

交易 API 功能說明文件

目錄

目錄.....	2
版本修訂.....	4
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.....	6
2.3. 元件相關說明.....	6
3. API 元件功能說明：.....	7
3.1. 設定相關功能.....	8
3.1.1. 元件初始化(Init).....	8
3.1.2. 結束使用元件(Fini).....	8
3.1.3. 登入(Login).....	9
3.1.4. 登出(Logout).....	9
3.1.5. 連線(Connect).....	10
3.1.6. 斷線(Disconnect).....	10
3.1.7. 整體初始化(Init2).....	11
3.1.8. 完整結束元件(Fini2).....	11
3.1.9. 取得帳號資訊(GetAccount).....	12
3.1.10. 設定 LOG 目錄(SetLogPath).....	13
3.1.11. 設定下單類型(SetExecOrderType).....	14
3.1.12. 設定交易單位(SetLotSizeData).....	14
3.1.13. 設定 API 回傳類型(SetEchoType).....	15
3.2. 交易相關功能.....	16
3.2.1. 證券.....	16
3.2.1.1. 證券新單(Stock_NewOrder).....	17
3.2.1.2. 證券改單(Stock_ModifyOrder).....	18
3.2.1.3. 證券刪單(Stock_CancelOrder).....	19
3.2.1.4. 證券委託查詢(Stock_QueryOrder).....	20
3.2.1.5. 證券成交查詢(Stock_QueryMatch).....	20
3.2.1.6. 證券庫存查詢(Stock_QueryPosition).....	21
3.2.2. 期權.....	22
3.2.2.1. 期權新單(FutOpt_NewOrder).....	22
3.2.2.2. 期權改單(FutOpt_ModifyOrder).....	23
3.2.2.3. 期權刪單(FutOpt_CancelOrder).....	24
3.2.2.4. 期權委託查詢(FutOpt_QueryOrder).....	25

3.2.2.5.	期權成交查詢(FutOpt_QueryMatch)	25
3.2.2.6.	期權未平倉查詢(FutOpt_QueryOI)	26
3.3.	帳務查詢(MIX_QueryData)	26
3.4.	元件的非同步回傳格式	28
3.4.1.	C# 建立 Call Back Function 方式	28
3.4.2.	VB 建立 Call Back Function 方式	28
3.4.3.	CallBack 事件分類	29
附錄 A.	錯誤代碼	29
附錄 B.	主動回報回傳格式	31
	證券欄位定義	31
	期權欄位定義	32
附錄 C.	一般回傳格式	34
	證券新單回傳格式	35
	證券改單回傳格式	35
	證券刪單回傳格式	35
	證券委託查詢回傳格式	36
	證券成交查詢回傳格式	38
	證券庫存查詢回傳格式	39
	期權新單回傳格式	40
	期權改單回傳格式	40
	期權刪單回傳格式	41
	期權委託查詢回傳格式	42
	期權成交查詢回傳格式	44
	期權未平倉查詢回傳格式	45
附錄 D.	帳號回傳格式	47

版本修訂

修訂人	修正日期	版本	說明
吳宗翰	2011/3/21	V1.01	建立說明文件

1. 目的：

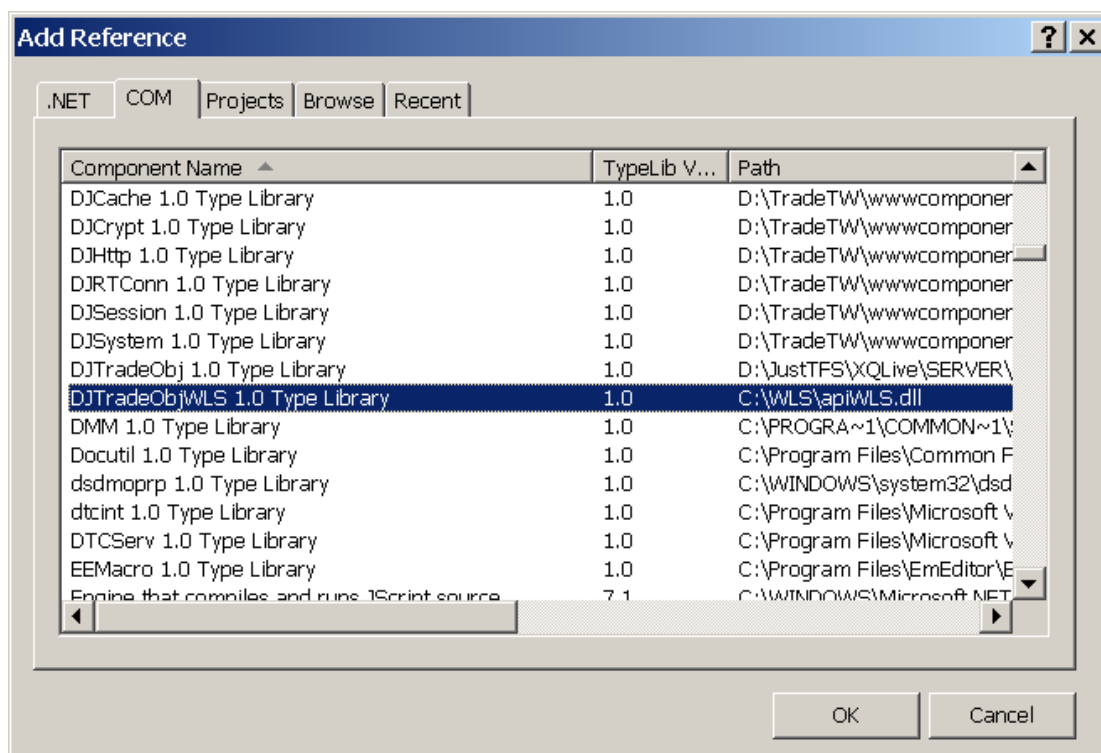
需安裝 XT AP,但使用時無需開啟 XT AP,提供交易 API 元件讓 Third Party 可以自行開發程式下單,使用 COM 方法,進行連接。

2. 元件使用說明：

安裝擁有交易 API 的 XT AP 以後,將會擁有 apiWLS.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 DJTradeObjWLS 1.0 Type Library 這個 COM 元件加入到 Reference 中。如下圖。

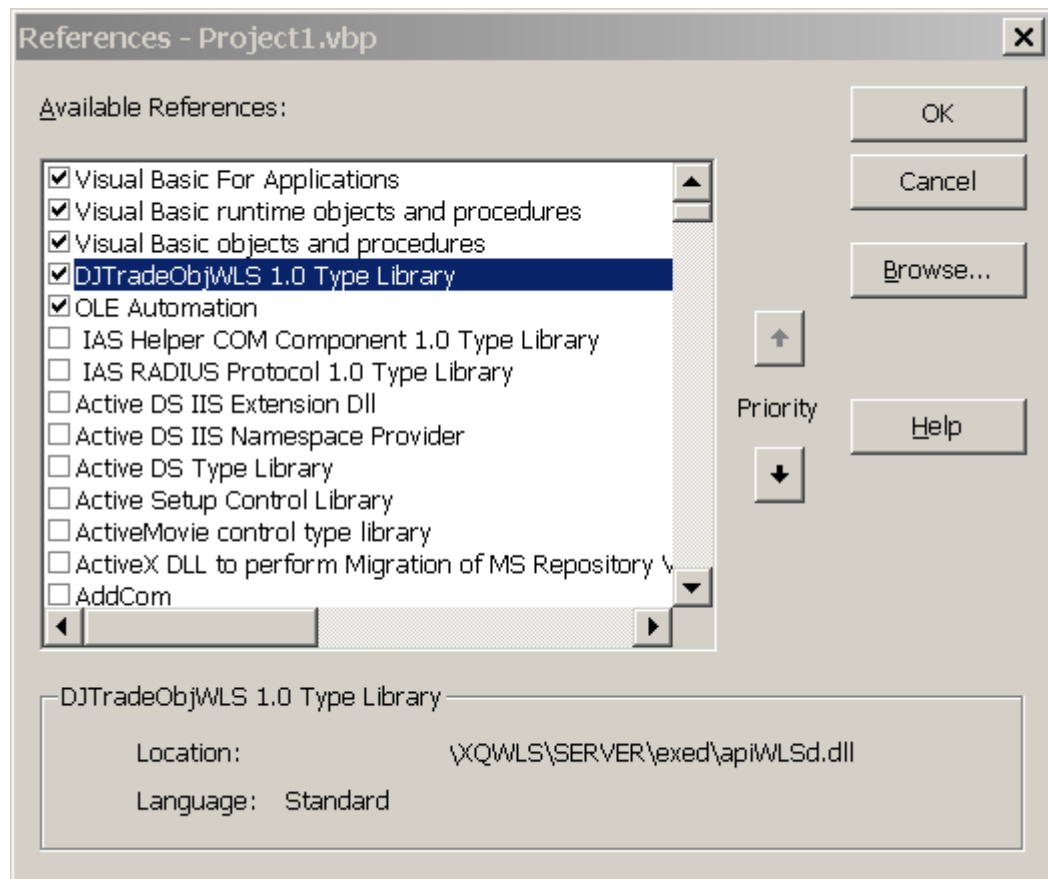


接著使用程式碼 using DJTRADEOBJLibWLS; 來使用 COM 元件。再使用 DJTRADEOBJLibWLS.TradeAppClass m_tradeApp; 宣告一個使用 interface 的物件。並且定義此物件 m_tradeApp = new TradeAppClass();。最後就可以使用元件所提供的 API。

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 DJTradeObjWLS 1.0 Type Library 這個 COM 元件加入到 Reference 中。如下圖。



接著宣告與定義，使用此 COM 的物件。

```
Dim WithEvents TradeApp1 As DJTRADEOBJLibWLS.TradeApp
```

```
Set TradeApp1 = New DJTRADEOBJLibWLS.TradeApp
```

最後使用 COM 元件提供的 API。

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

2.3. 元件相關說明

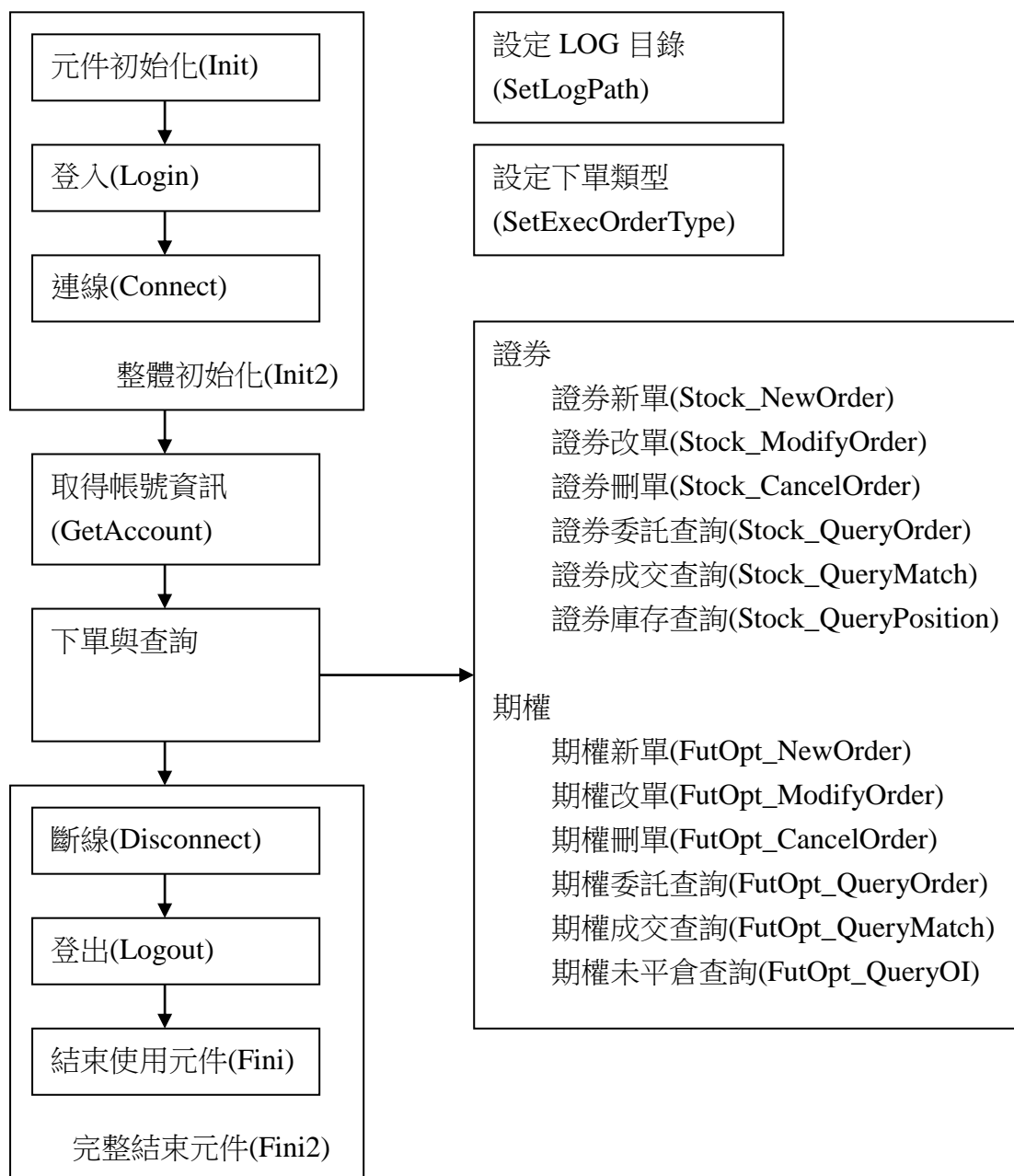
元件的變數型態 BSTR 對等 C# 和 VB 的 String 型態。

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，證券與期權的新單、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：

```
Init(  
    [in]BSTR sTradeDAS,  
    [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：

```
Init2(  
    [in]BSTR sTradeDAS,    // 連線主機 Ex. test.just.com/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，不分證券、期權。

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。不分證券、期權。

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。不分證券、期權。

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 IType)

名稱：SetExecOrderType

說明：設定是走 Http 下單，或是 Socket 下單。

參數：

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

回傳：無

C#

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

VB

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

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、期權的交易模組，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、未平倉查詢。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: 79Z-1234567

交易日期：合法交易日，格式為年月日 Ex: 20110313

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)  
    [in]long nTT, // 0: 普通/ 1: 零股/ 2: 盤後/ 5:興櫃  
    [in]long nOT, // 0: 現股/ 1:融資/ 2: 融券  
    [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  
    [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 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(Stock_ModifyOrder)

名稱：Stock_ModifyOrder

說明：證券改單。

參數：

```
Stock_ModifyOrder(  
    [in]BSTR 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)  
    [in]long nTT,        // 0: 普通/ 1: 零股/ 2: 盤後/ 5:興櫃  
    [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,       // 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]BSTR sNewPrice, // new price ( for 改價, 改量給空的 )  
    [in]BSTR sNewPriceType, // 0: 限價/ 1: 漲停/ 2: 跌停/ 3: 平盤  
    [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改單回傳格式](#)」

C#

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

VB

```
Dim Result As String
```

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

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)  
    [in]long nTT,         // 0: 普通/ 1: 零股/ 2: 盤後/ 5:興櫃  
    [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  
    [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)  
    [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)
```

3.2.2. 期權

使用參數說明：

期權帳號：券商分公司代號 - 期權帳號 Ex: 050-1234567

交易日期：合法交易日，格式為年月日 Ex: 20110313

商品代碼：後台所需要的交易商品代碼 Ex: TXFJ0

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：可以有小數點的數值，複式單可為負值 Ex: 90.5

3.2.2.1. 期權新單(FutOpt_NewOrder)

名稱：FutOpt_NewOrder

說明：期權新單，可以使用複式單。如果是單式單，TradeID2 可以填空。

參數：

FutOpt_NewOrder(

```
[in]BSTR sAccountID, // 期權帳號
[in]long nTradeDate, // 交易日期(西元年月日)
[in]long nTT,         // 期權商品類型 (0:期貨, 1:選擇權)
[in]BSTR sTradeID1,  // 商品代碼
[in]long nBS1,       // 買賣別 (1: 買進/ 2: 賣出)
[in]long nPT,        // PT (0:限價, 1:漲停, 2:跌停, 3:平盤, 4:市價,)
[in]BSTR sPrice,     // 價格
[in]long nQty,       // 數量
[in]long nOffset,    // 倉別(Offset) 0:新倉, 1:平倉, 2:自動, 4:當沖
[in]long nCond,      // 委託條件 (0:ROD, 1:FOK, 2:IOC)
[in]BSTR sTradeID2,  // 商品代碼
[in]long nBS2,       // 買賣別 (1: 買進/ 2: 賣出)
[in]long nPreOrder,  // 1: 預約單/ 0:盤中單
[out, retval]BSTR* psResult
);
```

回傳：請參照「[附錄 C. 一般回傳格式的期權新單回傳格式](#)」

C#

```
string strResult = m_tradeApp.FutOpt_NewOrder(szAccountID, m_nTradeDate, nTT, strTradeID1, nBS1,
                                              nPT, strPrice, nQty, nOffset, nCond, strTradeID2, nBS2, nPreOrder);
```

VB

Dim Result As String

```
Result = TradeApp1.FutOpt_NewOrder(AccountID, TradeDate, TT, strTradeID1, BS1, PT, strPrice, Qty, Offset, Cond,
                                   strTradeID2, BS2, PreOrder)
```

3.2.2.2. 期權改單(FutOpt_ModifyOrder)

名稱：FutOpt_ModifyOrder

說明：期權改單。

參數：

```
FutOpt_ModifyOrder(  
    [in]BSTR sAccountID,    // 期權帳號  
    [in]long nType,        // 0:改量, 2:改價  
    [in]long nTradeDate,   // 交易日期(西元年月日)  
    [in]BSTR sOID,        // 委託單編號  
    [in]BSTR sOrderNo,    // 委託書號  
    [in]long nQty,        // 取消數量  
    [in]long nTT,         // 期權商品類型 (0:期貨, 1:選擇權)  
    [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 nNewPT,      // 新 PT 改價才需要  
                                // (0:限價, 1:漲停, 2:跌停, 3:平盤, 4:市價)  
    [in]BSTR sNewPrice,   // 新價格 改價才需要  
    [in]long nNewCond,    // 新委託條件  
                                // 改價才需要(0:ROD, 1:FOK, 2:IOC)  
    [in]BSTR sCommID1,    // 商品代碼  
    [in]BSTR sCommID2,    // 商品代碼  
    [out, retval]BSTR* psResult  
);
```

回傳：請參照「[附錄 C. 一般回傳格式的期權改單回傳格式](#)」

C#

```
string strResult = m_tradeApp.FutOpt_ModifyOrder(szAccountID, nType, m_nTradeDate, szOID,  
    szOrderNo, nQty, nTT, nQCurrent, nQMatch, nPreOrder, nNewPT,  
    strNewPrice, nNewCond, strTradeID1, strTradeID2);
```

VB

Dim Result As String

```
Result = TradeApp1.FutOpt_ModifyOrder(AccountID, TradeDate, nType, strOID, strOrderNo, Qty, TT, QCurrent, QMatch,  
    PreOrder, NewPT, newprice, NewCond, strTradeID1, strTradeID2)
```

3.2.2.3. 期權刪單(FutOpt_CancelOrder)

名稱：FutOpt_CancelOrder

說明：期權刪單

參數：

```
FutOpt_CancelOrder(  
    [in]BSTR sAccountID, // 期權帳號  
    [in]long nTradeDate, // 交易日期(西元年月日)  
    [in]BSTR sOID,       // 委託單編號  
    [in]BSTR sOrderNo,   // 委託書號  
    [in]long nQty,       // 取消數量  
    [in]long nTT,        // 期權商品類型 (0:期貨, 1:選擇權)  
    [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]BSTR sCommID1,  // 商品代碼  
    [in]BSTR sCommID2,  // 商品代碼  
    [out, retval]BSTR* psResult  
);
```

回傳：請參照「[附錄 C. 一般回傳格式的期權刪單回傳格式](#)」

C#

```
string strResult = m_tradeApp.FutOpt_CancelOrder(szAccountID, m_nTradeDate, szOID,  
                                                  szOrderNo, nQty, nTT, nQCurrent, nQMatch, nPreOrder,  
                                                  strTradeID1, strTradeID2);
```

VB

Dim Result As String

```
Result = TradeApp1.FutOpt_CancelOrder(AccountID, TradeDate, strOID, strOrderNo, Qty, TT, QCurrent, QMatch, PreOrder,  
                                       strTradeID1, strTradeID2)
```


3.2.2.4. 期權委託查詢(FutOpt_QueryOrder)

名稱：FutOpt_QueryOrder

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

參數：

```
FutOpt_QueryOrder(  
    [in]BSTR sAccountID, // 期權帳號  
    [in]long nTT,          // 期權商品類型 (0:期貨, 1:選擇權)  
    [in]long nForceQuery, // 0:user cache/ 1:force query  
    [out, retval]BSTR* psResult  
);
```

回傳：請參照「[附錄 C. 一般回傳格式的期權委託查詢回傳格式](#)」

C#

```
string strResult = m_tradeApp.FutOpt_QueryOrder(szAccountID, nTT, nForceQuery);
```

VB

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

```
Result = TradeApp1.FutOpt_QueryOrder(AccountID, TT, ForceQuery)
```

3.2.2.5. 期權成交查詢(FutOpt_QueryMatch)

名稱：FutOpt_QueryMatch

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

參數：

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

回傳：請參照「[附錄 C. 一般回傳格式的期權成交查詢回傳格式](#)」

C#

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

VB

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

```
Result = TradeApp1.FutOpt_QueryMatch(AccountID, ForceQuery)
```

3.2.2.6. 期權未平倉查詢(FutOpt_QueryOI)

名稱：FutOpt_QueryOI

說明：期權未平倉查詢，可以使用 nForceQuery 設定是否使用快取資料。

參數：

```
FutOpt_QueryOI(  
    [in]BSTR sAccountID, // 期權帳號  
    [in]long nTradeDate, // 交易日期(西元年月日)  
    [in]long nForceQuery, // 0:user cache/ 1:force query  
    [out, retval]BSTR* psResult  
);
```

回傳：請參照「[附錄 C. 一般回傳格式的期權未平倉查詢回傳格式](#)」

C#

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

VB

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

```
Result = TradeApp1.FutOpt_QueryOI(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：

C#

VB

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 DJTRADEOBJLibWLS.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), 2(期權)
Status: 0(斷線), 1(連線)
- 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
- 事件代碼: 202
事件名稱: 資料改變通知

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

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, 04 期權
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'
F7	現股'0'/融資'3'/融券'4'
F8	股票代號
F9	價位: 9(6)V9(2)
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	改量前(only valid for 改量): 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 F4 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)
<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=902 F17=1 F18= F19=633350 F20= F21= F22=2 F23=0000000000 F24= F25= F26=0 F27= >	

期權欄位定義

F0	帳號
F1	回報類別 03 證券, 04 期權
F2	期貨'F'/選擇權'O'
F3	OP
F4	Broker ID
F5	Account ID
F6	委託書號
F7	B/S
F8	市價'M'/限價'L'
F9	委託條件: FOK'F'/IOCT'/ROD'R', 唯 市價 不支援 ROD
F10	開平倉碼, 開倉'0'/平倉'1'/當沖'2', (當沖 僅適用期貨)
F11	商品代號 1
F12	買賣別 1, B/S
F13	價位 1, S9(6)V9(3) 複式單委託, 為價差/價和
F14	商品代號 2 只用在複式單
F15	買賣別 2, B/S 只用在複式單
F16	價位 2, S9(6)V9(3) 只用在複式單
F17	數量, 改量: 更改後數量, 刪單: 取消數量, 均不含成交
F18	改量前(only valid for 改量): 改量前數量, 均不含成交
F19	委託/成交日期, YYYYMMDD
F20	委託/成交時間, HHMMSS
F21	營業員
F22	資料來源
F23	交易所成交序號 9(8)
F24	網路序號 X(8)
F25	智慧下單類型, 1:通知,2:一般單回報,3:智慧單回報
F26	智慧下單網路序號
F27	子帳號

F28	子帳號成交序號
F29	交易方式 '0':單式單 '1':複式第一隻腳 '2':複式第二隻腳 '3':複式單 '' :不做特別處理
F30	Error Message
<pre><F0=15000-3314559 F1=04 F2=F F3=11 F4=15000 F5=3314559 F6=X0007 F7=B F8=M F9=I F10=0 F11=TXFD1 F12=B F13=8866.000 F14= F15= F16= F17=1 F18=0 F19=20110408 F20=100730 F21= F22= F23= F24= F25=2 F26= F27= F28= F29=0 F30=></pre>	

OP 定義：

成功	11: 委託成功, 21: 刪單成功, 31: 改量成功, 40: 成交
失敗	02: 預約失敗, 12: 委託失敗, 22: 刪單失敗, 32: 改量失敗

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
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
"&"	"&"
"<"	"<"
">"	">"
" "	"&bar;"
"="	"&equ;"

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
F5	OrderStatus
F6	Code
F7	CodeMsg
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](#)」

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
F5	bValid
F6	Code
F7	CodeMsg
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
F2	nTT, 0: 普通/ 1: 零股/ 2: 盤後/ 5:興櫃
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
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, 有效張數
F19	nUpdateDate, 異動日期 (YYYYMMDD)
F20	nUpdateTime, 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=10
00|F9=X0001|F10=0|F11=0|F12=20110408|F13=104132|F14=委託成功
|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	strTradeDate 交易日期
F1	nTT, 0: 普通/ 1: 零股/ 2: 盤後/ 5:興櫃
F2	strStockID, 商品代號
F3	nBS, 買賣別 1: 買進/ 2: 賣出
F4	nOT, 0: 現股/ 1:融資/ 2: 融券
F5	strPrice, 價位
F6	nQty, 原委託張數/股數
F7	strAmount, 價金
F8	strOrderNo, 委託書號
F9	strMatchDate, 成交日期 (YYYYMMDD)
F10	strMatchTime, 成交時間 (HHMMSS)
F11	strFee, 手續費
F12	strTax, 交易稅
F13	strTradeID
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)
範例： <F0=20110408 F1=0 F2=1108 F3=1 F4=0 F5=7.38 F6=1 F7=7.38 F8=X0002 F9=20 110408 F10=105032 F11=10 F12=2 F13=20601-0101093 F14=633335 F15=2752 F1 6= F17= F18=>	

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	融資昨日庫存
F9	融資今日庫存
F10	融資今日買進委託
F11	融資今日買進成交
F12	融資今日賣出委託
F13	融資今日賣出成交
F14	融券昨日庫存
F15	融券今日庫存
F16	融券今日買進委託
F17	融券今日買進成交
F18	融券今日賣出委託
F19	融券今日賣出成交
F20	strCost, 成本
F21	strStockName, 股票名稱
F22	strRefPrice, 昨收
F23	strAMT, 昨收 * 昨日庫存
F24	strCRAmount, 融資金額
F25	strDBAmount, 融券保證金
F26	strDBGuarantee, 融券擔保金
<p>範例：</p> <pre><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=></pre>	

請參照「[3.2.1.6. 證券庫存查詢](#)」

期權新單回傳格式

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
F5	OrderStatus
F6	Code
F7	CodeMsg
範例：<F0=1 F1=622617 F2=20110411 F3=622617 F4=1 F5=1 F6= F7=>	

請參照「[3.2.2.1.期權新單](#)」

期權改單回傳格式

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
F5	OrderStatus
F6	Code
F7	CodeMsg
範例：<F0=1 F1=622617 F2=20110411 F3=622617 F4=1 F5=1 F6= F7=>	

請參照「[3.2.2.2.期權改單](#)」

期權刪單回傳格式

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
F5	bValid
F6	Code
F7	CodeMsg
範例：<F0=1 F1=20110411 F2=622617 F3=622617 F4=1 F5=1 F6= F7=>	

請參照「[3.2.2.3. 期權刪單](#)」

期權委託查詢回傳格式

F0	strTradeDate 交易日期
F1	strTradeID
F2	strOID, 編號
F3	strOrderNo, 委託書號
F4	nTT, 0:期貨, 1:選擇權, 2:選擇權複式, 3:期貨複式
F5	strStockID1, 商品代號
F6	strStockName2, 商品名稱
F7	nBS1, 買賣別 1: 買進/ 2: 賣出
F8	strStockID2, 商品代號
F9	strStockName2, 商品名稱
F10	nBS2, 買賣別 1: 買進/ 2: 賣出
F11	nPT, 委託價位別 0:限價, 1:漲停, 2:跌停, 3:平盤, 4:市價
F12	strPrice, 價格
F13	nOrderDate, 委託日期
F14	nOrderTime, 委託時間
F15	nCond, 委託方式(Cond) (0:ROD, 1:FOK, 2:IOC)
F16	nOffset, 倉別(Offset) 0:新倉, 1:平倉, 2:自動, 4:當沖
F17	nQty, 數量
F18	nQCurrent, 有效數量
F19	nQMatch, 成交數量
F20	nQCancel, 取消數量
F21	strOrderStatus, 錯誤狀態
F22	strStatusDesc, 委託狀態
F23	strOrderSource, 委託來源
F24	strAvgPrice, 成交均價
F25	nPreOrder, 預約單 or 盤中單
F26	nUpdateDate, 異動日期
F27	nUpdateTime, 異動時間
F28	nActionFlag,
F29	strOrderKey, 委託單 key
F30	strCodeMsg,
F31	strCode,
F32	strAmt, 成交價金
F33	cCid, 下單管道

F34	nConfirmDate,
F35	nConfirmTime,
F36	strOrgPrice, 原委託價位
F37	nOrgPT, 原委託價位別 0:限價, 1:漲停, 2:跌停, 3:平盤, 4:市價
F38	nOrgCond, 原委託條件 (0:ROD, 1:FOK, 2:IOC)
F39	nOrderType, 原預約單或盤中單
<p>範例：</p> <pre><F0=20110411 F1=15000-3314559 F2=622615 F3=622615 F4=0 F5=TXFD1 F6= F7=1 F8= F9= F10=0 F11=0 F12=8866.000 F13=20110408 F14=154541 F15=0 F16=2 F17=1 F18=1 F19=0 F20=0 F21= F22=委託預約中 F23=E O O F24=0 F25=1 F26=20110408 F27=154541 F28=3 F29=622615 F30= F31= F32= F33= F34=20110408 F35=154541 F36=8866.000 F37=0 F38=99 F39=2></pre>	

請參照「[3.2.2.4. 期權委託查詢](#)」

期權成交查詢回傳格式

F0	strTradeDate 交易日期
F1	nTT, 盤別 0:期貨, 1:選擇權, 2:選擇權複式, 3:期貨複式
F2	strStockID1, 商品代號
F3	strStockName1, 商品名稱
F4	nBS1, 買賣別 1: 買進/ 2: 賣出
F5	nPT, 委託價位別 0:限價, 1:漲停, 2:跌停, 3:平盤, 4:市價
F6	strPrice, 價格
F7	nQty, 數量
F8	nOffset, 倉別 0:新倉, 1:平倉, 2:自動, 4:當沖
F9	nCond, 委託條件 (0:ROD, 1:FOK, 2:IOC)
F10	strStockID2, 商品代號
F11	strStockName2, 商品名稱
F12	nBS2, 1: 買進/ 2: 賣出
F13	strMatchPrice, 成交價位
F14	nQMatch, 成交數量
F15	strOrderNo, 委託書號
F16	nMatchTime, 成交時間
F17	strOMID, 成交序號
F18	strFee, 手續費
F19	strTax, 交易稅
F20	strTradeID,
F21	strCurrency, 幣別
F22	strOID, 編號
F23	strOrderSource, 委託來源
F24	cCid, 下單管道
<p>範例：</p> <pre><F0=20110408 F1=0 F2=TXFD1 F3= F4=1 F5=0 F6=8866.000 F7=1 F8=0 F9=0 F10= F11= F12= F13=8866.000 F14=1 F15=I0001 F16=100718 F17=S001 F18=10 F19=2 F20=15000-1234567 F21= F22=6876542 F23= F24=></pre>	

請參照「[3.2.2.5. 期權成交查詢](#)」

期權未平倉查詢回傳格式

F0	nTT, 0:期貨, 1:選擇權, 2:選擇權複式, 3:期貨複式
F1	strStockID1, 商品代號
F2	strStockName1, 商品名稱
F3	nBS1, 買賣別 1: 買進/ 2: 賣出
F4	strStockID2, 商品代號
F5	strStockName2, 商品名稱
F6	nBS2, 1: 買進/ 2: 賣出
F7	nOpenInterest1, 未平倉口數
F8	strAvgPrice1, 成交均價
F9	nBalance1, 未平倉損益
F10	strSettlePrice1, 昨日結算價
F11	strDeposit1, 原始保證金
F12	nTradeDate 交易日期
F13	nOpenInterest2, 未平倉口數
F14	strAvgPrice2, 成交均價
F15	nBalance2, 未平倉損益
F16	strSettlePrice2, 昨日結算價
F17	strDeposit2, 原始保證金
F18	nPreTDate, 結算日期
F19	strCost1, 成本
F20	strCost2, 成本
F21	strCurrency, 幣別
F22	strSymb1,
F23	strSymb2,
F24	nBQty, 委買
F25	nSQty, 委賣
F26	nBMQty, 買成
F27	nSMQty, 賣成
F28	nChargeQty, 抵繳口數
F29	nMixQty, 混合口數
F30	nSetQty, 當沖口數
F31	cCid, 下單管道
F32	strSource, 委託來源
F33	strCurPrice, 參考現價

F34	nOffset, 倉別 0:新倉, 1:平倉, 2:自動, 4:當沖
範例： <F0=0 F1=TXFD1 F2= F3=1 F4= F5= F6= F7=1 F8=8866.000 F9=1000 F10=8860.000 F11=5000 F12=20110408 F13= F14= F15= F16= F17= F18=20110408 F19=10000 F20= F21= F22= F23= F24=5 F25=0 F26=1 F27=0 F28=0 F29=0 F30=0 F31= F32= F33= F34=0>	

請參照「[3.2.2.6. 期權未平倉查詢](#)」

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、期權帳號
Name	使用者名稱
UID	身份證帳號
Type	1 證券, 2 期權
範例： <ID=79Z-1234567 Name=證-安和 1234567-陳 XX UID=A123456789 Type=1>	