊息，訊息內容則為取消停損單狀態。

pnMessageBufferSize lpszMessage 的 Buffer 寬度，如果傳入的 Buffer 寬度不足，則訊息內容將會被截斷以符合 Buffer 的大小，回傳會回覆真正填入 Buffer 的訊息長度。

回傳值：

0 取消已經送出，詳細請參考回傳訊息說明。

SK_ERROR_INITIALIZE_FAIL 尚未完成 SKOrderLib_Initialize 初始動作。

SK_ERROR_ACCOUNT_NOT_EXIST 帶入的帳號是無效的，請檢查 lpszAccount 帶入的值

SK_ERROR_ORDER_SIGN_INVALID 委託單加簽失敗，請確認憑證安裝是否正確，或已經完成 ReadCert 的動作。

int RegisterOnStopLossReportCallBack(long lCallBackFunction)

說明：

註冊一個 Callback function，當有呼叫 GetStopLossReprot，即會透過此函式回傳帳號資訊。

函式宣告方式如下：

```
void __stdcall OnStopLossReport( BSTR bstrData);
```

如果回傳的第一筆資料是 "M" 開頭，表示為回傳訊息，後面接的就是訊息內容。回傳的字串內容，以「,」分隔每一個欄位，欄位依序為：

交易日期,	(此筆委託單有效交易日期)
市場別,	(TF 期貨 / TO 選擇權)
分公司+帳號,	(broker+cust_no)
委託序號,	(停損單編號 觸發後會變成委託書號)
商品,	(期貨：商品代號+商品年月 選擇權：商品代號 + CALL/PUT + 履約價 + 商品年月)
委託種類,	(第一碼：B 買/S 賣 第二碼：N 新倉/O 平倉 第三碼： L 限價/M 市價)
委託價格,	(0 為市價)
委託量,	
委託時間,	
觸發價,	(移動停損單尚未觸發時，此欄位會回傳觸發參考價)
觸發時間,	
狀態,	(01 尚未觸發 (已扣保證金的停損單但尚未觸發) 02 已取消 (已送交易所並取消或是 IOC 單的取消) 03 完全成交 04 部份成交 05 已觸發 (已扣保證金的停損單觸發) 06 送交易所失敗，原因：(選擇權停損單送單失敗) 07 未觸發已取消 08 委託成功尚未觸價 (選擇權停損單下單成功，尚未送至後台) 09 預約成功 (預約停損單下單成功) 10 預約送單失敗 11 已取消預約)
現價,	(目前為空值)
成交量,	(目前為空值)
委託條件,	(ROD/IOC/FOK)

當沖與否, (Y 當沖)
單別, (STP:停損 MST:移動 OST:選擇權停損)
移動點數, (移動停損單才會有 其餘為 0)
委託日期, (此筆委託單委託日期)
錯誤訊息
委託書號, (交易所委託書號)

傳入值：

ICallBackFunction 將 Callback 函式位址轉成 long 的表示方式，傳入 Library 內，完成註冊程序。

回傳值：

SK_SUCCESS 表示函式呼叫成功。

SK_ERROR_INITIALIZE_FAIL 尚未完成 SKOrderLib_Initialize 初始動作。

int GetStopLossReport([in] char* lpszAccount,[in] int nReportStatus, [in] char* lpszKind)

說明：

查詢停損委託內容。

傳入值：

lpszAccount 期貨帳號，IB + 帳號。

nReportStatus 0: 全部的委託單，1: 有效的委託單。

lpszKind STP:一般停損，MST:移動停損，OST:選擇權停損

回傳值：

SK_SUCCESS 表示函式呼叫成功，回報內容將從 Callback 回傳。

SK_ERROR_INITIALIZE_FAIL 尚未完成 SKOrderLib_Initialize 初始動作。

SK_ERROR_QUERY_IN_PROCESSING 前一次查詢尚未完成，必須等待前一次查詢完成方可繼續下一次的查詢。

SK_ERROR_ACCOUNT_NOT_EXIST 帶入的帳號是無效的，請檢查 lpszAccount 帶入的值。

SK_ERROR_ACCOUNT_MARKET_NOT_MATCH 帶入的帳號非期貨帳號。

SK_ERROR_REGISTER_CALLBACK; 尚未註冊 Call Back 函式。

複委託傳送委託 (美股與港股下單)

● 美股與港股下單檢查規則

1. 價格

1.1 會檢查使用者是否輸入高於或低於 10%的昨收價格

=>若高於 10%(輸入價格的 1.1) 或 低於 10%(輸入價格的 0.9) 則

1.1.1 回傳錯誤訊息提示使用者輸入價格超過或低於 10%範圍

=> (SK_ERROR_FOREIGNSTOCK_PRICE_OUT_OF_RANGE (1024))

1.1.2 不送出下單的委託

1.2 使用者不得輸入 負數價格 或 0

1.2.1 回傳錯誤訊息提示使用者價格輸入錯誤

=> (SK_ERROR_PRICE_INVALID (1016))

1.2.2 不送出下單的委託

2. 幣別

2.1 檢查使用者輸入幣別是否為指定幣別(HKD, NTD, USD, JPY, SGD, EUR, AUD)

2.1.1 回傳錯誤訊息提示使用者輸入非指定幣別

=> (SK_ERROR_FOREIGNSTOCK_UNDEFINE_COINTYPE (1025))

2.1.2 不送出下單的委託

2.2 檢查使用者輸入幣別是否有重複

(交割幣別 1, 交割幣別 2, 交割幣別 3)

交割幣別必須三者都不同, (ex. HKD, NTD, USD)或是

只設定交割幣別 1, 交割幣別 2,3 為空字串(“ ”) (ex. HKD, “ ”, “ ”)

2.2.1 回傳錯誤訊息提示使用者輸入幣別重複

=> (SK_ERROR_FOREIGNSTOCK_SAME_COINSTYPE (1026))

2.2.2 不送出下單的委託

2.3 檢查使用者送賣出委託幣別為原幣別(美股美金) (港股港幣)

賣出委託, 交割幣別 1 必須為(美股 USD)(港股 HKD), 交割幣別 2,3 可輸入其他幣別或輸入空字串(“ ”)

2.3.1 回傳錯誤訊息提示使用者為賣出委託時, 則必須輸入原幣別

(美股美金) (港股港幣)

=> (SK_ERROR_FOREIGNSTOCK_SALE_SHOULD_ORIGINAL_COIN (1027))

2.3.2 不送出下單的委託

- 3. 股數 (每個商品有不同的最小交易單位)
 - 3.1 買單輸入股數要是最小交易單位的倍數且不得小於等於 0
 - 3.1.1 回傳錯誤訊息提示使用者股數輸入錯誤
=> (SK_ERROR_FOREIGNSTOCK_TRADE_UNIT_INVALID (1028))
 - 3.1.2 不送出下單的委託
 - 3.2 賣單輸入股數要是大於等於最小交易單位，則必須為最小交易單位的倍數且不得小於等於 0
 - 3.2.1 回傳錯誤訊息提示使用者股數輸入錯誤
=> (SK_ERROR_FOREIGNSTOCK_TRADE_UNIT_INVALID (1028))
 - 3.2.2 不送出下單的委託
 - 3.3 賣單輸入股數要是小於最小交易單位，則為零股交易且不得小於等於 0
 - 3.3.1 回傳錯誤訊息提示使用者股數輸入錯誤
=> (SK_ERROR_FOREIGNSTOCK_TRADE_UNIT_INVALID (1028))
 - 3.3.2 不送出下單的委託
- 4. 可交易股票名稱
 - 4.1 使用者輸入的委託必須為當日可以交易之股票名稱
 - 4.1.1 回傳錯誤訊息提示使用者股票名稱輸入錯誤
=> (SK_ERROR_FOREIGNSTOCK_STOCKNO_INVALID (1029))
 - 4.1.2 不送出下單的委託

```

int __stdcall SendForeignStockOrder( [in] const TCHAR* lpszAccount,
                                     [in] const TCHAR* lpszStockNo,
                                     [in] const TCHAR* lpszExchangeNo,
                                     [in] unsigned short usBuySell,
                                     [in] const TCHAR* lpszPrice,
                                     [in] int nQty,
                                     [in] const TCHAR* lpszCurrency1,
                                     [in] const TCHAR* lpszCurrency2,
                                     [in] const TCHAR* lpszCurrency3,
                                     [in] int iAccountType
                                     [out] TCHAR* lpszMessage,
                                     [out] int* pnMessageBufferSize )

```

說明：

送出一筆 美股或港股 下單委託。

傳入值：

lpszAccount	複委託帳號，分公司＋帳號。
lpszStockNo	股票代號
lpszExchangeNo	交易所，美股：(US)， 港股：(HK)。
usBuySell	買賣別，0:買進 1:賣出。
lpszPrice	委託價格
nQty	委託量
lpszCurrency1	扣款幣別 幣別順序一 (HKD, NTD, USD, JPY, SGD, EUR, AUD)
lpszCurrency2	扣款幣別 幣別順序二 (HKD, NTD, USD, JPY, SGD, EUR, AUD)
lpszCurrency3	扣款幣別 幣別順序三 (HKD, NTD, USD, JPY, SGD, EUR, AUD)
iAccountType	專戶別種類: 1:外幣專戶 2:台幣專戶

扣款幣別順序

lpszCurrency1	幣別順序一
lpszCurrency2	幣別順序二
lpszCurrency3	幣別順序三

lpszCurrency1 一定要輸入值，不可帶入(””)空字串，至於 lpszCurrency2, lpszCurrency3 可帶入空字串

回傳值：

SK_SUCCESS 表示函式呼叫成功，回報內容將從 Callback 回傳。

SK_ERROR_INITIALIZE_FAIL 尚未完成 SKOrderLib_Initialize 初始動作。

SK_ERROR_FOREIGNSTOCK_INITIALIZE_FAIL 尚未完成

GetEfficientForeignStockForInitial 初始動作

SK_ERROR_QUERY_IN_PROCESSING 前一次查詢尚未完成，必須等待前一次查詢完成方可繼續下一次的查詢。

SK_ERROR_ACCOUNT_NOT_EXIST 帶入的帳號是無效的，請檢查 lpszAccount 帶入的值。

SK_ERROR_ACCOUNT_MARKET_NOT_MATCH 帶入的帳號非複委託帳號。

SK_ERROR_FOREIGNSTOCK_PRICE_OUT_OF_RANGE

輸入價格低於昨收價 10%或高於昨收價 10%

SK_ERROR_FOREIGNSTOCK_UNDEFINE_COINTYPE 輸入錯誤幣別

目前可使用的幣別只有(HKD, NTD, USD, JPY, SGD, EUR, AUD,"")

SK_ERROR_FOREIGNSTOCK_SAME_COINSTYPE 幣別順序中有相同幣別

SK_ERROR_FOREIGNSTOCK_SALE_SHOULD_ORIGINAL_COIN 賣出委託必須輸入原幣別(美股美金 USD)(港股港幣 HKD)

SK_ERROR_FOREIGNSTOCK_TRADE_UNIT_INVALID 輸入股數錯誤

股數規則:

1. 買(賣)單股數必須為最小交易單位倍數而且不可以小於或等於 0
2. 買單股數，不得小於最小交易單位

SK_ERROR_FOREIGNSTOCK_STOCKNO_INVALID 輸入股票代號目前不能交易

SK_ERROR_FOREIGNSTOCK_ACCOUNTTYPE_INVALID 交易專戶別輸入錯誤

使用範例:

Ex.

(1.) 美股

// 美股

```
SendForeignStockOrder( "91829801721", //複委託帳號，分公司+帳號  
                        "AAPL",          //股票代號 (蘋果電腦)  
                        "US",            //交易所  
                        0,                //0 買 1 賣  
                        "260.83",        //委託價 (昨收為 260)  
                        2,                //委託量  
                        "USD",           //幣別順序一  
                        "HKD",           //幣別順序二  
                        "NTD",           //幣別順序三  
                        1,                //1:外幣專戶 2:台幣專戶  
                        caBuf,  
                        &nSize);
```

(2.) 港股

```
SendForeignStockOrder( "91829801721", //複委託帳號，分公司+帳號  
                        "00119",         //股票代號 (保利香港)  
                        "HK",            //交易所  
                        0,                //0 買 1 賣  
                        5.88,            //委託價 (昨收為 6)  
                        1000,            //委託量  
                        "USD",           //幣別順序一  
                        "HKD",           //幣別順序二  
                        "NTD",           //幣別順序三  
                        1,                //1:外幣專戶 2:台幣專戶  
                        caBuf,  
                        &nSize);
```

```

int __stdcall SendForeignStockOrderAsync ( [in] const TCHAR* lpszAccount,
                                           [in] const TCHAR* lpszStockNo,
                                           [in] const TCHAR* lpszExchangeNo,
                                           [in] unsigned short usBuySell,
                                           [in] const TCHAR* lpszPrice,
                                           [in] int nQty,
                                           [in] const TCHAR* lpszCurrency1,
                                           [in] const TCHAR* lpszCurrency2,
                                           [in] const TCHAR* lpszCurrency3,
                                           [in] int iAccountType,
                                           [out] TCHAR* lpszMessage,
                                           [out] int* pnMessageBufferSize )

```

說明：

送出一筆 美股或港股 下單委託。(非同步傳送)

傳入值：

lpszAccount	複委託帳號，分公司＋帳號。
lpszStockNo	股票代號
lpszExchangeNo	交易所，美股：(US)，港股：(HK)。
usBuySell	買賣別，0:買進 1:賣出。
lpszPrice	委託價格
nQty	委託量
lpszCurrency1	扣款幣別 幣別順序一 (HKD, NTD, USD, JPY, SGD, EUR, AUD)
lpszCurrency2	扣款幣別 幣別順序二 (HKD, NTD, USD, JPY, SGD, EUR, AUD)
lpszCurrency3	扣款幣別 幣別順序三 (HKD, NTD, USD, JPY, SGD, EUR, AUD)
iAccountType	專戶別種類: 1:外幣專戶 2:台幣專戶

扣款幣別順序

lpszCurrency1	幣別順序一
lpszCurrency2	幣別順序二
lpszCurrency3	幣別順序三

lpszCurrency1 一定要輸入值，不可帶入(“ ”)空字串，至於 lpszCurrency2, lpszCurrency3 可帶入空字串

回傳值：

SK_SUCCESS 表示函式呼叫成功，詳細委託狀態，請參考 Log 內容說明。

SK_ERROR_INITIALIZE_FAIL 尚未完成 SKOrderLib_Initialize 初始動作。

SK_ERROR_FOREIGNSTOCK_INITIALIZE_FAIL 尚未完成

GetEfficientForeignStockForInitial 初始動作

SK_ERROR_ACCOUNT_NOT_EXIST 帶入的帳號是無效的，請檢查 lpseAccount 帶入的值。

SK_ERROR_ACCOUNT_MARKET_NOT_MATCH 所帶入的帳號與交易的市場別不符合，必須輸入複委託帳號。

SK_ERROR_BUYSELL_OUT_OF_RANGE 帶入的買賣別錯誤，必須是 0:買進 1:賣出。

SK_ERROR_ORDER_SIGN_INVALID 委託單加簽失敗，請確認憑證安裝是否正確，或已經完成 ReadCert 的動作。

SK_ERROR_ACCOUNT_MARKET_NOT_MATCH 帶入的帳號非複委託帳號。

SK_ERROR_FOREIGNSTOCK_PRICE_OUT_OF_RANGE

輸入價格低於昨收價 10%或高於昨收價 10%

SK_ERROR_FOREIGNSTOCK_UNDEFINE_COINTYPE 輸入錯誤幣別

目前可使用的幣別只有(HKD, NTD, USD, JPY, SGD, EUR, AUD,"")

SK_ERROR_FOREIGNSTOCK_SAME_COINSTYPE 幣別順序中有相同幣別

SK_ERROR_FOREIGNSTOCK_SALE_SHOULD_ORIGINAL_COIN 賣出委託必須輸入原幣別(美股美金 USD)(港股港幣 HKD)

SK_ERROR_FOREIGNSTOCK_TRADE_UNIT_INVALID 輸入股數錯誤

股數規則:

3. 買(賣)單股數必須為最小交易單位倍數而且不可以小於或等於 0

4. 買單股數，不得小於最小交易單位

SK_ERROR_FOREIGNSTOCK_STOCKNO_INVALID 輸入股票代號目前不能交易

SK_ERROR_FOREIGNSTOCK_ACCOUNTTYPE_INVALID 交易專戶別輸入錯誤

使用範例:

Ex.

(3.) 美股

// 美股

```
SendForeignStockOrderAsync ( "91829801721", //複委託帳號，分公司+帳號
                              "AAPL",         //股票代號 (蘋果電腦)
                              "US",           //交易所
                              0,               //0 買 1 賣
                              "260.83",       //委託價 (昨收為 260)
                              2,               //委託量
                              "USD",           //幣別順序一
                              "HKD",           //幣別順序二
                              "NTD",           //幣別順序三
                              1,               //1:外幣專戶 2:台幣專戶
                              caBuf,
                              &nSize);
```

(4.) 港股

```
SendForeignStockOrderAsync ( "91829801721", //複委託帳號，分公司+帳號
                              "00119",        //股票代號 (保利香港)
                              "HK",           //交易所
                              0,               //0 買 1 賣
                              5.88,           //委託價 (昨收為 6)
                              1000,           //委託量
                              "USD",           //幣別順序一
                              "HKD",           //幣別順序二
                              "NTD",           //幣別順序三
                              1,               //1:外幣專戶 2:台幣專戶
                              caBuf,
                              &nSize);
```

```
int __stdcall CancelForeignStockOrderByBookNo( [in] const TCHAR* lpszAccount,
                                                [in] const TCHAR* lpszBookNo,
                                                [in] const TCHAR* lpszExchangeNo,
                                                [out] TCHAR* lpszMessage,
                                                [out] int* pnMessageBufferSize )
```

說明：

美股或港股 根據委託書號取消委託。

傳入值：

lpszAccount 複委託帳號，分公司＋帳號。

lpszBookNo 委託書號。

lpszExchangeNo 交易所，美股：(US)，港股：(HK)。

回傳值：

0 取消已經送出，詳細請參考回傳訊息說明。

SK_ERROR_INITIALIZE_FAIL 尚未完成 SKOrderLib_Initialize 初始動作。

SK_ERROR_FOREIGNSTOCK_INITIALIZE_FAIL 尚未完成

GetEfficientForeignStockForInitial 初始動作

SK_ERROR_ACCOUNT_NOT_EXIST 帶入的帳號是無效的，請檢查 lpszAccount 帶入的值

SK_ERROR_ORDER_SIGN_INVALID 委託單加簽失敗，請確認憑證安裝是否正確，或已經完成 ReadCert 的動作。

```
int __stdcall CancelForeignStockOrderBySeqNo( [in] const TCHAR* lpszAccount,  
                                                [in] const TCHAR* lpszSeqNo,  
                                                [in] const TCHAR* lpszExchangeNo,  
                                                [out] TCHAR* lpszMessage,  
                                                [out] int* pnMessageBufferSize )
```

說明：

美股或港股 根據委託序號取消委託。

傳入值：

lpszAccount 複委託帳號，分公司＋帳號。
lpszSeqNo 委託序號。
lpszExchangeNo 交易所，美股：(US) ，港股：(HK)。

回傳值：

0 取消已經送出，詳細請參考回傳訊息說明。

SK_ERROR_INITIALIZE_FAIL 尚未完成 SKOrderLib_Initialize 初始動作。

SK_ERROR_FOREIGNSTOCK_INITIALIZE_FAIL 尚未完成

GetEfficientForeignStockForInitial 初始動作

SK_ERROR_ACCOUNT_NOT_EXIST 帶入的帳號是無效的，請檢查 lpseAccount 帶入的值

SK_ERROR_ORDER_SIGN_INVALID 委託單加簽失敗，請確認憑證安裝是否正確，或已經完成 ReadCert 的動作。

Trade Station 訊號接收

此項功能主要接收 Trade Station 發出買賣訊號，測試版本為 Omega Research ProSuite2000i Version: 5.00.0822。

首先必須先完成 SKOrder_InitializeTS 後再進行呼叫

RegisterOnTSFilledOrderCallBack

RegisterOnTSActiveOrderCallBack

RegisterOnTSCanceledOrderCallBack

註冊訊號回傳 CallBack 函式。

`int __stdcall SKOrderLib_InitializeTS ([in] char* lpszTSCOMName)`

說明：

初始化與 TradeStation 間的連接。

傳入值：

`lpszTSCOMName` 填入 TradeStation 在電腦中註冊的名稱，如果不確定內容，請直接帶入空字串。

回傳值：

`SKSUCCESS` 進行連線串接成功。

`SK_WARNING_TS_READY` 表示重複執行此初始函式，之前的執行已經成功初始完成。

`SK_ERROR_INITIALIZE_FAIL` 尚未完成 `SKOrderLib_Initialize` 初始動作。

`SK_ERROR_TS_INITIALIZE_FAIL` 初始失敗，無法與 TradeStation 進行連線，可能版本不同，此函式預設與 Omega Research ProSuite 2000i Version: 5.00.0822 進行串接。

其他負值的錯誤代碼為串接過程發生的錯誤代碼，此代碼為作業系統定義，請自行尋找定義內容，或是透過 Visual Studio 6.0 中工具程式 Error LookUp 查詢。

int __stdcall RegisterOnTSFilledOrderCallBack ([in] long ICallBackFunction)

說明：

定義 TradeStation 訊號接收函式，此 CallBack 函式定義如下：

```
void __stdcall OnTSFilledOrder( BSTR bstrSymbol, BSTR bstrDescription,  
                                BSTRbstrOrderType, BSTR bstrOrder, long lFillPrice, long lSlippage, double  
                                dTimePlace, double dTimeFilled, BSTR bstrStrategy, BSTR bstrSignal, BSTR  
                                bstrWorkspace, BSTR bstrInterval, BSTR bstrPositionNumber, BSTR  
                                bstrOrderNumber)
```

其中

bstrSymbol 欄位 Symbol

bstrDescription 欄位 Description

bstrOrderType 欄位 OrderType

bstrOrder 欄位 Order

lFillPrice 欄位 Fill Price

lSlippage 欄位 Slippage

dTimePlaced 欄位 Time Placed，需自行轉成字串表示。

dTimeFilled; 欄位 Time Filled，需自行轉成字串表示。

bstrStrategy 欄位 Strategy

bstrSignal 欄位 Signal

bstrWorkspace 欄位 WorkSpace

bstrInterval 欄位 Interval

bstrPositionNumber 欄位 Position Number

bstrOrderNumber 欄位 Order Number

傳入值：

ICallBackFunction 將 Callback 函式位址轉成 long 的表示方式，傳入 Library 內，完成註冊程序。。

回傳值：

SKSUCCESS 進行連線串接成功。

SK_ERROR_INITIALIZE_FAIL 尚未完成 SKOrderLib_Initialize 初始動作。

SK_ERROR_TS_INITIALIZE_FAIL 尚未完成 SKOrderLib_InitializeTS 初始動作。

int __stdcall RegisterOnTSActiveOrderCallBack ([in] long lCallbackFunction)

說明：

定義 TradeStation 訊號接收函式，此 Callback 函式定義如下：

```
void __stdcall OnTSActiveOrder( BSTR bstrSymbol, BSTR bstrDescription, BSTR  
    bstrOrderType, BSTR bstrOrder, long lLastPrice, double dTimePlaced, BSTR  
    bstrStrategy, BSTR bstrSignal, BSTR bstrWorkspace, BSTR bstrInterval, BSTR  
    bstrPositionNumber, BSTR bstrOrderNumber)
```

其中

bstrSymbol 欄位 Symbol

bstrDescription 欄位 Description

bstrOrderType 欄位 OrderType

bstrOrder 欄位 Order

lLastPrice 欄位 Last Price

dTimePlaced 欄位 Time Placed，需自行轉成字串表示。

bstrStrategy 欄位 Strategy

bstrSignal 欄位 Signal

bstrWorkspace 欄位 WorkSpace

bstrInterval 欄位 Interval

bstrPositionNumber 欄位 Position Number

bstrOrderNumber 欄位 Order Number

傳入值：

lCallbackFunction 將Callback 函式位址轉成 long 的表示方式，傳入Library 內，完成註冊程序。

回傳值：

SKSUCCESS 進行連線串接成功。

SK_ERROR_INITIALIZE_FAIL 尚未完成SKOrderLib_Initialize 初始動作。

SK_ERROR_TS_INITIALIZE_FAIL 尚未完成 SKOrderLib_InitializeTS 初始動作。

int __stdcall RegisterOnTSCanceledOrderCallBack ([in] long lCallbackFunction)

說明：

定義 TradeStation 訊號接收函式，此 Callback 函式定義如下：

```
void __stdcall OnTSCanceledOrder( BSTR bstrSymbol, BSTR bstrDescription, BSTR  
    bstrOrderType, BSTR bstrOrder, double dTimePlaced, double dTimeCanceled, BSTR  
    bstrStrategy, BSTR bstrSignal, BSTR bstrWorkspace, BSTR bstrInterval, BSTR  
    bstrPositionNumber, BSTR bstrOrderNumber, BSTR bstrCanceledNumber)
```

其中

bstrSymbol 欄位 Symbol

bstrDescription 欄位 Description

bstrOrderType 欄位 OrderType

bstrOrder 欄位 Order

dTimePlaced 欄位 Time Placed，需自行轉成字串表示。

dTimeCanceled 欄位 Time Canceled，需自行轉成字串表示。

bstrStrategy 欄位 Strategy

bstrSignal 欄位 Signal

bstrWorkspace 欄位 WorkSpace

bstrInterval 欄位 Interval

bstrPositionNumber 欄位 Position Number

bstrOrderNumber 欄位 Order Number

bstrCanceledNumber 欄位 Canceled Number

傳入值：

lCallbackFunction 將Callback 函式位址轉成 long 的表示方式，傳入Library 內，完成註冊程序。

回傳值：

SKSUCCESS 進行連線串接成功。

SK_ERROR_INITIALIZE_FAIL 尚未完成SKOrderLib_Initialize 初始動作。

SK_ERROR_TS_INITIALIZE_FAIL 尚未完成SKOrderLib_InitializeTS 初始動作。

2、回報 SKReplyLib.dll 使用說明

版本	日期	異動說明	備註
2.3.3	2010/7/7	以下函式已經失效，請勿再使用 1. SKReplyLib_Connect 2. SKReplyLib_Close 3.SKReplyLib_IsConnected	
		新增狀態 'P' 改價	

2.1 Report format

2.2 Export functions

2.1 Report Format

struct DataItem

```
{
    char caKeyNo[13];           // 委託序號
    char caMarketType[2];      // TS:證券 TA:盤後 TL:零股 TF:期貨 TO:選擇權
    char cType;                // N:委託 C:取消 U:改量 D:成交 Q:報價 P: 改價
    char cOrderErr;            // Y:失敗 T:逾時 N:正常
    char caBroker[7];          // TS,TA,TL: 分公司代號 unit no , TF,TO: IB 代號 broker id
    char caCustNo[7];          // 交易帳號
    char caBuySell[3];         // 證[0] B/S 買/賣, [1,2] 00 現股, 01 代資, 02 代券, 03 融資, 04 融券, 20 零股, 40 拍賣現股
                                // 期[0] B/S 買/賣, [1] Y 當沖, [2] I/R/F IOC / ROD / FOK
                                // 權[0] B/S 買/賣, [1] N/O 新倉 / 平倉, [2] I/R/F IOC / ROD / FOK

    char caExchangeID[7];      // 交易所
    char caComId[20];          // 前 13 碼商品代號 一個空白 後六碼商品年月
    char caStrikePrice[13];    // 履約價 七位整數
    char caOrderNo[5];         // 委託書號
    char caPrice[13];          // 證券兩位小數, 期貨三位小數,
                                // 期貨委託: 999999999999 為市價,
                                // 海外期貨, 八位整數五位小數
                                // 其餘為根據 Type 種類,「委託」為委託價;「成交」為成交價

    char caNumerator[5];       // 海外期貨回報用, 分子, 3 位整數, 2 位小數
    char caDenominator[4];     // 海外期貨回報用, 分母 4 位整數
    char caPrice1[13];         // 八位整數五位小數 委託: 觸發價 成交: 0000000000000
    char caNumerator1[5];      // 海外期貨回報用, 分子 三位整數二位小數
    char caDenominator1[4];    // 海外期貨回報用, 分母 四位整數
    char caPrice2[13];         // 八位整數五位小數 委託: 0000000000000 成交: 0000000000000
    char caNumerator2[5];      // 海外期貨回報用, 分子 三位整數二位小數
```

```
char caDenominator2[4];    // 海外期貨回報用，分母 四位整數
char caQty[8];              // TS TA 張數/ TL 股數/ TF TO 口數
char caBeforeQty[4];        // 參考欄位，異動變更前量
char caAfterQty[4];         // 參考欄位，異動變更後量
char caDate[8];             // 交易日期
char caTime[6];             // 交易時間
char caOkSeq[8];            // 成交序號
char caSubID[7];            // 子帳帳號
char caSaleNo[4];           // 營業員編號
char cAgent;                // 委託介面
};
```

2.2 Export functions

int SKReplyLib_Initialize(const TCHAR* lpszUserName, const TCHAR* lpszPassword)

說明：

在使用此 Library 前必須先通過使用者的身份認證，方可使用。此外，使用者必須先填寫申請開放使用外部 API 的申請書後才能通過初始程序，否則將會得到 SK_ERROR_PERMISSION_DENIED 的錯誤代碼。

傳入值

lpszUser 使用者代號

lpszPassword 密碼

回傳值：

0 表示初始化成功

SK_ERROR_INITIALIZE_FAIL，初始化失敗，可能是因為驗證伺服器未指定或無法與驗證伺服器做連線，所發生的錯誤

SK_ERROR_PERMISSION_DENIED，初始化的使用者沒有使用外部 API 的權限。

其他錯誤代碼，1~199 間的代碼為伺服器回傳的錯誤，請檢查 log 檔內將有錯誤原因的說明。

`int RegisterOnConnectCallBack(long lCallBackFunction)`

說明：

註冊一個 Callback function，當連線成功或失敗，會透過此 Callback 函式告知連線結果。
函式宣告方式如下：

`void __stdcall OnConnect(BSTR bstrID, int nErrorCode);`

bstrID 回傳該 callback 是通知哪一個帳號的連線狀態。

連線不論是成功或失敗，都可以透過 nErrorCode 瞭解錯誤原因。

傳入值：

lCallBackFunction 將 Callback 函式位址轉成 long 的表示方式，傳入 Library 內，完成註冊程序

回傳值：

1，表示成功

SK_ERROR_INITIALIZE_FAIL 表示尚未呼叫 SKReply_Initialize 或 使用者尚未認證成功。

OnConnect 中 nErrorCode 數值定義為：

0，Connection connected Successful

WSANOTINITIALISED，A successful AfxSocketInit must occur before using this API.

WSAENETDOWN，The Windows Sockets implementation detected that the network subsystem failed.

WSAEADDRINUSE，The specified address is already in use.

WSAEINPROGRESS，A blocking Windows Sockets call is in progress.

WSAEADDRNOTAVAIL，The specified address is not available from the local machine.

WSAEAFNOSUPPORT，Addresses in the specified family cannot be used with this socket.

WSAECONNREFUSED，The attempt to connect was rejected.

WSAEDESTADDRREQ，A destination address is required.

WSAEFAULT，The nSockAddrLen argument is incorrect.

WSAEINVAL，Invalid host address.

WSAEISCONN，The socket is already connected.

WSAEMFILE，No more file descriptors are available.

WSAENETUNREACH，The network cannot be reached from this host at this time.

WSAENOBUFS，No buffer space is available. The socket cannot be connected.

WSAENOTSOCK，The descriptor is not a socket.

WSAETIMEDOUT，Attempt to connect timed out without establishing a connection.

WSAEWOULDBLOCK，The socket is marked as nonblocking and the connection cannot be completed immediately.

以上參數定義在 `afxsock.h` 宣告中。

`int RegisterOnDataCallBack(long lCallbackFunction)`

說明：

註冊一個 Callback function，當有回報將主動呼叫函式，並通知委託的狀態。

函式宣告方式如下：

`Void __stdcall OnData(BSTR bstrData);`

回傳得字串內容，為 2.1 中的格式說明，2.1 格式中定義了每個欄位的大小，不包括字串終結字元 '\0' (NULL Terminated)，

傳入值：

lCallbackFunction 將 Callback 函式位址轉成 long 的表示方式，傳入 Library 內，完成註冊程序

回傳值：

1，表示成功

SK_ERROR_INITIALIZE_FAIL 表示尚未呼叫 SKReply_Initialize 或 使用者尚未認證成功。

OnData 回傳的內容 format，請參考 2.1 中的說明。

`int RegisterOnDisconnectCallBack(long lCallbackFunction)`

說明：

註冊一個 Callback function，當連線中斷將主動呼叫函式。

函式宣告方式如下：

`void __stdcall OnDisconnect(BSTR bstrID, int nErrorCode);`

bstrID 回傳該 callback 是通知哪一個帳號的連線狀態。

連線中斷，可以透過 nErrorCode 瞭解原因。

請注意：通知此事件時，底層尚未完成斷線的處理，因此請勿在此事件中立刻執行重新連線的動作。請收到此事件後啓用一個 **Timer**，等待數秒後再嘗試重新連線。

傳入值：

lCallbackFunction 將 Callback 函式位址轉成 long 的表示方式，傳入 Library 內，完成註冊程序。

回傳值：

1，表示成功

SK_ERROR_INITIALIZE_FAIL 表示尚未呼叫 SKReply_Initialize 或 使用者尚未認證成功。

OnDisconnect 中 nErrorCode 數值的定義：

0，Connection closed successfully.

WSAENETDOWN，The Windows Sockets implementation detected that the network subsystem failed.

WSAECONNRESET， The connection was reset by the remote side.

WSAECONNABORTED， The connection was aborted due to timeout or other failure.

以上參數定義在 afxsock.h 宣告中。

`int SKReplyLib_ConnectByID(char* lpszUserID)`

說明：

指定回報連線的使用者登入帳號。

傳入值：

`lpszUserID` 使用者 Login ID，執行前請先確認完成呼叫 `SKReply_Initialize` 的動作。

回傳值：

`SK_SUCCESS` 表示成功。

`SK_ERROR_ACCOUNT_NOT_EXIST` 指定連線的帳號不存在，請確認是否完成 `Initailze`。

int SKReplyLib_CloseByID(char* lpszUserID)

說明：

中斷指定的帳號回報連線。

傳入值：

lpszUserID 使用者 Login ID，執行前請先確認完成呼叫 SKReply_Initialize 的動作。

回傳值：

SK_SUCCESS 表示成功。

SK_FAIL 表示失敗。

SK_ERROR_ACCOUNT_NOT_EXIST 指定連線的帳號不存在，請確認是否完成 Initailze。

int SKReplyLib_IsConnectedByID(char* lpszUserID)

說明：

檢查輸入的帳號目前連線狀態。

傳入值：

lpszUserID 使用者 Login ID，執行前請先確認完成呼叫 SKReply_Initialize 的動作。

回傳值：

1 表示連線正常。

0 表示連線中斷。

SK_ERROR_ACCOUNT_NOT_EXIST 指定連線的帳號不存在，請確認是否完成 Initailze。

3、include 函式宣告

```
#define DLLImport _declspec(dllimport)
```

```
#define SK_SUCCESS 0
#define SK_FAIL -1
#define SK_ERROR_SERVER_NOT_CONNECTED -2
#define SK_ERROR_INITIALIZE_FAIL -4
#define SK_ERROR_ACCOUNT_NOT_EXIST 1001
#define SK_ERROR_ACCOUNT_MARKET_NOT_MATCH 1002
#define SK_ERROR_PERIOD_OUT_OF_RANGE 1003
#define SK_ERROR_FLAG_OUT_OF_RANGE 1004
#define SK_ERROR_BUYSELL_OUT_OF_RANGE 1005
#define SK_ERROR_ORDER_SERVER_INVALID 1006
#define SK_ERROR_PERMISSION_DENIED 1007
#define SK_ERROR_TRADE_TYPR_OUT_OF_RANGE 1008
#define SK_ERROR_DAY_TRADE_OUT_OF_RANGE 1009
#define SK_ERROR_ORDER_SIGN_INVALID 1010
#define SK_ERROR_NEW_CLOSE_OUT_OF_RANGE 1011
#define SK_ERROR_PRODUCT_INVALID 1012
#define SK_ERROR_QTY_INVALID 1013
#define SK_ERROR_DAYTRADE_DENIED 1014
#define SK_ERROR_SPICAL_TRADE_TYPE_INVALID 1015
#define SK_ERROR_PRICE_INVALID 1016
#define SK_ERROR_INDEX_OUT_OF_RANGE 1017
#define SK_ERROR_QUERY_IN_PROCESSING 1018
#define SK_ERROR_LOGIN_INVALID 1019
#define SK_ERROR_REGISTER_CALLBACK 1020
#define SK_ERROR_FUNCTION_PERMISSION_DENIED 1021
#define SK_ERROR_MARKET_OUT_OF_RANGE 1022
#define SK_ERROR_PERMISSION_TIMEOUT 1023
#define SK_ERROR_FOREIGNSTOCK_PRICE_OUT_OF_RANGE 1024
#define SK_ERROR_FOREIGNSTOCK_UNDEFINE_COINTYPE 1025
#define SK_ERROR_FOREIGNSTOCK_SAME_COINSTYPE 1026
#define SK_ERROR_FOREIGNSTOCK_SALE_SHOULD_ORIGINAL_COIN 1027
#define SK_ERROR_FOREIGNSTOCK_TRADE_UNIT_INVALID 1028
#define SK_ERROR_FOREIGNSTOCK_STOCKNO_INVALID 1029
```

#define SK_ERROR_FOREIGNSTOCK_ACCOUNTTYPE_INVALID	1030
#define SK_ERROR_FOREIGNSTOCK_INITIALIZE_FAIL	1031
#define SK_ERROR_TS_INITIALIZE_FAIL	1032
#define SK_ERROR_OVERSEA_TRADE_PRODUCT_FAIL	1033
#define SK_ERROR_OVERSEA_TRADE_DATA_NOT_COMPLETE	1034
#define SK_WARNING_OF_COM_DATA_MISSING	2001


```

// import from SKOrderLib.dll
DLLImport int __stdcall SKOrderLib_Initialize( char* pcUserName, char* pcPassword);
DLLEExport int __stdcall SKOrderLib_ReadCertByID( TCHAR* lpszUserID);
DLLImport int __stdcall SendStockOrder( const TCHAR* lpszAccount, const TCHAR*
    lpszStockNo, unsigned short usPeriod, unsigned short usFlag, unsigned short usBuySell,
    const TCHAR* lpszPrice, int nQty, TCHAR* lpszMessage, int* nMessageBufferSize);
DLLImport int __stdcall CancelOrderBySeqNo( const TCHAR* lpszAccount, const TCHAR*
    lpszSeqNo);
DLLImport int __stdcall SendFutureOrder( const TCHAR* lpszAccount, const TCHAR*
    lpszStockNo, unsigned short usTradeType, unsigned short usDayTrade, unsigned short
    usBuySell, const TCHAR* lpszPrice, int nQty, TCHAR* lpszMessage, int*
    nMessageBufferSize);
DLLImport int __stdcall SendOptionOrder( const TCHAR* lpszAccount, const TCHAR*
    lpszStockNo, unsigned short usTradeType, unsigned short usNewClose, unsigned short
    usBuySell, const TCHAR* lpszPrice, int nQty, TCHAR* lpszMessage, int&
    nMessageBufferSize);
DLLImport int __stdcall CancelOrderByStockNo( const TCHAR* lpszStockNo);
DLLImport int __stdcall DecreaseOrderBySeqNo( const TCHAR* lpszSeqNo, int
    nDecreaseQty);
DLLImport int __stdcall RegisterOnOpenInterestCallBack( long lCallbackFunction);
DLLImport int __stdcall GetOpenInterest( TCHAR* lpszAccount);
DLLImport int __stdcall RegisterOnAccountCallBack( long lCallbackFunction);
DLLImport int __stdcall GetUserAccount();
DLLImport int __stdcall SendForeignStockOrder( const TCHAR* lpszAccount, const
    TCHAR* lpszStockNo, const TCHAR* lpszExchangeNo,
    unsigned short usBuySell, const TCHAR* lpszPrice, int nQty, const TCHAR*
    lpszCurrency1, const TCHAR* lpszCurrency2, const TCHAR* lpszCurrency3, int
    iAccountType, TCHAR* lpszMessage, int* pnMessageBufferSize);

DLLImport int __stdcall SendForeignStockOrderAsync( const TCHAR* lpszAccount, const
    TCHAR* lpszStockNo, const TCHAR* lpszExchangeNo, unsigned short usBuySell,
    const TCHAR* lpszPrice, int nQty, const TCHAR* lpszCurrency1, const TCHAR*
    lpszCurrency2, const TCHAR* lpszCurrency3, int iAccountType);

DLLImport int __stdcall CancelForeignStockOrderByBookNo( const TCHAR* lpszAccount,
    const TCHAR* lpszBookNo, const TCHAR* lpszExchangeNo);

DLLImport int __stdcall CancelForeignStockOrderBySeqNo( const TCHAR* lpszAccount,

```

```
const TCHAR* lpszSeqNo,const  TCHAR* lpszExchangeNo);
DLLImport int __stdcall SKOrderLib_InitializeTS( TCHAR* lpszTSCOMName);

// import from SKReplyLib.dll
DLLImport int SKReplyLib_Initialize( const TCHAR* lpszUserName, const TCHAR*
lpszPassword);
DLLImport int __stdcall RegisterOnConnectCallBack( long lCallbackFunction);
DLLImport int __stdcall RegisterOnDataCallBack( long lCallbackFunction);
DLLImport int __stdcall RegisterOnDisconnectCallBack( long lCallbackFunction);
DLLImport int __stdcall SKReplyLib_ ConnectByID( char* lpszUserID);
DLLImport int __stdcall SKReplyLib_Close ByID( char* lpszUserID) ;
DLLImport int __stdcall SKReplyLib_IsConnectedByID( char* lpszUserID);
```