

第三版 SNS COM+ v3.3 及 v3.4 元件 使用說明手冊

2017/07/04

<u>目 錄</u>	<u>頁 次</u>
1. 簡介	2
2. 註冊元件方法	2
3. 此新版和舊版 SNS COM+ 元件差異	7
4. SNS COM+ v3 API 函式說明	8
5. 程式碼範例	23
5.1. C#.NET 視窗版單門號範例程式	23
5.2. C#.NET 視窗版多門號範例程式	37
5.3. ASP.NET 網頁範例程式	42
5.4. ASP 網頁範例程式	48
6. 常見問題及說明解決	59

1. 簡介

此第三版 SNS COM+ 元件完整支援 98 年版 SNS (Special Number Service, 簡訊特碼服務) 通訊協定實作，提供中華電信客戶透過此元件登入 SNS、發送簡訊、傳送長簡訊、查詢簡訊狀態、接收簡訊、保持連線、變更密碼、以及登出等功能。

為了使用長簡訊之新功能，本元件亦提供輔助函式，以便客戶容易使用及傳送包括：切割較長訊息至多則簡訊、以及如何傳送多則簡訊，讓受訊端手機可收到完整的單通長簡訊內容。另亦處理含特殊符號字元之純英文簡訊，使手機不會在特殊符號處出現三角型符號之亂碼。

可使用的發訊方、及受訊方號碼包括：行動電話號碼、及 emome 號碼，除了可發送國內門號之外，亦可發送國外門號 (包括大陸門號)。這些門號組合和使用者申請 SNS 特碼之權限有關，包括國外門號發送、長簡訊、及接收簡訊亦需要其權限開通，請洽本中華電信公司業務服務單位。

另外本說明文件亦提供多種程式語言範例，說明一般使用者如何使用此 SNS COM+ v3 元件，以便處理最常用的基本 API 用法，以便可快速開發客戶端自己的應用程式。

2. 註冊元件方法

若之前已安裝早期的第二版 SNS COM+ 元件，請先依之前版本作法移除該元件。注意的是，此第三版元件及之前舊版本之 API 用法不同，可參考本文件第 3 章之差異說明。

本 SNS COM+ v3 元件安裝需求為適用 Windows 主機，且須事先安裝 .NET Framework 2.0 平台，也可適用其包含版本之 .NET Framework 較高版本平台。

本 SNS COM+ v3 元件提供兩種版本 DLL，分別提供 SNS_COM+_v3.3.zip 及 SNS_COM+_v3.4.zip，其安裝方式相同，主要為是否為強式名稱 COM+ 元件，其 API 用法皆相同，主要為編譯您的應用程式環境之差異，請參考後述說明。

以 SNS_COM+_v3.4.zip 元件之安裝檔說明，它包括下列六個檔案，請先複製至某目錄，例如 C:\SNS_COM+ v3.4 目錄，它內含其中一個所要的 SnsComServer_v3.4.dll 檔元件、DLL 版本異動說明，另包括 SnsCom3Tester.exe 視窗版測試程式、以及此測試程式視窗之 C# 原始碼 Form1.cs 範例（參考後述第 5.1 章節說明）。



一般客戶開發 .NET 平台之 SNS 簡訊服務應用程式，包括：C#.NET、VB.NET 等應用程式、以及 ASP.NET 網頁專案，只需在 Visual Studio 2005（或以上）開發工具將 SnsComServer_v3.4 加入參考，即可直接引用此元件，不需另安裝程序作業。

但若是開發 ASP 網頁應用者，則須要執行下列的安裝程序，再於網頁引用參考的 CHT.SNSCOMSERVER.SnsComObject3 模組，參考後述第五章節之各程式語言範例說明。

若以前尚未安裝此 SNS COM+ v3 元件，請以系統管理員身份執行 reg_v3.3.bat 或 reg_v3.4.bat 指令，以註冊本元件 SnsComServer_v3.3.dll 或 SnsComServer_v3.4.dll 至主機中，操作如下：（下列以 v3.4 為例說明）



```
C:\Windows\System32\cmd.exe

C:\SNS_COM+ v3.4>echo off

磁碟區 C 中的磁碟是 HD_MIN7
磁碟區序號: 9670-351E

C:\SNS_COM+ v3.4 的目錄

2017/07/04 下午 05:22          45,056 SnsComServer_v3.4.dll
                1 個檔案          45,056 位元組
                0 個目錄    18,195,693,568 位元組可用

*** Install SnsComServer_v3.4.dll to .NET Framework 2.0

Microsoft (R) .NET Framework 服務安裝公用程式版本 2.0.50727.5483
Copyright (c) Microsoft Corporation. All rights reserved.

警告: 組件未宣告 ApplicationAccessControl 屬性。將依預設啟用應用程式安全性。
已安裝的組件:
    組件: C:\SNS_COM+ v3.4\SnsComServer_v3.4.dll
    應用程式: SnsComServer_v3.4
    TypeLib: C:\SNS_COM+ v3.4\SnsComServer_v3.4.tlb

磁碟區 C 中的磁碟是 HD_MIN7
磁碟區序號: 9670-351E

C:\SNS_COM+ v3.4 的目錄

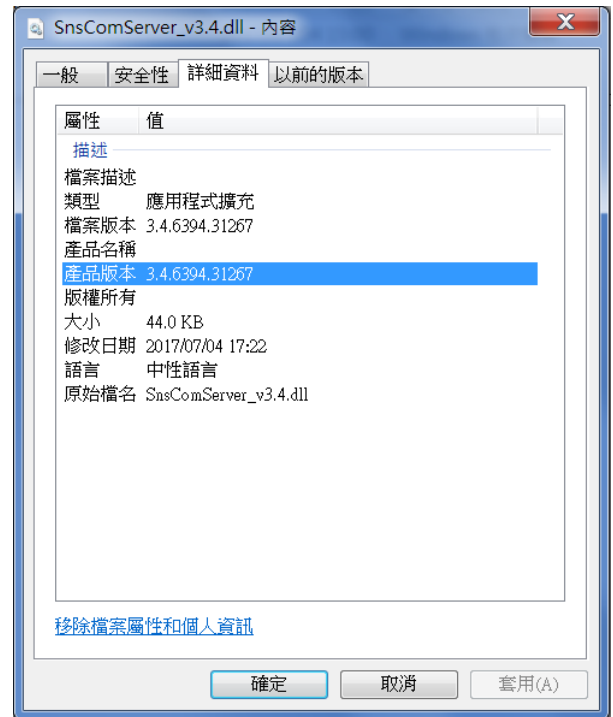
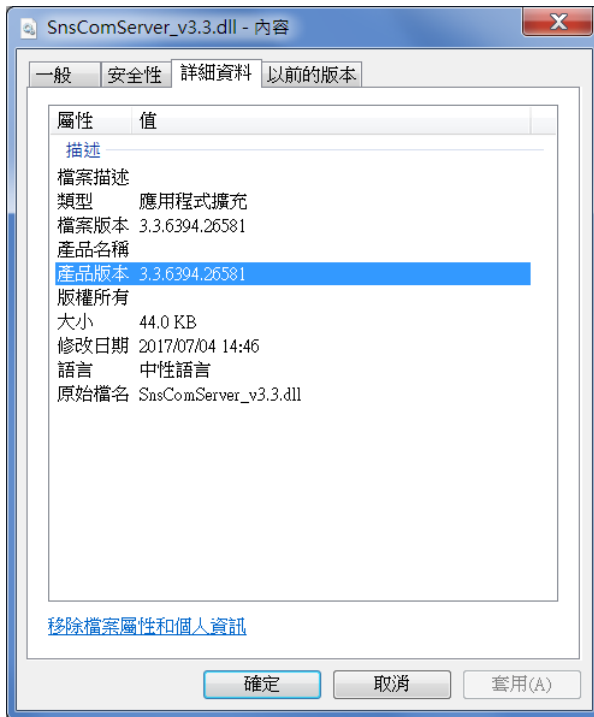
2017/07/04 下午 05:22          45,056 SnsComServer_v3.4.dll
2017/07/04 下午 06:29        10,508 SnsComServer_v3.4.tlb
                2 個檔案          55,564 位元組
                0 個目錄    18,195,648,512 位元組可用

*** 若有看到 SnsComServer_v3.4.tlb 檔，表示完成 SNS COM+ v3.4 安裝作業！
請按任意鍵繼續 . . .
```

當順利完成安裝後，此目錄新增一個 SnsComServer_v3.4.tlb 檔，使用者可依此快速識別出本元件是否已安裝至主機：

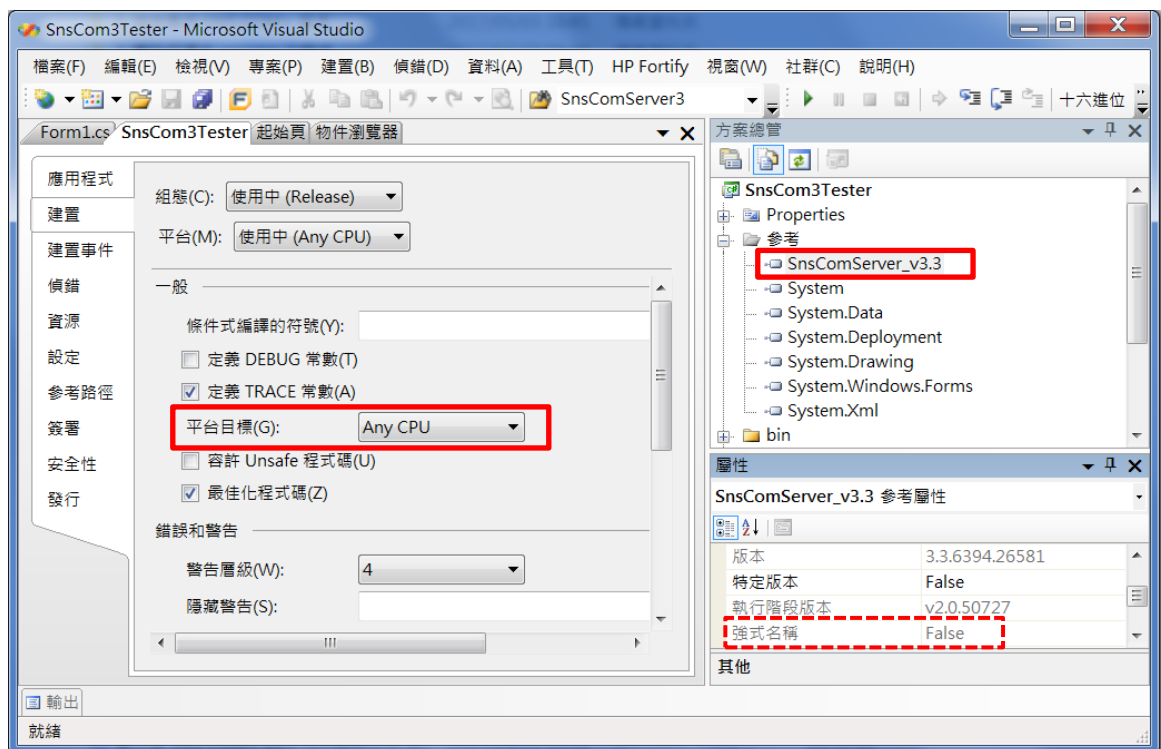


欲知 SNS COM+ v3 DLL 檔案版本，可透過滑鼠點選其 DLL 檔案，按下右鍵 → 內容 → 詳細資料即可得到其版本及修改日期等資訊，下圖為兩個版本功能皆相同，差異在於是否為強式名稱 SNS COM+ v3 DLL：

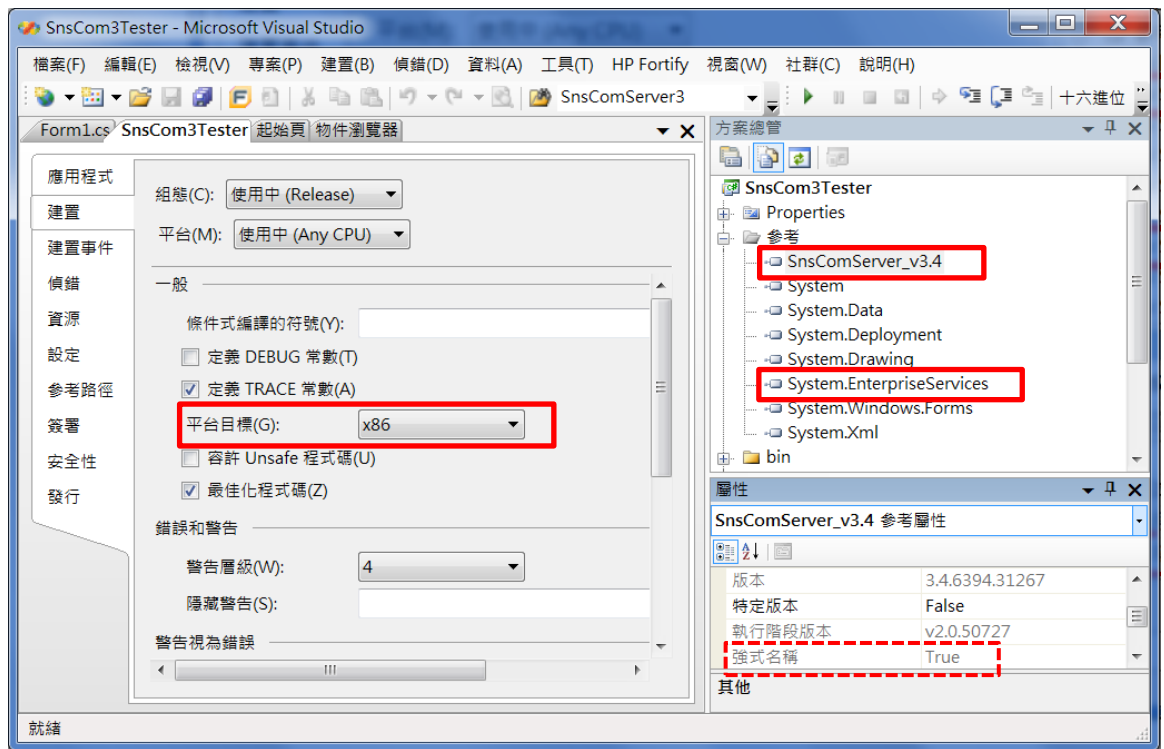


其中：

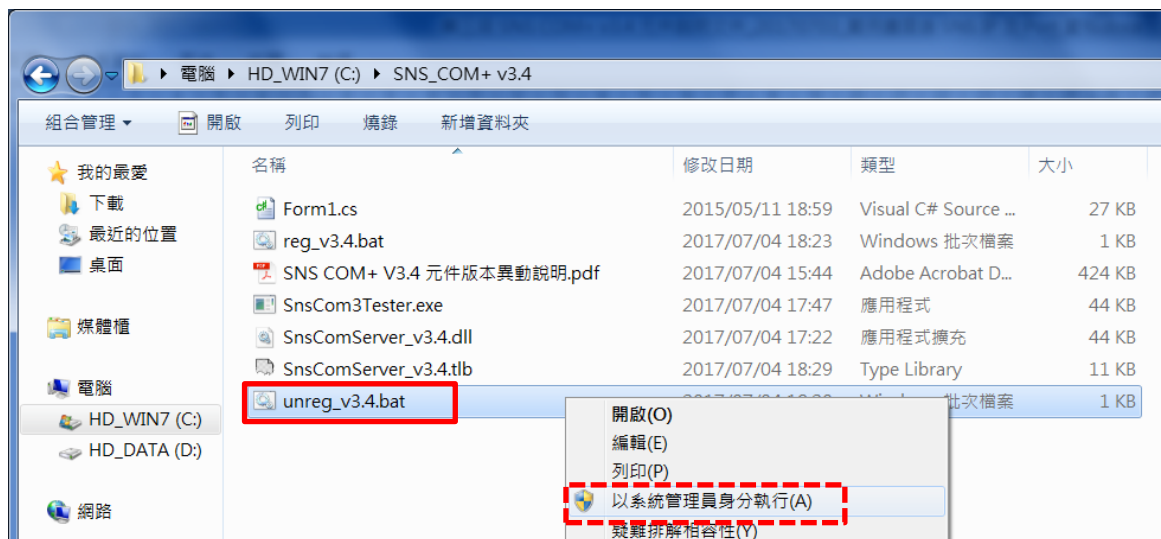
- 其 SNS COM+ DLL v3.3 為不含強式名稱版本者，使用 Visual Studio 2005 (或以上) 開發工具之平台目標可為 “Any CPU”：



- 而 SNS COM+ DLL v3.4 為含強式名稱版本者，使用 Visual Studio 2005 (或以上) 開發工具之相關應用程式部份，其平台目標須為 “x86”、且加入的參考除了包括 SnsComServer3 之外，另須包括 System.EnterpriseServices 方可正常使用：



當未來不用使用本 SNS COM+ 元件時、或有新版本要更新者，請先執行 unreg_v3.4.bat 及 iisreset 指令進行反註冊程序：



它會移除 SnsComServer_v3.4.tlb 檔如下：

```
C:\Windows\System32\cmd.exe

C:\SNS_COM+ v3.4>echo off

磁碟區 C 中的磁碟是 HD_WIN7
磁碟區序號: 9670-351E

C:\SNS_COM+ v3.4 的目錄
2017/07/04 下午 05:22          45,056 SnsComServer_v3.4.dll
2017/07/04 下午 06:29          10,508 SnsComServer_v3.4.tlb
                2 個檔案          55,564 位元組
                0 個目錄      18,175,090,688 位元組可用

*** Uninstall SnsComServer_v3.4.dll from .NET Framework 2.0
Microsoft (R) .NET Framework 服務安裝公用程式版本 2.0.50727.5483
Copyright (c) Microsoft Corporation. All rights reserved.

組件 C:\SNS_COM+ v3.4\SnsComServer_v3.4.dll 的解除安裝成功。

*** 請再下達 iisreset 使 Windows IIS 不再使用 SNS COM+ v3.4 DLL

磁碟區 C 中的磁碟是 HD_WIN7
磁碟區序號: 9670-351E

C:\SNS_COM+ v3.4 的目錄
2017/07/04 下午 05:22          45,056 SnsComServer_v3.4.dll
                1 個檔案          45,056 位元組
                0 個目錄      18,175,098,880 位元組可用

*** 若沒有看到 SnsComServer_v3.4.tlb 檔，表示完成 SNS COM+ v3.4 移除安裝作業！
請按任意鍵繼續 . . .
```

經完成上述反註冊程序，若已有裝 IIS 者，請續下達 iisreset 指令以重新啟動 IIS；若沒有安裝 IIS 者，建議請重新啟動電腦；以確認主機已無此 SNS COM+ v3 元件。

```
系統管理員: 命令提示字元

Microsoft Windows [版本 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>ver

Microsoft Windows [版本 6.1.7601]

C:\Users\Administrator>iisreset

正在嘗試停止...
網際網路服務已順利停止
正在嘗試啟動...
網際網路服務已順利重新啟動

C:\Users\Administrator>
```

3. 此版和舊版 SNS COM+ 元件差異

下表列出 SNS 簡訊服務最基本 API 用法差異，並請參考第 4 章之各 API 用法說明。其最大差異為新版 SNS COM+ v3 支援了變更密碼、長簡訊發送、及保持連線之功能：

差異項目	之前第二版舊版用法	此新第三版用法
元件模組 DLL	SnsComServer.dll	SnsComServer3.dll
使用元件	SNSCOMSERVER.SnsComObject	CHT.SNSCOMSERVER.SnsComObject3
適用網頁編碼	BIG-5	BIG-5 及 UTF-8
參數初始化	無	void InitRemoteServer(ip, port, account, passwd);
登入	int Login(ip, port, account, passwd); String RespMessage;	int Login(); 登入回應訊息：String GetRespDesc();
變更密碼	無	int EditPassword(newPasswd); 變更回應訊息：String GetRespDesc();
發送簡訊	int SubmitMessage(msisdn, text); String RespMessage;	int SubmitMessage(null, msisdn, text, 1440); ^(註1) 發訊成功取簡訊識別碼：String GetRespDesc(); 發訊失敗原因：String GetRespDesc();
發送長簡訊	無此功能 API	int SubmitLongMessage(null, msisdn, bUDHI, bEnglish, text, nSerno, nCurrSplit, nTotalSplit, 1440); 發訊成功取簡訊識別碼：String GetRespDesc(); 發訊失敗原因：String GetRespDesc();
查詢狀態	int QryMessageStatus(msisdn, msgid); String RespMessage;	int QueryMessageStatus(null, msisdn, msgid); 查詢回應訊息：String GetRespDesc();
接收簡訊	int GetMessage(); String FromMsisdn; String ToMsisdn; String RespMessage;	int GetMessage(); 有資料續取發訊方門號：String GetFromMsisdn(); 有資料續取受訊方門號：String GetToMsisdn(); 有資料續取收到簡訊內容：String GetRecvSMS(); 無資料之回應訊息：String GetRespDesc();
保持連線狀態	無	int EnquireLink(); 保持連線回應訊息：String GetRespDesc();
登出	void Logout();	void Logout();

註 1：相關 SNS COM+ v3 之 API 函式說明，請參考本文件第 4 章。

註 2：相關 SNS COM+ v3 之多種程式語言使用範例，請參考本文件第 5 章。

4. SNS COM+ v3 API 函式說明

此第三版本 SNS COM+ 元件提供項目（及 API 數）摘要如下：

- 初始化 (1)
- 取參數資料 (4)
- 特殊英文符號輔助處理 (2)

- 長簡訊輔助處理 (4)
- 中英文訊息及 byte 串列轉換處理 (8)
- 十六進位制 HEX 訊息字串轉換處理 (3)
- 執行結果描述 (1)
- 登入驗證 (1+1)：含判斷是否已連線
- 更改密碼 (1)
- 發訊 (9)：包括一般文字及二位元訊息發訊
- 查詢簡訊發送結果 (4)
- 接收簡訊 (1+3)：含接收成功要取回的發訊方號碼、受訊方號碼、及簡訊內容
- 保持連線 (1)
- 登出切線 (1)

一般呼叫 API 以執行 SNS 作業之回傳碼摘要如下：

- -2：表示用戶端給定 API 的參數錯誤
 - 請續呼叫 GetRespDesc() 取失敗原因
- -1：表示連線異常，用戶端需要重新連線
 - 請續呼叫 GetRespDesc() 取失敗原因
 - 若之前 InitRemoteServer() 指定多個 IP，則每次異常會自動切換伺服器網址
 - 請先 Logout() 後，再執行 Login() 重新連線
- 0：表示 SNS 作業成功
 - 可續呼叫 GetRespDesc() 取回應訊息
 - 發訊部份請用 GetRespDesc() 取簡訊識別碼
 - 查詢部份仍用 GetRespDesc() 取狀態結果訊息
 - 接收成功請用 GetFromMsisdn()、GetToMsisdn()、及 GetRecvSMS() 取簡訊資料
- >0：請參考 SNS 通訊協定回傳碼說明
 - 請續呼叫 GetRespDesc() 取回應訊息說明

可參考第 5.1 章節之視窗版測試程式範例，內含 SNS COM+ 提供的 API 用法說明，並提供客戶自行開發簡訊應用程式上之一般實務上參考建議。

注意本 SNS COM+ v3 元件之 API 不是 Thread-Safe，當使用者之多執行緒 (Multi-Thread)

應用程式利用本元件，須自行處理其 Lock 機制。

元件名稱：

✓ CHT.SNSCOMSERVER.SnsComObject3

初始化 (1)：

```
/**
 * 初始化 SNS 連線帳號資訊：
 *   SnsIPList:      若為 null，表系統內定伺服器網址，多個 IP 之間以逗號分隔；
 *   SnsPortList:    若為 null，表系統內定通訊埠。若多個 IP 有不同的通訊埠，
 *                   請用多個 Port 之間以逗號分隔，且和 SnsIPList 對應的 IP 相關。
 *   SnsAccount:     特碼帳號為必須輸入的資料，不可為 null。
 *   SnsPassword:    特碼密碼為必須輸入的資料，不可為 null。
 */
public void InitRemoteServer(String strSnsIPList, String strSnsPortList, String strSnsAccount,
                             String strSnsPassword);
```

取參數資料 (4)：

```
/**
 * 取得元件參考的 SNS 伺服器網址清單資訊
 */
public String GetSnsIPList();

/**
 * 取得元件參考的 SNS 伺服器通訊埠清單資訊
 */
public String GetSnsPortList();

/**
 * 目前使用的 SNS 伺服器位址
 */
public String GetServerIP();

/**
```

* 目前使用的 SNS 伺服器通訊埠

*/

```
public int GetServerPort();
```

特殊英文符號輔助處理 (2)：

/**

* 判斷是否純英數訊息，其輸入參數為一般文字訊息。

* 回傳 true 表為純英數訊息；false 表中文訊息。

*/

```
public bool IsPureEnglishMessage(String strMessage);
```

/**

* 判斷純英數訊息含多少個英文特殊符號 (部份 GSM 簡訊特殊碼 \n ^ { } \ [~] | 使用兩個 bytes 表示)。

* 註1：要使元件可正確送出含特殊符號之純英文簡訊，須將原英文簡訊長度扣除本 API

* 回傳的特殊符號數，再透過 SubmitMessage() 或其他 Submit 函式，受訊手機

* 才會正常顯示特殊符號字元。

* 註2：若傳送的英文訊息之長度，加上額外處理的 GSM 簡訊特殊碼數，超過 160 個 bytes，

* 則 Submit 函式不會進行轉換，受訊手機收到及顯示的特殊符號可能以三角型表示。

*/

```
public int CountPureEnglishSpecialSymbol(String strMessage);
```

長簡訊輔助處理 (4)：

/**

* 判斷長訊息可切割成多少則簡訊

* strLongMessage: 要判斷的完整訊息內容；

* bUDHI: 若為 true 表示為長簡訊 (手機整合多則簡訊至單通較長訊息)；

* 若為 false 表每則簡訊前面加上 n/m- 提示詞，表示共 m 通，此為

* 第 n 通；

* bEnglish: 是否純英文訊息，經由 IsPureEnglishMessage() 判斷。

* 注意：當 bUDHI=false 時，本元件最多支援切割至 9 則簡訊。

* 註：UDHI 為 User-Data-Header-Indicator 之簡稱，表示使用者資料標頭指示旗標，以便

* 加上額外訊息指示作何用途。

*/

```
public int GetTotalSplit(String strLongMessage, bool bUDHI, bool bEnglish);
```

```

/**
 * 取得指定則數之簡訊內容
 * strLongMessage: 要判別的訊息內容 ;
 * bUDHI:          若為 true 表示為長簡訊 (手機整合多則簡訊至單通較長訊息) ;
 *                若為 false 表每則簡訊前面加上 n/m- 提示詞，表示共 m 通, 此為
 *                第 n 通 ;
 * nCURR_SPLIT:    第幾則簡訊，須為 1-based 數值 ;
 * nTOTAL_SPLIT:   共多少則簡訊，此經由 GetTotalSplit() 得來 ;
 * bEnglish:       是否純英文訊息，經由 IsPureEnglishMessage() 判斷。
 */
public String GetSplitSMS(String strLongMessage, bool bUDHI, int nCURR_SPLIT, int
                        nTOTAL_SPLIT, bool bEnglish);

/**
 * 判斷十六進位制訊息可切割成多少則簡訊
 * nUDHI:          若為 1 (表示 true) 表示為長簡訊 (手機整合多則簡訊至單通較長訊息) ;
 *                若為 0 (表示 false) 表手機將收到多則簡訊 ;
 * strLongHexMessage: 呼叫端擬送出的十六進位制 HEX 字串訊息，其 HEX 字串可接受
 *                兩種格式，其中之一格式以空白分隔，如：12 34 56 78 9A；另一格
 *                式為不含空白，如：123456789A；
 * nLongHexMessageLen: 呼叫端指定的 HEX 字串之總長度。
 */
public int GetTotalSplitHexMessage(int nUDHI, String strLongHexMessage, int
                        nLongHexMessageLen);

/**
 * 取得指定則數之十六進位制 HEX 訊息內容。同一通長簡訊之多則簡訊需為同一個 nSerno
 * 序號值，不同長簡訊請給不同值。
 * nUDHI:          若為 1 (表示 true) 表示為長簡訊 (手機整合多則簡訊至單通較長訊息) ;
 *                若為 0 (表示 false) 表手機將收到多則簡訊 ;
 * strLongHexMessage: 呼叫端擬送出的十六進位制 HEX 字串訊息 ;
 * nLongHexMessageLen: 呼叫端指定的 HEX 字串之總長度 ;
 * nCURR_SPLIT:    第幾則簡訊，須為 1-based 數值 ;
 * nTOTAL_SPLIT:   共多少則簡訊，此經由 GetTotalSplitHexMessage() 得來。
 */
public String GetSplitHexMessage(int nUDHI, String strLongHexMessage, int
                        nLongHexMessageLen, int nSerno, int nCURR_SPLIT, int nTOTAL_SPLIT);

```

中英文訊息及 byte 串列轉換處理 (8)

```
/**
 * 判斷是否純英數訊息，其輸入參數為 byte 串列訊息。
 * 回傳 true 表為純英數訊息；false 表中文訊息。
 */
public bool IsPureEnglishByteArray(byte[] byteMessage);

/**
 * 將中文字串轉換成 UCS-16-BE 之 byte 串列
 */
public byte[] ToUcs16BeByteArray(String strMessage);

/**
 * 將英文字串轉換成 ASCII 之 byte 串列
 */
public byte[] ToAsciiByteArray(String strMessage);

/**
 * 將 byte 串列之每個 byte 轉換成兩個十六進位 HEX 字串，且以空白分隔
 */
public String GetHexStringFromByteArray(byte[] byteMessage);

/**
 * 從 HEX 字串轉換成對應的 byte 串列，其中 HEX 字串可接受兩種格式之一，其中之一
 * 格式以空白分隔，如：12 34 56 78 9A；另一格式為不含空白，如：123456789A。
 * 注意：若輸入的 strHexString 不是真正的 HEX 字串、或無資料，則回傳 null。
 */
public byte[] GetByteArrayFromHexString(String strHexString) {

/**
 * 將 UCS-16-BE 編碼 byte 串列轉換成中文訊息
 */
public String GetStringFromUcs16BeByteArray(byte[] byteMessage);

/**
 * 將 byte 串列轉換成英文訊息
 */
public String GetStringFromAsciiByteArray(byte[] byteMessage);
```

```

/*
 * 取得輸入的中文或英文訊息對應的 byte 串列之長度
 */
public int GetByteArrayLength(String strMessage);

```

十六進位制 HEX 訊息字串轉換處理 (3)

```

/**
 * 判斷是否純英數訊息，其輸入參數為 HEX 字串訊息。其中 HEX 字串可接受兩種格式，
 * 一為以空白分隔，如：12 34 56 78 9A；另一格式為不含空白，如：123456789A。
 * 回傳 true 表為純英數訊息；false 表中文訊息。
 */
public bool IsPureEnglishHexString(String strHexString);

/*
 * 將輸入的英文訊息轉換成 ASCII 編碼之對應的十六進位 HEX 字串
 */
public String GetAsciiHexString(String strMessage);

/*
 * 將輸入的中文訊息轉換成 UCS-16-BE 編碼之對應的十六進位 HEX 字串
 */
public String GetUcs16BeHexString(String strMessage);

```

執行結果描述 (1)：

```

/**
 * 取得 SNS 作業結果回應訊息
 */
public String GetRespDesc();

```

登入驗證 (1+1)：

```

/**
 * 是否已登入連線？
 */
public bool IsBind() {

```

```

/**
 * LOGIN: 連線及登入伺服器。
 * -2 表不正確的 IP 及 Port 筆數資訊 (當皆多個時)、初始化未指定帳號或密碼等;
 * -1 表無法連線;
 * 0 表通過登入驗證並連線成功
 * 1 表不正確的帳號/密碼; 其他值: 參考 SNS Protocol 文件
 * 註: 若連線但未通過登入驗證, 則自動切離 Socket 連線。
 */
public int Login();

```

更改密碼 (1):

```

/**
 * EDIT_PASSWD: 變更密碼
 * -2 表須指定新密碼資料、長度須為六~八碼、密碼須為英數混合字元;
 * -1 表連線異常;
 * 0 表變更成功且元件已參考此新密碼資訊; 其他值: 參考 SNS Protocol 文件
 */
public int EditPassword(String strNewPassword);

```

發訊 (9): (註: 其中長簡訊部份, 簡訊特碼帳號須先擁有其發送權限)

```

/**
 * SERV_SEND: 一般行動電話號碼發訊至一般行動電話號碼。
 * -2: 參數格式錯誤
 *
 * strFromMsisdn: 發訊方須為 null (表簡訊特碼對應的行動電話號碼)、或 09 開頭
 *
 *                  十碼之發訊方國內行動電話號碼;
 *
 * strToMsisdn: 受訊方須為 09 開頭十碼之受訊方國內行動電話號碼;
 *
 * strSMS: 須有文字內容資料;
 *
 * nExpiredMinutes: 重送期限範圍為 0 ~ 1440 分鐘 (0 亦表示最長一天 1440
 *
 *                  分鐘), 注意此數值為估計值, 實際可能會有一兩小時之誤差。
 *
 * -1: 連線異常;
 * 0: 傳送成功 (續呼叫 GetRespDesc() 取得訊息代碼);
 * 其他值: 參考 SNS Protocol 文件
 */
public int SubmitMessage(String strFromMsisdn, String strToMsisdn, String strSMS, int
nExpiredMinutes);

```



```

/**
 * SERV_SEND_WITH_SUBNO: emome 號碼發訊至一般行動電話號碼。
 * -2: 參數格式錯誤
 *     strFromSubno:    發訊方須為九碼、或含第十碼檢查碼之發訊方 emome 號碼；
 *     strToMsisdn:     受訊方須為 09 開頭十碼之受訊方國內行動電話號碼；
 *     strSMS:          須有文字內容資料；
 *     nExpiredMinutes: 重送期限範圍為 0 ~ 1440 分鐘 (0 亦表示最長一天 1440
 *                      分鐘)，注意此數值為估計值，實際可能會有一兩小時之誤差。
 * -1: 連線異常;
 * 0: 傳送成功 (續呼叫 GetRespDesc() 取得訊息代碼);
 * 其他值: 參考 SNS Protocol 文件
 */
public int SubmitMessageWithSubno(String strFromSubno, String strToMsisdn, String strSMS,
                                int nExpiredMinutes);

```

```

/**
 * SERV_SEND_WITH_RECV_SUBNO: 一般行動電話號碼發訊至 emome 號碼。
 * -2: 參數格式錯誤
 *     strFromMsisdn:    發訊方須為 null (表簡訊特碼對應的行動電話號碼)、或 09
 *                      開頭十碼之發訊方國內行動電話號碼；
 *     strToSubno:       受訊方須為九碼、或含第十碼檢查碼之受訊方 emome 號碼；
 *     strSMS:          須有文字內容資料；
 *     nExpiredMinutes: 重送期限範圍為 0 ~ 1440 分鐘 (0 亦表示最長一天 1440
 *                      分鐘)，注意此數值為估計值，實際可能會有一兩小時之誤差。
 * -1: 連線異常;
 * 0: 傳送成功 (續呼叫 GetRespDesc() 取得訊息代碼);
 * 其他值: 參考 SNS Protocol 文件
 */
public int SubmitMessageWithRecvSubno(String strFromMsisdn, String strToSubno, String
                                     strSMS, int nExpiredMinutes);

```

```

/**
 * SERV_SEND_WITH_ALL_SUBNO: emome 號碼發訊至 emome 號碼。
 * -2: 參數格式錯誤
 *     strFromSubno:    發訊方須為九碼、或含第十碼檢查碼之發訊方 emome 號碼；
 *     strToSubno:       受訊方須為九碼、或含第十碼檢查碼之受訊方 emome 號碼；
 *     strSMS:          須有文字內容資料；
 *     nExpiredMinutes: 重送期限範圍為 0 ~ 1440 分鐘 (0 亦表示最長一天 1440

```

```

*          分鐘)・注意此數值為估計值・實際可能會有一兩小時之誤差。
* -1: 連線異常;
* 0: 傳送成功 (續呼叫 GetRespDesc() 取得訊息代碼);
* 其他值: 參考 SNS Protocol 文件
*/
public int SubmitMessageWithAllSubno(String strFromSubno, String strToSubno, String
    strSMS, int nExpiredMinutes);

/**
* SERV_SEND: 一般行動電話號碼發訊至一般行動電話號碼。其中 UDHI=true 表示為
* 長簡訊 (手機整合多則簡訊至單通較長訊息); 若 UDHI=false 表每則簡訊前面
* 加上 n/m- 提示詞 (表示共 m 通, 此為第 n 通)。同一通長簡訊之多則簡訊需為
* 同一個序號 nSerno・不同長簡訊請給不同值。
* -2: 參數格式錯誤
*     strFromMsisdn: 發訊方須為 null (表簡訊特碼對應的行動電話號碼)、或 09 開頭
*                     十碼之發訊方國內行動電話號碼;
*     strToMsisdn:   受訊方須為 09 開頭十碼之受訊方國內行動電話號碼;
*     strSMS:        須有文字內容資料;
*     nCurrSplit:    第幾則・須為 1-based 序號;
*     nTotalSplit:   共多少則・須大於或等於 nCurrSplit 數值;
*     nExpiredMinutes: 重送期限範圍為 0 ~ 1440 分鐘 (0 亦表示最長一天 1440
*                     分鐘)・注意此數值為估計值・實際可能會有一兩小時之誤差。
* -1: 連線異常;
* 0: 傳送成功 (續呼叫 GetRespDesc() 取得訊息代碼);
* 其他值: 參考 SNS Protocol 文件
*/
public int SubmitLongMessage(String strFromMsisdn, String strToMsisdn, bool bUDHI, bool
    bEnglish, String strSMS, int nSerno, int nCurrSplit, int nTotalSplit, int
    nExpiredMinutes);

/**
* SERV_SEND_WITH_SUBNO: emome 號碼發送長簡訊至一般行動電話號碼。其中
* UDHI=true 表示為長簡訊 (手機整合多則簡訊至單通較長訊息); 若 UDHI=false 表
* 每則簡訊前面加上 n/m- 提示詞 (表示共 m 通, 此為第 n 通)。同一通長簡訊之多則
* 簡訊需為同一個序號 nSerno・不同長簡訊請給不同值。
* -2: 參數格式錯誤
*     strFromSubno: 發訊方須為九碼、或含第十碼檢查碼之發訊方 emome 號碼;
*     strToMsisdn:  受訊方須為 09 開頭十碼之受訊方國內行動電話號碼;
*     strSMS:       須有文字內容資料;

```

```

*      nCurrSplit:      第幾則，須為 1-based 序號；
*      nTotalSplit:     共多少則，須大於或等於 nCurrSplit 數值；
*      nExpiredMinutes: 重送期限範圍為 0 ~ 1440 分鐘 (0 亦表示最長一天 1440
*                      分鐘)，注意此數值為估計值，實際可能會有一兩小時之誤差。
* -1: 連線異常;
* 0: 傳送成功 (續呼叫 GetRespDesc() 取得訊息代碼);
* 其他值: 參考 SNS Protocol 文件
*/

public int SubmitLongMessageWithSubno(String strFromSubno, String strToMsisdn, bool
                                     bUDHI, bool bEnglish, String strSMS, int nSerno, int nCurrSplit, int
                                     nTotalSplit, int nExpiredMinutes);

/**
* SERV_SEND_WITH_RECV_SUBNO: 一般行動電話號碼發訊至 emome 號碼。其中
* UDHI=true 表示為長簡訊 (手機整合多則簡訊至單通較長訊息)；若 UDHI=false 表
* 每則簡訊前面加上 n/m- 提示詞 (表示共 m 通, 此為第 n 通)。同一通長簡訊之多則
* 簡訊需為同一個序號 nSerno，不同長簡訊請給不同值。
* -2: 參數格式錯誤
*      strFromMsisdn:    發訊方須為 null (表簡訊特碼對應的行動電話號碼)、或 09 開頭
*                      十碼之發訊方國內行動電話號碼；
*      strToSubno:       受訊方須為九碼、或含第十碼檢查碼之受訊方 emome 號碼；
*      strSMS:           須有文字內容資料；
*      nCurrSplit:       第幾則，須為 1-based 序號；
*      nTotalSplit:      共多少則，須大於或等於 nCurrSplit 數值；
*      nExpiredMinutes: 重送期限範圍為 0 ~ 1440 分鐘 (0 亦表示最長一天 1440
*                      分鐘)，注意此數值為估計值，實際可能會有一兩小時之誤差。
* -1: 連線異常;
* 0: 傳送成功(續呼叫GetRespDesc() 取得訊息代碼);
* 其他值: 參考SNS Protocol 文件
*/

public int SubmitLongMessageWithRecvSubno(String strFromMsisdn, String strToSubno,
                                           bool bUDHI, bool bEnglish, String strSMS, int nSerno, int nCurrSplit, int
                                           nTotalSplit, int nExpiredMinutes);

/**
* SERV_SEND_WITH_ALL_SUBNO: emome 號碼發送長簡訊至 emome 號碼。其中
* UDHI=true 表示為長簡訊 (手機整合多則簡訊至單通較長訊息)；若 UDHI=false 表
* 每則簡訊前面加上 n/m- 提示詞(表示共 m 通, 此為第 n 通)。同一通長簡訊之多則
* 簡訊需為同一個序號 nSerno，不同長簡訊請給不同值。

```

```

* -2: 參數格式錯誤
*   strFromSubno: 發訊方須為九碼、或含第十碼檢查碼之發訊方 emome 號碼；
*   strToSubno:  受訊方須為九碼、或含第十碼檢查碼之受訊方 emome 號碼；
*   strSMS:      須有文字內容資料；
*   nCurrSplit:  第幾則，須為 1-based 序號；
*   nTotalSplit: 共多少則，須大於或等於 nCurrSplit 數值；
*   nExpiredMinutes: 重送期限範圍為 0 ~ 1440 分鐘 (0 亦表示最長一天 1440
*                   分鐘)，注意此數值為估計值，實際可能會有一兩小時之誤差。
* -1: 連線異常;
*   0: 傳送成功 (續呼叫 GetRespDesc() 取得訊息代碼);
*   其他值: 參考 SNS Protocol 文件
*/

```

```

public int SubmitLongMessageWithAllSubno(String strFromSubno, String strToSubno, bool
                                         bUDHI, bool bEnglish, String strSMS, int nSerno, int nCurrSplit, int
                                         nTotalSplit, int nExpiredMinutes);

```

```

/**

```

```

* 傳送十六進位制之二位元圖鈴下載訊息簡訊 (僅適用支援 SmartMessaging 規格之
*   Nokia 手機)、或自行傳送十六進位制文字編碼訊息 (適用所有手機)。
* -2: 參數格式錯誤
*   strFromMsisdn: 發訊方須為 null (表簡訊特碼對應的行動電話號碼)、或 09 開頭
*                   十碼之發訊方國內行動電話號碼；
*   strToMsisdn:   受訊方須為 09 開頭十碼之受訊方國內行動電話號碼；
*   nUDHI:         其值須為 0 (表示 false)、或 1 (表示 true)；
*   nCoding:       其值範圍須 0 ~ 255 之間，其中常用的值包括：
*                   0 (即 0x00) 表示 strHexString 除了 ASCII 編碼 HEX 字串之
*                   外，亦包含特殊碼之 HEX 字串；
*                   1 (即 0x01) 表示 strHexString 為 ASCII 編碼字串；
*                   8 (即 0x08) 表示 strHexString 為 UCS-16-BE 編碼字串；
*                   248 (即 0xF8) 表示 strHexString 為傳送 SmartMessaging
*                   規格之 HEX 字串；
*   strHexString   須有內容且不超過 160 個位元組資料，每個資料以兩個 HEX
*                   表示。每個 HEX 範圍為 0~9、A~F、或 a~f。每個位元組資料
*                   之間可用空白分隔、或連續訊息。例如：12 34 56 78 9A 或
*                   123456789A 等兩種格式之一；
*   nHexStringLen: 其長度不可超過 strHexString 之最大長度。正常情況為相
*                   同於 strHexString 之長度，例如：12 34 56 78 9A 及
*                   123456789A 之長度分別為 14 及 10；

```

```

*      nExpiredMinutes: 重送期限範圍為 0 ~ 1440 分鐘 (0 亦表示最長一天 1440
*                      分鐘) , 注意此數值為估計值 , 實際可能會有一兩小時之誤差。
* -1: 連線異常;
* 0: 傳送成功 (續呼叫 GetRespDesc() 取得訊息代碼);
* 其他值: 參考 SNS Protocol 文件
* 註 1: 要傳送 binary 訊息, 請將 bUDHI 設為 true, 並將 byteCoding=0xF5(binary) ;
* 註 2: 若要傳送編碼文字, 請依單則訊息或長簡訊需求設定 bUDHI=false 或 true。
*      一般用法為 byteCoding=0x08(Unicode, 文字可用 UCS16-BE 編碼之); 或
*      0x01 (ASCII), 皆表示不含中文之英文及數字訊息。
*/

```

```

public int SubmitBinaryMessage(String strFromMsisdn, String strToMsisdn, int nUDHI, int
                                nCoding, String strHexMessage, int nHexMessageLen, int
                                nExpiredMinutes);

```

查詢簡訊發送結果 (4) :

```

/**
* SERV_QUERY: 查詢行動電話號碼發訊 (表簡訊特碼對應的行動電話號碼) 發訊至行動
* 電話號碼之最終狀態
* -2: 參數格式錯誤
*      strFromMsisdn: 須為 null (表簡訊特碼對應的行動電話號碼)、或 09 開頭十碼之
*                      發訊方國內行動電話號碼 ;
*      strToMsisdn:    須為 09 開頭十碼之受訊方國內行動電話號碼 ;
*      strMsgID:       須有訊息識別碼資料。
* -1: 無法連線;
* 其他值: 參考 SNS Protocol 文件 (0: 成功送出; 1: 傳送中; 2: 錯誤資料;
*      3: 傳送失敗; 4: 暫時無法查詢; ...)
*/

```

```

public int QueryMessageStatus(String strFromMsisdn, String strToMsisdn, String strMsgID);

```

```

/**
* SERV_QUERY_WITH_SUBNO: 查詢 Emome ID 發訊至行動電話號碼之最終狀態
* -2: 參數格式錯誤
*      strFromSubno: 須為九碼、或含第十碼檢查碼之發訊方 emome 號碼 ;
*      strToMsisdn: 須為 09 開頭十碼之受訊方國內行動電話號碼 ;
*      strMsgID:    須有訊息識別碼資料。
* -1: 無法連線;
* 其他值: 參考 SNS Protocol 文件 (0: 成功送出; 1: 傳送中; 2: 錯誤資料;

```

```

*          3: 傳送失敗; 4: 暫時無法查詢; ...)
*/
public int QueryMessageStatusWithSubno(String strFromSubno, String strToMsisdn, String
    strMsgID);

/**
 * SERV_QUERY_WITH_RECV_SUBNO: 查詢行動電話號碼 含特碼) 發訊至 Emome ID
 * 之最終狀態
 * -2: 參數格式錯誤
 * strFromMsisdn: 須為 null (表簡訊特碼對應的行動電話號碼)、或 09 開頭十碼之
 * 發訊方國內行動電話號碼 ;
 * strToSubno: 須為九碼、或含第十碼檢查碼之受訊方 emome 號碼 ;
 * strMsgID: 須有訊息識別碼資料。
 * -1: 無法連線;
 * 其他值: 參考 SNS Protocol 文件 (0: 成功送出; 1: 傳送中; 2: 錯誤資料;
 *          3: 傳送失敗; 4: 暫時無法查詢; ...)
 */
public int QueryMessageStatusWithRecvSubno(String strFromMsisdn, String strToSubno,
    String strMsgID);

/**
 * SERV_QUERY_WITH_ALL_SUBNO: 查詢 Emome ID 發訊至 Emome ID 之最終狀態
 * -2: 參數格式錯誤
 * strFromSubno: 須為九碼、或含第十碼檢查碼之發訊方 emome 號碼 ;
 * strToSubno: 須為九碼、或含第十碼檢查碼之受訊方 emome 號碼 ;
 * strMsgID: 須有訊息識別碼資料。
 * -1: 無法連線;
 * 其他值: 參考 SNS Protocol 文件 (0: 成功送出; 1: 傳送中; 2: 錯誤資料;
 *          3: 傳送失敗; 4: 暫時無法查詢; ...)
 */
public int QueryMessageStatusWithAllSubno(String strFromSubno, String strToSubno, String
    strMsgID);

```

接收簡訊 (1+3) :

```

/**
 * SERV_GET: 接收手機端送來的簡訊
 * -2: 訊息轉碼失敗 ;

```

```

* -1: 無法連線
* 0: 接收到上傳簡訊 (續呼叫 GetFromMsisdn()、GetToMsisdn()、及 GetRecvSMS());
* 其他值: 參考 SNS Protocol 文件
*/
public int GetMessage();

/**
* 取得 SNS 接收簡訊之發訊方號碼
*/
public String GetFromMsisdn();

/**
* 取得 SNS 接收簡訊之受訊方號碼
*/
public String GetToMsisdn();

/**
* 取得 SNS 接收簡訊之文字內容
*/
public String GetRecvSMS();

```

保持連線 (1)：

```

/**
* LINK: 與伺服器保持連線狀態，以免因閒置過久 (約三分鐘) 而被切線。
* -1: 無法連線 SNS ;
* 0: 成功;
* 其他值: 參考 SNS Protocol 文件
*/
public int EnquireLink();

```

登出切線 (1)：

```

/**
* 切離與遠端的 SNS 伺服器連線
*/
public void Logout();

```

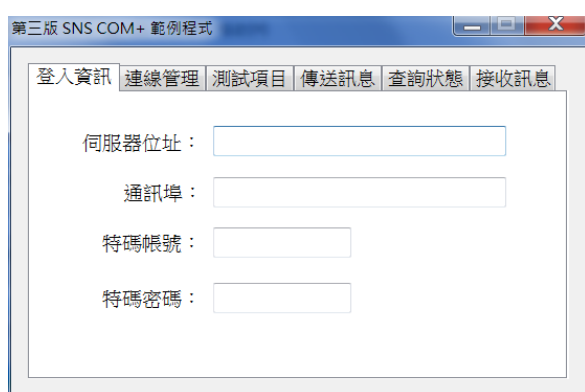

5. 程式碼範例

5.1 C#.NET 視窗版單門號範例程式

本範例相關程式碼可參考“CSharp.NET 單門號範例.zip”壓縮檔。

使用者可執行本元件附上的 SnsCom3Tester.exe 程式範例，以便使用同個目錄之 SNS COM+ 第三版元件之大部份 API 功能，包括：

- 登入測試 (請先輸入您申請的 SNS 帳號及密碼)
- 連線管理測試 (包括：登入連線、保持連線、切離連線、及變更密碼)
- 測試項目 (包括 SNS 可服務的多種對象方式，一般最常用為內定第一個測試項目，即：特碼發訊至一般 09 開頭之受訊手機號碼)
- 傳送訊息測試 (請輸入要受訊的受訊端號碼，可發送包括：單則簡訊及包含兩則以上長度之長簡訊等)
- 查詢狀態測試 (即可查詢已送出簡訊之受訊號碼端結果狀態)
- 接收訊息 (即接收手機端手機使用者上傳的訊息，可為單則簡訊及包含兩則以上長度之長簡訊)



第三版 SNS COM+ 範例程式

登入資訊 | 連線管理 | 測試項目 | 傳送訊息 | 查詢狀態 | 接收訊息

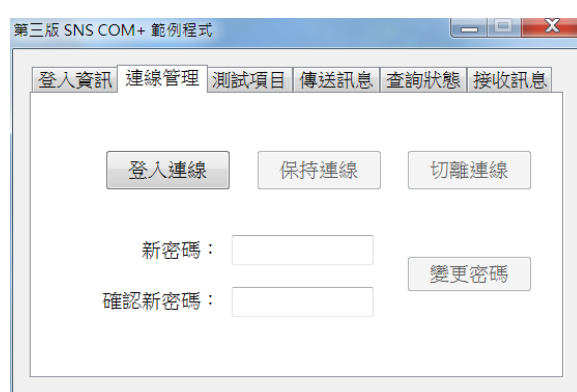
伺服器位址:

通訊埠:

特碼帳號:

特碼密碼:

(連線參數初始化，多個 IP 以，分隔)
輸入已申請的 SNS 特碼帳號及密碼



第三版 SNS COM+ 範例程式

登入資訊 | 連線管理 | 測試項目 | 傳送訊息 | 查詢狀態 | 接收訊息

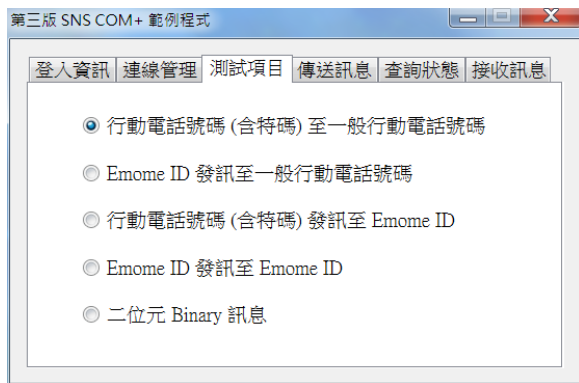
登入連線 保持連線 切離連線

新密碼:

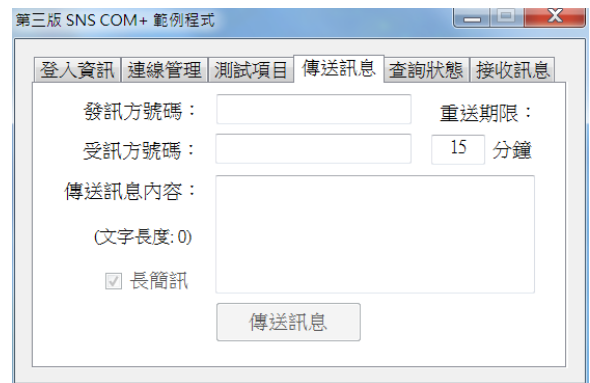
確認新密碼:

變更密碼

(登入連線、保持連線、切離連線、及變更密碼)



(提供前四種一般文字訊息、及二位元 Binary 訊息
發訊及查詢功能測試)

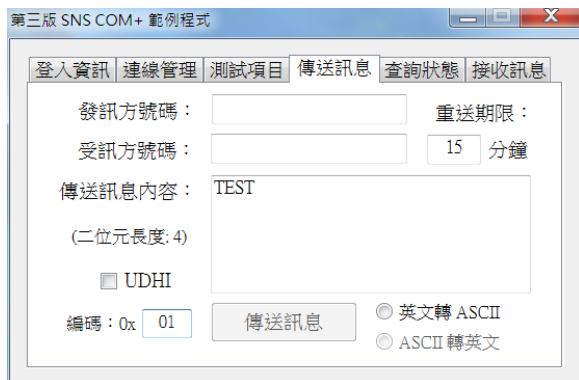


(當選擇前四種其中一個文字測試項目)

直接輸入一般文字訊息

→ 若是切割至多則簡訊，可選擇分多個單則簡訊
或整合至單通長簡訊

→ 送出簡訊/長簡訊至受訊一般手機號碼或 Emome ID
(API 自動處理編碼及訊息轉換等細節)



(當選擇第五種 Binary 訊息測試項目)

直接英文輸入訊息 → 點選英文轉 ASCII

→ 間接得 HEX 字串

→ 送出編碼 0x00 (Default) 或 0x01 (ASCII) 之
Binary 訊息至一般手機號碼

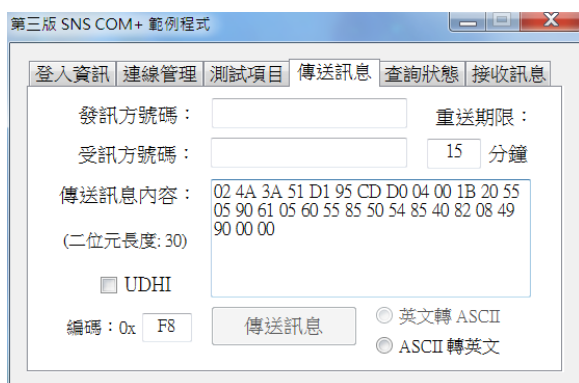


(當選擇第五種 Binary 訊息測試項目)

直接輸入中文訊息 → 點選中文轉 UCS-16-BE

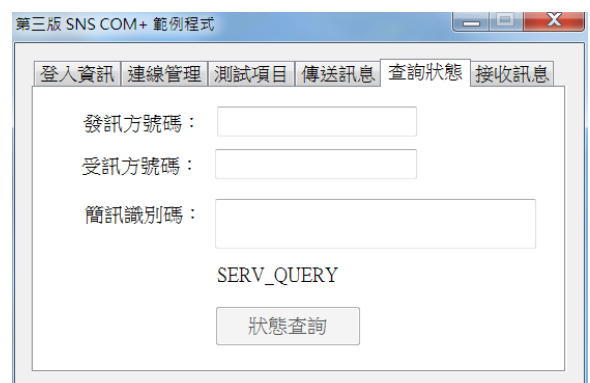
→ 間接得 HEX 字串

→ 送出編碼 0x08 (Unicode) 之 Binary 訊息
至一般手機號碼



(當選擇第五種 Binary 訊息測試項目)

直接輸入 HEX 字串



(查詢已送出簡訊之最終狀態測試)

→ 送出編碼 0xF8 之 Binary 圖鈴下載訊息範例
(限發訊至有支援 SmartMessaging 規格之 Nokia 手機
可正常收訊，此功能限發一般手機號碼)

本單門號視窗程式實作範例可參考 C# 完整程式碼如下 (可參考其壓縮檔並解壓之，以便使用其測試專案檔範例)，可參考後述相關程式片段、及一般實務用法建議說明：

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Globalization;

namespace SnsCom3Tester
{
    public partial class Form1 : Form
    {
        //宣告一物件變數為 SnsComObject3 元件
        private CHT.SNSCOMSERVER.SnsComObject3 m_sns = new CHT.SNSCOMSERVER.SnsComObject3();

        //提供五種發訊及查詢功能測試
        private static readonly int TEST_DEFAULT = 0;
        private static readonly int TEST_RECV_SUBNO = 1;
        private static readonly int TEST_SUBNO = 2;
        private static readonly int TEST_ALL_SUBNO = 3;
        private static readonly int TEST_BINARY = 4;

        //選用的功能測試項目
        private int nType = TEST_DEFAULT;
        private String strSendType = "SERV_SEND";

        //發送長簡訊之參考序號
        private int nSerno = new Random().Next(0x100); //範圍: 0x00 ~ 0xFF

        public Form1()
        {
            InitializeComponent();
        }

        private void CheckButtonState()
        {
            bool b = m_sns.IsBind();

            LoginButton.Enabled = !b;
            LinkButton.Enabled = b;
            LogoutButton.Enabled = b;
            EditPasswdButton.Enabled = b;
            SubmitButton.Enabled = b;
            QueryButton.Enabled = b;
            GetButton.Enabled = b;
        }
    }
}
```

```

private void Form1_Load(object sender, EventArgs e)
{
    SubmitMessage_TextChanged(sender, e);

    CheckButtonState();
    TestDefault();
}

//當按下「登入連線」鍵
private void LoginButton_Click(object sender, EventArgs e)
{
    //連線參數初始化
    m_sns.InitRemoteServer(SnsIPList.Text, SnsPortList.Text, SnsAccount.Text, SnsPassword.Text);

    //登入 SNS 伺服器
    int nSnsCode = m_sns.Login();

    switch (nSnsCode)
    {
        case -2:
            MessageBox.Show "[" + nSnsCode + "]" + m_sns.GetRespDesc(), "參數錯誤");
            break;
        case -1:
            MessageBox.Show "[" + nSnsCode + "]" + m_sns.GetRespDesc(), "無法連線 SNS");
            break;
        case 0:
            MessageBox.Show "[" + nSnsCode + "]" + m_sns.GetRespDesc(), "登入驗證成功");
            break;
        case 1:
            MessageBox.Show "[" + nSnsCode + "]" + m_sns.GetRespDesc(), "帳號/密碼輸入錯誤");
            break;
        default:
            MessageBox.Show "[" + nSnsCode + "]" + m_sns.GetRespDesc(), "請參考 SNS Protocol 文件");
            break;
    }

    CheckButtonState();
}

//當按下「變更密碼」鍵
private void EditPasswdButton_Click(object sender, EventArgs e)
{
    if (!NewPassword.Text.Equals(ConfNewPassword.Text)) {
        MessageBox.Show("兩個密碼內容不一致，請重新輸入！", "錯誤");
        return;
    }

    //變更密碼
    int nSnsCode = m_sns.EditPassword(NewPassword.Text);

    switch (nSnsCode) {
        case -2:
            MessageBox.Show "[" + nSnsCode + "]" + m_sns.GetRespDesc(), "參數錯誤");
            break;
        case -1:
            MessageBox.Show "[" + nSnsCode + "]" + m_sns.GetRespDesc(), "無法連線 SNS");
            break;
        case 0:
            MessageBox.Show "[" + nSnsCode + "]" + m_sns.GetRespDesc() + "\n\n下次請用此新密碼登入！",
                "變更密碼成功");
    }
}

```

```

        break;
    default:
        MessageBox.Show("[ " + nSnsCode + " ] " + m_sns.GetRespDesc(), "請參考 SNS Protocol 文件");
        break;
    }

    CheckButtonState();
}

private void TestDefault()
{
    nType = TEST_DEFAULT;
    strSendType = "SERV_SEND";
    LabelQuery.Text = "SERV_QUERY";

    radioButton5.Visible = false;
    radioButton6.Visible = false;
    LabelCoding.Visible = false;
    Coding.Visible = false;
    udhi.Text = "長簡訊";
    udhi.Checked = true;

    SubmitMessage_TextChanged(null, null);
}

//當點選第一個測試項目：行動電話號碼（含特碼）至一般行動電話號碼
private void radioButton0_Click(object sender, EventArgs e)
{
    MessageBox.Show("傳送：SERV_SEND 及\n\n查詢：SERV_QUERY", "測試項目");

    TestDefault();
}

//當點選第二個測試項目：Emome ID 發訊至一般行動電話號碼
private void radioButton1_Click(object sender, EventArgs e)
{
    MessageBox.Show("傳送：SERV_SEND_WITH_SUBNO 及\n\n查詢：SERV_QUERY_WITH_SUBNO", "測試項目");

    nType = TEST_SUBNO;
    strSendType = "SERV_SEND_WITH_SUBNO";
    LabelQuery.Text = "SERV_QUERY_WITH_SUBNO";

    radioButton5.Visible = false;
    radioButton6.Visible = false;
    LabelCoding.Visible = false;
    Coding.Visible = false;
    udhi.Text = "長簡訊";
    udhi.Checked = true;

    SubmitMessage_TextChanged(sender, e);
}

//當點選第三個測試項目：行動電話號碼（含特碼）發訊至 Emome ID
private void radioButton2_Click(object sender, EventArgs e)
{
    MessageBox.Show("傳送：SERV_SEND_WITH_RECV_SUBNO 及\n\n查詢：SERV_QUERY_WITH_RECV_SUBNO",
        "測試項目");

    nType = TEST_RECV_SUBNO;
    strSendType = "SERV_SEND_WITH_RECV_SUBNO";
    LabelQuery.Text = "SERV_QUERY_WITH_RECV_SUBNO";
}

```

```

radioButton5.Visible = false;
radioButton6.Visible = false;
LabelCoding.Visible = false;
Coding.Visible = false;
udhi.Text = "長簡訊";
udhi.Checked = true;

SubmitMessage_TextChanged(sender, e);
}

//當點選第四個測試項目：Emome ID 發訊至 Emome ID
private void radioButton3_Click(object sender, EventArgs e)
{
    MessageBox.Show("傳送：SERV_SEND_WITH_ALL_SUBNO 及\n\n查詢：SERV_QUERY_WITH_ALL_SUBNO", "測試項目");

    nType = TEST_ALL_SUBNO;
    strSendType = "SERV_SEND_WITH_ALL_SUBNO";
    LabelQuery.Text = "SERV_QUERY_WITH_ALL_SUBNO";

    radioButton5.Visible = false;
    radioButton6.Visible = false;
    LabelCoding.Visible = false;
    Coding.Visible = false;
    udhi.Text = "長簡訊";
    udhi.Checked = true;

    SubmitMessage_TextChanged(sender, e);
}

//當點選第五個測試項目：二位元 Binary 訊息
private void radioButton4_Click(object sender, EventArgs e)
{
    MessageBox.Show("傳送：SERV_SEND_WITH_BINARY 及\n\n查詢：SERV_QUERY", "測試項目");

    nType = TEST_BINARY;
    strSendType = "SERV_SEND_WITH_BINARY";
    LabelQuery.Text = "SERV_QUERY";

    radioButton5.Visible = true;
    radioButton6.Visible = true;
    LabelCoding.Visible = true;
    Coding.Visible = true;
    udhi.Text = "UDHI";
    udhi.Enabled = true;
    udhi.Checked = false;

    SubmitMessage_TextChanged(sender, e);
}

//當點選傳送訊息：「中文轉 Unicode」或「英文轉 ASCII」之輔助轉換
private void radioButton5_Click(object sender, EventArgs e)
{
    byte[] byteMessage = null;

    if(m_sns.IsPureEnglishMessage(SubmitMessage.Text)) {
        byteMessage = m_sns.ToAsciiByteArray(SubmitMessage.Text);
        Coding.Text = "01";
    } else {
        byteMessage = m_sns.ToUcs16BeByteArray(SubmitMessage.Text);
        Coding.Text = "08";
    }
}

```

```

    }

    int nLen = (byteMessage != null) ? byteMessage.Length : 0;

    SubmitMessage.Text = m_sns.GetHexStringFromByteArray(byteMessage);

    //是否超過一則訊息之最大長度
    udhi.Checked = (nLen > 160) ? true : false;

    radioButton5.Enabled = false;
    radioButton6.Enabled = true;
}

//當點選傳送訊息：「Unicode 轉中文」或「ASCII 轉英文」之輔助轉換
private void radioButton6_Click(object sender, EventArgs e) {
    byte[] byteMessage = m_sns.GetByteArrayFromHexString(SubmitMessage.Text);

    if(m_sns.IsPureEnglishByteArray(byteMessage)) {
        SubmitMessage.Text = m_sns.GetStringFromAsciiByteArray(byteMessage);
    } else {
        SubmitMessage.Text = m_sns.GetStringFromUcs16BeByteArray(byteMessage);
    }

    radioButton6.Enabled = false;
    radioButton5.Enabled = true;
}

//輸入傳送訊息即時判斷作業
private void SubmitMessage_TextChanged(object sender, EventArgs e) {
    //若是第五個測試項目：二位元 Binary 訊息
    if(nType == TEST_BINARY) {
        byte[] byteMessage = null;
        bool bEnglish = true;

        //當之前已點「英文轉 ASCII」、或「中文轉 UCS-16-BE」；或為以空白分隔之 Hex 字串
        if(radioButton5.Checked || (SubmitMessage.Text.Length > 2)) {
            byteMessage = m_sns.GetByteArrayFromHexString(SubmitMessage.Text); //輸入訊息可能為 Hex 字串
        }

        if(byteMessage == null) { //表示 SubmitMessage 不是真正的 Hex 字串
            if(m_sns.IsPureEnglishMessage(SubmitMessage.Text)) {
                byteMessage = m_sns.ToAsciiByteArray(SubmitMessage.Text); //輸入訊息為英文
                bEnglish = true;
            } else {
                byteMessage = m_sns.ToUcs16BeByteArray(SubmitMessage.Text); //輸入訊息為中文
                bEnglish = false;
            }

            if(bEnglish) {
                radioButton5.Text = "英文轉 ASCII";
                radioButton6.Text = "ASCII 轉英文";
            } else {
                radioButton5.Text = "中文轉 UCS-16-BE";
                radioButton6.Text = "UCS-16-BE 轉中文";
            }

            radioButton5.Enabled = true;
            radioButton6.Enabled = false;
            radioButton5.Checked = false;
        } else {
            bEnglish = m_sns.IsPureEnglishByteArray(byteMessage); //自動判斷輸入為中文或英文訊息

```



```

        radioButton5.Enabled = false;
        radioButton6.Enabled = true;
    }

    LabelLength.Text = "(二位元長度: " + ((byteMessage != null) ? byteMessage.Length : 0) + ")";

    udhi.Enabled = true;

} else { //否則為前四個一般文字訊息測試項目
    int nLen = SubmitMessage.Text.Length;
    bool bEnglish = m_sns.IsPureEnglishMessage(SubmitMessage.Text); //自動判斷輸入為中文或英文訊息
    int nTotalSplit = m_sns.GetTotalSplit(SubmitMessage.Text, udhi.Checked, bEnglish); //可切割多少則簡訊

    LabelLength.Text = "(文字長度: " + nLen + ")";

    udhi.Enabled = (nTotalSplit > 1) ? true : false; //若超過一則簡訊，則啟用 UDHI 選項 (多則整合至單一長簡訊)
}
}

//當按下「傳送訊息」鍵以發送單則或多則簡訊至單一門號
private void SubmitButton_Click(object sender, EventArgs e)
{
    //重送期限 (0 ~ 1440 分鐘)
    int nExpiredMinutes = 0; //表示系統最長 1440 分鐘

    if(SubmitExpired.Text.Length > 0) {
        if(!Int32.TryParse(SubmitExpired.Text, out nExpiredMinutes)) {
            MessageBox.Show("重送期限須為數字，範圍為 0 ~ 1440 分鐘！", "無法發送");
            return;
        }
    }

    bool bEnglish = false;
    int nTotalSplit = 1;

    DialogResult dr;
    if(nType == TEST_BINARY) { //輸入訊息須為 HEX 字串
        byte[] byteMessage = m_sns.GetByteArrayFromHexString(SubmitMessage.Text);

        if(byteMessage != null) {
            if(!Int32.TryParse(Coding.Text, NumberStyles.HexNumber, null, out nCoding) ||
                (nCoding < 0x00) || (nCoding > 0xFF)) {
                MessageBox.Show("請輸入正確的編碼資訊，範圍 00 ~ FF。", "無法發送");
                return;
            }
            } else if(nTotalSplit > 1) {
                dr = MessageBox.Show("即將發送含 " + nTotalSplit + " 則二位元訊息：\n\n使用的編碼為 0x"
                    + Coding.Text + "\n\n確認送出？", strSendType, MessageBoxButtons.OKCancel);
            } else {
                dr = MessageBox.Show("即將發送二位元訊息：\n\n使用的編碼為 0x" + Coding.Text
                    + "\n\n確認送出？", strSendType, MessageBoxButtons.OKCancel);
            }
        } else {
            MessageBox.Show("沒有輸入或錯誤的 HEX 字串資料！", "無法發送");
            return;
        }
    } else {
        bEnglish = m_sns.IsPureEnglishMessage(SubmitMessage.Text);
        nTotalSplit = m_sns.GetTotalSplit(SubmitMessage.Text, udhi.Checked, bEnglish);

        if(nTotalSplit == 1) {
            dr = MessageBox.Show("即將發送單通簡訊？", strSendType, MessageBoxButtons.OKCancel);

```

```

    } else {
        if(udhi.Checked) {
            dr = MessageBox.Show("即將發送含 " + nTotalSplit + " 則之單通長簡訊?", strSendType,
                                MessageBoxButtons.OKCancel);
        } else {
            dr = MessageBox.Show("即將發送含 " + nTotalSplit + " 則簡訊?", strSendType,
                                MessageBoxButtons.OKCancel);
        }
    }
}

if(dr != DialogResult.OK) {
    return;
}

String strFromMsisdn = SubmitFromMsisdn.Text;
String strToMsisdn = SubmitToMsisdn.Text;
StringBuilder sbMsgidList = new StringBuilder();

nSerno = (nSerno + 1) % 0x100;

//傳送則數為 1-based
for(int nCurrSplit = 1; nCurrSplit <= nTotalSplit; nCurrSplit++) {
    int nSnsCode = 0;

    if(nTotalSplit == 1) {
        if(nType == TEST_BINARY) {
            String strHexMessage = SubmitMessage.Text;
            int nHexLen = strHexMessage.Length;
            int nUDHI = udhi.Checked ? 1 : 0;
            int nCoding = Convert.ToInt32(Coding.Text, 16);

            nSnsCode = m_sns.SubmitBinaryMessage(strFromMsisdn, strToMsisdn, nUDHI, nCoding,
                                                strHexMessage, nHexLen, nExpiredMinutes);
        } else {
            String strMessage = SubmitMessage.Text;

            if(nType == TEST_DEFAULT) {
                nSnsCode = m_sns.SubmitMessage(strFromMsisdn, strToMsisdn, strMessage, nExpiredMinutes);
            } else if(nType == TEST_SUBNO) {
                nSnsCode = m_sns.SubmitMessageWithSubno(strFromMsisdn, strToMsisdn, strMessage,
                                                        nExpiredMinutes);
            } else if(nType == TEST_RECV_SUBNO) {
                nSnsCode = m_sns.SubmitMessageWithRecvSubno(strFromMsisdn, strToMsisdn, strMessage,
                                                            nExpiredMinutes);
            } else if(nType == TEST_ALL_SUBNO) {
                nSnsCode = m_sns.SubmitMessageWithAllSubno(strFromMsisdn, strToMsisdn, strMessage,
                                                            nExpiredMinutes);
            }
        }
    } else {
        bool bUDHI = udhi.Checked;

        if(nType == TEST_BINARY) {
            String strLongHexMessage = SubmitMessage.Text;
            int nLongHexLen = byteLongMessage.Length;
            int nUDHI = bUDHI ? 1 : 0;

            String strSplitHexMessage = m_sns.GetSplitHexMessage(nUDHI, strLongHexMessage,
                                                                nLongHexLen, nSerno, nCurrSplit, nTotalSplit);
            int nHexLen = (strSplitHexMessage != null) ? strSplitHexMessage.Length : 0;
            int nCoding = Convert.ToInt32(Coding.Text, 16);

```

```

        nSnsCode = m_sns.SubmitBinaryMessage(strFromMsisdn, strToMsisdn, nUDHI,
                                             nCoding, strSplitHexMessage, nHexLen, nExpiredMinutes);
    } else {
        String strLongMessage = SubmitMessage.Text;
        String strSplitSMS = m_sns.GetSplitSMS(strLongMessage, bUDHI, nCurrSplit, nTotalSplit, bEnglish);

        if(nType == TEST_DEFAULT) {
            nSnsCode = m_sns.SubmitLongMessage(strFromMsisdn, strToMsisdn, bUDHI, bEnglish, strSplitSMS,
                                                nSerno, nCurrSplit, nTotalSplit, nExpiredMinutes);
        } else if(nType == TEST_SUBNO) {
            nSnsCode = m_sns.SubmitLongMessageWithSubno(strFromMsisdn, strToMsisdn, bUDHI, bEnglish,
                                                         strSplitSMS, nSerno, nCurrSplit, nTotalSplit, nExpiredMinutes);
        } else if(nType == TEST_RECV_SUBNO) {
            nSnsCode = m_sns.SubmitLongMessageWithRecvSubno(strFromMsisdn, strToMsisdn, bUDHI,
                                                             bEnglish, strSplitSMS, nSerno, nCurrSplit, nTotalSplit, nExpiredMinutes);
        } else { //TEST_ALL_SUBNO
            nSnsCode = m_sns.SubmitLongMessageWithAllSubno(strFromMsisdn, strToMsisdn, bUDHI, bEnglish,
                                                            strSplitSMS, nSerno, nCurrSplit, nTotalSplit, nExpiredMinutes);
        }
    }
}

String strSnsDesc = m_sns.GetRespDesc();

String strPrefix = (nTotalSplit == 1) ? "" : ("第 " + nCurrSplit + " 則 : ");

switch(nSnsCode) {
    case -2:
        MessageBox.Show(strPrefix + "[" + nSnsCode + "]" + strSnsDesc, "參數錯誤");
        break;
    case -1:
        MessageBox.Show(strPrefix + "[" + nSnsCode + "]" + strSnsDesc, "無法連線 SNS");
        break;
    case 0:
        MessageBox.Show(strPrefix + "(得簡訊識別碼) " + strSnsDesc, "簡訊傳送成功");

        if(sbMsgidList.Length > 0) {
            sbMsgidList.Append(", ");
        }

        sbMsgidList.Append(strSnsDesc); //簡訊識別碼
        break;
    default:
        MessageBox.Show(strPrefix + "[" + nSnsCode + "]" + strSnsDesc, "請參考 SNS Protocol 文件");
        break;
}

if(sbMsgidList.Length > 0) {
    QueryFromMsisdn.Text = SubmitFromMsisdn.Text;
    QueryToMsisdn.Text = SubmitToMsisdn.Text;
    MsgidList.Text = sbMsgidList.ToString();
}

CheckButtonState();
}

private static String GetCategory(int nSnsCode, String strSnsDesc)
{
    /*

```

```

* 當系統已到達最終狀態時 (code 為 0、3、8、9、11、12、13、32)，將會包含訊息最終狀態之時間，
* 並以：做為區隔字元，時間格式為 yyyyymmddHHMMSS。
*/
String strStatusType = null;

switch(nSnsCode) {
    case 0: //Successful:yyyyymmddHHMMSS
        strStatusType = "發訊成功";
        break;
    case 1: //Message is processing
    case 31: //Message is submitting
        strStatusType = "發訊中 (稍後再查)";
        break;
    case 2: //System contains no data
    case 4: //System error
    case 5: //Message status unknown
    case 21: //System error
        strStatusType = "無法得知 (建議多查幾次確認)";
        break;
    case 3: //Message can not send to GSM/Pager:yyyyymmddHHMMSS
    case 8: //SIM card memory full:yyyyymmddHHMMSS
    case 9: //Destination number is unavailable:yyyyymmddHHMMSS
    case 11: //Destination number is error:yyyyymmddHHMMSS
    case 12: //Mobile equipment can not receive SMS:yyyyymmddHHMMSS
    case 13: //Mobile equipment error:yyyyymmddHHMMSS
    case 32: //Message send to SMSC fail:yyyyymmddHHMMSS
    default:
        strStatusType = "發訊失敗";
        break;
}

return strStatusType;
}

//當按下「狀態查詢」鍵以查詢已送簡訊傳送狀態
private void QueryButton_Click(object sender, EventArgs e)
{
    String[] saMsgID = MsgidList.Text.Split(",".ToCharArray());

    for(int i = 0; i < saMsgID.Length; i++) {
        if(saMsgID[i].Trim().Trim().Length > 0) {
            int nSnsCode = 0;

            if((nType == TEST_DEFAULT) || (nType == TEST_BINARY)) {
                nSnsCode = m_sns.QueryMessageStatus(QueryFromMsisdn.Text, QueryToMsisdn.Text, saMsgID[i].Trim());
            } else if(nType == TEST_SUBNO) {
                nSnsCode = m_sns.QueryMessageStatusWithSubno(QueryFromMsisdn.Text, QueryToMsisdn.Text,
                    saMsgID[i].Trim());
            } else if(nType == TEST_RECV_SUBNO) {
                nSnsCode = m_sns.QueryMessageStatusWithRecvSubno(QueryFromMsisdn.Text, QueryToMsisdn.Text,
                    saMsgID[i].Trim());
            } else { //TEST_ALL_SUBNO
                nSnsCode = m_sns.QueryMessageStatusWithAllSubno(QueryFromMsisdn.Text, QueryToMsisdn.Text,
                    saMsgID[i].Trim());
            }

            String strSnsDesc = m_sns.GetRespDesc();

            String strPrefix = (saMsgID.Length == 1) ? "" : ("第 " + (i+1) + " 則：");

            switch(nSnsCode) {

```

```

        case -2:
            MessageBox.Show(strPrefix + "[" + nSnsCode + "]" + strSnsDesc, "參數錯誤");
            break;
        case -1:
            MessageBox.Show(strPrefix + "[" + nSnsCode + "]" + strSnsDesc, "無法連線 SNS");
            break;
        default:
            String strResult = GetCategory(nSnsCode, strSnsDesc);
            MessageBox.Show(strPrefix + "[" + nSnsCode + "]" + strSnsDesc, strResult);
            break;
    }
}

}

CheckButtonState();
}

//當按下「接收訊息」鍵以接收手機上傳的簡訊
private void GetButton_Click(object sender, EventArgs e)
{
    int nSnsCode = m_sns.GetMessage();

    switch (nSnsCode)
    {
        case -2:
            MessageBox.Show "[" + nSnsCode + "]" + m_sns.GetRespDesc(), "參數錯誤");
            break;
        case -1:
            MessageBox.Show "[" + nSnsCode + "]" + m_sns.GetRespDesc(), "無法連線 SNS");
            break;
        case 0:
            MessageBox.Show("接收到上傳簡訊", "收訊成功");

            RecvFromMsisdn.Text = m_sns.GetFromMsisdn();
            RecvToMsisdn.Text = m_sns.GetToMsisdn();
            RecvMessage.Text = m_sns.GetRespDesc();

            //若要取得長簡訊真正的分則訊息內容
            {
                bool bUDHI = false; //若為 true 表示是長簡訊
                int nSerno = 0; //若是長簡訊，則單一長簡訊之多則簡訊序號應相同。
                int nCurrSplit = 0; //若是長簡訊，則此為第幾則簡訊 (1-based)
                int nTotalSplit = 0; //若是長簡訊，則共含多少則簡訊。
                byte[] byteArray = null;
                int nStartIdx = 0;

                if (m_sns.GetIsEnglish()) {
                    byteArray = m_sns.ToAsciiByteArray(RecvMessage.Text);
                } else {
                    byteArray = m_sns.ToUcs16BeByteArray(RecvMessage.Text);
                }

                if (byteArray != null) {
                    if (!bUDHI && (byteArray.Length > 6) &&
                        (byteArray[0] == 0x05) && (byteArray[1] == 0x00) && (byteArray[2] == 0x03)) {
                        byte b3 = byteArray[3];
                        byte b4 = byteArray[4];
                        byte b5 = byteArray[5];
                        nSerno = (int)b3;
                        nTotalSplit = (int)b4;
                        nCurrSplit = (int)b5;
                    }
                }
            }
        }
    }
}

```

```

        bUDHI = true;
        nStartIdx = 6; //byteArray[6] 之後為該分則簡訊之二位元編碼訊息內容
    }
    if (!bUDHI && (byteArray.Length > 7) &&
        (byteArray[0] == 0x06) && (byteArray[1] == 0x08) && (byteArray[2] == 0x04)) {
        byte b3 = byteArray[3];
        byte b4 = byteArray[4];
        byte b5 = byteArray[5];
        byte b6 = byteArray[6];
        nSerno = (int)(b3 * 0xFF) + (int)b4;
        nTotalSplit = (int)b5;
        nCurrSplit = (int)b6;
        bUDHI = true;
        nStartIdx = 7; //byteArray[7] 之後為該分則簡訊之二位元編碼訊息內容
    }
}

if (bUDHI) {
    byte[] byteSplit = null; //取得所要的分則簡訊二位元編碼訊息內容

    byteSplit = new byte[byteArray.Length - nStartIdx];
    Array.Copy(byteArray, nStartIdx, byteSplit, 0, byteArray.Length - nStartIdx);

    //實際應用應是根據 nCurrSplit，依序整合同個 nSerno 所有則數 byteSplit 至較大的 byte 串列，
    //再續下列 API 取得完整的長簡訊內容。
    String strSplitSMS = null;

    if (m_sns.IsPureEnglishByteArray(byteSplit)) {
        strSplitSMS = m_sns.GetStringFromAsciiByteArray(byteSplit);
    } else {
        strSplitSMS = m_sns.GetStringFromUcs16BeByteArray(byteSplit);
    }

    MessageBox.Show(strSplitSMS, "收到序號 0x" + nSerno.ToString("X2") + " 長簡訊之第 "
        + nCurrSplit + "/" + nTotalSplit + " 則簡訊");
}
}
break;
default:
    MessageBox.Show("[ " + nSnsCode + " ] " + m_sns.GetRespDesc(), "請參考 SNS Protocol 文件");
    break;
}

CheckButtonState();
}

//當按下「保持連線」鍵
private void LinkButton_Click(object sender, EventArgs e)
{
    int nSnsCode = m_sns.EnquireLink();

    switch(nSnsCode) {
        case -1:
            MessageBox.Show("[ " + nSnsCode + " ] " + m_sns.GetRespDesc(), "無法連線 SNS");
            break;
        case 0:
            MessageBox.Show("[ " + nSnsCode + " ] " + m_sns.GetRespDesc(), "保持連線成功");
            break;
        default:
            MessageBox.Show("[ " + nSnsCode + " ] " + m_sns.GetRespDesc(), "請參考 SNS Protocol 文件");
            break;
    }
}

```

```

    }

    CheckButtonState();
}

//當按下「登出連線」鍵
private void LogoutButton_Click(object sender, EventArgs e)
{
    m_sns.Logout();

    MessageBox.Show("切斷與 SNS Server 的連線", "結束作業");

    CheckButtonState();
}
}
}

```

當客戶使用本 SNS COM+ v3 元件，要開發自己的簡訊應用程式時，請注意下列幾點：(茲以本章節之視窗測試程式說明為例)

(1) 使用本 SNS COM+ v3 服務，要先執行連線初始化 (參考 SNS COM+ 之 *InitRemoteServer()* API 說明)；

(2) 開始使用 SNS 服務時，需先進行登入驗證，通過驗證後方可使用下列服務 (參考本視窗範例之 *LoginButton_Click()* 用法說明)；

註： 您可至本公司服務窗口，申請您的 SNS 帳號專屬允入連線 IP 範圍，進一步保護您的帳號使用安全，以及可另設定額外的發訊、及接收相關服務權限。

- a. 需要時，請執行密碼異動作業 (參考 *EditPasswdButton_Click()* 用法範例)；
- b. 若有待送訊息資料時，請先判斷是否為長簡訊，再以一或多則發送簡訊，並取得各則之簡訊識別碼，以確保受訊端收到完整的訊息內容 (參考 *SubmitButton_Click()* 用法範例；若有發送長簡訊、網外或國外門號者，您的 SNS 帳號需要有此權限)；
- c. 若要查詢已送出簡訊的發訊結果狀態時，進行簡訊發送結果查詢 (參考 *QueryButton_Click()* 用法範例)；
- d. 若有接收手機上傳簡訊服務之需求時，進行接收簡訊作業 (參考 *GetButton_Click()* 用法範例；您的 SNS 帳號需要有此權限)；
- e. 若一段時間未使用該 SNS 連線時，建議每隔兩三分鐘定期執行保持連線作業，以避免連線被 SNS 伺服器切斷 (參考 *LinkButton_Click()* 用法範例；若您的自己程式採用 Multi-Thread 程式架構，可定期執行已連線的 SNS COM+ 之 *EnquireLink()* 作業)；
- f. 使用上述作業後，檢查 SNS 連線是否仍存在，若被中斷再重新執行切線及連線作業 (參考 SNS COM+ 之 *IsBind()* API)；

g. 若經重新連線後，需要繼續之前原本未完成的發訊 / 查詢 / 接收作業時，請至多執行一或兩次即可（請參考 SNS COM+ 回傳的 SNS Protocol 錯誤碼及訊息說明，以免可能重覆無效的作業）。

(3) 最後，當您的應用程式不再使用本 SNS 服務時，請執行登出連線作業（參考 LogoutButton_Click() 用法範例）。

一般實務上，使用 SNS COM+ v3 元件，也是可以建立各自的 SNS 連線專司處理發訊 / 查詢 / 接收之簡訊服務作業，僅可能使用到最少的連線數，因各個 SNS 帳號連線數有限制。

因本 SNS COM+ 元件不是 thread-safe，若您的應用程式是多執行緒作業時，但共使用一個 SNS 連線，須自行處理 lock SNS 連線機制，以避免多個執行緒同時處理同一個 SNS 連線時，可能造成您的程式發生未預期的情況。

在您自行開發簡訊應用程式中，需要時也可加入資料庫，將所要的發訊及查詢作業寫入資料庫及查詢，再由另一個 SNS COM+ v3 應用程式定期讀取資料庫資料，透過提供的 API 進行發訊、及簡訊狀態查詢，整個作業皆保持 SNS 連線，並於閒置一段時間無作業時（如：兩三分鐘）定期執行保持連線作業，以便提供穩定的 SNS 連線品質相關作業。

本說明文件不提供 SNS COM+ v3 元件結合資料庫之簡訊應用範例相關說明。

5.2 C#.NET 視窗版多門號範例程式

本範例相關程式碼可參考“CSharp.NET 多門號範例.zip”壓縮檔。

使用者可執行範例內的 SnsCom3App.exe 程式範例，以使用同個目錄之 SNS COM+ 第三版元件之所有 API 功能，其相關畫面類似 5.1 章節，主要差異為可同時發送多個門號之簡訊/長簡訊、查詢多個門號發送結果。

下圖為和 5.1 章節操作畫面差異之處：

一次發送多個門號之訊息範例
(各個門號之間以 , 或 ; 符號分隔)

一次查詢多個受訊門號之發訊結果範例
其單一資料格式為：受訊門號(簡訊識別碼清單，多個清單之間以 , 符號分隔)
(各個資料之間以 ; 符號分隔)

本多門號視窗程式之 C# 部份程式碼範例如下 (大部份程式範例和 5.1 章節相同，以下僅摘錄多個號碼發送及狀態查詢部份，並只含一般簡訊/長簡訊發送部份，且只截取單純受訊電話號碼者片段；若要完整的程式碼，可參考其壓縮檔並解壓之，以便使用其測試專案檔範例)，並請參考前述第 5.1 章節後述之注意事項、及一般實務用法建議說明：

//當按下「傳送訊息」鍵以發送單則或多則簡訊至一或多個受訊門號

```
private void SubmitButton_Click(object sender, EventArgs e) {
```

```
    //重送期限 (0 ~ 1440 分鐘)
```

```
    int nExpiredMinutes = 0; //表示系統最長 1440 分鐘
```

```
    ... (略)...
```

```
    bool bEnglish = false;
```

```
    int nTotalSplit = 1;
```

```
    DialogResult dr;
```

```
    if(m_nType == TEST_BINARY) { //輸入訊息須為 HEX 字串
```

```
        ... (略)...
```

```
    } else {
```

```
        bEnglish = m_sns.IsPureEnglishMessage(SubmitMessage.Text);
```

```
        nTotalSplit = m_sns.GetTotalSplit(SubmitMessage.Text, udhi.Checked, bEnglish);
```

```
        if(nTotalSplit == 1) {
```

```
            dr = MessageBox.Show("即將發送單通簡訊？", m_strSendType, MessageBoxButtons.OKCancel);
```

```
        } else {
```

```

        if(udhi.Checked) {
            dr = MessageBox.Show("即將發送含 " + nTotalSplit + " 則之單通長簡訊?", m_strSendType,
                                MessageBoxButtons.OKCancel);
        } else {
            dr = MessageBox.Show("即將發送含 " + nTotalSplit + " 則簡訊?", m_strSendType,
                                MessageBoxButtons.OKCancel);
        }
    }
}

if(dr != DialogResult.OK) {
    return;
}

String strFromMsisdn = SubmitFromMsisdn.Text;
String[] saToMsisdn = SubmitToMsisdnList.Text.Split(";", ToCharArray()); //受訊門號以 , 或 ; 分隔
StringBuilder sbMsisdnMsgidList = new StringBuilder();
StringBuilder sbResult = new StringBuilder();
int nOK = 0;
int nFailed = 0;

m_sbSubmitDebug.Remove(0, m_sbSubmitDebug.Length);
m_nSerno = (m_nSerno + 1) % 0x100;

for(int i = 0; i < saToMsisdn.Length; i++) {
    String strToMsisdn = saToMsisdn[i].Trim();
    StringBuilder sbMsgidList = new StringBuilder();
    bool bOK = true;
    String strTitle = "(" + (((i+1) < 10) ? "0" : "") + (i+1) + ") " + strToMsisdn;

    if(i > 0) {
        m_sbSubmitDebug.Append("\n");
    }
    m_sbSubmitDebug.Append(strTitle + " : ");

    //傳送則數為 1-based
    for(int nCurrSplit = 1; nCurrSplit <= nTotalSplit; nCurrSplit++) {
        int nSnsCode = 0;

        if(nTotalSplit == 1) {
            if(m_nType == TEST_BINARY) {
                ...(略)...
            } else {
                String strMessage = SubmitMessage.Text;

                if(m_nType == TEST_DEFAULT) {
                    nSnsCode = m_sns.SubmitMessage(strFromMsisdn, strToMsisdn, strMessage, nExpiredMinutes);
                } else ... {
                    ...(略)...
                }
            }
        }
    }
} else {
    bool bUDHI = udhi.Checked;

    if(m_nType == TEST_BINARY) {
        ...(略)...
    } else {
        String strLongMessage = SubmitMessage.Text;
        String strSplitSMS = m_sns.GetSplitSMS(strLongMessage, bUDHI, nCurrSplit, nTotalSplit,
                                                bEnglish);
    }
}

```

```

        if(m_nType == TEST_DEFAULT) {
            nSnsCode = m_sns.SubmitLongMessage(strFromMsisdn, strToMsisdn, bUDHI,
                                                bEnglish, strSplitSMS, m_nSerno, nCurrSplit, nTotalSplit,
                                                nExpiredMinutes);

            } else ... {
                ...(略)...
            }
        }
    }

    String strSnsDesc = m_sns.GetRespDesc();

    if(nTotalSplit > 1) {
        m_sbSubmitDebug.Append("\n        第 " + nCurrSplit + " 則 : ");
    }

    switch(nSnsCode) {
        case -2:
            m_sbSubmitDebug.Append("參數錯誤: [" + nSnsCode + "] " + strSnsDesc);
            bOK = false;
            break;
        case -1:
            m_sbSubmitDebug.Append("無法連線: [" + nSnsCode + "] " + strSnsDesc);
            bOK = false;
            break;
        case 0:
            m_sbSubmitDebug.Append("傳送成功: [" + nSnsCode + "] " + strSnsDesc);

            if(sbMsgidList.Length > 0) {
                sbMsgidList.Append(", ");
            }

            sbMsgidList.Append(strSnsDesc); //簡訊識別碼
            break;
        default:
            m_sbSubmitDebug.Append("傳送失敗: [" + nSnsCode + "] " + strSnsDesc);
            bOK = false;
            break;
    }
}

if(sbMsisdnMsgidList.Length > 0) {
    sbMsisdnMsgidList.Append("; ");
}

if(sbResult.Length > 0) {
    sbResult.Append("\n");
}

if(bOK || (sbMsgidList.Length > 0)) {
    nOK++;
    sbMsisdnMsgidList.Append(strToMsisdn + "(" + sbMsgidList.ToString() + ")");
    sbResult.Append(strTitle + ": " + (bOK ? "OK" : "部份則數 OK"));
} else {
    nFailed++;
    sbResult.Append(strTitle + ": FAILED");
}
}

if(checkSubmitDebug.Checked) {
    MessageBox.Show(m_sbSubmitDebug.ToString(), "偵錯");
}

```

```

}

MessageBox.Show("共發送 " + (nOK + nFailed) + " 通：成功 " + nOK + " 通，失敗 " + nFailed + " 通。 \n\n"
    + sbResult.ToString(), "發訊結果");

if(sbMsisdnMsgidList.Length > 0) {
    QueryFromMsisdn.Text = SubmitFromMsisdn.Text;
    QueryMsisdnMsgidList.Text = sbMsisdnMsgidList.ToString();
}

CheckButtonState();
}

//當按下「狀態查詢」鍵以查詢已送簡訊傳送狀態
private void QueryButton_Click(object sender, EventArgs e) {

    //其多個資料（受訊門號及簡訊識別碼）以，或；分隔
    String[] saMsisdnMsgidList = QueryMsisdnMsgidList.Text.Split(";;".ToCharArray());

    m_sbQueryDebug.Remove(0, m_sbQueryDebug.Length);

    for(int i = 0; i < saMsisdnMsgidList.Length; i++) {
        String strMsisdnMsgidList = saMsisdnMsgidList[i].Trim();

        int nIdx = strMsisdnMsgidList.IndexOf("(");
        int nIdx2 = strMsisdnMsgidList.IndexOf(")");

        if((nIdx2 > nIdx) && (nIdx >= 10)) {
            String strToMsisdn = strMsisdnMsgidList.Substring(0, nIdx).Trim();
            String[] saMsgID = strMsisdnMsgidList.Substring(nIdx + 1, nIdx2 - nIdx - 1).Trim().Split(",".ToCharArray());

            String strTitle = "(" + ((i + 1) < 10) ? "0" : "" + (i + 1) + ") " + strToMsisdn;

            if(m_sbQueryDebug.Length > 0) {
                m_sbQueryDebug.Append("\n");
            }

            m_sbQueryDebug.Append(strTitle + " : ");

            for(int j = 0; j < saMsgID.Length; j++) {
                String strMsgID = saMsgID[j].Trim();

                if(strMsgID.Length > 0) {
                    int nSnsCode = 0;

                    if((m_nType == TEST_DEFAULT) || (m_nType == TEST_BINARY)) {
                        nSnsCode = m_sns.QueryMessageStatus(QueryFromMsisdn.Text, strToMsisdn, strMsgID);
                    } else ... {
                        ...(略)...
                    }

                    String strSnsDesc = m_sns.GetRespDesc();

                    m_sbQueryDebug.Append("\n      ");
                    if(saMsgID.Length > 1) {
                        m_sbQueryDebug.Append("第 " + (j+1) + " 則 : ");
                    }

                    switch(nSnsCode) {
                        case -2:
                            m_sbQueryDebug.Append("參數錯誤: [" + nSnsCode + "] " + strSnsDesc);

```

```

        break;
    case -1:
        m_sbQueryDebug.Append("無法連線: [" + nSnsCode + "]" + strSnsDesc);
        break;
    default:
        String strResult = GetCategory(nSnsCode, strSnsDesc);
        m_sbQueryDebug.Append(strResult + ": [" + nSnsCode + "]" + strSnsDesc);
        break;
    }
}
}
}

MessageBox.Show(m_sbQueryDebug.ToString(), "查詢結果");

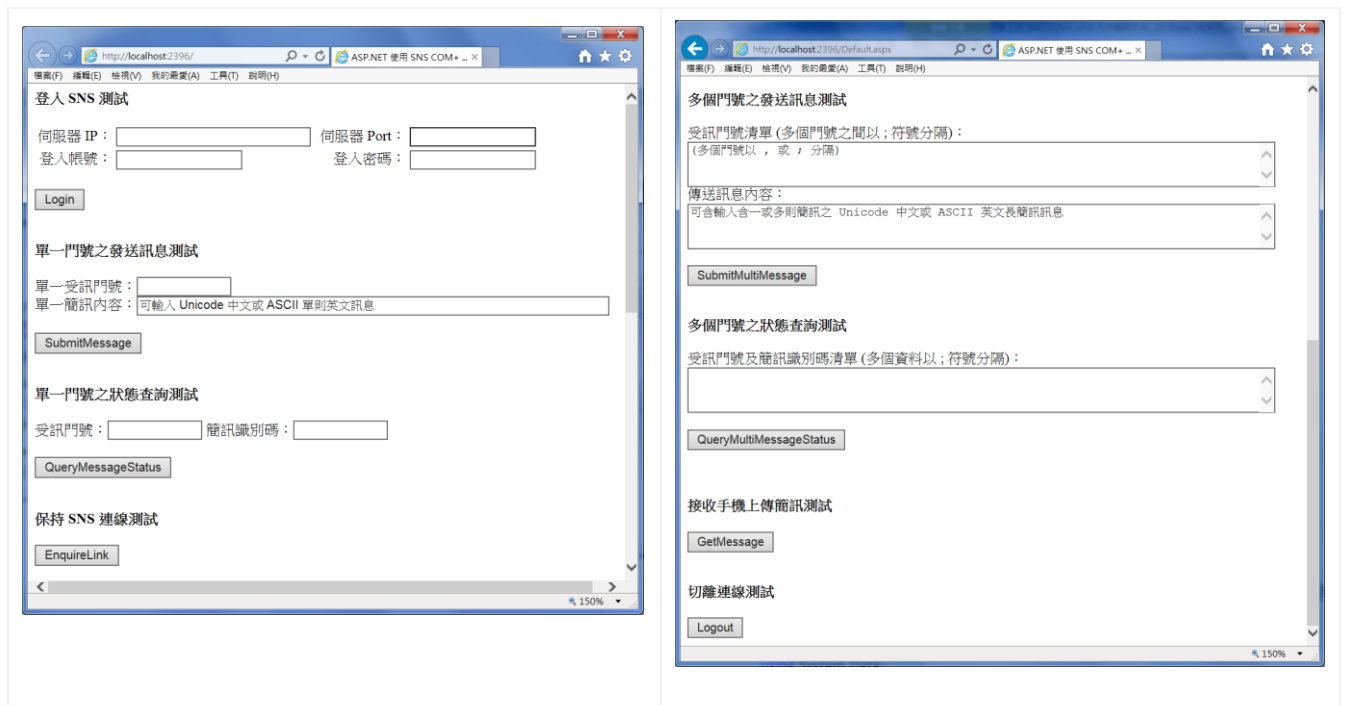
CheckButtonState();
}

```

5.3 ASP.NET 網頁範例程式

本範例相關程式碼可參考“ASP.NET 網頁範例.zip”壓縮檔。

其參考相關 ASP.NET 範例網頁畫面如下，當使用者完成輸入資料後及按鍵送出，即會將 SNS COM+ V3 元件之作業結果輸出至該鍵下方：



以下為完整的 ASP.NET 網頁之實際處理的程式碼範例，執行每個作業皆包含建立、及切離 SNS 連線。

```

using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Text;

namespace SamplePage
{
    public partial class _Default : System.Web.UI.Page
    {
        //建立 SNS COM 物件
        private CHT.SNSCOMSERVER.SnsComObject3 m_sns = new CHT.SNSCOMSERVER.SnsComObject3();

        //發送長簡訊之參考序號
        private static int m_nSerno = new Random().Next(0x100); //範圍: 0x00 ~ 0xFF

        //清除原輸出訊息作業結果內容
        protected void Page_Load(object sender, EventArgs e)
        {
            ResultLogin.Text = "";
            ResultSubmitMessage.Text = "";
            ResultQryMessageStatus.Text = "";
            ResultGetMessage.Text = "";
            ResultLogout.Text = "";
        }

        //登入 SNS 測試
        protected void LoginButton_Click(object sender, EventArgs e)
        {
            m_sns.InitRemoteServer(ip.Text, port.Text, account.Text, passwd.Text);

            int nResult = m_sns.Login();

            ResultLogin.Text = "[" + nResult + "]" + m_sns.GetRespDesc();

            m_sns.Logout();
        }

        //單一門號之發送訊息測試
        protected void SubmitMessage_Click(object sender, EventArgs e)
        {
            m_sns.InitRemoteServer(ip.Text, port.Text, account.Text, passwd.Text);

            int nResult = m_sns.Login();

            if (nResult == 0) {
                //傳送手機號碼, 訊息內容 (使用預設發訊號碼, 及重送期限使用預設一天 1440 分鐘)
                nResult = m_sns.SubmitMessage(null, msisdn.Text, text.Text, 1440);

                //若發訊成功, 則更新至狀態查詢測試的兩個欄位, 以便進一步查詢發送結果。
                if (nResult == 0) {
                    msisdn2.Text = msisdn.Text;
                    msgid.Text = m_sns.GetRespDesc();
                }
            }
        }
    }
}

```

```

        ResultSubmitMessage.Text = "[" + nResult + "]" + m_sns.GetRespDesc();

        m_sns.Logout();
    }

    //單一門號之狀態查詢測試
    protected void QueryMessageStatus_Click(object sender, EventArgs e)
    {
        m_sns.InitRemoteServer(ip.Text, port.Text, account.Text, passwd.Text);

        int nResult = m_sns.Login();

        if (nResult == 0) {
            nResult = m_sns.QueryMessageStatus(null, msisd2.Text, msgid.Text);
        }

        ResultQryMessageStatus.Text = "[" + nResult + "]" + m_sns.GetRespDesc();

        m_sns.Logout();
    }

    //一或多個門號之發送訊息測試 (簡訊或長簡訊)
    protected void SubmitMultiMessage_Click(object sender, EventArgs e)
    {
        m_sns.InitRemoteServer(ip.Text, port.Text, account.Text, passwd.Text);

        int nResult = m_sns.Login();

        if (nResult == 0) {
            String[] saToMsisdn = msisd3.Text.Split(",;".ToCharArray()); //受訊門號以 , 或 ; 分隔
            StringBuilder sbHtmlResult = new StringBuilder();

            String strFromMsisdn = null; //使用預設發訊號碼
            int nExpiredMinutes = 1440; //使用重送期限使用預設一天 1440 分鐘

            String strLongText = text3.Text; //要傳送的含一或多則簡訊之完整訊息內容
            bool bEnglish = m_sns.IsPureEnglishMessage(strLongText); //判斷是否純英文訊息
            Boolean bUDHI = true; //使用長簡訊
            int nTotalSplit = m_sns.GetTotalSplit(strLongText, bUDHI, bEnglish); //共多少則

            StringBuilder sbMultiMsgid = new StringBuilder();

            for (int i = 0; i < saToMsisdn.Length; i++) {
                String strToMsisdn = saToMsisdn[i];

                if ((strToMsisdn != null) && (strToMsisdn.Trim().Length > 0)) {
                    if (sbHtmlResult.Length > 0) {
                        sbHtmlResult.Append("<br/>"); //若超過一個以上受訊門號時
                    }

                    StringBuilder sb = new StringBuilder();

                    if (nTotalSplit == 1) {
                        //傳送手機號碼, 單則簡訊內容
                        nResult = m_sns.SubmitMessage(strFromMsisdn, strToMsisdn.Trim(), strLongText,
nExpiredMinutes);

                        //發訊結果
                        sbHtmlResult.Append "[" + nResult + "]" + m_sns.GetRespDesc();

```



```

        //若發訊成功，可便進一步以簡訊識別碼查詢發送結果。
        if (nResult == 0) {
            sb.Append(m_sns.GetRespDesc()); //發送成功之簡訊識別碼
        }
    }
    else //分多則簡訊發送長簡訊
    {
        StringBuilder sbSplitResult = new StringBuilder();

        //確保每次發送長簡訊，使用不同的長簡訊序號 (若是同一批不同門號發送，也可使用同一
        //個長簡訊序號，但下一批需使用新的長簡訊序號)
        m_nSerno = (m_nSerno + 1) % 0x100;

        sbHtmlResult.Append(strToMsisdn.Trim() + " (長簡訊序號 " + m_nSerno + "，含 " + nTotalSplit +
        " 則簡訊):<br/>"); //該門號之發送記錄

        for (int nCurrSplit = 1; nCurrSplit <= nTotalSplit; nCurrSplit++) { //使用 1-based 則數
            //取得其長簡訊之第幾則簡訊內容
            String strSplitSMS = m_sns.GetSplitSMS(strLongText, bUDHI, nCurrSplit, nTotalSplit,
            bEnglish);

            //傳送手機號碼，單則簡訊內容 (使用預設發訊號碼，及重送期限使用預設一天 1440
            //分鐘)

            nResult = m_sns.SubmitLongMessage(strFromMsisdn, strToMsisdn.Trim(), bUDHI, bEnglish,
            strSplitSMS,

            m_nSerno, nCurrSplit, nTotalSplit, nExpiredMinutes);

            //發訊結果
            sbSplitResult.Append("第 " + nCurrSplit + " 則 [" + nResult + "] " + m_sns.GetRespDesc() +
            "<br/>");

            //若發訊成功，可便進一步以簡訊識別碼以一個一個查詢發送結果。
            if (nResult == 0) {
                if (sb.Length > 0) {
                    sb.Append(";"); //多個簡訊識別碼以，分隔
                }

                sb.Append(m_sns.GetRespDesc()); //發送成功之簡訊識別碼
            }
        }

        sbHtmlResult.Append(sbSplitResult.ToString());
    }

    if (sb.Length > 0) {
        if (sbMultiMsgid.Length > 0) {
            sbMultiMsgid.Append(";"); //以；分隔多個發訊成功之門號及簡訊識別碼時
        }

        sbMultiMsgid.Append(strToMsisdn.Trim() + "(" + sb.ToString() + ")"); //該門號稍後要查的發訊
        //結果
    }
}

ResultSubmitMultiMessage.Text = sbHtmlResult.ToString();

MultiMsgid.Text = sbMultiMsgid.ToString(); //稍後要查的發送結果
}
else
{

```

```

        ResultSubmitMultiMessage.Text = "[" + nResult + "]" + m_sns.GetRespDesc();
    }

    m_sns.Logout();
}

//多個門號及一或多個簡訊識別碼之狀態查詢測試
protected void QueryMultiMessageStatus_Click(object sender, EventArgs e)
{
    m_sns.InitRemoteServer(ip.Text, port.Text, account.Text, passwd.Text);

    StringBuilder sbHtmlStatus = new StringBuilder();

    int nResult = m_sns.Login();

    if (nResult == 0) {
        //多個資料 (門號及一或多個簡訊識別碼) 以 ; 分隔
        String[] saMsisdnMsgid = MultiMsgid.Text.Split(";".ToCharArray());

        if ((saMsisdnMsgid != null) && (saMsisdnMsgid.Length > 0)) {
            for (int i = 0; i < saMsisdnMsgid.Length; i++)
            {
                String[] sa = saMsisdnMsgid[i].Split("(").ToCharArray();
                String strMsisdn = sa[0];

                if ((strMsisdn != null) && (strMsisdn.Length > 0)) {
                    if (sbHtmlStatus.Length > 0) {
                        sbHtmlStatus.Append("<br/>");
                    }

                    sbHtmlStatus.Append(strMsisdn.Trim() + ":<br/>");

                    for (int j = 1; j < sa.Length; j++) {
                        String strMsgid = sa[j];

                        if ((strMsgid != null) && (strMsgid.Trim().Length > 0)) {
                            nResult = m_sns.QueryMessageStatus(null, strMsisdn.Trim(), strMsgid.Trim());

                            sbHtmlStatus.Append("第 " + j + " 則 " + strMsgid.Trim() + " => [" + nResult + "]" +
                                m_sns.GetRespDesc().Trim() + "<br/>");
                        }
                    }
                }
            }
        }
        else {
            sbHtmlStatus.Append "[" + nResult + "]" + m_sns.GetRespDesc();
        }

        ResultQryMultiMessageStatus.Text = sbHtmlStatus.ToString();

        m_sns.Logout();
    }

    //保持連線測試
    protected void EnquireLink_Click(object sender, EventArgs e)
    {
        m_sns.InitRemoteServer(ip.Text, port.Text, account.Text, passwd.Text);

        int nResult = m_sns.Login();
    }
}

```

```

        if (nResult == 0) {
            nResult = m_sns.EnquireLink();
        }

        ResultEnquireLinkStatus.Text = "[" + nResult + "]" + m_sns.GetRespDesc();

        m_sns.Logout();
    }

    //接收手機上傳簡訊測試
    protected void GetMessage_Click(object sender, EventArgs e)
    {
        m_sns.InitRemoteServer(ip.Text, port.Text, account.Text, passwd.Text);

        int nResult = m_sns.Login();

        if (nResult == 0) {
            nResult = m_sns.GetMessage();

            if (nResult == 0) {
                ResultGetMessage.Text = "[" + nResult + "]" + m_sns.GetFromMsisdn() + " | "
                    + m_sns.GetToMsisdn() + " | " + m_sns.GetRecvSMS();
            } else {
                ResultGetMessage.Text = "[" + nResult + "]" + m_sns.GetRespDesc();
            }
        } else {
            ResultGetMessage.Text = "[" + nResult + "]" + m_sns.GetRespDesc();
        }

        m_sns.Logout();
    }

    //切離連線測試
    protected void Logout_Click(object sender, EventArgs e)
    {
        m_sns.Logout();

        ResultLogout.Text = "(無回傳值)";
    }
}
}

```

以下為透過 ASP.NET 網頁範例執行 SNS 連線，並一次發送長簡訊至兩個門號，以及切線之範例：

受訊門號清單 (多個門號之間以 ; 符號分隔) :

09XXXXXXXX; 09XXXXXXXX

傳送訊息內容 :

此第三版 SNS COM+ 元件完整支援 98 年版 SNS (Special Number Service, 簡訊特碼服務) 通訊協定實作, 提供中華電信客戶透過此元件登入 SNS、發送簡訊、傳送長簡訊、查詢簡訊狀態、接收簡訊、保持連線、變更密碼、以及登出等功能。

SubmitMultiMessage

09XXXXXXXX (長簡訊序號 110, 含 2 則簡訊):

第 1 則 [0] 5F24332B

第 2 則 [0] 7024332B

09XXXXXXXX (長簡訊序號 111, 含 2 則簡訊):

第 1 則 [0] 7724332B

第 2 則 [0] 0825332B

註：若該批發送不含重覆門號者，多個門號也可使用同一個長簡訊序號發送。

以下為透過 ASP.NET 網頁範例執行 SNS 連線，查詢兩個門號之長簡訊發送結果，以及切線之範例：

多個門號之狀態查詢測試

受訊門號及簡訊識別碼清單 (多個資料以 ; 符號分隔) :

09XXXXXXXX (5F24332B, 7024332B); 09XXXXXXXX (7724332B, 0825332B)

QueryMultiMessageStatus

09XXXXXXXX:

第 1 則 5F24332B => [0] Successful:20170508172429

第 2 則 7024332B => [0] Successful:20170508172437

09XXXXXXXX:

第 1 則 7724332B => [0] Successful:20170508172437

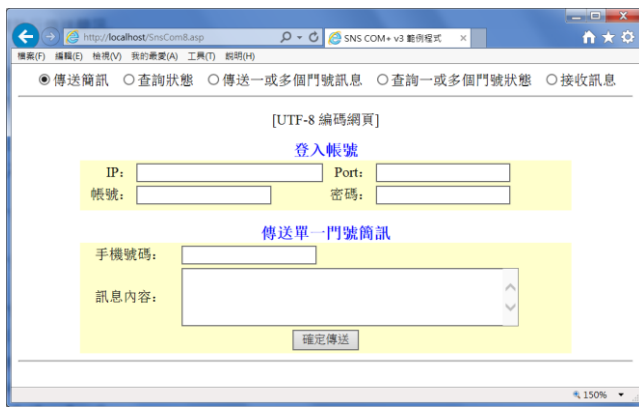
第 2 則 0825332B => [0] Successful:20170508172438

在您自行開發的 ASP.NET 簡訊應用網頁，在專案內需要包括此 SNS COM+ v3 DLL 元件，並請參考前述第 5.1 章節後述之注意事項、及一般實務用法建議說明。

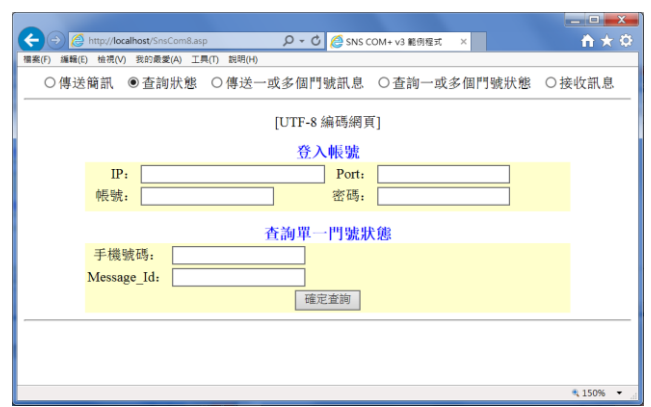
5.4 ASP 網頁範例程式

本範例相關程式碼可參考“ASP 網頁範例.zip”壓縮檔。

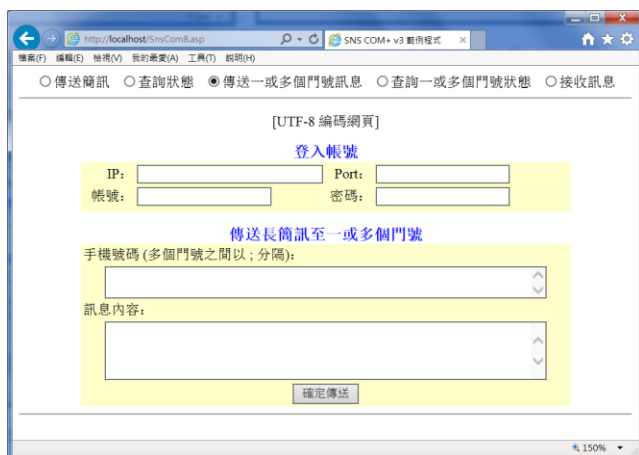
其參考相關 ASP 網頁範例畫面如下，當使用者完成輸入資料後及按鍵送出，即會經由 SNS COM+ V3 之作業回應結果輸出至該鍵下方，其相關畫面如同前述章節 ASP.NET 網頁範例者：



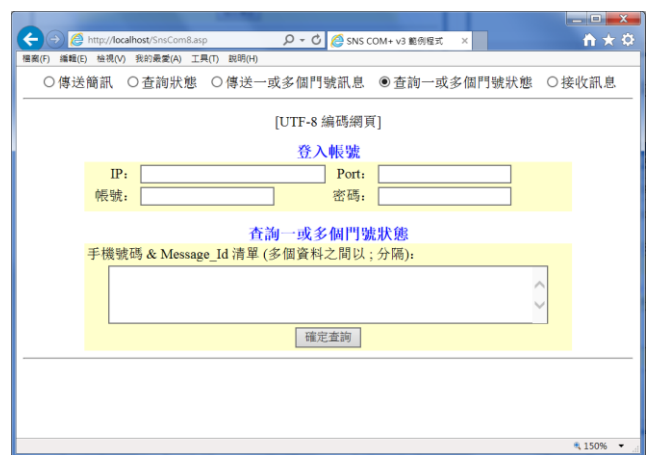
傳送簡訊至單一門號



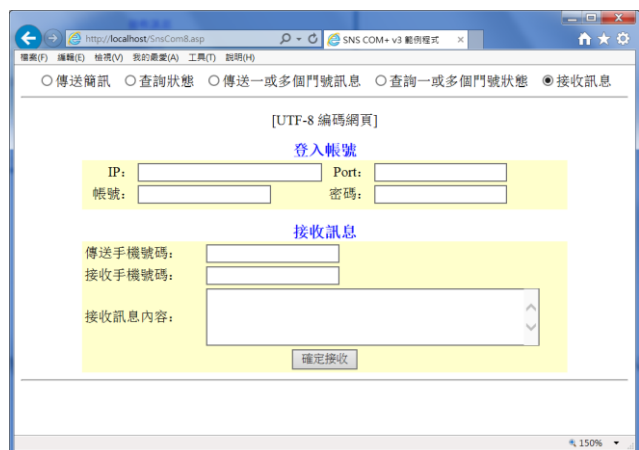
查詢簡訊狀態



傳送含一或多則簡訊之長簡訊至多個門號



查詢多個門號多則簡訊之狀態



接收手機上傳簡訊

以下為完整的 ASP 網頁程式碼範例，執行每個作業皆包含建立、及切離 SNS 連線。

<!DOCTYPE html>

```

<html lang="zh">
<head>
<meta http-equiv="Content-Language" content="zh-tw"/>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<script language="JavaScript">
<!--
function Show(n){
    if(n == 1){
        document.all["SendForm"].style.display="block";
        document.all["QueryForm"].style.display="none";
        document.all["SendMultiForm"].style.display="none";
        document.all["QueryMultiForm"].style.display="none";
        document.all["RecvForm"].style.display="none";
    } else if(n == 2){
        document.all["SendForm"].style.display="none";
        document.all["QueryForm"].style.display="block";
        document.all["SendMultiForm"].style.display="none";
        document.all["QueryMultiForm"].style.display="none";
        document.all["RecvForm"].style.display="none";
    } else if(n == 3){
        document.all["SendForm"].style.display="none";
        document.all["QueryForm"].style.display="none";
        document.all["SendMultiForm"].style.display="block";
        document.all["QueryMultiForm"].style.display="none";
        document.all["RecvForm"].style.display="none";
    } else if(n == 4){
        document.all["SendForm"].style.display="none";
        document.all["QueryForm"].style.display="none";
        document.all["SendMultiForm"].style.display="none";
        document.all["QueryMultiForm"].style.display="block";
        document.all["RecvForm"].style.display="none";
    } else if(n == 5){
        document.all["SendForm"].style.display="none";
        document.all["QueryForm"].style.display="none";
        document.all["SendMultiForm"].style.display="none";
        document.all["QueryMultiForm"].style.display="none";
        document.all["RecvForm"].style.display="block";
    }
}
//-->
</script>
<title>SNS COM+ v3 範例程式</title>
<meta http-equiv="Content-Type" content="text/html; charset=big5">
</head>
<body>
<%
    Dim Func
    Dim SendPhone, Message, QryPhone, MsgId, SendMsisdn, RecvMsisdn, RecvMsg
    Dim SendMultiPhone, strLongText, QryMultiPhoneMsgId

```

'宣告一物件變數為 CHT.SNSComServer 元件

'若瀏覽器出現下列畫面訊息:

```

'
' 伺服器物件 錯誤 'ASP 0177 : 8000ffff'
' Server.CreateObject 失敗
' /ASP/SnsCom8.asp, 行 48
' '8000ffff

```

```
'
'解決方式:
' 1. 若尚未安裝 SnsComServer3.dll 檔，請先執行 reg3.bat 檔註冊此元件;
' 2. 若已安裝及註冊 SnsComServer3.dll 檔，請參考 http://support.microsoft.com/kb/945701/en-us
' 加入機碼以排除問題
```

```
Set testCom = Server.CreateObject("CHT.SNSCOMSERVER.SnsComObject3")
Dim retCode_1
Dim retCode_2
```

```
Dim SNS_IP, SNS_Port, Account, Password
Dim ResponseMsg
Dim ShowSendForm, ShowQryForm, ShowRecvForm
```

```
'預設登入 IP 為 203.66.172.133, Port 為 8001
SNS_IP = request("IP")
SNS_Port = request("Port")
```

```
'輸入您的帳號和密碼
Account = request("Account")
Password = request("Password")
```

```
'預設為傳送訊息的畫面
ShowSendForm = "display:block"
ShowQryForm = "display:none"
ShowSendMultiForm = "display:none"
ShowQryMultiForm = "display:none"
ShowRecvForm = "display:none"
```

```
'依使用者點選功能
Func = request("Func")
```

```
Dim nExpiredMinutes
```

```
nExpiredMinutes = 1440 '預設重送期限為一天 1440 分鐘
```

```
ResponseMsg = "" '傳送訊息或查詢狀態的結果
```

```
If Func = "1" Then '選擇傳送單一門號簡訊的功能
    SendPhone = request("SendPhoneNum")
    Message = request("Message")
```

```
'登入 SNS Server
testCom.InitRemoteServer SNS_IP, SNS_Port, Account, Password
retCode_1 = testCom.Login()
```

```
'如果登入成功,才繼續執行傳送簡訊
```

```
If retCode_1 = 0 Then
    '傳送手機號碼, 訊息內容
    retCode_2 = testCom.SubmitMessage(null, SendPhone, Message, nExpiredMinutes)
```

```
'依據回應碼判斷傳送結果
Select Case retCode_2
    Case -2
        ResponseMsg = "參數錯誤:" & testCom.GetRespDesc()
    Case -1
```

```

        ResponseMsg = "連線失敗:" & testCom.GetRespDesc()
    Case 0
        ResponseMsg = "傳送成功;Message_id=" & testCom.GetRespDesc()
        '查詢狀態中的手機號碼預設為傳送手機號碼
        QryPhone = SendPhone
        '查詢狀態中的 Message_Id 預設為回傳的訊息 · 以備查詢
        MsgId = testCom.GetRespDesc()
    Case Else
        ResponseMsg = "傳送失敗(" & retCode_2 & ");原因=" & testCom.GetRespDesc()
    End Select

    '登出
    testCom.Logout
Else '登入失敗
    ResponseMsg = "Login fail :" & testCom.GetRespDesc()
End If

'控制在傳送訊息的畫面
ShowSendForm = "display:block"
ShowQryForm = "display:none"
ShowSendMultiForm = "display:none"
ShowQryMultiForm = "display:none"
ShowRecvForm = "display:none"
End If

If Func = "2" Then '選擇查詢單一門號狀態的功能

    QryPhone = request("QryPhoneNum")
    MsgId = request("MsgId")

    '登入 SNS Server
    testCom.InitRemoteServer SNS_IP, SNS_Port, Account, Password
    retCode_1 = testCom.Login()

    '如果登入成功,才繼續執行簡訊狀態查詢
    If retCode_1 = 0 Then
        '查詢手機號碼, Message_Id
        retCode_2 = testCom.QueryMessageStatus(null, QryPhone, MsgId)

        '依據回應碼判斷查詢結果
        If retCode_2 = -2 Then
            ResponseMsg = "參數錯誤 :" & testCom.GetRespDesc()
        ElseIf retCode_2 = -1 Then
            ResponseMsg = "連線失敗 :" & testCom.GetRespDesc()
        Else
            ResponseMsg = "訊息代碼=" & retCode_2 & ";狀態=" & testCom.GetRespDesc()
        End If

        '登出
        testCom.Logout
    Else '登入失敗
        ResponseMsg = "Login fail :" & testCom.GetRespDesc()
    End If

    '控制在查詢狀態的畫面

```



```

ShowSendForm = "display:none"
ShowQryForm = "display:block"
ShowSendMultiForm = "display:none"
ShowQryMultiForm = "display:none"
ShowRecvForm = "display:none"
End If

If Func = "3" Then    '選擇傳送簡訊/長簡訊至一或多個門號的功能
    SendMultiPhone = request("SendMultiPhoneNum")
    strLongText = request("MultiMessage")

    '登入 SNS Server
    testCom.InitRemoteServer SNS_IP, SNS_Port, Account, Password
    retCode_1 = testCom.Login()

    '如果登入成功,才繼續執行傳送簡訊
    If retCode_1 = 0 Then
        '總則數
        Dim bEnglish, bUDHI, nCurrSplit, nTotalSplit
        Dim i, saToMsisdn, sastrSplitSMS, strFromMsisdn, nSerno

        bEnglish = testCom.IsPureEnglishMessage(strLongText)
        bUDHI = True
        nTotalSplit = testCom.GetTotalSplit(strLongText, bUDHI, bEnglish)
        strFromMsisdn = null

        nSerno = Fix(255 * Rnd) + 1

        '多個受訊門號項目
        saToMsisdn = Split(SendMultiPhone, ";")

        QryMultiPhoneMsgId = ""

        for i = 0 to UBound(saToMsisdn)
            If Len(saToMsisdn(i)) > 0 Then
                ResponseMsg = ResponseMsg & "<br/>" & saToMsisdn(i) + ":"

                If Len(QryMultiPhoneMsgId) > 0 Then
                    QryMultiPhoneMsgId = QryMultiPhoneMsgId & ";"
                End If

                QryMultiPhoneMsgId = QryMultiPhoneMsgId + saToMsisdn(i) & "("

                '傳送分則簡訊
                for nCurrSplit = 1 to nTotalSplit
                    '取得各則簡訊
                    strSplitSMS = testCom.GetSplitSMS(strLongText, bUDHI, nCurrSplit, nTotalSplit, bEnglish)

                    '傳送手機號碼, 訊息內容
                    retCode_2 = testCom.SubmitLongMessage(strFromMsisdn, saToMsisdn(i), bUDHI, bEnglish, strSplitSMS, _
                        nSerno, nCurrSplit, nTotalSplit, nExpiredMinutes)

                    ResponseMsg = ResponseMsg & "<br/>" & "第 " & nCurrSplit & " 則 =&gt;"

                '依據回應碼判斷傳送結果
                Select Case retCode_2
                    Case -2
                        ResponseMsg = ResponseMsg & "參數錯誤:" & testCom.GetRespDesc()

```

```

Case -1
    ResponseMsg = ResponseMsg & "連線失敗:" & testCom.GetRespDesc()
Case 0
    '查詢狀態中的手機號碼預設為傳送手機號碼
    QryPhone = SendPhone
    '查詢狀態中的 Message_Id 預設為回傳的訊息，以備查詢
    MsgId = testCom.GetRespDesc()

    If nCurrSplit > 1 Then
        QryMultiPhoneMsgId = QryMultiPhoneMsgId & ","
    End If

    QryMultiPhoneMsgId = QryMultiPhoneMsgId & MsgId

    ResponseMsg = ResponseMsg & "傳送成功;Message_id=" & testCom.GetRespDesc()
Case Else
    ResponseMsg = ResponseMsg & "傳送失敗(" & retCode_2 & ");原因=" & testCom.GetRespDesc()
End Select
next

    QryMultiPhoneMsgId = QryMultiPhoneMsgId & ")"
End If
next

'登出
testCom.Logout
Else '登入失敗
    ResponseMsg = "Login fail :" & testCom.GetRespDesc()
End If

'控制在傳送訊息的畫面
ShowSendForm = "display:none"
ShowQryForm = "display:none"
ShowSendMultiForm = "display:block"
ShowQryMultiForm = "display:none"
ShowRecvForm = "display:none"
End If

If Func = "4" Then '選擇查詢多個門號分則簡訊狀態的功能

    QryMultiPhoneMsgId = request("QryMultiPhoneMsgId")

    '登入 SNS Server
    testCom.InitRemoteServer SNS_IP, SNS_Port, Account, Password
    retCode_1 = testCom.Login()

    '如果登入成功,才繼續執行簡訊狀態查詢
    If retCode_1 = 0 Then
        Dim saQryInfo

        saQryInfo = Split(QryMultiPhoneMsgId, ";")

        for i = 0 to UBound(saQryInfo)
            If Len(saQryInfo(i)) > 0 Then
                Dim saField, j

                saField = Split(saQryInfo(i), "(")

```

```

If UBound(saField) = 1 Then
    Dim saMsgid

    QryPhone = saField(0)
    saMsgid = Split(Mid(saField(1), 1, Len(saField(1))-1), ",")

    ResponseMsg = ResponseMsg & "<br/>" & QryPhone & ":"

    for j = 0 to UBound(saMsgid)
        Msgid = saMsgid(j)
        If Len(Msgid) > 0 Then
            '查詢手機號碼, Message_Id
            retCode_2 = testCom.QueryMessageStatus(null, QryPhone, Msgid)

            ResponseMsg = ResponseMsg & "<br/>第 " & (j+1) & "則 " & Msgid & "=> "

            '依據回應碼判斷查詢結果
            If retCode_2 = -2 Then
                ResponseMsg = ResponseMsg & "參數錯誤 :" & testCom.GetRespDesc()
            ElseIf retCode_2 = -1 Then
                ResponseMsg = ResponseMsg & "連線失敗 :" & testCom.GetRespDesc()
            Else
                ResponseMsg = ResponseMsg & "訊息代碼=" & retCode_2 & ";狀態=" & testCom.GetRespDesc()
            End If
        End If
    next
End If

'登出
testCom.Logout
Else '登入失敗
    ResponseMsg = "Login fail :" & testCom.GetRespDesc()
End If

'控制在查詢狀態的畫面
ShowSendForm = "display:none"
ShowQryForm = "display:none"
ShowSendMultiForm = "display:none"
ShowQryMultiForm = "display:block"
ShowRecvForm = "display:none"
End If

If Func = "5" Then '選擇接收訊息的功能

    '登入 SNS Server
    testCom.InitRemoteServer SNS_IP, SNS_Port, Account, Password
    retCode_1 = testCom.Login()

    '如果登入成功,才繼續執行簡訊狀態查詢
    If retCode_1 = 0 Then
        '呼叫接收訊息
        retCode_2 = testCom.GetMessage()

        '依據回應碼判斷傳送結果
        Select Case retCode_2

```



```

        <td width="44%" align="left"><input type="text" name="Port" size="20" value="<%=SNS_Port%>"></td>
    </tr>
    <tr>
        <td width="15%" align="right">帳號 : </td>
        <td width="25%" align="left"><input type="text" name="Account" size="20" value="<%=Account%>"></td>
        <td width="14%" align="right">密碼 : </td>
        <td width="44%" align="left"><input type="text" name="Password" size="20" value="<%=Password%>"></td>
    </tr>
    <tr>
        <td width="99%" align="center" colspan="4"></td>
    </tr>
</table>
<p>

```

```

<div id="SendForm" style="<%=ShowSendForm%>">
    <table border="0" width="80%" align="center" style="background-color: #FFFFCC">
        <caption><font size="4" color="#0000FF"><b>傳送單一門號簡訊</b></font></caption>
        <tr align="center">
            <td width="20%">手機號碼 : </td>
            <td width="80%" align="left"><input type="text" name="SendPhoneNum" size="20"></td>
        </tr>
        <tr align="center">
            <td width="20%">訊息內容 : </td>
            <td width="80%" align="left"><textarea name="Message" rows="4" cols="45"></textarea></td>
        </tr>
        <tr align="center">
            <td width="100%" align="center" colspan="2"><input type="submit" value="確定傳送" name="submit"></td>
        </tr>
    </table>
</div>

```

```

<div id="QueryForm" style="<%=ShowQryForm%>">
    <table border="0" width="80%" align="center" style="background-color: #FFFFCC">
        <caption><font color="#0000FF" size="4"><b>查詢單一門號狀態</b></font></caption>
        <tr align="center">
            <td width="15%">手機號碼 : </td>
            <td width="85%" align="left"><input type="text" name="QryPhoneNum" size="20" value="<%=QryPhone%>"></td>
        </tr>
        <tr align="center">
            <td width="15%">Message_Id : </td>
            <td width="85%" align="left"><input type="text" name="MsgId" size="20" value="<%=MsgId%>"></td>
        </tr>
        <tr align="center">
            <td width="100%" align="center" colspan="2"><input type="submit" value="確定查詢" name="submit"></td>
        </tr>
    </table>
</div>

```

```

<div id="SendMultiForm" style="<%=ShowSendMultiForm%>">
    <table border="0" width="80%" align="center" style="background-color: #FFFFCC">
        <caption><font size="4" color="#0000FF"><b>傳送長簡訊至一或多個門號</b></font></caption>
        <tr>
            <td align="left">手機號碼 (多個門號之間以 ; 分隔) : </td>
        </tr>
        <tr>
            <td align="center"><textarea name="SendMultiPhoneNum" rows="2" cols="60"></textarea></td>
        </tr>
    </table>

```

```

<tr>
  <td align="left">訊息內容：</td>
</tr>
<tr>
  <td align="center"><textarea name="MultiMessage" rows="4" cols="60"></textarea></td>
</tr>
<tr>
  <td width="100%" align="center"><input type="submit" value="確定傳送" name="submit"></td>
</tr>
</table>
</div>

<div id="QueryMultiForm" style="<%=ShowQryMultiForm%>">
  <table border="0" width="80%" align="center" style="background-color: #FFFFCC">
    <caption><font color="#0000FF" size="4"><b>查詢一或多個門號狀態</b></font></caption>
    <tr>
      <td align="left">手機號碼 & Message_Id 清單 (多個資料之間以 ; 分隔)：</td>
    </tr>
    <tr>
      <td align="center"><textarea name="QryMultiPhoneMsgid" rows="4"
cols="60"><%=QryMultiPhoneMsgid%></textarea></td>
    </tr>
    <tr>
      <td width="100%" align="center"><input type="submit" value="確定查詢" name="submit"></td>
    </tr>
  </table>
</div>

<div id="RecvForm" style="<%=ShowRecvForm%>">
  <table border="0" width="80%" align="center" style="background-color: #FFFFCC" id="table1">
    <caption><font size="4" color="#0000FF"><b>接收訊息</b></font></caption>
    <tr>
      <td width="25%">傳送手機號碼：</td>
      <td width="75%" align="left"><input type="text" name="SendMsisdn" size="20" value="<%=SendMsisdn%>" readonly></td>
    </tr>
    <tr>
      <td width="25%">接收手機號碼：</td>
      <td width="75%" align="left"><input type="text" name="RecvMsisdn" size="20" value="<%=RecvMsisdn%>" readonly></td>
    </tr>
    <tr>
      <td width="25%">接收訊息內容：</td>
      <td width="75%" align="left"><textarea rows="4" name="RecvMsg" cols="45" readonly><%=RecvMsg%></textarea></td>
    </tr>
    <tr>
      <td width="100%" align="center" colspan="2"><input type="submit" value="確定接收" name="submit"></td>
    </tr>
  </table>
</div>

</form>
<hr>
<%
'輸出結果
If(ResponseMsg <> "") Then
  Response.Write(ResponseMsg)
End If
%>
</body>

```

</html>

在您自行開發的 ASP 簡訊應用網頁，其所在主機需要執行第一章之 reg3.bat 程式，以便安裝 SNS COM+ v3 DLL 元件，並請參考前述第 5.1 章節後述之注意事項、及一般實務用法建議說明。

6. 常見問題及說明解決

【問題一】 那些程式語言可以使用 SNS COM+ v3 元件，需要進行安裝程序？

- 說明：
- (1) 以本 API 說明文件已舉例者，SNS COM+ v3 元件可支援至 C#.NET、ASP.NET、及 ASP 等程式語言。
 - (2) 至於本文件未舉例者，其 SNS COM+ v3 元件也是應該可以另支援 VB.NET 等 Microsoft Visual Studio 相關開發語言使用，可參考本書第 5 章之相關範例，自行將 C#.NET 轉換至對應的 VB.NET 等程式語法。
 - (3) 參考本書第 1 章說明，只有 ASP 方要執行 reg3.bat 安裝指令，安裝其 SNS COM+ v3 DLL 檔至主機程序，以便匯入至 .NET Framework v2.0 平台；本作業只需執行一次即可。
 - (4) 參考本書第 1 章說明，其 C#.NET、ASP.NET、及相關 Microsoft Visual Studio 相關開發語言，因可在其專案直接加入 SNS COM+ v3 DLL 檔作參考，可不需執行 reg3.bat 安裝指令。