

台新證券
證券 API 下單
規格說明書
Ver 22.04.06

2022 年 04 月 06 日

目錄

1. 目的：	5
2. 元件使用說明：	5
2.1.C# 撰寫說明	5
2.2.VB 撰寫說明	7
2.3.Excel VBA 撰寫說明	8
2.4.元件相關說明	8
3. API 元件功能說明：	9
3.1.設定相關功能	10
3.1.1. 元件初始化(Init)	10
3.1.2. 結束使用元件(Fini)	10
3.1.3. 登入(Login)	11
3.1.4. 登出(Logout)	11
3.1.5. 連線(Connect)	12
3.1.6. 斷線(Disconnect)	12
3.1.7. 整體初始化(Init2)	13
3.1.8. 完整結束元件(Fini2)	13
3.1.9. 取得帳號資訊(GetAccount)	14
3.1.10. 設定 LOG 目錄(SetLogPath)	15
3.1.11. 設定下單類型(SetExecOrderType)	16
3.1.12. 設定交易單位(SetLotSizeData)	16
3.1.13. 設定 API 回傳類型(SetEchoType)	17
3.2.交易相關功能	18
3.2.1. 證券	18
3.2.1.1. 證券新單(Stock_NewOrder)	19
3.2.1.2. 證券改單、改價(Stock_ModifyOrder)	20
3.2.1.3. 證券刪單(Stock_CancelOrder)	21
3.2.1.4. 證券委託查詢(Stock_QueryOrder)	22
3.2.1.5. 證券成交查詢(Stock_QueryMatch)	22
3.2.1.6. 證券庫存查詢(Stock_QueryPosition)	23
3.3.帳務查詢(MIX_QueryData)	24
3.4.元件的非同步回傳格式	25
3.4.1. C# 建立 Call Back Function 方式	25
3.4.2. VB 建立 Call Back Function 方式	25
3.4.3. Callback 事件分類	26
附錄 A. 錯誤代碼	27
附錄 B. 主動回報回傳格式	28
附錄 C. 一般回傳格式	30
附錄 D. 帳號回傳格式	37

1. 目的：

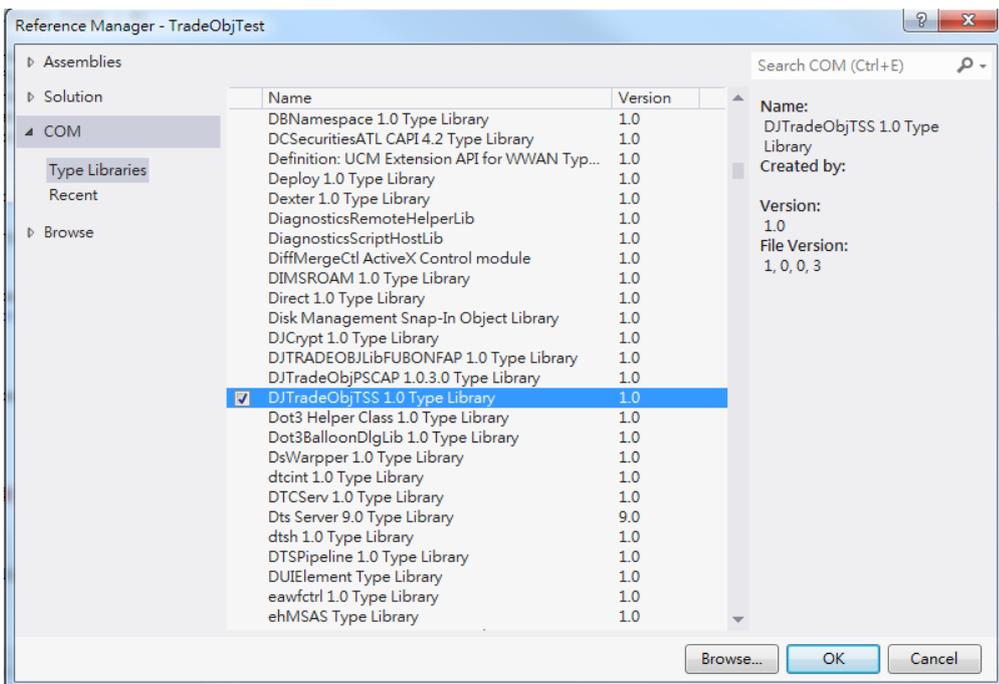
需安裝超級智多星，但使用時無需開啟超級智多星，提供交易 API 元件讓 Third Party 可以自行開發程式下單，使用 COM 方法，進行連接。

2. 元件使用說明：

安裝擁有交易 API 的超級智多星以後，將會擁有 apiTSS.dll 此檔案。安裝的同時會將此 COM 元件註冊。撰寫程式時，只要將此 COM 元件進行引用，就可以使用此元件，進行交易與查詢。

2.1.C# 撰寫說明

使用 C# 語言進程式撰寫，必須先將 DJTradeObjTSS 1.0 Type Library 這個 COM 元件加入到 Reference 中。如下圖。



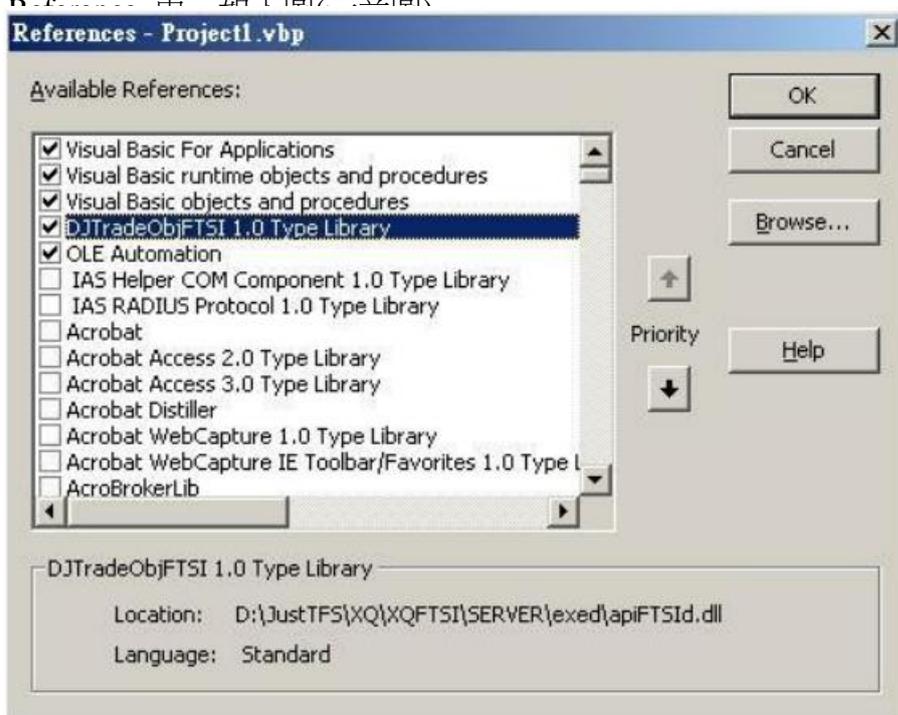
接著使用程式碼 `using DJTRADEOBJLibTSS;` 來使用 COM 元件。再使用

DJTRADEOBJLibTSS.TradeAppClass m_tradeApp; 宣告一個使用 **interface** 的物件。並且定義此物件 `m_tradeApp = new TradeAppClass();`。最後就可以使用元件所提供的 **API**。

Ex: `m_tradeApp.Init(strTradeDAS, out nErrCode, out sErrMsg);`

2.2.VB 撰寫說明

使用 VB 語言進行程式撰寫，必須先將 DJTradeObjTSS 1.0 Type Library 這個 COM 元件加入到



接著宣告與定義，使用此 COM 的物件。

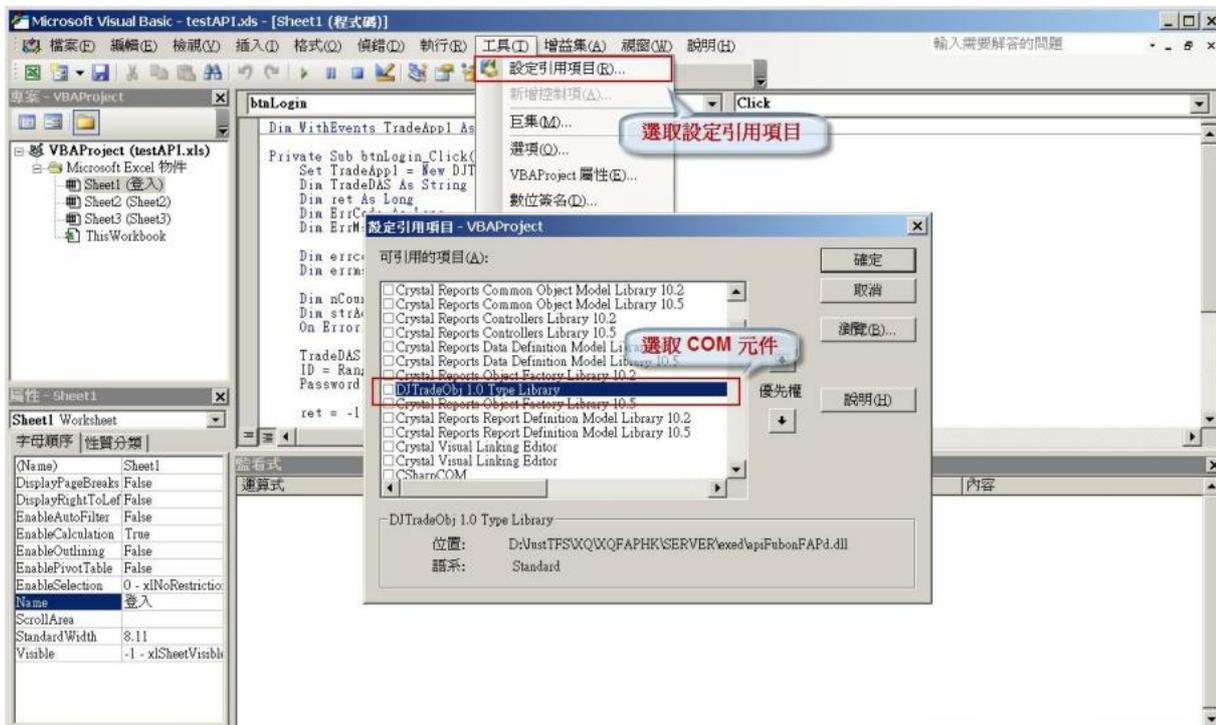
```
Dim WithEvents TradeApp1 As DJTRADEOBJLibTSS.TradeApp
```

Set TradeApp1 = New DJTRADEOBJLibTSS.TradeApp 最後使用 COM 元件提供的 API。

```
Ex: ret = TradeApp1.Init2(TradeDAS, txtUID, txtPwd)
```

2.3.Excel VBA 撰寫說明

使用 Excel VBA 方法類似 VB 撰寫方法。在新增 COM 到 Reference 的方法，請參考下圖。



2.4.元件相關說明

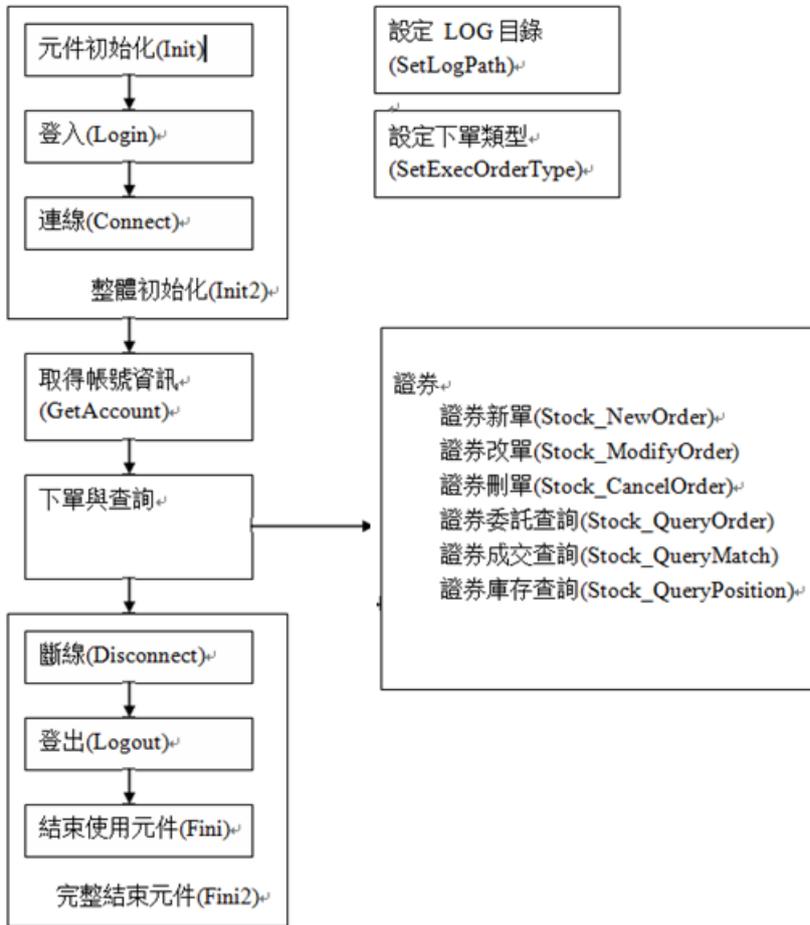
元件的變數型態 BSTR 對等 C# 和 VB 的 String 型態。

3. API 元件功能說明：

API 元件功能分為三大類：設定相關、交易相關、帳務相關。設定相關功能，主要負責，初始化，登入登出，主動回報連線，帳號取得，Log 目錄設定，Socket 下單設定。交易相關包涵，證券的新單、刪改單、委託查詢、成交查詢、庫存查詢、未平倉查詢。帳務相關則是帳務類別的查詢。

流程運作可分為三部份：初始化元件、交易與查詢、結束元件。初始化元件包涵 Init、Login、Connect 這三部份可以直接呼叫 Init2 一次完成。交易與查詢，則需要取得帳號並且呼叫提供的交易 API 與查詢 API。結束元件則需要 Disconnect、Logout、Fini 這三個動作，也可以直接呼叫 Fini2 會直接完成此三個步驟。

API 運作流程如下：



3.1. 設定相關功能

初始元件呼叫順序必須依照：Init、Login，接著才可以呼叫 Connect、GetAccount、SetLogPath、SetExecOrder，結束時必須依照順序，Disconnect、Logout、Fini。

初始元件也可以直接呼叫 Init2，這個 Function 將會依照順序呼叫，Init、Login、Connect，一次做完。而結束時可呼叫 Fini2，將會依照順序呼叫 Disconnect、Logout、Fini。

3.1.1. 元件初始化(Init)

名稱：Init

說明：元件開始初始化，除了 SetLogPath 以外，其他的呼叫都必須在這個完成之後才可以呼叫。

參數：連線主機為取得交易主機位置的伺服器，通常與交易主機同一台。

Init(

```
[in]BSTR sTradeDAS,    // 連線主機 Ex. goldtest.ftsi.com.tw/tradedas
[out]long* pnErrCode,
[out]BSTR* psErrMsg,
[out, retval]long* pnRet );
```

回傳：1 成功, 0 失敗

C#

```
int nRet = m_tradeApp.Init(strTradeDAS, out nErrCode, out sErrMsg);
```

VB

```
Dim ret As Long, TradeDAS As String, ErrCode As String, ErrMsg As String ret =
TradeApp1.Init(TradeDAS, ErrCode, ErrMsg)
```

3.1.2. 結束使用元件(Fini)

名稱：Fini

說明：元件使用完，要結束程式時，使用這個可以釋放記憶體。完整結束參數：無

回傳：無

C#

```
m_tradeApp.Fini();
```

VB

```
TradeApp1.Fini
```

3.1.3. 登入(Login)

名稱：Login

說明：Init 呼叫完畢後才可以執行此 API，功能為登入交易主機，取得交易設定，主動回報伺服器位址，以及交易帳號。

參數：

```
Login(  
    [in]BSTR sUID,    // 帳號  
    [in]BSTR sPassword, // 密碼  
    [out]long* pnErrCode, // 回傳錯誤碼  
    [out]BSTR* psErrMsg, // 回傳錯誤訊息  
    [out, retval]long* pnRet  
);
```

回傳：1 成功, 0 失敗

C#

```
int nRet = m_tradeApp.Login(txtUID.Text, txtPwd.Text, out nErrCode, out sErrMsg)
```

VB

```
Dim ret As Long, ErrCode As String, ErrMsg As String  
ret = TradeApp1.Login(txtUID.Text, txtPwd.Text, ErrCode, ErrMsg)
```

3.1.4. 登出(Logout)

名稱：Logout

說明：向交易主機送出登出訊息。結束交易連線。

參數：

```
Logout(  
    [in]BSTR sUID    // 帳號  
);
```

回傳：無

C#

```
m_tradeApp.Logout(txtUID.Text);
```

VB

```
ret = TradeApp1.Logout(txtUID.Text)
```

3.1.5. 連線(Connect)

名稱：Connect

說明：主動回報連線，Socket 下單連線，必須在 Login API 呼叫完畢後才可執行。

參數：

```
Connect(  
    [out, retval]long* pnRet  
);
```

回傳：1 成功, 0 失敗

C#

```
int nRet = m_tradeApp.Connect();
```

VB

```
Dim ret As Long
```

```
ret = TradeApp1.Connect
```

3.1.6. 斷線(Disconnect)

名稱：Disconnect

說明：登出主動回報連線。參數：無

回傳：無

C#

```
m_tradeApp.Disconnect();
```

VB

```
TradeApp1.Disconnect
```

3.1.7. 整體初始化(Init2)

名稱：Init2

說明：一次完成 Init、Login、Connect 三個 API 所做的事情。可以搭配 Fini2 來執行。

參數：連線主機為取得交易主機位置的伺服器，通常與交易主機同一台。

```
Init2(  
    [in]BSTR sTradeDAS, // 連線主機 Ex. goldtest.ftsi.com.tw/tradedas  
    [in]BSTR sUID, // 帳號  
    [in]BSTR sPassword, // 密碼  
    [out]long* pnErrCode, // 錯誤代碼  
    [out]BSTR* psErrMsg, // 錯誤訊息  
    [out, retval]long* pnRet  
);
```

回傳：1 成功, 0 失敗

C#

```
int nRet = m_tradeApp.Init2(strTradeDAS, txtUID.Text, txtPwd.Text, out nErrCode, out sErrMsg);
```

VB

```
Dim ret As Long, TradeDAS As String, ErrCode As String, ErrMsg As String  
ret = TradeApp1.Init2(TradeDAS, txtUID.Text, txtPwd.Text, ErrCode, ErrMsg)
```

3.1.8. 完整結束元件(Fini2)

名稱：Fini2

說明：一次完成 Disconnect、Logout、Fini 三個 API 所做的事情。可搭配 Init2 來執行。

參數：

```
Fini2(  
    [in]BSTR sUID // 帳號  
);
```

回傳：無

C#

```
m_tradeApp.Fini2(txtUID.Text);
```

VB

```
TradeApp1.Fini2 (txtUID.Text)
```

3.1.9. 取得帳號資訊(GetAccount)

帳號取得有兩個方法，一個是先取得帳號數量，接著一筆一筆取出，可以依照不同的 **Index** 取得相對應的帳號。方法二是一次取得所有的帳號。可以直接在外面對帳號進行處理。

名稱：GetAccountCount

說明：取得所有帳號的數量。

參數：

```
GetAccountCount(  
    [out, retval]long* pnRet  
);
```

回傳：整數數量

C#

```
int nCount = m_tradeApp.GetAccountCount();
```

VB

```
Dim nCount As Long
```

```
nCount = TradeApp1.GetAccountCount
```

名稱：GetAccount

說明：取得 **idx** 相對應的帳號。

參數：

```
GetAccount(  
    [in]long idx,  
    [out, retval]BSTR* psResult  
);
```

回傳：請參照「附錄 D. 帳號回傳格式」

C#

```
int i;
```

```
string AccountData = m_tradeApp.GetAccount(i);
```

VB

```
Dim i As Long, strAccount As String strAccount = TradeApp1.GetAccount(i)
```

名稱：GetAccountList

說明：一次取出所有帳號資料。

參數：

```
GetAccountList(  
    [out, retval]BSTR* psResult  
);
```

回傳：請參照「附錄 D. 帳號回傳格式」

C#

```
string AccountData = m_tradeApp.GetAccountList;
```

VB

```
strAccount As String
```

```
strAccount = TradeApp1.GetAccountList
```

3.1.10. 設定 LOG 目錄(SetLogPath)

名稱：SetLogPath

說明：必須在呼叫 Init 以前設定，不然就會使用預設路徑。

參數：

```
SetLogPath(  
    [in]BSTR sLogPath,    // 絕對路徑  
    [out, retval]BSTR* psError  
);
```

回傳：空字串成功，其他為錯誤訊息。

C#

```
string sLogPath;
```

```
string sResult = m_tradeApp.SetLogPath(sLogPath);
```

VB

```
Dim sResult As String, sLogPath As String sResult = TradeApp1.SetLogPath(sLogPath)
```

3.1.11. 設定下單類型(SetExecOrderType)

SetExecOrderType(long lType)名稱：SetExecOrderType

說明：設定是走 Http 下單，或是 Socket 下單。

參數：

```
SetExecOrderType(  
    [in]long lType // 0: Auto, 1: HTTP, 2: Socket  
);
```

回傳：無

C#

```
int nType = 0;  
m_tradeApp.SetExecOrderType(nType);
```

VB

```
Dim lType As Long TradeApp1.SetExecOrderType(lType)
```

3.1.12. 設定交易單位(SetLotSizeData)

名稱：SetLotSizeData

說明：交易單位非千股的商品，可以在此設定。

參數：

```
SetLotSizeData(  
    [in]BSTR sLotSizeData // Data: StockID1=LotSize1|StockID2=LotSize2  
);
```

回傳：無

C#

```
string strLotSizeData = "0050=300|0028=200";  
m_tradeApp.SetLotSizeData(strLotSizeData);
```

VB

```
Dim strLotSizeData As String  
TradeApp1.SetLotSizeData(strLotSizeData)
```

3.1.13. 設定 API 回傳類型(SetEchoType)

名稱：SetEchoType

說明：API 呼叫的回傳類型分為兩類：同步與非同步。採用同步回傳時，呼叫會等 API 執行完後，回傳完整資料。採用非同步回傳時，當呼叫 API 後，COM 元件會產生一個 cookie，等執行完後，會執行 Call Back Function，將結果傳回。此時再依照 cookie 將執行的 API 與對應的結果做結合。API 呼叫類型分為三類：分別是 Order(下單)、QueryTrade(當日交易相關查詢)、QueryData(其他查詢)。可針對不同的類型做不同的回傳設定。

參數：

```
SetEchoType(  
    [in]long lCmdType, // 1: Order, 2: QueryTrade, 3: QueryData  
    [in]long lEchoType // 0: Sync, 1: ASync  
);
```

回傳：無

C#

```
int nCmdType = 1, nEchoType = 0;  
m_tradeApp.SetEchoType(nCmdType, nEchoType);
```

VB

```
Dim CmdType As Long, EchoType As Long  
TradeApp1.SetEchoType(CmdType, EchoType)
```

3.2.交易相關功能

目前提供證券的交易模組，

提供項目為

新單、

刪改單、

當日委託查詢、

當日成交查詢、

即時庫存查詢。

其他帳務查詢，必須提出新需求，才提供相關類型查詢。

3.2.1. 證券

使用參數說明：

證券帳號：券商分公司代號 - 證券帳號 Ex: 538L-1234567

交易日期：合法交易日，格式為年月日 Ex: 20110615

股號：後台所需要的交易商品 Ex: 0050

強制查詢：nForceQuery，不使用元件快取資料，

一定會向伺服器拿取資料價格：可以有小數點的字串 Ex: 90.5

3.2.1.1. 證券新單(Stock_NewOrder)

名稱：Stock_NewOrder

說明：證券下單。根據不同下單類別型態，輸入不同參數。

參數：

```
Stock_NewOrder(  
    [in]BSTR sAccountID,    // 證券帳號  
    [in]long lTradeDate,    // 交易日期(西元年月日) 預約單帶下個交易日  
    [in]long nTT,           // 0: 普通/ 1:盤後零股/ 2:盤後/ 5:興櫃/ 7:盤中零股  
    [in]long nOT,           // 0: 現股/ 1:融資/ 2: 融券/ 16:現股先賣  
    [in]long nBS,           // 1: 買進/ 2: 賣出/ 3:超買/ 4:超賣 (3/4 for 興櫃)  
    [in]BSTR sStockID,     // 股號  
    [in]long lQty,          // 普通&盤後(張數) / 零股(股數) / 興櫃(股數)  
    [in]long nPT,           // 0: 限價/ 1: 漲停/ 2: 跌停/ 3: 平盤/ 4: 市價  
    [in]BSTR sPrice,        // 價格: 非限價帶 0  
    [in]BSTR sBroker,       // 推薦券商 for 興櫃  
    [in]long nPayType,       // 0:餘額交割/ 1:逐筆交割 (for 興櫃)  
    [in]long nOrderType,    // 0: 預約單/ 1:盤中單  
    [in]long nCond,         // 0: ROD/ 1:FOK/ 2: IOC  
    [out, retval]BSTR* psResult  
);
```

回傳：請參照「附錄 C. 一般回傳格式的證券新單回傳格式」

C#

```
string strResult = m_tradeApp.Stock_NewOrder(szAccountID, m_nTradeDate, nTT, nOT, nBS,  
szStockID, nQty, nPT, szPrice, szBroker, nPayType);
```

VB

```
Dim Result As String
```

```
Result = TradeApp1.Stock_NewOrder(AccountID, TradeDate, TT, OT, BS, StockID, Qty, PT,  
Price, Broker, PayType)
```

3.2.1.2. 證券改單、改價(Stock_ModifyOrder)

名稱：Stock_ModifyOrder 說明：證券改單、改價。

參數：

```

Stock_ModifyOrder(
    [in]BSTRT sAccountID, // 證券帳號
    [in]long nType,       // 0:改量 2:改價
    [in]long nTradeDate, // 交易日期(西元年月日) 預約單帶下個交易日
    [in]long nTT,        // 0: 普通/ 1: 零股/ 2: 盤後/ 5:興櫃/ 7:盤中零股
    [in]long nOT,        // 0: 現股/ 1:融資/ 2: 融券
    [in]BSTRT sOID,      // 委託單編號
    [in]BSTRT sOrderNo,  // 委託書號
    [in]BSTRT sStockID,  // 股號
    [in]long nBS,        // 1: 買進/ 2: 賣出/ 3:超買/ 4:超賣 (3/4 for 興櫃)
    [in]long nQty,       // 取消數量/取消後股數,
                        // 預約單/零股為取消後數量, 0=刪單
    [in]long nQcurrent,  // 有效張數
    [in]long nQmatch,    // 成交張數
    [in]long nPreOrder,  // 1: 預約單/ 0:盤中單
    [in]BSTRT sPrice,    // 改價:新價位 改量:原價位
    [in]BSTRT sPriceType, // 0: 限價/ 1: 漲停/ 2: 跌停/ 3: 平盤
                        // 改價:新價位旗標 改量:原價位旗標
    [in]long nCond,      // 0: ROD/ 1:FOK/ 2: IOC
                        // 改價:新 Cond 改量:原 Cond

    [out, retval]BSTRT* psResult
);
    
```

回傳：請參照「附錄 C. 一般回傳格式的證券改單回傳格式」

C#

```

string strResult = m_tradeApp.Stock_ModifyOrder(szAccountID, 0, m_nTradeDate, nTT,
nOT,szOID, szOrderNo, txtStockID.Text, nBS, nQty, nQCurrent, nQMatch, nPreOrder, szPrice,
nPT,nCond);
    
```

VB

```
Dim Result As String
```

```
Result = TradeApp1.Stock_ModifyOrder(AccountID, TradeDate, TT, OT, szOID, szOrderNo,
szStockID, BS, Qty, QCurrent, QMatch, PreOrder, NewPrice)
```

3.2.1.3. 證券刪單(Stock_CancelOrder)

名稱：Stock_CancelOrder 說明：證券刪單

參數：

```
Stock_CancelOrder(  
    [in]BSTR sAccountID, // 證券帳號  
    [in]long nTradeDate, // 交易日期(西元年月日) 預約單帶下個交易日  
    [in]long nTT, // 0: 普通/ 1: 零股/ 2: 盤後/ 5:興櫃/ 7:盤中零股  
    [in]long nOT, // 0: 現股/ 1:融資/ 2: 融券  
    [in]BSTR sOID, // 委託單編號  
    [in]BSTR sOrderNo, // 委託書號  
    [in]BSTR sStockID, // 股號  
    [in]long nBS, // 1: 買進/ 2: 賣出/ 3:超買/ 4:超賣 (3/4 for 興櫃)  
    [in]long nQty, // 0=刪單  
    [in]long nQcurrent, // 有效張數  
    [in]long nQmatch, // 成交張數  
    [in]long nPreOrder, // 1: 預約單/ 0: 盤中單  
    [in]long nCond, // 0: ROD/ 1:FOK/ 2: IOC  
    [in]BSTR sPrice, // 委託單 價格  
    [out, retval]BSTR* psResult  
);
```

回傳：請參照「附錄 C. 一般回傳格式的證券刪單回傳格式」

C#

```
string strResult = m_tradeApp.Stock_CancelOrder(szAccountID, m_nTradeDate, nTT, nOT,  
szOID, szOrderNo, txtStockID.Text, nBS, nQty, nQCurrent, nQMatch, nPreOrder);
```

VB

```
Dim Result As String
```

```
Result = TradeApp1.Stock_CancelOrder(AccountID, TradeDate, TT, OT, szOID, szOrderNo,  
szStockID, BS, Qty, QCurrent, QMatch, PreOrder)
```

3.2.1.4. 證券委託查詢(Stock_QueryOrder)

名稱：Stock_QueryOrder

說明：證券委託查詢，可以使用 nForceQuery 設定是否使用快取資料。

參數：

```
Stock_QueryOrder(  
    [in]BSTR sAccountID, // 證券帳號  
    [in]long nForceQuery, // 0:user cache/ 1:force query [out, retval]BSTR* psResult  
);
```

回傳：請參照「附錄 C. 一般回傳格式的證券委託查詢回傳格式」

```
C#  
string strResult = m_tradeApp.Stock_QueryOrder(szAccountID, nForceQuery);  
VB  
Dim Result As String, ForceQuery As Long  
Result = TradeApp1.Stock_QueryOrder(AccountID, ForceQuery)
```

3.2.1.5. 證券成交查詢(Stock_QueryMatch)

名稱：Stock_QueryMatch

說明：證券成交查詢，可以使用 nForceQuery 設定是否使用快取資料。

參數：

```
Stock_QueryMatch(  
    [in]BSTR sAccountID, // 證券帳號  
    [in]long nForceQuery, // 0:user cache/ 1:force query [out, retval]BSTR* psResult  
);
```

回傳：請參照「附錄 C. 一般回傳格式的證券成交查詢回傳格式」

```
C#  
string strResult = m_tradeApp.Stock_QueryMatch(szAccountID, nForceQuery);  
VB  
Dim Result As String, ForceQuery As Long  
Result = TradeApp1.Stock_QueryMatch(AccountID, ForceQuery)
```

3.2.1.6. 證券庫存查詢(Stock_QueryPosition)

名稱：Stock_QueryPosition

說明：證券庫存查詢，可以使用 nForceQuery 設定是否使用快取資料。

參數：

```
Stock_QueryPosition(  
    [in]BSTR sAccountID, // 證券帳號  
    [in]long nTradeDate, // 交易日期(西元年月日) 可帶今天  
    [in]long nForceQuery, // 0:user cache/ 1:force query [out, retval]BSTR* psResult  
);
```

回傳：請參照「附錄 C. 一般回傳格式的證券庫存查詢回傳格式」

C#

```
string strResult = m_tradeApp.Stock_QueryPosition(szAccountID, m_nTradeDate, nForceQuery);
```

VB

```
Dim Result As String, ForceQuery As Long
```

```
Result = TradeApp1.Stock_QueryPosition(AccountID, TradeDate, ForceQuery)
```

3.3.帳務查詢(MIX_QueryData)

名稱：MIX_QueryData

說明：綜合查詢，必須提出新需求，再提供所需要的參數。

參數：

```
MIX_QueryData(  
    [in]BSTR sAccountID, // 帳號  
    [in]BSTR sAPI, // API 名稱  
    [in]BSTR sParamString, // 參數字串  
    [out, retval] BSTR* psResult  
);
```

回傳：

--

3.4.元件的非同步回傳格式

非同步回傳是當呼叫 API 的時候，最終結果不會直接從呼叫 API 的結果中得到。API 呼叫的結果會得到一個 cookie 值，等到 COM 元件執行此 Request 結束後，才會透過 Call Back Function 將結果傳回來。因此必須實做 Call Back Function。

Call Back Function 的 Interface 定義如下 OnDataResponse([in]long nEventID, [in]BSTR sResponseData);

3.4.1. C# 建立 Call Back Function 方式

可以使用 C# Delegate 的方式來撰寫。

```
m_tradeApp.OnDataResponse += new  
_ITradeAppEvents_OnDataResponseEventHandler(m_tradeApp_OnDataResponse);
```

說明：

m_tradeApp.OnDataResponse：這個事件是 COM 元件所提供的事件 Interface。此名稱不可作修改。

_ITradeAppEvents_OnDataResponseEventHandler：此為 C# 經過處理的事件處理的 Class。此名稱不可作修改。

m_tradeApp_OnDataResponse：此為自訂的 Call Back Function 名稱。

可以自行定義所需要的名稱。此 Function 的功能為處理回傳的資料。
參數有兩個 int nEventID, string sResponseData。

3.4.2. VB 建立 Call Back Function 方式

宣告制式化的 Function 名稱，如果宣告的 member 是 Dim WithEvents TradeApp1 As DJTRADEOBJLibFTSI.TradeApp，則 Function Name 必須宣告為 TradeApp1_OnDataResponse(ByVal nEventID As Long, ByVal sResponseData As String) 如此就可以實做 Call Back Function 的內容。

3.4.3. Callback 事件分類

COM 要求 Call Back Function 的參數有兩個。分別是 int nEventID 和 string sResponseData。EventID 是事件代碼，ResponseData 是此事件回傳的資料。

事件分類如下：

- 事件代碼：1
事件名稱：主動回報連線狀態
資料內容：AccType,Status (Ex:1.1)
AccType: 1(證券)
Status: 0(斷線), 1(連線)
- 事件代碼：10
事件名稱：商品檔更新
資料內容：請參照「附錄 E. 商品檔格式」
- 事件代碼：100
事件名稱：主動回報
資料內容：請參照「附錄 B. 主動回報回傳格式」
- 事件代碼: 101
事件名稱: 下單 回傳(非同步)
資料內容: 請參照「附錄 C. 一般回傳格式」
- 事件代碼: 102
事件名稱: 交易查詢 回傳(非同步)
資料內容: 請參照「附錄 C. 一般回傳格式」
- 事件代碼: 103
事件名稱: 帳務查詢 回傳(非同步)
資料內容: 請參照「附錄 C. 一般回傳格式」
- 事件代碼: 201
事件名稱: Sunrise(清盤)
事件說明: 清盤時間，此時回傳資料僅剩預約單。
- 事件代碼: 202
事件名稱: 資料改變通知
事件說明: 預約轉盤中資料類型轉變

附錄 A. 錯誤代碼

錯誤代碼	代表意義
0	無錯誤
1	執行逾時
9000	未定義的錯
9001	系統內部錯
9002	不合法的參數
9003	未支援的功能
9100	網路錯誤
9200	結果錯誤

附錄 B. 主動回報回傳格式

主動回報狀態請參考證券 F2 欄位的 OP。根據不同的 OP 會有不同的內容（譬如 OP: 40 成交回報時，證券主動回報 F18 才会有值）。此外只有盤中單才会有主動回報。預約單不会有主動回報。

OP 定義：

成功	11: 委託成功, 21: 刪單成功, 31: 改量成功, 40: 成交 61: 改價成功 01: 預約成功, 03: 預約取消 04: 預約刪單成功, 05: 預約刪單失敗, 06: 預約刪改取消 07: 預約改單成功, 08: 預約改單失敗
失敗	02: 預約失敗, 12: 委託失敗, 22: 刪單失敗, 32: 改量失敗 62: 改價失敗

回傳格式：<F0=...|F1=...|F2=...|...|Fn=...>

證券欄位定義

F0	帳號
F1	回報類別 03 證券
F2	OP
F3	Broker ID
F4	Account ID
F5	委託書號
F6	普通'0'/零股'2'/盤後'3'/興櫃'4'/盤中零股'7'
F7	現股'0'/融資'3'/融券'4'
F8	股票代號
F9	價位
F10	限價'0'/漲停價'1'/跌停價'2'/平盤價'3'/市價'4'/超限價'5'
F11	B/S
F12	數量, 改量: 更改後數量, 刪單: 取消數量, 均不含成交
F13	只有改量有此欄位, 改量前數量, 均不含成交
F14	委託/成交回報日期
F15	委託/成交回報時間
F16	營業員
F17	資料來源
F18	交易所成交序號 9(8)
F19	網路序號 X(8)
F20	'1'餘額交割/'2'逐筆交割
F21	推薦券商代號
F22	智慧下單類型, 1:通知,2:一般單回報,3:智慧單回報
F23	智慧下單網路序號
F24	子帳號
F25	子帳號成交序號
F26	'1': 表成交回報時皆為股數, '0':表成交回報時根據 F6 決定
F27	Error Message
F28	委託條件: FOK'F'/IOCT'/ROD'R'
F29	委託/成交回報時間至毫秒 (HHMMSS.fff)
<p><F0=20601-0101093 F1=03 F2=11 F3=9661 F4=0101093 F5=X0001 F6=0 F7=0 F8=1108 F9=00000743 F10=0 F11=B F12=00000001 F13=00000000 F14=20110411 F15=094922 F16=9 02 F17=1 F18= F19=633350 F20= F21= F22=2 F23=0000000000 F24= F25= F26=0 F27=></p>	

附錄 C. 一般回傳格式

回傳格式：

```
<rc=#code|cookie=#cookie|err=#errcode|msg=#message|count=#record_count>
<#record0><#record1><#record2>...
```

格式說明：

rc	1 表示成功, 0 表示失敗
cookie	Request 編號: cookie 是每一次重啟 API, 對每一次 Request 都編一個號碼, 用於紀錄 Request。編號在一次啟動 API 之內是連續, 且不重複的。
err	錯誤碼
msg	訊息, 可能是錯誤訊息或警示訊息
count	資料筆數
record	資料內容

資料內容格式：<#record> = <F0=...|F1=...|F2=...|...|Fn=...>

資料編碼處理：編碼為 Big5 (可包涵中文字)

編碼前	編碼後
"&"	"&#38;"
"<"	"<"
">"	">"
" "	"|"
"="	"="

證券新單回傳格式

F0	1 預約單, 0 盤中單
F1	交易日期
F2	委託書號
F3	委託編號
範例: <F0=0 F1=20110408 F2=X0001 F3=12345>	

請參照「3.2.1.1 證券新單」

證券改單回傳格式

F0	1 預約單, 0 盤中單
F1	委託書號
F2	交易日期
F3	委託編號
F4	Mode 委託方式
F5	OrderStatus 委託狀態
F6	Code 錯誤編碼
F7	CodeMsg 錯誤訊息
範例: <F0=0 F1=20110408 F2=X0001 F3=12345 F4=1 F5=1 F6= F7=>	

請參照「3.2.1.2 證券改單、改價」

證券刪單回傳格式

F0	1 預約單, 0 盤中單
F1	交易日期
F2	委託書號
F3	委託編號
F4	Mode 委託方式
F5	bValid
F6	Code 錯誤編碼
F7	CodeMsg 錯誤訊息
範例: <F0=0 F1=20110408 F2=X0001 F3=12345 F4=1 F5=1 F6= F7=>	

請參照「3.2.1.3 證券刪單」

證券委託查詢回傳格式

F0	szAccountID 證券帳號
F1	nTDate 交易日期，預約單為下個交易日
F2	nTT, 0: 普通/ 1: 盤後零股/ 2: 盤後/ 5:興櫃 / 7:盤中零股
F3	nOT, 0: 現股/ 1:融資/ 2: 融券
F4	nBS, 買賣別 1: 買進/ 2: 賣出
F5	szStockID, 商品代號
F6	nPT, 0: 限價/ 1: 漲停/ 2: 跌停/ 3: 平盤/ 4: 市價
F7	szPrice, 價位
F8	nQty, 原委託張數/股數 (除零股為股數外，其餘為張數。)
F9	szOrderNo, 委託書號
F10	nQtyMatch, 已成交張數
F11	nQtyCancel, 已取消張數
F12	nOrderDate, 委託日期 (YYYYMMDD)
F13	nOrderTime, 委託時間 (HHMMSS)
F14	szStatusDesc, FALSE 失敗
F15	szOID, 編號
F16	nPreOrder, 預約單
F17	szAvgPrice, 均價
F18	nQcurrent, 有效數量，包涵成交 (除零股為股數外，其餘為張數。)
F19	無作用
F20	nConfirmTime 異動時間(HHMMSS)
F21	nPayType, 0:餘額交割/ 1:逐筆交割
F22	szBroker, 推薦券商 for 興櫃
F23	nActionFlag
F24	CID 來源別
F25	OrderSource 委託來源
F26	nOrderStatus 1:錯誤單
F27	szCode
F28	szCodeMsg 錯誤訊息
F29	szExtraID 為了延伸 OrderNo, OID 譬如玉山的 ROID
F30	Cond 委託條件 - 0: ROD, 1: FOK, 2: IOC, 99: NULL
F31	nOrderTimeMS 委託時間到毫秒 (HHMMSS.fff)
F32	nConfirmTimeMS 異動時間到毫秒 (HHMMSS.fff)
範例： <F0=20601-0101093 F1=20110408 F2=0 F3=0 F4=1 F5=1108 F6=0 F7=7.38 F8=1000 F9=X0001 F10=0 F11=0 F12=20110408 F13=104132 F14=委託成功	

台新證券 證券 API 下單規格說明書

|F15=633330|F16=0|F17=0|F18=1000|F19=20110408|F20=104132|F21=0|F22=>

請參照「3.2.1.4 證券委託查詢」

證券成交查詢回傳格式

F0	strTradeID
F1	strTradeDate 交易日期
F2	nTT, 0: 普通/ 1: 盤後零股/ 2: 盤後/ 5:興櫃/ 7:盤中零股
F3	strStockID, 商品代號
F4	nBS, 買賣別 1: 買進/ 2: 賣出
F5	nOT, 0: 現股/ 1:融資/ 2: 融券
F6	strPrice, 價位
F7	nQty, 原委託張數/股數 (除零股為股數外, 其餘為張數。)
F8	strAmount, 價金
F9	strOrderNo, 委託書號
F10	strMatchDate, 成交日期 (YYYYMMDD)
F11	strMatchTime, 成交時間 (HHMMSS)
F12	strFee, 手續費
F13	strTax, 交易稅
F14	strOID, 編號
F15	strOMID, 交易編號
F16	strStockName, 股票名稱
F17	strOrderSource, 委託來源
F18	cCid, 下單管道
F19	無作用
F20	strMatchTimeMS, 成交時間到毫秒 (HHMMSSfff)
<p>範例:</p> <pre><F0=20110408 F1=0 F2=1108 F3=1 F4=0 F5=7.38 F6=1 F7=7.38 F8=X0002 F9=20110408 F10=105032 F11=10 F12=2 F13=20601-0101093 F14=633335 F15=2752 F16= F17= F18=></pre>	

請參照「3.2.1.5 證券成交查詢」

證券庫存查詢回傳格式

F0	nTT, 0: 普通/ 1: 零股/ 2: 盤後/ 5:興櫃
F1	strStockID, 商品代號
F2	集保昨日庫存
F3	集保今日庫存
F4	集保今日買進委託
F5	集保今日買進成交
F6	集保今日賣出委託
F7	集保今日賣出成交
F8	調整
F9	融資昨日庫存
F10	融資今日庫存
F11	融資今日買進委託
F12	融資今日買進成交
F13	融資今日賣出委託
F14	融資今日賣出成交
F15	資償
F16	融券昨日庫存
F17	融券今日庫存
F18	融券今日買進委託
F19	融券今日買進成交
F20	融券今日賣出委託
F21	融券今日賣出成交
F22	券償
F23	零股昨日庫存
F24	零股今日庫存
F25	零股今日買進委託
F26	零股今日買進成交
F27	零股今日賣出委託
F28	零股今日賣出成交
F29	調整
F30	strCost, 成本
F31	strStockName, 股票名稱
F32	strRefPrice, 昨收
F33	strAMT, 昨收 * 昨日庫存
F34	無作用
F35	無作用
F36	無作用

台新證券 證券 API 下單規格說明書

範例：

```
<F0=0|F1=2330|F2=1000|F3=1000|F4=0|F5=0|F6=0|F7=0|F8=0|F9=0|F10=0|F11=0|F12=0|F13=0|F14=0|F15=0|F16=0|F17=0|F18=0|F19=0|F20=0|F21=台積電|F22=|F23=|F24=|F25=|F26=|F27=|F28=|F29=|F30=>
```

請參照「3.2.1.6. 證券庫存查詢」

附錄 D. 帳號回傳格式

回傳格式：<#account0><#account1><#account2>...

帳號格式：<#account> = <ID=...|Name=...|UID=...|Type=...>

ID	證券帳號
Name	使用者名稱
UID	身份證帳號
Type	1 證券
Credit	
SFID	
CanHedge	
範例： <ID=538L-1234567 Name=證-華山 1234567-陳 XX UID=A123456789 Type=1>	

附錄 E. 商品檔格式

回傳格式：商品檔存放絕對路徑

E-1：證券商品檔格式

```
<?xml version="1.0" encoding="big5" ?>
<Result Code="0" ServerTime="20111230-192230">
  <Symbol ExpireTime="20111231-082000" Exch="1">
    <Item ID="1101" Name="台泥" LT="1000" P="35" U="37.45"
      D="32.55" TDate="20111230" />
    <Item ID="1102" Name="亞泥" LT="1000" P="33.8" U="36.15" D="31.45"
      TDate="20111230" />
    <Item ID="1103" Name="嘉泥" LT="1000" P="11.85" U="12.65"
      D="11.05" TDate="20111230" />
    .....
  </Symbol>
</Result>
```

證券欄位說明：

Code	0: 正常 其他為失敗
ExpireTime	商品檔有效時間
Exch	商品檔市場 1:證券
ID	商品代碼
Name	商品名稱
LT	交易單位股數
P	平盤價
U	漲停價
D	跌停價
TDate	交易日

