

# TC4 APP

## *Developer's Guide*

***Version 2.0***

## ***Table of Contents***

### [Introduction](#)

#### [About TC4](#)

#### [At a Glance](#)

### [Chapter 1 TC4 APP Folder Structure](#)

### [Chapter 2 System & Developer Requirements](#)

### [Chapter 3 Creating a New Project](#)

### [Chapter 4 Modifying App Configuration](#)

#### [4.1 Changing App Name](#)

#### [4.2 Changing App Icon](#)

#### [4.3 Changing Default Culture](#)

#### [4.4 Setting App Single Instance](#)

### [Chapter 5 Adding User Control to Your Project](#)

### [Chapter 6 Main Window Layout & Main Page](#)

#### [6.1 The Title Bar Keyword in Main Window](#)

#### [6.2 Main Content Page in Mainw Window](#)

### [Chapter 7 I4Framework Initializing Events](#)

#### [7.1 I4Framework Initial Startup Events](#)

#### [7.2 I4Framework Custom Command Event](#)

### [Chapter 8 I4Framework Inter-Instance Communication](#)

#### [8.1 Register to receive Inter-Message](#)

#### [8.2 Unregisetering to stop receiving Inter-Message](#)

#### [8.3 Send Inter-Message to all registered components](#)

#### [8.4 Receiving registration Inter-Message](#)

### [Chapter 9 I4Framework APIs Tutorial](#)

#### [9.1 Using I4Framework.Prop.\\*](#)

#### [9.2 Using I4Framework.Func.\\*](#)

##### [9.2.1 Getting Properties](#)

##### [9.2.2 Setting Properties](#)

##### [9.2.3 Getting Permanent Data Properties](#)

##### [9.2.4 Setting Permanent Data Properties](#)

##### [9.2.5 Getting Global Permanent Data Properties](#)

##### [9.2.6 Setting Global Permanent Data Properties](#)

##### [9.2.7 System Logger](#)

##### [9.2.8 Dynamic Load Assembly](#)

#### [9.3 How to Get Image from Resources](#)

#### [9.4 How to Get String from Resources](#)

#### [9.5 How to Get Style from Resources](#)

### [Chapter 10 How to Use TCORE Quote API](#)

#### [10.1 Add a COM TCQuoteWrapperAPILib Reference to Project](#)

#### [10.2 Create a COM QuoteAPI Object](#)

#### [10.3 Add Event Handlers](#)

#### [10.4 Create a Connection with TCore](#)

#### [10.5 Subscribe Symbol](#)

[10.6 Unsubscribe Symbol](#)

[10.7 Symbol Lookup](#)

[10.8 Quote API Event Processing](#)

#### [Chapter 11 How to Use TCORE Trade API](#)

[11.1 Add a COM TCTradeWrapperAPILib Reference to Project](#)

[11.2 Create a COM TradeAPI Object](#)

[11.3 Add Event Handlers](#)

[11.4 Create a Connection with TCore](#)

[11.5 NewOrder](#)

[11.6 NewOCOProtectOrder](#)

[11.7 CancelOrder](#)

[11.8 ReplaceOrder](#)

[11.9 Trade API Event Processing](#)

#### [Chapter 12 How to Creating a Chart Component](#)

[12.1 Create a New Project](#)

[12.2 Modify Project Properties](#)

[12.3 Add References to Project](#)

[12.4 Create a new chart component](#)

[12.5 Add a Chart Component to the List of Charts](#)

[12.6 Classes of the TechLine.Components.Base](#)

[12.7 Properties of BaseControl](#)

[12.8 Methods of BaseControl](#)

#### [Chapter 13 TCORE Quoting API Reference](#)

[13.1 Quoting Methods](#)

[13.1.1 Connect](#)

[13.1.2 Disconnect](#)

[13.1.3 SubQuote](#)

[13.1.4 UnsubQuote](#)

[13.1.5 SymbolLookup](#)

[13.1.6 DoSomething](#)

[13.1.7 GetSymbolClassifications](#)

[13.1.8 GetProductCurrency](#)

[13.1.9 GetCurrencyExchangeRate](#)

[13.1.10 GetGeneralService](#)

[13.1.11 SetGeneralService](#)

[13.1.12 TopicPublish](#)

[13.1.13 TopicSub](#)

[13.1.14 TopicUnsub](#)

[13.1.15 GeneralQuery](#)

[13.1.16 GetInstrumentInfo](#)

[13.2 Quoting Events](#)

[13.2.1 OnCommandMsg](#)

[13.2.2 OnQuoteData](#)

[13.2.3 OnSymbolClassificationsUpdate](#)

[13.2.4 OnProductCurrencyUpdate](#)

[13.2.5 OnCurrencyExchangeRateUpdate](#)

[13.2.6 OnGeneralServiceUpdate](#)

[13.2.7 OnSetTopic](#)

[13.2.8 OnGeneralQueryReturn](#)

[13.2.9 OnInstrumentInfoUpdate](#)

[13.3 Quoting Properties](#)

[13.3.1 DataParameters](#)

[13.3.2 QuoteDataFuture](#)

[13.3.3 QuoteDataStock](#)

[Chapter 14 TCORE Trading API Reference](#)

[14.1 Trading Methods](#)

[14.1.1 Connect](#)

[14.1.2 Disconnect](#)

[14.1.3 DoSomething](#)

[14.1.4 GetAccountData](#)

[14.1.5 NewOrder](#)

[14.1.6 ReplaceOrder](#)

[14.1.7 CancelOrder](#)

[14.1.8 GetReportData](#)

[14.1.9 GetProductCurrency](#)

[14.1.10 GetCurrencyExchangeRate](#)

[14.1.11 GetGeneralService](#)

[14.1.12 SetGeneralService](#)

[14.1.13 TopicSub](#)

[14.1.14 TopicUnsub](#)

[14.1.15 TopicPublish](#)

[14.1.16 GeneralQuery](#)

[14.1.17 GetInstrumentInfo](#)

[14.1.18 NewOCOProtectOrder](#)

[14.1.19 NewAlertOrder](#)

[14.1.20 SetAccountSubscriptionLevel](#)

[14.1.21 QueryAccountData](#)

[14.2 Trading Events](#)

[14.2.1 OnCommandMsg](#)

[14.2.2 OnAccountUpdate](#)

[14.2.3 OnExecutionReport](#)

[14.2.4 OnCancelReplaceError](#)

[14.2.5 OnReportReset](#)

[14.2.6 OnProductCurrencyUpdate](#)

[14.2.7 OnCurrencyExchangeRateUpdate](#)

[14.2.8 OnGeneralServiceUpdate](#)

[14.2.9 OnSetTopic](#)

[14.2.10 OnGeneralQueryReturn](#)

[14.2.11 OnInstrumentInfoUpdate](#)

[14.3 Trading Properties](#)

[14.3.1 NOPFuture](#)

[14.3.2 NOPStock](#)

[14.3.3 ADIAccount](#)

[14.3.4 ADIFutureMargin](#)

[14.3.5 ADIFuturePosition](#)

[14.3.6 RPTExecutionReport](#)

[14.3.7 RPTFilledOrder](#)

[14.3.8 NOPReplaceOrder](#)

[14.3.9 RPTReports](#)

[Chapter 15 Packing an App for TC AppStore](#)

[Chapter 16 Publishing to the TC App Store](#)

[Appendix](#)

[A.1 Converting C++ Data Types to C#](#)

[A.2 C#, WPF Related Websites](#)

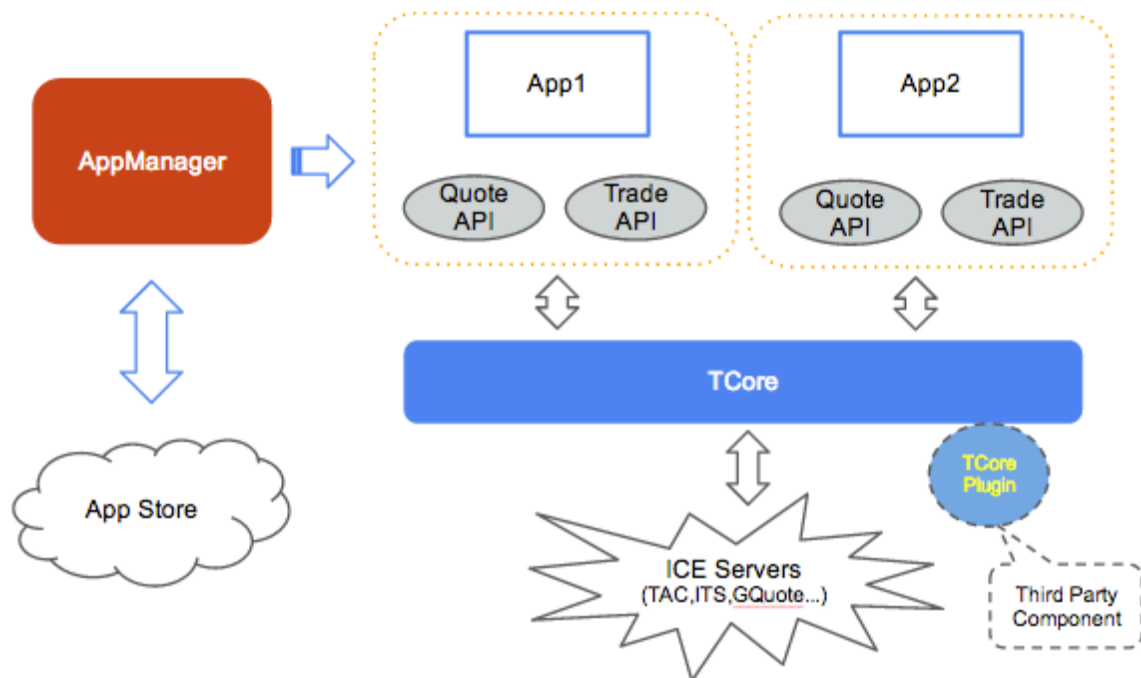
[A.3 C#, WPF Related Books](#)

[Document Revision History](#)

## Introduction

### About TC4

TC4 架構，包含 4 個主要部分，App Manager、App Store、TCore、TC App。



### At a Glance

- **I4Framework** 是用 C# & WPF & .NET 4.0 所開發出來的應用程式平台，提供『語系字串、元件樣式、屬性管理、元件內部通訊』等功能，開發者可利用 I4Framework 快速開發 TC App。
- **TCore** 主要是報價及交易核心，利用 TCore 所提供的 Quote API & Trade API 就可以取得核心報價及交易的資訊與方法，也提供 App 之間 Topic 訂閱通訊功能。
- **AppManager** 負責 App 的下載、更新、執行管理，並是 TCore 運作及更新的管理中心，使用者也可以使用 AppManager 送出特定 App 的 Feedback。
- **AppStore** 是開發者專屬的網站，提供開發者上載自行開發 App，並有 App 的審核機制，開發者可利用此網站管理自己的 App 上下架狀態，此網站也包含使用者 Feedback 的資訊收集。

## Chapter 1 TC4 APP Folder Structure

TC4 App				
	<i>I4Framework.exe</i>			
	<i>I4Framework.exe.config</i>			
	<i>I4Framework.App.xpl</i>			
	<b>COM</b>			
	<b>Shared Components</b>			
	<b>Services</b>			
	<b>Components</b>			
	<b>Logs</b>			
	<b>zh-CHT</b>	<b>Configures</b>	<b>Defaults</b>	<i>I4Framework.App.xpl</i>
		<b>Dictionaries</b>	<b>Strings</b>	<i>I4Framework.xaml</i>
			<b>Styles</b>	<i>I4Framework.xaml</i>
			<b>Images</b>	<i>AppLogo.ico</i>
			<i>ResourceDictionary.xaml</i>	
		<b>Windows</b>	<i>MainWindow.xaml</i>	
			<i>MainPage.xaml</i>	
			<del><i>SplashScreen.xaml</i></del>	
			<i>CloseWindow.xaml</i>	
	<b>zh-CHS</b>	Same as above		
	<b>en_US</b>	Same as above		

**Caution: Files encoding must be saved as UTF-8**

## Chapter 2 System & Developer Requirements

- ✓ Visual Studio 2010
- ✓ AppManager Installer
- ✓ **TC4AppTemplate** Template Development Kit
- ✓ Login Account & Password
- ✓ App Service Key

## Chapter 3 Creating a New Project

1. Install Visual Studio 2010 and AppManager
2. 以『系統管理員身分』執行 Visual Studio 2010
3. 加入新專案並選擇 WPF User Control Library
4. Copy all files from **TC4AppTemplate\bin** folder to **專案目錄\bin** folder
5. Add References to Project
  - a. **專案目錄\bin\Debug\I4Framework.exe** 並設定 **Copy Local** 為 **False**
  - b. **專案目錄\bin\Debug\Common Components\TCoreAPI.dll** 並設定 **Copy Local** 為 **False**
6. Modify Project Properties
  - a. **Application tab**
    - Target Framework 選擇『**.NET Framework 4.0 Client Profile**』
  - b. **Build tab**
    - Output path 設定為 **專案目錄\bin\Debug\Components\**
  - c. **Debug tab**
    - Start external program 設定為 **專案目錄\bin\Debug\I4Framework.exe**
    - Working directory 設定為 **專案目錄\bin\Debug**



## Chapter 4 Modifying App Configuration

### 4.1 Changing App Name

在 語系\Dictionaryes\Strings\I4Framwork.xml 中，修改 Key = App.Name 的 Value

### 4.2 Changing App Icon

在 語系\Dictionaryes\Images\AppLogo.ico 將其替代

### 4.3 Changing Default Culture

此設定用來指定 App 的預設語系

在 *I4Framework.exe.config* 中，修改 Key = DefaultCulture 的值  
( 如：zh-CHT、zh-CHS、en-US )

**Caution:** 此檔中其他參數不可亂改

### 4.4 Setting App Single Instance

在 *I4Framework.exe.config* 中，修改 Key = SingleInstance 的值為 True

**Caution:** 此檔中其他參數不可亂改

## Chapter 5 Adding User Control to Your Project

### 5.1 Create a WPF User Control

修改 **AssemblyInfo** 中相關屬性的值。

如：

- AssemblyTitle
- AssemblyDescription
- AssemblyCompany
- AssemblyProduct
- AssemblyVersion
- AssemblyFileVersion

### 5.2 將程式會使用的字串加入至 所有語系\Dictionaryes\Strings\ProjectName.xaml

```
<ResourceDictionary xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" ..>
    <sys:String x:Key="MyProject.Key">custom string </sys:String>
</ResourceDictionnary>
```

利用 *GetResource* API 取出 ([refer to](#))

### 5.3 將程式會使用的 Style 加入至 所有語系\Dictionaryes\Styles\ProjectName.xaml

```
<ResourceDictionary xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" ..>
    <Style x:Key="MyStyle" TargetType="Label">
        <Setter Property="Foreground" Value="White" />
        ....
    </Style>
</ResourceDictionary>
```

利用 *GetResource* API 取出 ([refer to](#))

### 5.4 在 語系\Windows\MainPage.xaml 檔案中加入元件，如下

```
<Page xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" ...
    xmlns:myUC="clr-namespace:mynamespace;assembly=myassembly"
    ... >
    <Grid>
        <myUC:myUserControl />
    </Grid>
</Page>
```

## Chapter 6 Main Window Layout & Main Page

### 6.1 The Title Bar Keyword in Main Window

( MainWindow Layout 在 語系\Windows\MainWindow.xaml )

- **I4F\_MainWindowTitle**                      標題名稱
- **I4F\_MainWindowTitleBar**                標題列
- **I4F\_MainWindowMinimize**               縮小按鍵
- **I4F\_MainWindowMaximize**              放大按鍵
- **I4F\_MainWindowClose**                  關閉按鍵

### 6.2 Main Content Page in Mainw Window

( MainPage Layout 在 語系\Windows\MainPage.xaml )

- **I4F\_MainWindowDock** 做為主體內容，App 版面及元件都應該在此檔中呈現

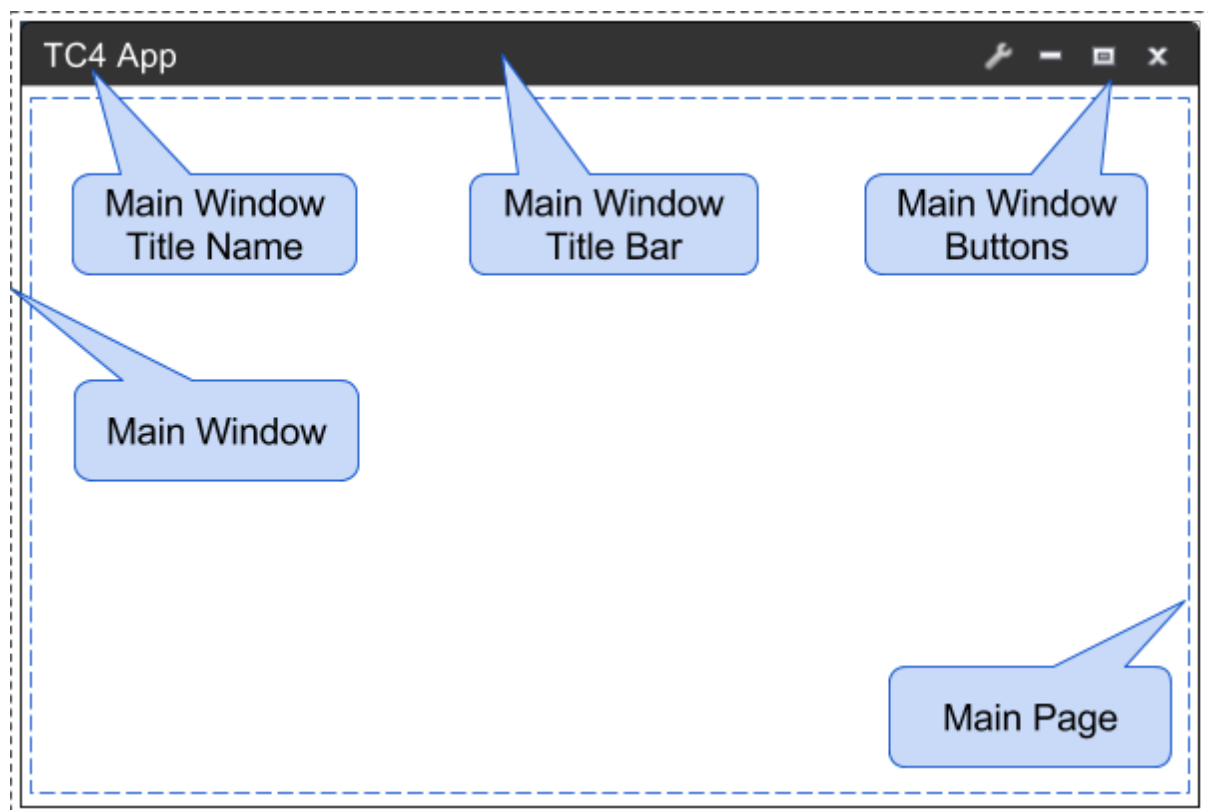


Figure 6-1

## Chapter 7 I4Framework Initializing Events

### 7.1 I4Framework Initial Startup Events

```
public static void I4EventInitialize(I4Framework.I4EventArgs i4e)
```

實作此方法將會收到 I4Framework 的初始的步驟通知，

依序為 ServciesReady -> ComponentsReady -> MainWindowReady -> MainMenuReady -> PagesLoaded -> MainWindowShow -> MainWindowActivated -> SystemReady。

- 事件參數定義

```
public class I4EventArgs : EventArgs {  
    public Define.Status Status;           // 事件狀態  
    public Object Data;                   // 事件資料物件  
}
```

- 事件狀態定義

```
public enum Status {  
    ServicesReady,           // 載入 Servers 目錄下的所有元件就緒  
    ComponentsReady,        // 載入 Components 目錄下的所有元件就緒  
    MainWindowReady,        // 主視窗就緒  
    MainMenuReady,          // 主選單就緒  
    SystemReady,            // 系統就緒  
    CustomCommand,          // 自定命令  
    EventArgs,              // 自訂事件  
    Exception,              // 例外事件  
    MainWindowShow,         // 主視窗顯示  
    MainWindowActivated,    // 主視窗活動  
    PagesLoaded             // 所有頁面已載入  
};
```

### 7.2 I4Framework Custom Command Event

```
public static void I4EventCustomCommand(I4Framework.I4EventArgs i4e)
```

實作此方法可接收來至KeyBinding 的按鍵事件

如下：

```
<KeyBinding  
    Modifiers="Control" Key="S"  
    Command="{x:Static i4f:App.CustomCommand}"  
    CommandParameter="測試,ForTest,ctrol,s"  
>
```

## Chapter 8 I4Framework Inter-Instance Communication

### 8.1 Register to receive Inter-Message

```
I4Framework.Func.RegisterMessage(Object owner, String groupName = "")
```

### 8.2 Unregistering to stop receiving Inter-Message

```
I4Framework.Func.UnRegisterMessage(Object owner, String groupName = "")
```

### 8.3 Send Inter-Message to all registered components

```
I4Framework.Func.SendMessage (  
    Object owner,                // 傳送的元件  
    Define.MessageType MsgType,  // 訊息的形式，如下說明  
    Object MsgData,              // 訊息物件  
    Object Target = null         // Group Name  
)  
  
    ○ MessageType :  
        Frameowrk                // 平台相關  
        Compnent                 // 元件相關  
        Command                  // 命令訊息  
        Message                  // 一般訊息  
        Alert                    // 警示訊息  
        Change                   // 改變訊息  
        Show                     // 顯示訊息  
        Hidden                   // 隱藏訊息  
        Close                    // 關閉訊息  
        None                     // 未定義
```

### 8.4 Receiving registration Inter-Message

實作如下方法即可。

```
public void OnFrameworkMessageEX (  
    Object sender,                // 傳送者  
    Define.MessageType MsgType,  // 訊息的形式，如上說明  
    Object MsgData,              // 訊息物件  
    Object GroupName             // 群組名稱  
)
```

## Chapter 9 I4Framework APIs Tutorial

### 9.1 Using I4Framework.Prop.\*

9.1.1 Prop.AppID	// 應用程式代碼
9.1.2 Prop.AppVersion	// 應用程式版本
9.1.3 Prop.AppManagerRegion	// AppManager 所屬區域
9.1.4 Prop.AppManagerPath	// AppManager 的路徑
9.1.5 Prop.AppServiceKey	// 應用程式的服務識別碼
9.1.6 Prop.AppCulture	// 應用程式目前的語系
9.1.7 Prop.AppRootPath	// 應用程式的根目錄
9.1.8 Prop.AppCulturePath	// 應用程式目前語系目錄
9.1.9 Prop.AppConfiguresPath	// 預設檔的路徑
9.1.10 Prop.AppUsersRootPath	// 應用程式存放設定檔，相關使用者的根目錄
9.1.11 Prop.AppUsersPath	// 應用程式存放設定檔，相關使用者的語系目錄
9.1.12 Prop.AppImagesPath	// 應用程式根目錄下的 Images 目錄
9.1.13 Prop.AppDictionariesAPath	// 應用程式目前語系下的資源目錄 (Dictionaries)絕對路徑
9.1.14 Prop.AppDictionariesRPath	// 應用程式目前語系下的資源目錄 (Dictionaries)相對路徑
9.1.15 Prop.AppSerialNumber	// 應用程式目前的序號，當多個相同App 啟動時，可識別
9.1.16 Prop.CommonComPath	// Common Components 的路徑
9.1.17 Prop.MainWindow	// 主視窗
9.1.18 Prop.MainWindowDock	// (I4F_MainWindowDock)
9.1.19 Prop.MainPage	// Main Page
9.1.20 Prop.MainWindowTitle	// 主視窗標題
9.1.21 Prop.IsUacEnable	// 判斷UAC是否啟動
9.1.22 Prop.IsExitingWithoutConfirmation	// 可設定或判斷應用程式關閉時，是否顯示確認視窗

## 9.2 Using I4Framework.Func.\*

### 9.2.1 Getting Properties

在語系目錄下的 Configures\Defaults\I4Framework.App.xpl  
Object Func.GetAppProperty(String key, String defaultValue = "")

### 9.2.2 Setting Properties

存放在 Users 目錄下的相關位置  
Func.SetAppProperty(String key, String value)

### 9.2.3 Getting Permanent Data Properties

在 App 根目錄下的 I4Framework.PermanentData.xml  
String Func.GetAppPermanentData(String key, String defaultValue = "")

### 9.2.4 Setting Permanent Data Properties

存放在 Users 目錄下的相關位置  
Func.SetAppPermanentData(String key, String value)

### 9.2.5 Getting Global Permanent Data Properties

在 App Manager 或 App 根目錄下的 I4Framework.PermanentData.xml  
String Func.GetAppGlobalPermanentData(String key, String defaultValue = "")

### 9.2.6 Setting Global Permanent Data Properties

在 App Manager 或 App 根目錄的 I4Framework.PermanentData.xml  
Func.SetAppGlobalPermanentData(String key, String value)

### 9.2.7 System Logger

void Func.SystemLogger(Define.LogStatus status, Object owenr, params String[] args)

### 9.2.8 Dynamic Load Assembly

Object Func.LoadAssemblyClass(String assemblyName, String className, String path = "")

如下：

```
assemblyName = "WpfControlLibrar1.dll"  
className = "WpfControlLibrary1.UserControl1"  
path = "Comonents\\"
```

## 9.3 How to Get Image from Resources

Resources in 語系/Dictionaries/Images

### 9.3.1 在程式碼中

```
I4Framework.Func.GetImageFromFile(this, I4Framework.Prop.AppDictionariesRPath +  
    "Images/image.png", ImageFormat.Png);
```

### 9.3.2 在 XAML 中

```
Source = "pack://siteoforigin:,,,/Images/AppLogo.ico"
```

or

*/\*\* 如果圖檔存放在 語系\Dictionaries\Images , 要用此方法 \*/*

```
{Binding Converter={StaticResource I4F_DImgConv}, ConverterParameter='image.png'}
```

## 9.4 How to Get String from Resources

Resources in 語系/Dictionaries/Strings

```
<ResourceDictionary xmlns="....">  
    <sys:String x:Key="MyStringKey">I4Framework App</sys:String>  
</ResourceDictionary>
```

### 9.4.1 在程式碼中

```
String str = Func.GetResource("MyStringKey").toString();
```

### 9.4.2 在 XAML 中

```
Content = "{DynamicResource MyStringKey}"
```

## 9.5 How to Get Style from Resources

Resources in 語系/Dictionaries/Styles

```
<Style TargetType="Control" x:Key="MyStyleKey">  
    <Setter Property="FontSize" Value="13"/>  
</Style>
```

### 9.5.1 在程式碼中

```
Style myStyle = Func.GetResource("MyStyleKey") as Style;  
Control.Style = myStyle;
```

### 9.5.2 在 XAML 中

```
Style = "{DynamicResource MyStleKey}"
```



## Chapter 10 How to Use TCORE Quote API

### 10.1 Add a COM TCQuoteWrapperAPILib Reference to Project

### 10.2 Create a COM QuoteAPI Object

```
TCQuoteWrapperAPILib.TCQuoteAPIEx QuoteAPI = new TCQuoteWrapperAPILib.TCQuoteAPIEx();
```

### 10.3 Add Event Handlers

```
QuoteAPI.OnCommandMsg += new  
    TCQuoteWrapperAPILib._ITCQuoteAPIExEvents_OnCommandMsgEventHandler(QuoteAPI_On  
    nCommandMsg);  
QuoteAPI.OnQuoteData += new  
    TCQuoteWrapperAPILib._ITCQuoteAPIExEvents_OnQuoteDataEventHandler(QuoteAPI_OnQ  
    uoteData);  
QuoteAPI.OnSymbolClassificationsUpdate += new  
    TCQuoteWrapperAPILib._ITCQuoteAPIExEvents_OnSymbolClassificationsUpdateEventHandl  
    er(QuoteAPI_OnSymbolClassificationsUpdate);
```

### 10.4 Create a Connection with TCore

```
TCoreAPI.Quote.Connect(QuoteAPI);  
OR  
QuoteAPI.Connect(/*Host Address*/, /*System Name*/, /*App Service Key */, 1);
```

### 10.5 Subscribe Symbol

```
TCoreAPI.Quote.SubSymbol(QuoteAPI, TCoreAPI.Defines.DATA_TYPE_REALTIME, symbol);  
OR  
TCQuoteWrapperAPILib.DataParameters DataParam = new  
    TCQuoteWrapperAPILib.DataParameters();  
DataParam.SecurityType = 9 /*SECURITY_TYPE_SYMBOL*/;  
DataParam.Symbol = symbol;  
QuoteAPI.SubQuote(DataType, DataParam); ( Please refer to 13.1.3 )
```

### 10.6 Unsubscribe Symbol

```
TCoreAPI.Quote.UnSubSymbol(QuoteAPI, TCoreAPI.Defines.DATA_TYPE_REALTIME, symbol);  
OR  
TCQuoteWrapperAPILib.DataParameters DataParam = new  
    TCQuoteWrapperAPILib.DataParameters();  
DataParam.SecurityType = 9 /*SECURITY_TYPE_SYMBOL*/;  
DataParam.Symbol = symbol;  
QuoteAPI.UnsubQuote(DataType, DataParam); ( Please refer to 13.1.4 )
```

## 10.7 Symbol Lookup

```
// 商品名稱查詢, 設定是否同步處理 (true:同步處理, 會等待function回復return值 false:非同步處理, 會從
// QuoteAPI_OnCommandMsg回覆) (Please refer to 13.1.5)
bool isSync = false;
List<string> ret = TCoreAPI.Quote.SymbolLookup(QuoteAPI,
    TCoreAPI.Defines.SymbolLookup_FullMatch_All, position.Symbol, isSync); // 完全比對

// 若有查詢過的商品, 或是設定為同步處理, 則會有return值
if (ret != null && ret.Count > 0)
{
    string s = ret[0];

    // 2,TC.F.CBOT.ZN.201309,美10年期国债201309,ZN,CBOT,芝加哥商品期货交易所,201309,,,,;
    String[] ss = s.Split(',');
    if (ss.Length > 2)
    {
        if (ss[2].ToLower() == "(null)") return; // 查無商品
        string symbol = ss[1];
        string name = ss[2];
    }
}
```

## 10.8 Quote API Event Processing

```
// 10.8.1 Quote API 命令訊息事件 (Please refer to 13.2.1)
void QuoteAPI_OnCommandMsg(int MsgType, int MsgCode, string MsgString)
{
    // TCore 的連線狀態
    if (MsgType == 2 /*TCoreAPI.Defines.COMMAND_MSG_SYSTEM_STATUS*/)
    {
        switch (MsgCode)
        {
            case TCoreAPI.Defines.CONNECT_STATUS_CONNECTED: // TCore已連線
                break;
            case TCoreAPI.Defines.CONNECT_STATUS_DISCONNECTED: // TCore已斷線
                break;
        }
    }
    // 來自 SymbolLookup 非同步查詢結果
    else if (MsgType == 3 /*TCoreAPI.Defines.COMMAND_MSG_SECURITY_LOOKUP*/)
    {
        // 2,TC.F.CBOT.ZN.201309,美10年期国债201309,ZN,CBOT,芝加哥商品期货交易所,201309,,,;
        String[] ss = MsgString.Split(',');
        if (ss.Length > 2) {
            if (ss[2].ToLower() == "(null)") return; // 查無商品
            string symbol = ss[1];
            string name = ss[2];
        }
    }
}
```

// 10.8.2 報價 即時、回補 相關資訊 (**Please refer to 13.2.2**)

```
void QuoteAPI_OnQuoteData(int SecurityType, int DataType, object QuoteData)
{
    if (DataType == TCoreAPI.Defines.DATA_TYPE_REALTIME)
    {
        switch (SecurityType)
        {
            case TCoreAPI.Defines.SECURITY_TYPE_STOCK:
            {
                // 取得證券報價元件
                TCQuoteWrapperAPILib.QuoteDataStock quote =
                    (TCQuoteWrapperAPILib.QuoteDataStock)QuoteData;
            }
            break;
            case TCoreAPI.Defines.SECURITY_TYPE_FUTURE:
            {
                // 取得期貨報價元件
                TCQuoteWrapperAPILib.QuoteDataFuture quote =
                    (TCQuoteWrapperAPILib.QuoteDataFuture)QuoteData;
            }
            break;
        }
    }
    // 分時表 (成交明細)
    else if (DataType == TCoreAPI.Defines.DATA_TYPE_MATCH_LIST)
    {
        TCQuoteWrapperAPILib.TimeAndSales TimeAndSales =
            (TCQuoteWrapperAPILib.TimeAndSales)QuoteData;
        TCQuoteWrapperAPILib.TimeAndSalesItem item = new
            TCQuoteWrapperAPILib.TimeAndSalesItem();
        for (int i = 0; i < TimeAndSales.ItemCount && i < 100; ++i)
        {
            TimeAndSales.GetItem(i, item);
        }
    }
    // 分價表
    else if (DataType == TCoreAPI.Defines.DATA_TYPE_PRICE_LIST)
    {
        TCQuoteWrapperAPILib.PriceByVolume PriceByVolume =
            (TCQuoteWrapperAPILib.PriceByVolume)QuoteData;
        TCQuoteWrapperAPILib.PriceByVolumeItem item = new
            TCQuoteWrapperAPILib.PriceByVolumeItem();
        for (int i = 0; i < PriceByVolume.ItemCount && i < 50; ++i)
        {
            PriceByVolume.GetItem(i, item);
        }
    }
    // 1分K
    else if (DataType == TCoreAPI.Defines.DATA_TYPE_1MK)
    {

```

```
TCQuoteWrapperAPILib.TechnicalAnalysis Technical =
    (TCQuoteWrapperAPILib.TechnicalAnalysis)QuoteData;
TCQuoteWrapperAPILib.TechnicalAnalysisItem item = new
    TCQuoteWrapperAPILib.TechnicalAnalysisItem();
for (int i = 0; i < Technical.ItemCount && i < 50; ++i)
{
    Technical.GetItem(i, item);
}
}
// 1日K
else if (DataType == TCoreAPI.Defines.DATA_TYPE_1DK)
{
    TCQuoteWrapperAPILib.TechnicalAnalysis Technical =
        (TCQuoteWrapperAPILib.TechnicalAnalysis)QuoteData;
    TCQuoteWrapperAPILib.TechnicalAnalysisItem item = new
        TCQuoteWrapperAPILib.TechnicalAnalysisItem();
    for (int i = 0; i < Technical.ItemCount && i < 100; ++i)
    {
        Technical.GetItem(i, item);
    }
}
}

// 10.8.3 分類表更新事件，分類表必須在此事件發生後取得 (Please refer to 13.1.7)
void QuoteAPI_OnSymbolClassificationsUpdate()
{
    string param1 = ""; // LV1 key
    string param2 = ""; // LV2 key
    string param3 = ""; // LV3 key
    string s = QuoteAPI.GetSymbolClassifications(I4Framework.Prop.AppCulture, param1,
        param2, param3);
    // ex: EXCHANGE_8,利率|EXCHANGE_7,貴金屬|EXCHANGE_5,基本金屬|EXCHANGE_4,外匯
    |EXCHANGE_6,能源|EXCHANGE_3,農產品|EXCHANGE_1,LME|EXCHANGE_2,股指|HKEX,香港期
    貨交易所
}
```

## Chapter 11 How to Use TCORE Trade API

### 11.1 Add a COM TCTradeWrapperAPILib Reference to Project

### 11.2 Create a COM TradeAPI Object

```
TCTradeWrapperAPILib.CTCTradeAPIEx TradeAPI = new TCTradeWrapperAPILib.CTCTradeAPIEx();
```

### 11.3 Add Event Handlers

```
TradeAPI.OnCommandMsg += new  
    TCTradeWrapperAPILib._ICTCTradeAPIExEvents_OnCommandMsgEventHandler(TradeAPI_OnC  
ommandMsg);  
TradeAPI.OnAccountUpdate += new  
    TCTradeWrapperAPILib._ICTCTradeAPIExEvents_OnAccountUpdateEventHandler(TradeAPI_On  
AccountUpdate);  
TradeAPI.OnExecutionReport += new  
    TCTradeWrapperAPILib._ICTCTradeAPIExEvents_OnExecutionReportEventHandler(TradeAPI_O  
nExecutionReport);  
TradeAPI.OnReportReset += new  
    TCTradeWrapperAPILib._ICTCTradeAPIExEvents_OnReportResetEventHandler(TradeAPI_OnRe  
portReset);  
TradeAPI.OnCancelReplaceError += new  
    TCTradeWrapperAPILib._ICTCTradeAPIExEvents_OnCancelReplaceErrorEventHandler(TradeAPI  
_OnCancelReplaceError);
```

### 11.4 Create a Connection with TCore

```
TCoreAPI.Trade.Connect(TradeAPI);  
OR  
TradeAPI.Connect(/*Host Address*/, /*System Name*/, /*App Service Key */, 1);
```

### 11.5 NewOrder

```
// 建立封包期貨物件，並設定屬性值。  
// Please refer to NOPFuture of Trading Properties of Chapter 14  
TCTradeWrapperAPILib.NOPFuture future = new TCTradeWrapperAPILib.NOPFuture();  
future.Symbol  
future.BrokerID  
future.Account  
future.Price  
future.StopPrice // Order Type : 3, 4 時需要帶入  
future.OrderType // 1:Market 2:Limit 3:StopMarket 4:StopLimit  
future.TimeInForce // 1:ROD 2:IOC 3:FOK  
future.OrderQty // 委託口數  
future.Side // buy:1 sell:2  
future.Strategy // 策略名稱或元件名稱  
future.UserKey1 // 請帶入交易帳號  
future.UserKey2 // 請帶入 AppName  
future.PositionEffect // 4:Auto 1:Close  
// 送出委託單  
TradeAPI.NewOrder(TCoreAPI.Defines.SECURITY_TYPE_SYMBOL /*9*/, future);
```

## 11.6 NewOCOProtectOrder

```
// 下單追加停損停利
// 封包期貨物件，並設定屬性值。Please refer to NOPFuture of Trading Properties of Chapter 14
TCTradeWrapperAPILib.NOPFuture future = new TCTradeWrapperAPILib.NOPFuture();
future.Symbol
future.BrokerID
future.Account
future.Price
future.StopPrice      // Order Type : 3, 4 時需要帶入
future.OrderType      // 1:Market 2:Limit 3:StopMarket 4:StopLimit
future.TimeInForce    // 1:ROD 2:IOC 3:FOK
future.OrderQty       // 委託口數
future.Side           // buy:1 sell:2
future.Strategy        // 策略名稱或元件名稱
future.UserKey1       // 請帶入交易帳號
future.UserKey2       // 帶入 AppName
future.PositionEffect // 4:Auto
int StopLossPrice     // 檔位 or 價格的數值，如不使用其值設為0
int TakeProfitPrice   // 檔位 or 價格的數值，如不使用其值設為0
int ByTickSize        // 0:價位 1:檔位
int ProtectMarket     // 0:表示 保護單觸價後 以限價送出，1:表示 保護單觸價後 以市價送出
// 送出追加停損停利委託單
TradeAPI.NewOCOProtectOrder(TCoreAPI.Defines.SECURITY_TYPE_SYMBOL /*9*/, future,
StopLossPrice, TakeProfitPrice, ByTickSize, ProtectMarket);
```

## 11.7 CancelOrder

```
// 刪單帶入 Report ID
TradeAPI.CancelOrder(ReportId, "");
```

## 11.8 ReplaceOrder

```
TCTradeWrapperAPILib.NOPReplaceOrder param = new
TCTradeWrapperAPILib.NOPReplaceOrder();
param.ReplaceExecType = ReplaceExecType;      // 0:改價 1:改量 2:改價&量
param.ReportID = report_id;
if (ReplaceExecType == 1 || ReplaceExecType == 2)
{
    param.OrderQty = qty;
}
if (ReplaceExecType == 0 || ReplaceExecType == 2)
{
    param.Price = price;
    param.StopPrice = stop_price;
}
// 送出改單
TradeAPI.ReplaceOrder(param);
```

## 11.9 Trade API Event Processing

// 11.9.1 Trade API 命令訊息事件 (**Please refer to 14.2.1**)

```
void TradeAPI_OnCommandMsg(int MsgType, int MsgCode, string MsgString)
{
    if (MsgType == 2 /*TCoreAPI.Defines.COMMAND_MSG_SYSTEM_STATUS*/)
    {
        switch (MsgCode)
        {
            case TCoreAPI.Defines.CONNECT_STATUS_CONNECTED:
                // Connected
                break;
            case TCoreAPI.Defines.CONNECT_STATUS_DISCONNECTED:
                // Disconnected
                break;
        }
    }
}
```

// 11.9.2 帳務相關事件 (**Please refer to 14.2.2**)

```
void TradeAPI_OnAccountUpdate(int dataType, string accMask, int itemCount)
{
    if (dataType == TCoreAPI.Defines.ACCOUNT_DATA_TYPE_ACCOUNT)
    {
        // 帳號資料
        TTradeWrapperAPILib.ADIAccount account = new
            TTradeWrapperAPILib.ADIAccount();
        TradeAPI.GetAccountData(TCoreAPI.Defines.ACCOUNT_DATA_TYPE_ACCOUNT, i, "",
            account);
    }
    else if (dataType ==
        TCoreAPI.Defines.ACCOUNT_DATA_TYPE_FUTURE_REALTIME_POSITION)
    {
        // 持倉資料
        for (int i = 0; i < itemCount; ++i)
        {
            TTradeWrapperAPILib.ADIFuturePosition position = new
                TTradeWrapperAPILib.ADIFuturePosition();
            TradeAPI.GetAccountData(TCoreAPI.Defines.ACCOUNT_DATA_TYPE_FUTURE_RE
                ALTIME_POSITION, i, accMask, position);
        }
    }
    else if (dataType ==
        TCoreAPI.Defines.ACCOUNT_DATA_TYPE_FUTURE_REALTIME_MARGIN)
    {
        // 帳務資料
        for (int i = 0; i < itemCount; ++i)
        {
            TTradeWrapperAPILib.ADIFutureMargin margin = new
```

```
TCTradeWrapperAPILib.ADIFutureMargin();
TradeAPI.GetAccountData(TCoreAPI.Defines.ACCOUNT_DATA_TYPE_FUTURE_RE
    ALTIME_MARGIN, i, accMask, margin);
    }
}

// 11.9.3 交易委託成交回報事件 ( 即時 )
void TradeAPI_OnExecutionReport(string ReportID)
{
    TCTradeWrapperAPILib.RPTExecutionReport report = new
        TCTradeWrapperAPILib.RPTExecutionReport();
    // 依 ReportID 取得回報物件 (Please refer to 14.1.8)
    TradeAPI.GetReportData(0 /*By Report ID*/, reportID, report);
}

// 11.9.4 交易委託成交回報事件 ( 回補 , 當第一次連到 TCore 或 TCore 重送 )
void TradeAPI_OnReportReset()
{
    TCTradeWrapperAPILib.RPTReports reports = new TCTradeWrapperAPILib.RPTReports();
    TradeAPI.GetReportData(3 /*All Orders*/, "", reports); // Please refer to 14.1.8
    // 依筆數抓出所有的回報資訊
    for (int i = 0; i < reports.Count; i++)
    {
        string reportID = reports.GetReportID(i);
        TCTradeWrapperAPILib.RPTExecutionReport report = new
            TCTradeWrapperAPILib.RPTExecutionReport();
        TradeAPI.GetReportData(0 /*By Report ID*/, reportID, report);
    }
}

// 11.9.5 刪改單錯誤回報事件
void TradeAPI_OnCancelReplaceError(string ReportID)
{
}
```



## Chapter 12 How to Creating a Chart Component

### 12.1 Create a New Project

1. 選擇 WPF Custom Control Library
2. 將範例專案 `TC4AppSample\TechLineSample\bin\Debug` 目錄內所有檔案複製到專案\bin\Debug

### 12.2 Modify Project Properties

1. 將 Output path 改為 `bin\Debug\Components\`
2. 啟動應用程式設為 專案目錄\bin\Debug\I4Framework.exe
3. 工作目錄設為 專案目錄\bin\Debug\

### 12.3 Add References to Project

( 以下皆須要將 Copy Local 屬性設訂為 False )

1. bin\Debug\I4Framework.exe
2. bin\Debug\Common Components\TCoreAPI
3. bin\Debug\Components\TechLine.dll
4. bin\Debug\Components\TechApp.dll

Screenshot of the Chart App



Figure 12-1

## 12.4 Create a new chart component

新增指標 Class，並繼承 **TechLine.Components.Base.BaseControl**，實作如下

```
public partial class MyIndex : TechLine.Components.Base.BaseControl
{
    // 資訊欄顯示元件，顯示目前十字座標所在指標的值
    TechLine.Components.Base.InfoItem myInfoItem =
        new TechLine.Components.Base.InfoItem();

    // 用來指定的全域屬性，(請參考 12.6)
    // 此型態為 Brush，這裡為 上漲、下跌、平盤 顏色
    Base.BaseProperty<Brush> m_upBrush, m_dnBrush, m_parBrush;

    // 自定繪圖元件，用來綁定 上漲、下跌、平盤 顏色
    Shape drawShapeUp = new drawShapeUp;
    Shape drawShapeDn = new drawShapeUp;
    Shape drawShapePar = new drawShapeUp;

    // 提供設定視窗綁定的屬性，型態為 int
    Base.BaseProperty<int> myInteger = new Base.BaseProperty<int>(0);

    // 提供設定視窗綁定的屬性，型態為 Boolean
    Base.BaseProperty<Boolean> isShowTag = new BaseProperty<Boolean>(false);

    // 初始預設參數
    override public void TC_SetComponent(String param)
    {
        /** param 為自行定義的參數字串，用來設定指標預設值，如：顏色、指標的參數...等，
            來自，語系\Configures\TechLineList.txt 的預設值 或 透過 TC_GetParameter的儲存值 */

        /** 取得指定的全域屬性 (Please refer to 12.8.6) */
        //上漲顏色
        m_upBrush = GetGlobalProperty(GlobalSettingType.BrushProperty,
            GlobalSettingKey.ChartUpColor, Brushes.Red) as
            Base.BaseProperty<Brush>;
        // 下跌顏色
        m_dnBrush = GetGlobalProperty(GlobalSettingType.BrushProperty,
            GlobalSettingKey.ChartDnColor, Brushes.Green) as
            Base.BaseProperty<Brush>;
        // 平盤顏色
        m_parBrush = GetGlobalProperty(GlobalSettingType.BrushProperty,
            GlobalSettingKey.ChartParColor, Brushes.White) as
            Base.BaseProperty<Brush>;

        /** 自訂 Shape 與 全域屬性 的綁定 */
        // 上漲顏色
        drawShapeUp.SetBinding(Shape.StrokeProperty, GetBindingObject(m_upBrush));
        // 下跌顏色
        drawShapeDn.SetBinding(Shape.StrokeProperty, GetBindingObject(m_dnBrush));
    }
}
```

```
// 平盤顏色
drawShapePar .SetBinding(Shape.StrokeProperty, GetBindingObject(m_parBrush));
}

// 回傳儲存所需預設參數字串
override public String TC_GetParameter()
{
    // 傳回自行定義的參數字串，如：顏色、指標的參數...等，
    // 供下次載入 ( TC_SetComponent ) 預設值用。
    return param;
}

// 計算技術線所需資料和畫面最高最低值，
// 當 refreshData == true 表示資料有異動，必需重算資料
// 如果只有 scroll bar 移動 refreshData == false
override public void TC_SetTechData(bool refreshData)
{
    // 利用 GetBarData(index) & TotalBarCount 取得 KBar 資料，
    // 將邏輯運算後，結果存到自定的變數中，以供 TC_DrawTech 繪圖時使用
    // 並將 View 中 最高值，最低值 存放在 TC_viewMax, TC_viewMin
    // 若指標資料存放為List<double>型態，可使用setDataLimit(List<double>, CharType)
}

// 要求重畫指標圖
override public void TC_DrawTech()
{
    // 從自行運算結果的變數中取出欲作畫的值，
    // 並利用 ViewBarIndex、ViewBarCount、BarWidth & BarGap，
    // 得知 起始的 Bar index、畫面的 Bar 數、Bar 寬度、Bar 的間隙，
    // 利用 Tech_Canvas 及 全域或自定屬性 作畫。
}

// 即時資料更新通知，isAdd=true 為新增一筆 Kbar
override public void TC_SetRealTimeData(bool isAdd = false)
{
    // 可利用 GetLastBarData(0) 取得最後一筆 KBar 資料
}

// 當 TC_SetRealTimeData 被呼叫後，會呼叫此方法
override public void TC_RedrawLastBar()
{
    // 當畫面最後一根為即時 Chart 圖時，並且只有更新此 KBar 的資訊，
    // 利用在 TC_SetRealTimeData 運算好的值，重畫最後一根 Bar 圖，
    // 避免重畫整個 Chart 圖
}

// 資訊欄，項目元件列表
override public List<TechLine.Components.Base.InfoItem> TC_GetInfoItemList()
{
    List<TechLine.Components.Base.InfoItem> list =
```

```
new List<TechLine.Components.Base.InfoItem>();
// 資訊欄，顯示的名稱
myInfoItem.text_title.Text = "MyChart";
list.Add(myInfoItem); // 加入欲顯示至資訊欄的元件，可多個。
return list;
}

// 指標元件自定的參數項目列表，供使用者改變設定用
override public List<ParamControl> TC_ShowSettingItem()
{
    List<ParamControl> list = new List<ParamControl>();

    /** 自定屬性與設定畫面元件的綁定，
    提供的方法與說明，請參考 12.9 Methods of BaseControl */
    list.Add(SettingBindingBrush(m_upBrush, "上漲顏色"));
    list.Add(SettingBindingBrush(m_dnBrush, "下跌顏色"));
    list.Add(SettingBindingBrush(m_parBrush, "平盤顏色"));

    list.Add(SettingBindingInteger(myInteger, "週期", 1, 100));
    list.Add(SettingBindingBoolean(isShowTag, "是否顯示標籤"));
    return list;
}

// 被要求資訊欄資訊的更新，當十字座標 KBar 的位置改變時就會被呼叫
override public void TC_ShowInfo(int idx)
{
    // 改變顯示資訊欄的值
    myInfoItem.CO = /*根據 idx 顯示對應的值 */;
}

// 被要求清除指標畫面
public override void TC_ClearView()
{
    base.TC_ClearView();
    // 在此加入指標清除 Tech_Canvas，及技術資料的方法
}
}
```

## 12.5 Add a Chart Component to the List of Charts

指標元件的列表設定檔在 ( 語系\Confignures\TechLineList.txt )

a. 說明：

類別 ( 0 價格，1 量，2 趨勢 ) : 名稱縮寫(資訊欄顯示): 名稱: 元件名稱 [: 指標預設參數]

b. 範例：

2: 指標: 第三方指標: MyNamespace.Index[:15,#0000FF]

## 12.6 Classes of the TechLine.Components.Base

BaseControl	// 指標元件父類別
BaseProperty<T>	// 用來建立可綁定的屬性物件，T 可為 int, Boolean, Brush
ParamControl	// 指標設定視窗項目元件
InfoItem	// 指標資訊欄項目元件

## 12.7 Properties of BaseControl

int DecimalCnt	// 顯示小數位數（預設值 4）
int ChartType	// 指標類別 0：價, 1：量, 2：趨勢（此值為設定檔定義）
ScaleInterface TC_ScaleInterface	// 設置或取得刻度畫布
Canvas Tech_Canvas	// Chart 畫布
double CanvasBoundOffset	// 下方邊界偏移值
bool IsLastBarInView	// 可視畫面上最後一筆 KBar 是否為即時資料
int TimeType	// KBar 時間類別
int MinuteType	// KBar 時間間距
int ViewBarIndex	// 左邊第一根 KBar 在全部資料的位置
int TotalBarCount	// 全部 KBar 資料筆數
int ViewBarCount	// 畫面上 KBar 資料筆數
double BarWidth	// KBar 寬度
double BarGap	// KBar 左右間距
double TC_viewMax	// 設置或取得畫面上自己資料最大值
double TC_viewMin	// 設置或取得畫面上自己資料最小值
double TC_scaleMax	// 設置或取得畫面上刻度最大值
double TC_scaleMin	// 設置或取得畫面上刻度最小值
long MaxPrice	// 取得畫面上資料價格最大值
long MinPrice	// 取得畫面上資料價格最小值
int MaxVolume	// 取得畫面上資料成交量最大值
int MinVolume	// 取得畫面上資料成交量最小值
double UnitHeight	// 獲得單位刻度對應高度

## 12.8 Methods of BaseControl

// 12.8.1 獲得第 idx 根 KBar 資訊 ( 如 : 開,高,低,收,量,時間 )

TechLine.Controller.KData GetBarData(int idx)

// 12.8.2 獲得由後面往前第 idx 根 KBar 資訊 , idx == 0 表示最後一根 KBar

TechLine.Controller.KData GetLastBarData(int idx = 0)

// 12.8.3 轉換為顯示的字串格式

string TransValueFormat(double dval)

// 12.8.4 獲得 Y 軸 , 所對應的刻度數值

double GetValueFromHeight(double dHeight)

// 12.8.5 獲得List中最大和最小值 , 並設給TC\_viewMax,TC\_viewMin ,

void setDataLimit(List<double> datalist, int itype = 0, bool isMulti = false, int iperiod = 0)

// 12.8.6 獲得全域變數屬性 , Key可帶入自訂或系統預設GlobalSettingKey , DefaultValue預設值

Object GetGlobalProperty(GlobalSettingType type, Object Key, Object DefaultValue)

// 12.8.7 獲得全域變數的綁定 , 用來與自訂元件綁定

Binding GetBindingObject(Object source)

/\*\* 如下方法提供設定畫面元件的綁定 \*\*/

// 12.8.8 帶入標題文字 , 產生說明文字項目

ParamControl SettingLabelString(string s\_title)

// 12.8.9 帶入 IntegerProperty、標題文字、最低、最高值 , 產生數值設定元件

ParamControl SettingBindingInteger(IntegerProperty source, string s\_title, int imin = int.MinValue,  
int imax = int.MaxValue)

// 12.8.10 帶入 BrushProperty、標題文字 , 產生設定顏色元件

ParamControl SettingBindingBrush(BrushProperty source, string s\_title)

// 12.8.11 帶入 BooleanProperty、標題文字 , 產生CheckBox設定元件

ParamControl SettingBindingBoolean(BooleanProperty source, string s\_title)

// 12.8.12 帶入 IntegerProperty、標題文字、內容列表(用 , 分隔) , 產生ComboBox元件

ParamControl SettingBindingComboBox(IntegerProperty source, string s\_title, string ComboList)

// 12.8.13 帶入 Shape 元件、標題文字、種類(0:Stroke, 1:Fill) , 產生設定顏色元件

ParamControl SettingBindingColor(Shape source, string s\_title, int type = 0)

// 12.8.14 帶入 Shape 元件、標題文字、最大值 , 產生設定線寬數值元件

ParamControl SettingBindingStrokeThickness(Shape source, string s\_title, int imax =  
int.MaxValue)

## Chapter 13 TCORE Quoting API Reference

### 13.1 Quoting Methods

#### 13.1.1 Connect

函式宣告	Connect([in] BSTR HostAddress, [in] BSTR SystemName, [in] BSTR ServiceKey,[in] int ConnectType, [out,retval] LONG* pRet);
說明	Connect TCore。若TCore未啟動，則會先啟動TCore，再進行Connect。
參數	HostAddress：TCore位址。 SystemName：TCore系統名稱。 ServiceKey: App Service Key。 ConnectType：1。 pRet：回傳值。定義如下 <pre> #define CONNECT_RETURN_FAIL 0 #define CONNECT_RETURN_CONNECTING 1 #define CONNECT_RETURN_CONNECTED 2 </pre>

#### 13.1.2 Disconnect

函式宣告	Disconnect(void);
說明	Disconnect TCore。
參數	

#### 13.1.3 SubQuote

函式宣告	SubQuote([in] LONG DataType, [in] IDispatch* DataParameters, [out,retval] LONG* pSubCount);
說明	訂閱報價資料。
參數	DataType：定義說明如後。 DataParameters：訂閱資料物件( <b>DataParameters</b> )。 pSubCount：回傳值。0:失敗 1:成功
備註	DataType定義 <pre> #define DATA_TYPE_REALTIME 1 // 即時報價 #define DATA_TYPE_MATCH_LIST 2 // 分時表 #define DATA_TYPE_PRICE_LIST 3 // 分價表 #define DATA_TYPE_1K 4 // 1分K #define DATA_TYPE_DK 5 // 1日K </pre>

### 13.1.4 UnsubQuote

函式宣告	UnsubQuote([in] LONG DataType, [in] IDispatch* DataParameters, [out,retval] LONG* pSubCount);
說明	解除訂閱報價資料。
參數	DataType : 定義說明如後。 DataParameters : 訂閱資料物件( <b>DataParameters</b> )。 pSubCount : 回傳值。 0:失敗 1:成功
備註	DataType定義 <pre>           #define DATA_TYPE_REALTIME      1      // 即時報價           #define DATA_TYPE_MATCH_LIST    2      // 分時表           #define DATA_TYPE_PRICE_LIST    3      // 分價表           #define DATA_TYPE_1K            4      // 1分K           #define DATA_TYPE_DK            5      // 1日K           </pre>

### 13.1.5 SymbolLookup

函式宣告	SymbolLookup([in] LONG SymbolType, [in] BSTR Keyword, [out,retval] BSTR* pstrRet)
說明	商品查詢。
參數	SymbolType : 定義說明如後。 Keyword : 商品關鍵字。 pstrRet : 回傳值。 0:失敗 1:成功
備註	SymbolType定義 <pre>           #define SymbolLookup_All          0           #define SymbolLookup_PartialMatch_STK 1           #define SymbolLookup_PartialMatch_FUT 2           #define SymbolLookup_PartialMatch_OPT 3           #define SymbolLookup_PartialMatch_EXCHANGE 5           #define SymbolLookup_FullMatch_STK 11           #define SymbolLookup_FullMatch_FUT 12           #define SymbolLookup_FullMatch_OPT 13           #define SymbolLookup_FullMatch_EXCHANGE 15           </pre>

### 13.1.6 DoSomething

函式宣告	DoSomething([in] LONG FuncType, [in] LONG Param1, [in] BSTR Param2, [in] VARIANT* Param3);
說明	傳送訊息給TCore，要求TCore進行某些預定義的動作。
參數	FuncType : 訊息Type。



	Param1：數值參數資料。 Param2：字串參數資料。 Param3：variant參數資料。
--	---

### 13.1.7 GetSymbolClassifications

函式宣告	GetSymbolClassifications([in] BSTR Language, [in] BSTR Classify, [in] BSTR Exchange, [in] BSTR Symbol, [out,retval] BSTR* RetVal);
說明	傳送訊息給TCore，要求TCore進行某些預定義的動作。
參數	Language：語系。 Classify：分類表ID。 Exchange：分類交易所。 Symbol：分類商品。
備註	使用說明 1. Classify Exchange Symbol均帶空白，可取得分類表ID。 2. Classify帶入分類表ID，可取得分類交易所。 3. Classify帶入分類表ID，Exchange帶入分類交易所，可取得分類商品。 4. Classify帶入分類表ID，Exchange帶入分類交易所，Symbol帶入分類商品，可取得該商品有哪些交易月份。

### 13.1.8 GetProductCurrency

函式宣告	GetProductCurrency([in] BSTR SymbolID, [out,retval] BSTR* Currency);
說明	取得商品幣別。
參數	SymbolID：商品ID。 Currency：回傳幣別。

### 13.1.9 GetCurrencyExchangeRate

函式宣告	GetCurrencyExchangeRate([in] BSTR Currency, [out,retval] BSTR* ExchangeRate);
說明	取得商品幣別。
參數	Currency：幣別對應，ex.US/HK。 ExchangeRate：回傳匯率。

### 13.1.10 GetGeneralService

函式宣告	GetGeneralService([in] BSTR Key, [out,retval] BSTR* RetVal);
說明	取得存放在 TCore 中的設定資料。
參數	Key : 欲取得資料對應的Key。 RetVal : 回傳資料。

### 13.1.11 SetGeneralService

函式宣告	SetGeneralService([in] BSTR Key, [in] BSTR Val);
說明	儲存自行定義的資料至 TCore 中，可供其他App存取。
參數	Key : 欲取資料對應的Key。 Val : 欲取的資料內容。

### 13.1.12 TopicPublish

函式宣告	TopicPublish([in] BSTR Topic, [in] LONG IParam, [in] BSTR strParam, [in] VARIANT* pvParam);
說明	Apps間，透過TCore傳送訊息，該訊息會傳送給，有訂閱該Topic的元件。
參數	Topic : 訊息傳送的Topic。 IParam : 數值參數資料。 strParam : 字串參數資料。 pvParam : variant參數資料。

### 13.1.13 TopicSub

函式宣告	TopicSub([in] BSTR strTopic);
說明	訂閱Topic訊息。
參數	Topic : 訂閱訊息Topic。

### 13.1.14 TopicUnsub

函式宣告	TopicUnsub([in] BSTR strTopic);
說明	解除訂閱Topic訊息。
參數	Topic：解除訂閱訊息Topic。

### 13.1.15 GeneralQuery

函式宣告	GeneralQuery([in] LONG lIndex, [in] BSTR strQueryName, [in] BSTR strQueryParam, [in] LONG lContentEncodeType, [in] BSTR strAcctMask);
說明	通用CGI查詢。
參數	lIndex：由Apps自行編入，做查詢回覆時識別用。 strQueryName：用作CGI查詢的名稱，用預先設定於TAC中。 strQueryParam：CGI參數資料。 lContentEncodeType：server回傳資料的編碼格式。定義如下。 strAcctMask：若要依據帳號的交易連線查詢，則須帶AcctMask，否則帶空白。格式為[BrancdID-CustID]，ex.000-1234567。
備註	lContentEncodeType定義： 1：明碼 2：須先Unzip後為明碼 3：Msg2

### 13.1.16 GetInstrumentInfo

函式宣告	GetInstrumentInfo([in] LONG lType, [in] BSTR bstrSymbolID, [out,retval] BSTR* bstrRetVal);
說明	取得商品的交易設定資料。
參數	lType：定義說明如後。 bstrSymbolID：商品ID。 bstrRetVal：回傳資料。
備註	lType定義 1：上手分母值 2：跳動檔位 3：交易時間 4：顯示乘數 5：顯示分母 6：點值

## 13.2 Quoting Events

### 13.2.1 OnCommandMsg

函式宣告	OnCommandMsg([in] LONG MsgType, [in] LONG MsgCode, [in] BSTR MsgString);
說明	TCore傳送訊息。
參數	<p>MsgType : 訊息Type。</p> <pre> #define COMMAND_MSG_TYPE_ERROR          1 #define COMMAND_MSG_SYSTEM_STATUS      2 #define COMMAND_MSG_SECURITY_LOOKUP    3    // SymbolLookup 的回應 </pre> <p>MsgCode : 訊息數值資料。</p> <pre> #define CONNECT_STATUS_DISCONNECTED    0 #define CONNECT_STATUS_CONNECTING     1 #define CONNECT_STATUS_CONNECTED      2 #define SYSTEM_STATUS_LOCKSCREEN      7    // 系統通知鎖定App #define SYSTEM_STATUS_UNLOCKSCREEN    8    // 系統通知解鎖App </pre> <p>MsgString : 訊息字串資料。</p>

### 13.2.2 OnQuoteData

函式宣告	OnQuoteData([in] LONG SymbolType, [in] LONG DataType, [in] IDispatch* DataParameters);
說明	訂閱報價資料更新通知。
參數	<p>SymbolType : 定義說明如後。</p> <p>DataType : 定義說明如後。</p> <p>DataParameters : 報價資料物件 (refer to <b>DataParameters</b>)。</p>
備註	<p>1. SymbolType定義</p> <pre> #define SECURITY_TYPE_STOCK      1 #define SECURITY_TYPE_FUTURE    2 #define SECURITY_TYPE_OPTION    3 </pre> <p>2. DataType定義</p> <pre> #define DATA_TYPE_REALTIME      1    // 即時報價 #define DATA_TYPE_MATCH_LIST    2    // 分時表 #define DATA_TYPE_PRICE_LIST    3    // 分價表 #define DATA_TYPE_1K            4    // 1分K #define DATA_TYPE_DK            5    // 1日K </pre>

### 13.2.3 OnSymbolClassificationsUpdate

函式 宣告	OnSymbolClassificationsUpdate (void);
說明	商品分類表更新通知。
參數	

### 13.2.4 OnProductCurrencyUpdate

函式 宣告	OnProductCurrencyUpdate (void);
說明	商品幣別更新通知。
參數	

### 13.2.5 OnCurrencyExchangeRateUpdate

函式 宣告	OnCurrencyExchangeRateUpdate (void);
說明	匯率更新通知。
參數	

### 13.2.6 OnGeneralServiceUpdate

函式 宣告	OnGeneralServiceUpdate (void);
說明	TCore中的一些通用設定資料更新通知。
參數	

### 13.2.7 OnSetTopic

函式 宣告	OnSetTopic([in] BSTR Topic, [in] LONG lParam, [in] BSTR strParam, [in] VARIANT* pvParam);
說明	當有訂閱Topic，該Topic有新訊息傳送時。
參數	Topic：訊息傳送的Topic。 lParam：數值參數資料。

	strParam : 字串參數資料。 pvParam : variant參數資料。
--	--

### 13.2.8 OnGeneralQueryReturn

函式宣告	OnGeneralQueryReturn([in] LONG lIndex, [in] BSTR strQueryName, [in] VARIANT* varQueryResult);
說明	GeneralQuery通用CGI查詢回覆通知。
參數	lIndex : 對應原先GeneralQuery傳入的Index，供Apps識別用。 strQueryName : 對應原先GeneralQuery傳入的QueryName，供Apps識別用。 varQueryResult : CGI回傳資料。

### 13.2.9 OnInstrumentInfoUpdate

函式宣告	OnInstrumentInfoUpdate (void);
說明	商品的交易設定資料更新通知。
參數	

## 13.3 Quoting Properties

### 13.3.1 DataParameters

SecurityType	LONG	商品類別： 0：全部 1：證券 2：期貨 3：選擇權
Security	BSTR	交易所名稱
Exchange	BSTR	交易所代碼
Month	LONG	
Month2	LONG	
CallPut	BSTR	
CallPut2	BSTR	
StrikePrice	LONGLONG	
StrikePrice2	LONGLONG	
StartTime	LONG	YYMMDDHH
EndTime	LONG	YYMMDDHH

### 13.3.2 QuoteDataFuture

Exchange	BSTR	交易所代碼
ExchangeName	BSTR	交易所名稱
Security	BSTR	商品ID
Month	BSTR	商品月份
SecurityName	BSTR	商品名稱
SecurityType	LONG	商品類別
FilledTime	LONG	成交時間
TradingPrice	LONGLONG	成交價
TradeQuantity	LONG	成交單量
Change	LONGLONG	漲跌
TradeVolume	LONG	累計成交量

OpeningPrice	LONGLONG	開盤價
HighPrice	LONGLONG	當日最高價
LowPrice	LONGLONG	當日最低價
ClosingPrice	LONGLONG	收盤價
ReferencePrice	LONGLONG	參考價
UpperLimitPrice	LONGLONG	漲停價
LowerLimitPrice	LONGLONG	跌停價
YClosedPrice	LONGLONG	昨收
YTradeVolume	LONG	昨量
OpenInterest	LONG	今日未平倉量
YOpenInterest	LONG	昨日未平倉量
TradeDate	LONG	交易日期

FlagOfBuySell	LONG	內外盤
OpenTime	LONG	開盤時間
CloseTime	LONG	收盤時間
Bid	LONGLONG	買價#1
Bid1	LONGLONG	買價#2
Bid2	LONGLONG	買價#3
Bid3	LONGLONG	買價#4
Bid4	LONGLONG	買價#5
Bid5	LONGLONG	買價#6
Bid6	LONGLONG	買價#7
Bid7	LONGLONG	買價#8
Bid8	LONGLONG	買價#9
Bid9	LONGLONG	買價#10
BidVolume	LONG	買量#1
BidVolume1	LONG	買量#2
BidVolume2	LONG	買量#3
BidVolume3	LONG	買量#4



BidVolume4	LONG	買量#5
BidVolume5	LONG	買量#6
BidVolume6	LONG	買量#7
BidVolume7	LONG	買量#8
BidVolume8	LONG	買量#9
BidVolume9	LONG	買量#10
Ask	LONGLONG	賣價#1
Ask1	LONGLONG	賣價#2
Ask2	LONGLONG	賣價#3
Ask3	LONGLONG	賣價#4
Ask4	LONGLONG	賣價#5
Ask5	LONGLONG	賣價#6
Ask6	LONGLONG	賣價#7
Ask7	LONGLONG	賣價#8
Ask8	LONGLONG	賣價#9
Ask9	LONGLONG	賣價#10
AskVolume	LONG	賣量#1
AskVolume1	LONG	賣量#2
AskVolume2	LONG	賣量#3
AskVolume3	LONG	賣量#4
AskVolume4	LONG	賣量#5
AskVolume5	LONG	賣量#6
AskVolume6	LONG	賣量#7
AskVolume7	LONG	賣量#8
AskVolume8	LONG	賣量#9
AskVolume9	LONG	賣量#10
FirstDerivedBid	LONGLONG	台灣期貨交易所衍伸一檔買價
FirstDerivedBidVolume	LONG	台灣期貨交易所衍伸一檔買量
FirstDerivedAsk	LONGLONG	台灣期貨交易所衍伸一檔賣價

FirstDerivedAskVolume	LONG	台灣期貨交易所衍伸一檔賣量
TotalBidCount	LONG	委託買進總筆數
TotalBidVolume	LONG	委託買進總數量
TotalAskCount	LONG	委託賣出總筆數
TotalAskVolume	LONG	委託賣出總數量
OpenInterest	LONG	未平倉量(當日收盤)
YOpenInterest	LONG	未平倉量(前一交易日)
BuyCount	LONG	累進買進成交筆數
SellCount	LONG	累進賣出成交筆數
SettlementPrice	LONGLONG	結算價

### 13.3.3 QuoteDataStock

Exchange	BSTR	交易所代碼
ExchangeName	BSTR	交易所名稱
Security	BSTR	商品ID
SecurityName	BSTR	商品名稱
SecurityType	LONG	商品類別
FilledTime	LONG	成交時間
TradingPrice	LONGLONG	成交價
TradeQuantity	LONG	成交單量
Change	LONGLONG	漲跌
TradeVolume	LONG	累計成交量
OpeningPrice	LONGLONG	開盤價
HighPrice	LONGLONG	當日最高價
LowPrice	LONGLONG	當日最低價
ClosingPrice	LONGLONG	收盤價
ReferencePrice	LONGLONG	參考價
UpperLimitPrice	LONGLONG	漲停價
LowerLimitPrice	LONGLONG	跌停價

YClosedPrice	LONGLONG	昨收
YTradeVolume	LONG	昨量
TradeDate	LONG	交易日期
FlagOfBuySell	LONG	內外盤
OpenTime	LONG	開盤時間
CloseTime	LONG	收盤時間
Bid	LONGLONG	買價#1
Bid1	LONGLONG	買價#2
Bid2	LONGLONG	買價#3
Bid3	LONGLONG	買價#4
Bid4	LONGLONG	買價#5
Bid5	LONGLONG	買價#6
Bid6	LONGLONG	買價#7
Bid7	LONGLONG	買價#8
Bid8	LONGLONG	買價#9
Bid9	LONGLONG	買價#10
BidVolume	LONG	買量#1
BidVolume1	LONG	買量#2
BidVolume2	LONG	買量#3
BidVolume3	LONG	買量#4
BidVolume4	LONG	買量#5
BidVolume5	LONG	買量#6
BidVolume6	LONG	買量#7
BidVolume7	LONG	買量#8
BidVolume8	LONG	買量#9
BidVolume9	LONG	買量#10
Ask	LONGLONG	賣價#1
Ask1	LONGLONG	賣價#2
Ask2	LONGLONG	賣價#3

Ask3	LONGLONG	賣價#4
Ask4	LONGLONG	賣價#5
Ask5	LONGLONG	賣價#6
Ask6	LONGLONG	賣價#7
Ask7	LONGLONG	賣價#8
Ask8	LONGLONG	賣價#9
Ask9	LONGLONG	賣價#10
AskVolume	LONG	賣量#1
AskVolume1	LONG	賣量#2
AskVolume2	LONG	賣量#3
AskVolume3	LONG	賣量#4
AskVolume4	LONG	賣量#5
AskVolume5	LONG	賣量#6
AskVolume6	LONG	賣量#7
AskVolume7	LONG	賣量#8
AskVolume8	LONG	賣量#9
AskVolume9	LONG	賣量#10
TotalBidCount	LONG	委託買進總筆數
TotalBidVolume	LONG	委託買進總數量
TotalAskCount	LONG	委託賣出總筆數
TotalAskVolume	LONG	委託賣出總數量
BidSize	LONG	內盤量
AskSize	LONG	外盤量

## Chapter 14 TCORE Trading API Reference

### 14.1 Trading Methods

#### 14.1.1 Connect

函式宣告	Connect([in] BSTR HostAddress, [in] BSTR SystemName, [in] BSTR ServiceKey, [in] int ConnectType, [out,retval] LONG* plRet);
說明	Connect TCore。若TCore未啟動，則會先啟動TCore，再進行Connect。
參數	HostAddress : TCore位址 ( ".")。 SystemName : TCore系統名稱。 ServiceKey : App Service Key。 Connect Type : 1。 plRet : 回傳值。定義如下 <div style="margin-left: 40px;"> #define CONNECT_RETURN_FAIL 0  #define CONNECT_RETURN_CONNECTING 1  #define CONNECT_RETURN_CONNECTED 2 </div>

#### 14.1.2 Disconnect

函式宣告	Disconnect(void);
說明	Disconnect TCore。
參數	

#### 14.1.3 DoSomething

函式宣告	DoSomething([in] LONG FuncType, [in] LONG Param1, [in] BSTR Param2, [in] VARIANT* Param3);
說明	傳送訊息給TCore，要求TCore進行某些預定義的動作。
參數	FuncType : 訊息Type。 Param1 : 數值參數資料。 Param2 : 字串參數資料。 Param3 : variant參數資料。

### 14.1.4 GetAccountData

函式宣告	GetAccountData([in] LONG Type, [in] LONG Index, [in] BSTR AcctMask, [in] IDispatch* AccountDataItem, [out,retval] LONG* lpRet);
說明	取得帳號相關資料。
參數	<p>Type : 訊息Type。定義如下</p> <p>Index : 欲取得第幾筆資料。</p> <p>AcctMask : 欲取得哪個帳號資料，格式為[BrancdID-CustID]，ex.000-1234567。</p> <p>AccountDataItem : 資料回傳物件。</p> <p>lpRet : 回傳值。 1:成功</p>
備註	<p>1. Type定義</p> <pre> #define ACCOUNT_DATA_TYPE_ACCOUNT      1      // 帳號 #define ACCOUNT_DATA_TYPE_GROUP        2      // 群組帳號 #define ACCOUNT_DATA_TYPE_FUT_MARGIN   3      // 期貨權益數 #define ACCOUNT_DATA_TYPE_FUT_POSITION 4      // 期貨部位 #define ACCOUNT_DATA_TYPE_FUT_CHECK_SHEET 5    // 期貨對帳單 #define ACCOUNT_DATA_TYPE_STK_MARGIN   6      // 證券信用資料 #define ACCOUNT_DATA_TYPE_STK_POSITION 7      // 證券庫存 #define ACCOUNT_DATA_TYPE_STK_CHECK_SHEET 8    // 證券對帳單 #define ACCOUNT_DATA_TYPE_FUT_RT_MARGIN 9      // 即時 期貨權益數 #define ACCOUNT_DATA_TYPE_FUT_RT_POSITION 10   // 即時 期貨部位 </pre> <p>2. 若欲取得帳號資料，AcctMask帶空白。</p> <p>3. AccountDataItem 依 Type 定義</p> <pre> ACCOUNT_DATA_TYPE_ACCOUNT      refer to ADIAccount ACCOUNT_DATA_TYPE_FUT_MARGIN   refer to ADIFutureMargin ACCOUNT_DATA_TYPE_FUT_POSITION refer to ADIFuturePosition ACCOUNT_DATA_TYPE_FUT_RT_MARGIN refer to ADIFutureMargin ACCOUNT_DATA_TYPE_FUT_RT_POSITION refer to ADIFuturePosition </pre>

### 14.1.5 NewOrder

函式宣告	NewOrder([in] LONG SecurityType, [in] IDispatch* NewOrderParameters, [out,retval] LONG* RetVal);
說明	下一筆新單。
參數	<p>SecurityType : 下單商品Type。</p> <p>NewOrderParameters : 下單資料物件 (<b>refer to NOPFuture</b>)。</p> <p>RetVal : 回傳值。 0:失敗 1:成功</p>
備註	<p>SecurityType定義</p> <pre> #define SECURITY_TYPE_STOCK      1 #define SECURITY_TYPE_FUTURE     2 #define SECURITY_TYPE_OPTION     3 #define SECURITY_TYPE_SYMBOL     9 </pre>

### 14.1.6 ReplaceOrder

函式宣告	ReplaceOrder([in] IDispatch* ReplaceOrderParameters, [out,retval] LONG* RetVal);
說明	改單。
參數	ReplaceOrderParameters : 欲改單資料物件 ( <b>refer to NOPReplaceOrder</b> )。 RetVal : 回傳值。 0:失敗 1:成功

### 14.1.7 CancelOrder

函式宣告	CancelOrder([in] BSTR ReportID, [in] BSTR Key, [out,retval] LONG* RetVal);
說明	刪單。
參數	ReportID : 欲刪單的ReportID。 Key : 刪單的Key, 由Apps自行指定使用。 RetVal : 回傳值。 0:失敗 1:成功

### 14.1.8 GetReportData

函式宣告	GetReportData([in] LONG ReportType, BSTR ReportID, [in] IDispatch* ReportItems, [out,retval] LONG* RetVal);
說明	取得回報資料。
參數	ReportType : 定義說明如後。 ReportID : 欲取得該筆回報的ReportID。 ReportItems : 資料回傳物件 ( <b>refer to RPTReports</b> )。 RetVal : 回傳值。 0:失敗 1:成功
備註	ReportType定義 0 : By Report ID 1 : All Active Orders 2 : All Filled Orders 3 : All Orders

### 14.1.9 GetProductCurrency

函式宣告	GetProductCurrency([in] BSTR SymbolID, [out,retval] BSTR* Currency);
------	--

說明	取得商品幣別。
參數	SymbolID : 商品ID。 Currency : 回傳幣別。

#### 14.1.10 GetCurrencyExchangeRate

函式宣告	GetCurrencyExchangeRate([in] BSTR Currency, [out,retval] BSTR* ExchangeRate);
說明	取得商品幣別。
參數	Currency : 幣別對應, ex.US/HK。 ExchangeRate : 回傳匯率。

#### 14.1.11 GetGeneralService

函式宣告	GetGeneralService([in] BSTR Key, [out,retval] BSTR* RetVal);
說明	取得存放在 TCore 中的設定資料。
參數	Key : 欲取得資料對應的Key。 RetVal : 回傳資料。

#### 14.1.12 SetGeneralService

函式宣告	SetGeneralService([in] BSTR Key, [in] BSTR Val);
說明	儲存自行定義的資料至 TCore 中, 可供其他App存取。
參數	Key : 欲取資料對應的Key。 Val : 欲取的資料內容。

#### 14.1.13 TopicSub

函式宣告	TopicSub([in] BSTR strTopic);
說明	訂閱Topic訊息。



參數	Topic : 訂閱訊息Topic。
----	--------------------

#### 14.1.14 TopicUnsub

函式宣告	TopicUnsub([in] BSTR strTopic);
說明	解除訂閱Topic訊息。
參數	Topic : 解除訂閱訊息Topic。

#### 14.1.15 TopicPublish

函式宣告	TopicPublish([in] BSTR Topic, [in] LONG IParam, [in] BSTR strParam, [in] VARIANT* pvParam);
說明	Apps間，透過TCore傳送訊息，該訊息會傳送給，有訂閱該Topic的元件。
參數	Topic : 訊息傳送的Topic。 IParam : 數值參數資料。 strParam : 字串參數資料。 pvParam : variant參數資料。

#### 14.1.16 GeneralQuery

函式宣告	GeneralQuery([in] LONG IIndex, [in] BSTR strQueryName, [in] BSTR strQueryParam, [in] LONG IContentEncodeType, [in] BSTR strAcctMask);
說明	通用CGI查詢。
參數	IIndex : 由Apps自行編入，做查詢回覆時識別用。 strQueryName : 用作CGI查詢的名稱，用預先設定於TAC中。 strQueryParam : CGI參數資料。 IContentEncodeType : server回傳資料的編碼格式。定義如下。 strAcctMask : 若要依據帳號的交易連線查詢，則須帶AcctMask，否則帶空白。格式為 [BrancdID-CustID]，ex.000-1234567。
備註	IContentEncodeType定義： 1 : 明碼 2 : 須先Unzip後為明碼 3 : Msg2

### 14.1.17 GetInstrumentInfo

函式宣告	GetInstrumentInfo([in] LONG IType, [in] BSTR bstrSymbolID, [out,retval] BSTR* bstrRetVal);
說明	取得商品的交易設定資料。
參數	IType：定義說明如後。 bstrSymbolID：商品ID。 bstrRetVal：回傳資料。
備註	IType定義 1：上手分母值 2：跳動檔位 3：交易時間 4：顯示乘數 5：顯示分母 6：點值

### 14.1.18 NewOCOProtectOrder

函式宣告	NewOCOProtectOrder([in] LONG SecurityType, [in] IDispatch* NewOrderParameters, [in] LONGLONG StopLossPrice, [in] LONGLONG TakeProfitPrice, [in] LONG ByTickSize, [in] LONG ProtectMarket, [out,retval] LONG* RetVal);
說明	下一筆新單，且成交後，追加下停損停利單。
參數	SecurityType：下單商品Type。 NewOrderParameters：下單資料物件 ( <b>refer to NOPFuture</b> )。 StopLossPrice：停損價位 或 停損檔位。 TakeProfitPrice：停利價位 或 停利檔位。 ByTickSize： 0：表示 StopLossPrice TakeProfitPrice 為價位 1：表示 StopLossPrice TakeProfitPrice 為檔位 ProtectMarket： 0：表示 保護單觸價後 以限價送出 1：表示 保護單觸價後 以市價送出 RetVal：回傳值 ( 0:失敗 1:成功 )
備註	1. SecurityType定義 #define SECURITY_TYPE_STOCK 1 #define SECURITY_TYPE_FUTURE 2 #define SECURITY_TYPE_OPTION 3 #define SECURITY_TYPE_SYMBOL 9 2. 可只做停損 或 只做停利 StopLossPrice = 0 表不停損 TakeProfitPrice = 0 表不停利 若同時停損停利，則TCore會將兩筆單，下成OCO單。

### 14.1.19 NewAlertOrder

函式宣告	NewAlertOrder([in] LONG SecurityType, [in] IDispatch* NewOrderParameters, [in] LONG AlertField, [in] LONG AlertCondition, [in] LONGLONG AlertPrice, [out,retval] LONG* RetVal);
說明	觸價下單。
參數	<p>SecurityType：下單商品Type。</p> <p>NewOrderParameters：下單資料物件 (<b>refer to NOPFuture</b>)。</p> <p>AlertField：條件欄位，定義說明如後。</p> <p>AlertCondition：條件 ( 1:&gt; 2:&gt;= 3:= 4:&lt;= 5:&lt; )</p> <p>AlertPrice：觸發價格。</p> <p>RetVal：回傳值 ( 0:失敗 1:成功 )</p>
備註	<p>1. SecurityType定義</p> <pre>#define SECURITY_TYPE_STOCK          1 #define SECURITY_TYPE_FUTURE          2 #define SECURITY_TYPE_OPTION          3 #define SECURITY_TYPE_SYMBOL          9    // 使用Symbol</pre> <p>2. AlertField 定義</p> <p>1：成交價</p> <p>2：買價</p> <p>3：賣價</p>

### 14.1.20 SetAccountSubscriptionLevel

函式宣告	SetAccountSubscriptionLevel([in] LONG Level);
說明	帳務資料變動通知設定。
參數	<p>Level：</p> <p>0：不通知</p> <p>1：帳號變動通知 ( 預設值 )</p> <p>2：帳號 及 查詢的帳務 變動通知</p> <p>3：帳號 及 查詢的帳務 及 TCore即時運算帳務 變動通知</p>

### 14.1.21 QueryAccountData

函式宣告	QueryAccountData([in] LONG IType, [in] BSTR BrokerID, [in] BSTR Account);
說明	帳務資料查詢，觸發 OnAccountUpdate 事件。
參數	IType：定義說明如後。 BrokerID： Account：帳號
備註	Type定義 <pre>           #define ACCOUNT_DATA_TYPE_ACCOUNT          1          // 帳號           #define ACCOUNT_DATA_TYPE_GROUP            2          // 群組帳號           #define ACCOUNT_DATA_TYPE_FUT_MARGIN        3          // 期貨權益數           #define ACCOUNT_DATA_TYPE_FUT_POSITION      4          // 期貨部位           #define ACCOUNT_DATA_TYPE_FUT_CHECK_SHEET  5          // 期貨對帳單           #define ACCOUNT_DATA_TYPE_STK_MARGIN        6          // 證券信用資料           #define ACCOUNT_DATA_TYPE_STK_POSITION      7          // 證券庫存           #define ACCOUNT_DATA_TYPE_STK_CHECK_SHEET  8          // 證券對帳單           #define ACCOUNT_DATA_TYPE_FUT_RT_MARGIN     9          // 即時 期貨權益數           #define ACCOUNT_DATA_TYPE_FUT_RT_POSITION  10         // 即時 期貨部位           </pre>

## 14.2 Trading Events

### 14.2.1 OnCommandMsg

函式宣告	OnCommandMsg([in] LONG MsgType, [in] LONG MsgCode, [in] BSTR MsgString);
說明	TCore傳送訊息。
參數	<p>MsgType : 訊息Type。</p> <pre> #define COMMAND_MSG_TYPE_ERROR          1 #define COMMAND_MSG_SYSTEM_STATUS      2 </pre> <p>MsgCode : 訊息數值資料。</p> <pre> #define CONNECT_STATUS_DISCONNECTED    0 #define CONNECT_STATUS_CONNECTING     1 #define CONNECT_STATUS_CONNECTED2 #define SYSTEM_STATUS_LOCKSCREEN      7      // 系統通知鎖定App #define SYSTEM_STATUS_UNLOCKSCREEN    8      // 系統通知解鎖App </pre> <p>MsgString : 訊息字串資料。</p>

### 14.2.2 OnAccountUpdate

函式宣告	OnAccountUpdate([in] LONG Type, [in] BSTR AcctMask, [in] LONG Count);
說明	帳號及帳務資料更新通知。
參數	<p>Type : 定義說明如後。</p> <p>AcctMask : 更新帳務的帳號。</p> <p>Count : 資料筆數。</p>
備註	<p>1. Type定義</p> <pre> #define ACCOUNT_DATA_TYPE_ACCOUNT      1      // 帳號 #define ACCOUNT_DATA_TYPE_GROUP        2      // 群組帳號 #define ACCOUNT_DATA_TYPE_FUT_MARGIN   3      // 期貨權益數 #define ACCOUNT_DATA_TYPE_FUT_POSITION 4      // 期貨部位 #define ACCOUNT_DATA_TYPE_FUT_CHECK_SHEET 5    // 期貨對帳單 #define ACCOUNT_DATA_TYPE_STK_MARGIN   6      // 證券信用資料 #define ACCOUNT_DATA_TYPE_STK_POSITION 7      // 證券庫存 #define ACCOUNT_DATA_TYPE_STK_CHECK_SHEET 8    // 證券對帳單 #define ACCOUNT_DATA_TYPE_FUT_RT_MARGIN 9      // 即時 期貨權益數 #define ACCOUNT_DATA_TYPE_FUT_RT_POSITION 10   // 即時 期貨部位 </pre> <p>2. 若為帳號資料更新(ACCOUNT_DATA_TYPE_ACCOUNT)，則AcctMask為空白。</p> <p>3. 利用 GetAccountData Method 取得對應資料。(Please refer to 14.1.4)</p>

### 14.2.3 OnExecutionReport

函式宣告	OnExecutionReport([in] BSTR ReportID);
說明	回報資料更新通知。
參數	ReportID：該筆回報的RportID。

### 14.2.4 OnCancelReplaceError

函式宣告	OnCancelReplaceError([in] BSTR ReportID);
說明	刪改單錯誤回報通知。
參數	ReportID：該筆回報的RportID。
備註	目前並無使用，刪改單錯誤回報，會從OnExecutionReport通知。

### 14.2.5 OnReportReset

函式宣告	OnReportReset(void);
說明	所有回報更新通知。
參數	

### 14.2.6 OnProductCurrencyUpdate

函式宣告	OnProductCurrencyUpdate (void);
說明	商品幣別更新通知。
參數	

### 14.2.7 OnCurrencyExchangeRateUpdate

函式宣告	OnCurrencyExchangeRateUpdate (void);
說明	匯率更新通知。
參數	

### 14.2.8 OnGeneralServiceUpdate

函式宣告	OnGeneralServiceUpdate (void);
說明	TCore中的一些通用設定資料更新通知。
參數	

### 14.2.9 OnSetTopic

函式宣告	OnSetTopic([in] BSTR Topic, [in] LONG lParam, [in] BSTR strParam, [in] VARIANT* pvParam);
說明	當有訂閱Topic，該Topic有新訊息傳送時。
參數	Topic：訊息傳送的Topic。 lParam：數值參數資料。 strParam：字串參數資料。 pvParam：variant參數資料。

### 14.2.10 OnGeneralQueryReturn

函式宣告	OnGeneralQueryReturn([in] LONG lIndex, [in] BSTR strQueryName, [in] VARIANT* varQueryResult);
說明	GeneralQuery通用CGI查詢回覆通知。
參數	lIndex：對應原先GeneralQuery傳入的Index，供Apps識別用。 strQueryName：對應原先GeneralQuery傳入的QueryName，供Apps識別用。 varQueryResult：CGI回傳資料。

### 14.2.11 OnInstrumentInfoUpdate

函式 宣告	OnInstrumentInfoUpdate (void);
說明	商品的交易設定資料更新通知。
參數	

## 14.3 Trading Properties

### 14.3.1 NOPFuture

Broker	BSTR	券商
Account	BSTR	帳號
Exchange	BSTR	交易所
Security	BSTR	商品ID
Month	LONG	月份 YYYYMM
Month2	LONG	月份2 YYYYMM
Price	LONGLONG	委託價格
Side	LONG	買賣別 1 : Buy 2 : Sell
StopPrice	LONGLONG	停損價
TimeInForce	LONG	0 : None 0 None 1 : ROD Day order 2 : IOC Immediate or Cancel 3 : FOK Fill or Kill 8 : TAIFEX Auto expired Quote.(20sec)
OrderType	LONG	0 : None 1 : Market order 2 : Limit order 3 : Stop order 4 : Stop limit order 9 : ExtOrder 補單
OrderQty	LONG	下單口數
PositionEffect	LONG	0 : Open position



		1 : Close position 2 : Day trade 3 : Day trade open position 4 : Auto select Open/Close position
ExtCommands	BSTR	延伸下單參數
Strategy	BSTR	策略名稱
UserKey1	BSTR	交易帳號
UserKey2	BSTR	AppName
UserKey3	BSTR	自定字串

### 14.3.2 NOPStock

Broker	BSTR	券商
Account	BSTR	帳號
Exchange	BSTR	交易所
Security	BSTR	商品ID
MaxPriceLevels	LONG	最多撮合價位
Price	LONGLONG	委託價格
Side	LONG	買賣別 1 : Buy 2 : Sell
StopPrice	LONGLONG	停損價
TimeInForce	LONG	0 : None 1 : ROD Day order
OrderType	LONG	0: None 1 : Market 2 : Limit 3 : Stop 4 : StopLimit 5 : UpperLimit 6 : LowerLimit 7 : 平盤價 8 : 對方最佳價格 9 : 己方最佳價格 10 : 市價轉限價 11 : LimitAfterClose
TradeType	LONG	0 : Normal

		1 : Purchase on Margin 2 : ShortSell
ExCode	LONG	0 : 現股 1 : 零股 2 : 盤後
OrderQty	LONG	下單口數
ExtCommands	BSTR	延伸下單參數
Strategy	BSTR	策略名稱
UserKey1	BSTR	交易帳號
UserKey2	BSTR	AppName
UserKey3	BSTR	自定字串

### 14.3.3 ADIAccount

LoginID	BSTR	登入帳號
BrokerID	BSTR	券商 (對應下單與回報欄位)
Account	BSTR	帳號 (對應下單與回報欄位)
UserName	BSTR	登入帳號名稱
AccountName	BSTR	帳號名稱
BrokerName	BSTR	券商名稱
Status	LONG	帳號狀態(0:尚未登入 1:登入完成)
AccountType	BSTR	S/F/O
ItemType	LONG	固定1

### 14.3.4 ADIFutureMargin

LoginID	BSTR	登入帳號
UserName	BSTR	登入帳號名稱
BrokerID	BSTR	券商 (對應下單與回報欄位)
BrokerName	BSTR	券商名稱
Account	BSTR	帳號 (對應下單與回報欄位)

AccountName	BSTR	帳號名稱
TransactDate	LONG	更新日期 UTC+0
TransactTime	LONG	更新時間 UTC+0
ItemType	LONG	固定3
BeginningBalance	LONGLONG	期初結存
Commissions	LONGLONG	交易手續費
ExchangeClearinigFee	LONGLONG	交易所費用與結算費用
BrokerageFee	LONGLONG	經紀商費用
GrossPL	LONGLONG	已實現損益(未扣除費用)
OptionPremium	LONGLONG	選擇權權利金收入
NetPL	LONGLONG	已實現損益(扣除費用並加上選擇權權利金收入)
CashActivity	LONGLONG	今日出入金
EndingBalance	LONGLONG	期末結存
OpenTradeEquity	LONGLONG	未平倉損益(包含選擇權和期貨)
TotalEquity	LONGLONG	總權益數
OptionNetMarketValue	LONGLONG	選擇權市值
AccountValueAtMarket	LONGLONG	帳戶市值
InitialMarginRequirement	LONGLONG	初始保證金
MaintenanceMarginRequirement	LONGLONG	維持保證金
ExcessEquity	LONGLONG	可用餘額
MarginDeficit	LONGLONG	追繳保證金
CurrencyToSystem	BSTR	系統幣別
CurrencyConversionRate	LONGLONG	幣別轉換比率(轉換到帳戶幣別使用)
CurrencyToClient	BSTR	帳戶幣別
ConvertedAccountValueAtMkt	LONGLONG	幣別轉換後帳戶市值

### 14.3.5 ADIFuturePosition

LoginID	BSTR	
ReportID	BSTR	
OrderID	BSTR	
BrokerID	BSTR	
BrokerName	BSTR	
Account	BSTR	
AccountName	BSTR	
TransactDate	LONG	
TransactTime	LONG	
ItemType	LONG	固定4
Exchange	BSTR	
Security	BSTR	
Month	LONG	
Month2	LONG	
CallPut	BSTR	
CallPut2	BSTR	
StrikePrice	LONGLONG	
StrikePrice2	LONGLONG	
Side	LONG	買賣方向
Quantity	LONG	數量
AvgPrice	LONGLONG	成交成本均價
UnrealizedPL	LONGLONG	損益
CurrencyToSystem	BSTR	系統幣別

### 14.3.6 RPTExecutionReport

ItemType	LONG	物件類別 (get only) = 1
ReportID	BSTR	回報編號
Account	BSTR	帳號 (對應下單與回報欄位)
BrokerID	BSTR	券商
SecurityType	BSTR	S/F/O
Exchange	BSTR	交易所
Security	BSTR	商品ID
Month	LONG	月份
Month2	LONG	月份2
CallPut	BSTR	CP
CallPut2	BSTR	CP2
StrikePrice	LONGLONG	履約價
StrikePrice2	LONGLONG	履約價2
Side	LONG	買賣別 1:Buy 2:Sell
Side2		
OriginalQty	LONG	new order 時候的委託數量
OrderQty	LONG	實際委託數量=成交數量+剩餘有效委託口數
Price	LONGLONG	委託價格
StopPrice	LONGLONG	停損價
OrderType	LONG	0 : None Unknown 1 : Market Market order 2 : Limit Limit order 3 : Stop Stop order 4 : StopLimit Stop limit order 5 : Stop Stop order 6 : StopLimit Stop limit order
TradeType	LONG	0 : Normal 1 : Purchase on Margin 2 : ShortSell
PositionEffect	LONG	0 : Open Open position 1 : Close Close position

		2 : Rolled Day trade 3 : DayTradeOpen Day trade open position 4 : Auto Auto select Open/Close position
TimeInForce	LONG	0 : None None 1 : ROD Day order 2 : IOC Immediate or Cancel 3 : FOK Fill or Kill 8 : TFXQ TAIFEX Auto expired Quote.(20sec)
Strategy	BSTR	策略名稱
UserKey1	BSTR	對應委託參數的UserKey1欄位資料
UserKey2	BSTR	對應委託參數的UserKey2欄位資料
UserKey3	BSTR	對應委託參數的UserKey3欄位資料
CumQty	LONG	已成交數量
LeavesQty	LONG	委託剩餘有效量
ExecType	LONG	委託執行狀態 0 : 委託成功(包含改價改量) 1 : 部份委託成功其餘處理中 2 : 部份委託成功其餘錯誤 3 : 全部成交 4 : 部份成交其餘委託處理中 5 : 部份成交其餘刪單 6 : 部分成交尚有有效單 7 : 部份成交其餘錯誤 8 : 完全刪單成功 9 : 部份刪單成功 10 : 委託失敗 11 : 委託處理中 12 : 刪改單錯誤 13 : 洗價中 14 : ITS主機已收單(增加中)
TransactDate	LONG	更新日期
TransactTime	LONG	更新時間 (UpdateTime)
AvgPrice	LONGLONG	成交均價
ExecTypeText	BSTR	委託執行狀態描述[server錯誤訊息]
OrderStatusCount	LONG	委託回報數量
FilledOrdersCount	LONG	成交回報數量

### 14.3.7 RPTFilledOrder

ItemType	LONG	物件類別(get only) = 3
ReportID	BSTR	
Account	BSTR	帳號 (對應下單與回報欄位)
SecurityType	BSTR	S/F/O
BrokerID	BSTR	券商
Exchange	BSTR	交易所
Security	BSTR	
MatchedPrice	LONGLONG	本次成交價格
MatchedQty	LONG	本次成交數量
Month	LONG	月份
Month2	LONG	月份2
CallPut	BSTR	
CallPut2	BSTR	
StrikePrice	LONGLONG	
StrikePrice2	LONGLONG	
OrderID	BSTR	委託書編號
PositionEffect	LONG	0:Open Open position 1:Close Close position 2:Rolled Day trade 3:DayTradeOpen Day trade open position 4:Auto Auto select Open/Close position
Side	LONG	買賣別 1:Buy 2:Sell
Side2	LONG	
TransactDate	LONG	
TransactTime	LONG	交易時間
Strategy	BSTR	
UserKey1	BSTR	
UserKey2	BSTR	
UserKey3	BSTR	

### 14.3.8 NOPReplaceOrder

ReportID	BSTR	回報編號
ReplaceExecType	LONG	0:改價 1:改量
OrderQty	LONG	使用者期望有效委託量
Price	LONGLONG	使用者期望的委託價格
StopPrice	LONGLONG	

### 14.3.9 RPTReports

AddReportID	METHOD	
Count	LONG	資料筆數
GetReportID	METHOD	參數為 0 開始的數值，可讀筆數為 Count。 傳回 ReportID。
ItemType	LONG	



## Chapter 15 Packing an App for TC AppStore

1. 將 **Debug** 目錄下所有檔案複製至 **Release** 目錄
2. Rebuild Release Project 建立出 Release 版的元件
3. 將 資訊檔 ( AppInfo.xml )、圖檔 ( AppImg.ico )、目錄 ( Release )，壓縮成 ZIP 檔

a. 資訊檔：AppInfo.xml 內容如下

```
<root>
  <property key="APPDIR" value="Release" />
  <property key="APPEXE" value="I4Framework.exe" />
  <!-- 如下屬性，可提供非I4應用程式CommandLine參數(AppID,ServiceKey,語系,登入帳號) -->
  <property key="NON-I4" value="1" />
  <!-- 如下屬性，非必要，目的是提供 AppManager FeedBack 參考用 -->
  <property key="FEEDBACKPATH" value="" />
  <property key="FEEDBACKFILE" value="" />
</root>
```

b. 圖檔：AppImg.ico

c. 目錄：包含 Release 目錄及其中所需檔案

- i. I4Framework.exe
- ii. I4Framework.exe.config
- iii. **Components** 目錄下所有 DLL
- iv. **Shared Components** 目錄下 DLL ( 註：如果有用到才需要 )
- v. 所有語系目錄 ( 如：en-US, zh-CHT, zh-CHS ) 下，所有檔案

**Caution：**不要包含如下

- **Common Components** 目錄
- I4Framework.PermanentData.xml
- I4Framework.XML
- 非 App 執行會用到的檔案，盡可能不要包含，減少上下載頻寬



Figure 15-1

**Tip：**建議使用 7zip 壓縮工具，將壓縮後的附檔名改成 .zip 即可。

## Chapter 16 Publishing to the TC App Store

1. Acquire account for TC App Store
2. 建立 256x256 PNG 的應用程式圖檔，供 TC App Store 顯示用
3. 至開發者網站 (<http://app.icetech.com.cn/AppManager/>) 上傳圖檔及APP壓縮檔(Zip)

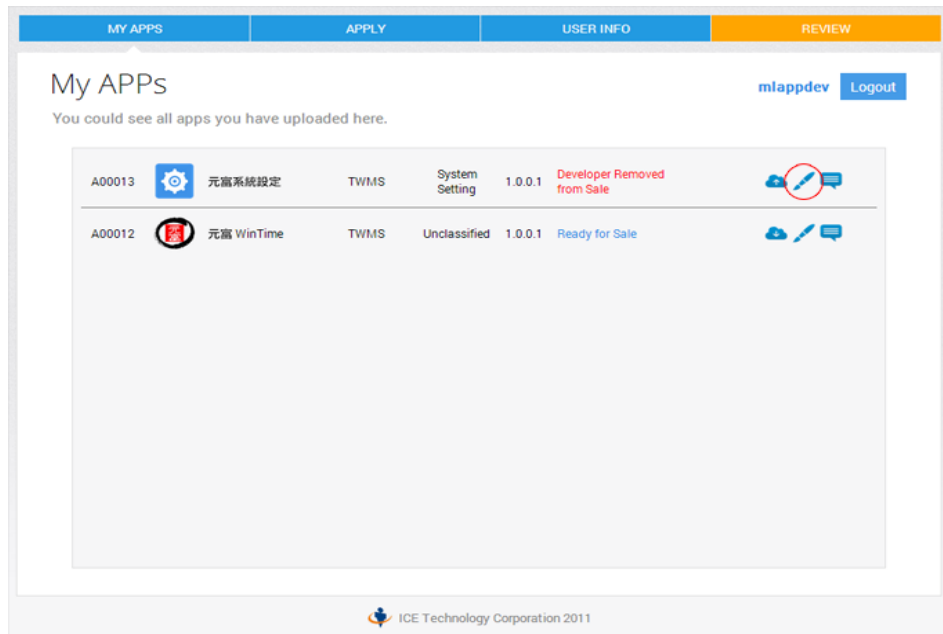


Figure 16-1

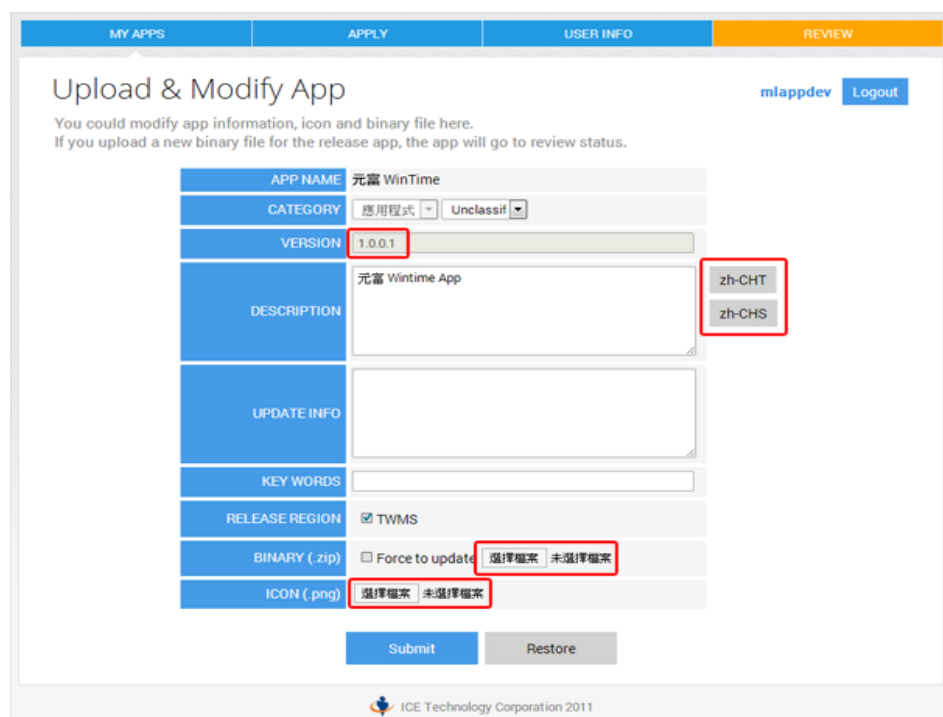


Figure 16-2

## Appendix

### A.1 Converting C++ Data Types to C#

C++ Type	C# Type	Size
BOOL	bool	1 byte
BYTE	byte	1 byte
DECIMAL	Decimal	16 bytes
DOUBLE	double	8 bytes
DWORD	uint, UInt32	4 bytes
FLOAT	float, single	4 bytes
INT, signed int	int, Int32	4 bytes
INT16, signed short int	short, Int16	2 bytes
INT32, signed int	int, Int32	4 bytes
INT64	long, Int64	8 bytes
LONG	int, Int32	4 bytes
LONG32, signed int	int, Int32	4 bytes
LONG64	long, Int64	8 bytes
LONGLONG	long, Int64	8 bytes
SHORT, signed short int	short, Int16	2 bytes
UINT, unsigned int	uint, UInt32	4 bytes
UINT16	ushort, UInt16	2 bytes
UINT32, unsigned int	uint, UInt32	4 bytes
UINT64	ulong, UInt64	8 bytes
ULONG, unsigned long	uint, UInt32	4 bytes
ULONG32	uint, UInt32	4 bytes
ULONG64	ulong, UInt64	8 bytes
ULONGLONG	ulong, UInt64	8 bytes
void*, pointers	IntPtr	x86=4 bytes, x64=8 bytes

## A.2 C#、WPF Related Websites

- WPFToolkit.Extended (<http://wpftoolkit.codeplex.com>)
- WPF Tutorial Beginning (<http://www.codeproject.com/Articles/140611/WPF-Tutorial-Beginning>)
- <http://msdn.microsoft.com/en-us/library/>
- <http://msdn.microsoft.com/en-us/vcsharp>
- <http://www.wpftutorial.net>
- <http://archive.cnblogs.com/a/1789619>
- <http://vbcity.com>
- <http://www.c-sharpcorner.com>
- <http://code.logos.com/blog/c/>

## A.3 C#、WPF Related Books

Book Title	Author	Publisher
Windows Presentation Foundation 新一代使用體驗開發實務	Adam Nathan 著、鄭淑芬 譯	悅知
WPF 與 XAML 應用程式講座	川西裕幸、高橋忍、鄧瑋敦	博碩文化
Essential WPF 精要剖析	Chris Anderson 著、劉祖亮 譯	旗標
專業WPF程式開發指南	蕭振亞	上奇科技
Visual C# 2010 程式設計速學對策	文淵閣工作室	碁峰
Visual C# 2010 程式設計與應用	許華青	藍海文化
Visual C# 2010 入門與應用	李春雄	上奇科技

## Document Revision History

Date	Revision	Author	Comments
2013/08/01	1.0	hanwen	Document Creation.
2013/08/27	1.0.1	hanwen	Added TCORE Quoting API Reference chapter. Added TCORE Trading API Reference chapter.
2013/08/28	1.0.2	hanwen	Updated TC4 App Directory Structure.
2013/08/29	1.0.3	hanwen	Added How to Use Quote API chapter. Added How to Use Trade API chapter. Added How to Creating a Chart Component chapter.
2013/09/02	1.0.4	peter	Updated How to Creating a Chart Component.
2013/09/03	1.0.5	hanwen	Updated Packing an App for TC4 AppStore.
2013/09/04	1.0.6	htyang	Updated Publishing to the TC4 AppStore.
2013/09/04	1.0.7	htyang	Updated Main Window Layout & Main Page.
2013/09/06	1.0.8	hanwen	Added about SymbolLookup on How to Use Quote API. Added about NewOCOPProtectOrder on How to Use Trade API.
2013/09/06	1.0.9	hanwen	Updated connect method on TCORE Quoting API Reference. Updated connect method on TCORE Trading API Reference.
2013/09/06	1.1.1	hanwen	Added about SubQuote & UnsubQuote description on chapter 10.
2013/09/09	1.2	hanwen	Updated chapter 13, chapter 14, chapter 15.
2013/09/10	1.2.1	hanwen	Change the layout of this document. Added SetGeneralService description on chapter 13, 14.
2013/09/11	1.2.2	hanwen	Updated NewOCOPProtectOrder description on chapter 11, 14.
2013/09/13	1.2.3	peter hanwen	Modified chapter 12. Updated GetInstrumentInfo description on chapter 13, 14.
2013/09/23	2.0	hanwen	Updated chapter 7, 8, 9 and chapter 10, 11 explanatory supplement and other minor update.