

網路通訊程式實例(三種程式語言)

James chou 2019/04

本文將介紹 Windows 下的 TCP/IP 網路通訊程式設計

程式使用以下三種程式語言, 五個範例

1. Visual Studio 2017 C#
 - (1) 使用內建 TcpClient / TcpListener 的範例 (範例一)
 - (2) 使用內建 Socket 的範例 (範例二)
2. Lazarus 2.0
 - (1) 使用外掛 INDY 元件的範例 (範例三)
3. RAD Studio 10.3.1 Rio C++ Builder
 - (1) 使用內建 INDY 元件的範例 (範例四)
 - (2) 使用 TClientSocket / TServerSocket 的範例 (範例五)

通訊架構設計構想如下 (本文所有範例都依這設計構想而撰寫)

1. Client 端每支程式只有一個 CLIENT 通訊通道 (若有多個 REQUEST 就排隊發送, 未使用 THREAD), 可多支程式一起執行
2. Server 端使用 Thread/Blocking 模式, 以多執行緒分別對應每個 CLIENT
3. 實作四大基本通訊功能
 - (1) SND_STR(): CLIENT 向 SERVER 端發送一組字串
 - (2) GET_STR(): CLIENT 向 SERVER 要求回復一組字串
 - (3) SND_FILE(): CLIENT 向 SERVER 端發送一個檔案
 - (4) GET_FILE(): CLIENT 向 SERVER 要求回傳檔案, 若只有一個檔案就直接回傳, 若有多個檔案就壓縮成一個檔案再回傳(CLIENT 端收到再解壓縮)
4. 所有的 REQUEST 都是由 CLIENT 發出, SERVER “不會主動” 發送資料給各個 CLIENT

四大基本通訊功能在 POS 程式設計應用上舉例如下

1. SND_STR(): 測試 SERVER 是否存在(是否連線中), 或 POS(CLIENT) 送出自身版本號碼 (送出即可, 不用等待回傳資料)
2. GET_STR(): POS(CLIENT) 跟後台 SERVER 查詢會員卡號, 退貨資料查詢, 取得 SERVER 時間字串來對時...等等, 要等待及接收回傳資料的相關應用
3. SND_FILE(): POS(CLIENT) 交易資料檔上傳
4. GET_FILE(): POS(CLIENT) 向後台 SERVER 取得配信資料(如商品主檔, 商品變價檔等等)

四大基本通訊功能的函式結構及資料結構

SND_STR 函式結構

```
String SND_STR(String command_id2, String command_id3, String  
command_text, String command_text2)
```

SND_STR 資料結構 (SERVER 端讀到第一碼為 “1” 則做 SND_STR 相關處理)

CLIENT 發出：

每段固定長度 1024 bytes					
command_id 固定為 “1”+ “\n”	command_id2 不定長字串+ “\n”	command_id3 不定長字串+ “\n”	command_text 不定長字串+ “\n”	command_text2 不定長字串+ “\n”	剩餘空字 串補足 1024 bytes

CLIENT 收回：無

舉例：CLIENT 發出 `SND_STR("1234", "POS_MSG", "HELLO", "");` //向 SERVER 送出一個
“HELLO”

GET_STR 函式結構

```
String GET_STR(String command_id2, String command_id3, String  
command_text, String command_text2)
```

GET_STR 資料結構 (SERVER 端讀到第一碼為 "2" 則做 GET_STR 相關處理)

CLIENT 發出 :

每段固定長度 1024 bytes					
command_id 固定為 "2"+ "\n"	command_id2 不定長字串+ "\n"	command_id3 不定長字串+ "\n"	command_text 不定長字串+ "\n"	command_text2 不定長字串+ "\n"	剩餘空字 串補足 1024 bytes

CLIENT 收回 :

每段固定長度 1024 bytes
不定長字串 + 補滿 1024 BYTE 的空資料

舉例 : CLIENT 發出 GET_STR("1234", "SRV_TIME", "", ""); //取得後台時間

SND_FILE 函式結構

String SND_FILE(String command_id2, String command_id3, String command_text, String command_text2)

SND_FILE 資料結構 (SERVER 端讀到第一碼為 "3" 則做 SND_FILE 相關處理)

CLIENT 發出 :

每段固定長度 1024 bytes					
command_id 固定為 "3"+ "\n"	command_id2 不定長字串+ "\n" 通常放檔名	command_id3 不定長字串+ "\n" 通常放檔案 SIZE 字串	command_text 不定長字串+ "\n"	command_text2 不定長字串+ "\n"	剩餘空字 串補足 1024 bytes
1024 bytes 的檔案串流(Stream)資料 (第一段)					
1024 bytes 的檔案串流(Stream)資料 (第二段)					
.....					
1024 bytes 的檔案串流(Stream)資料 (第 N 段)					
檔案尾段不足 1024 bytes 的剩餘串流(Stream)資料 + 補足 1024 bytes 的空資料					

CLIENT 收回 : 無

舉例 : CLIENT 發出

```
SND_FILE("1234", "", "C:\\ABACUS\\1234.SAL", "C:\\POSSVR\\DAT\\WORK\\1234.SAL");
```

把 CLIENT 端的 C:\\ABACUS\\1234.SAL 送到 SERVER 端的
"C:\\POSSVR\\DAT\\WORK\\1234.SAL"

GET_FILE 函式結構

String GET_FILE(String command_id2, String command_id3, String command_text, String command_text2)

GET_FILE 資料結構 (SERVER 端讀到第一碼為 "4" 則做 GET_FILE 相關處理)

CLIENT 發出 :

每段固定長度 1024 bytes					
command_id 固定為 "4"+ "\n"	command_id2 不定長字串+ "\n"	command_id3 不定長字串+ "\n"	command_text 不定長字串+ "\n"	command_text2 不定長字串+ "\n"	剩餘空字 串補足 1024 bytes

CLIENT 收回 :

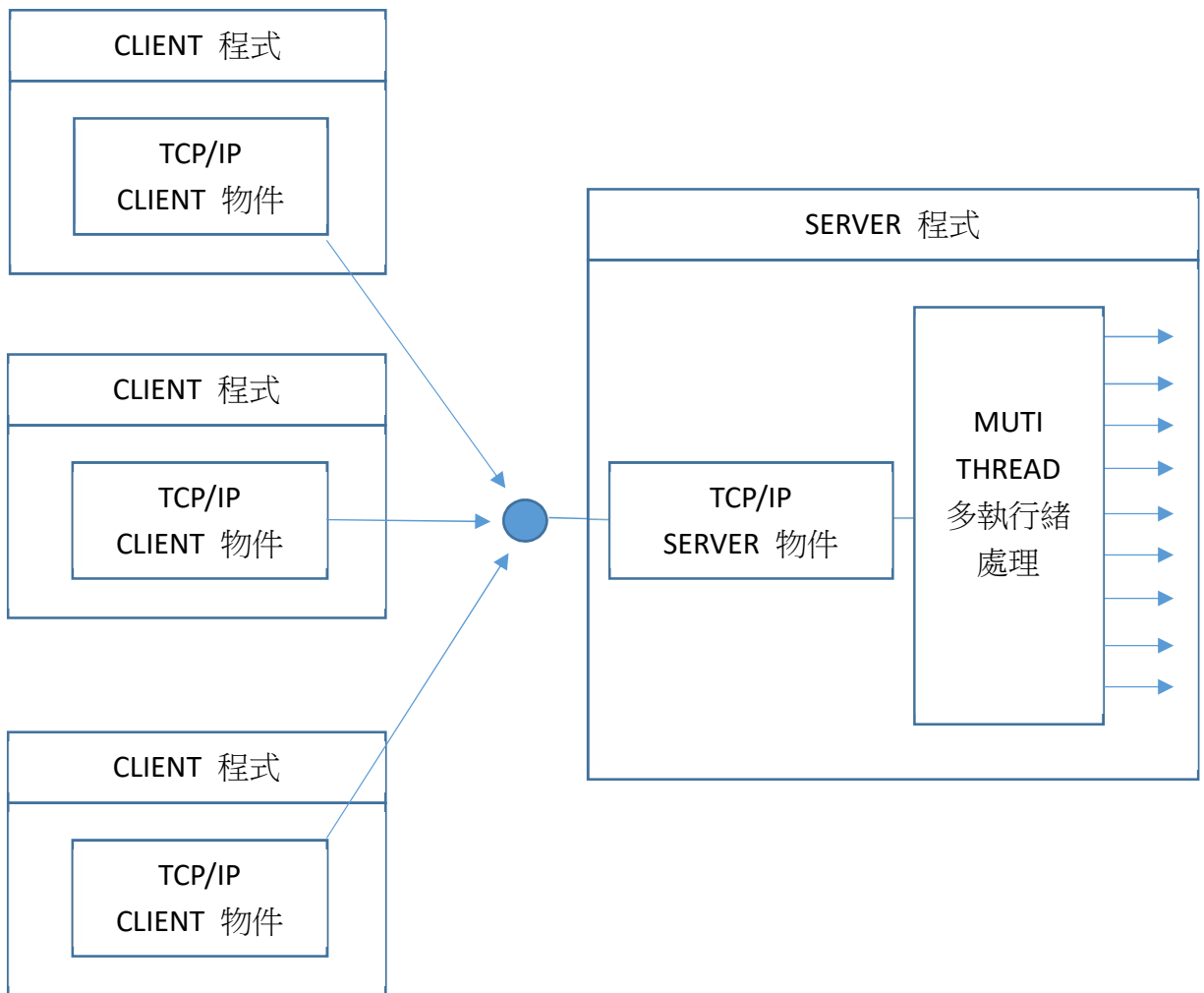
每段固定長度 1024 bytes		
回傳檔名(不 定長字串)+ "\n"	檔案 SIZE (不 定長字串)+ "\n"	補足 1024 bytes 的空資料
1024 bytes 的檔案串流(Stream)資料 (第一段)		
1024 bytes 的檔案串流(Stream)資料 (第二段)		
.....		
1024 bytes 的檔案串流(Stream)資料 (第 N 段)		
檔案尾段不足 1024 bytes 的剩餘串流(Stream)資料 + 補足 1024 bytes 的空資料		

舉例 : CLIENT 發出

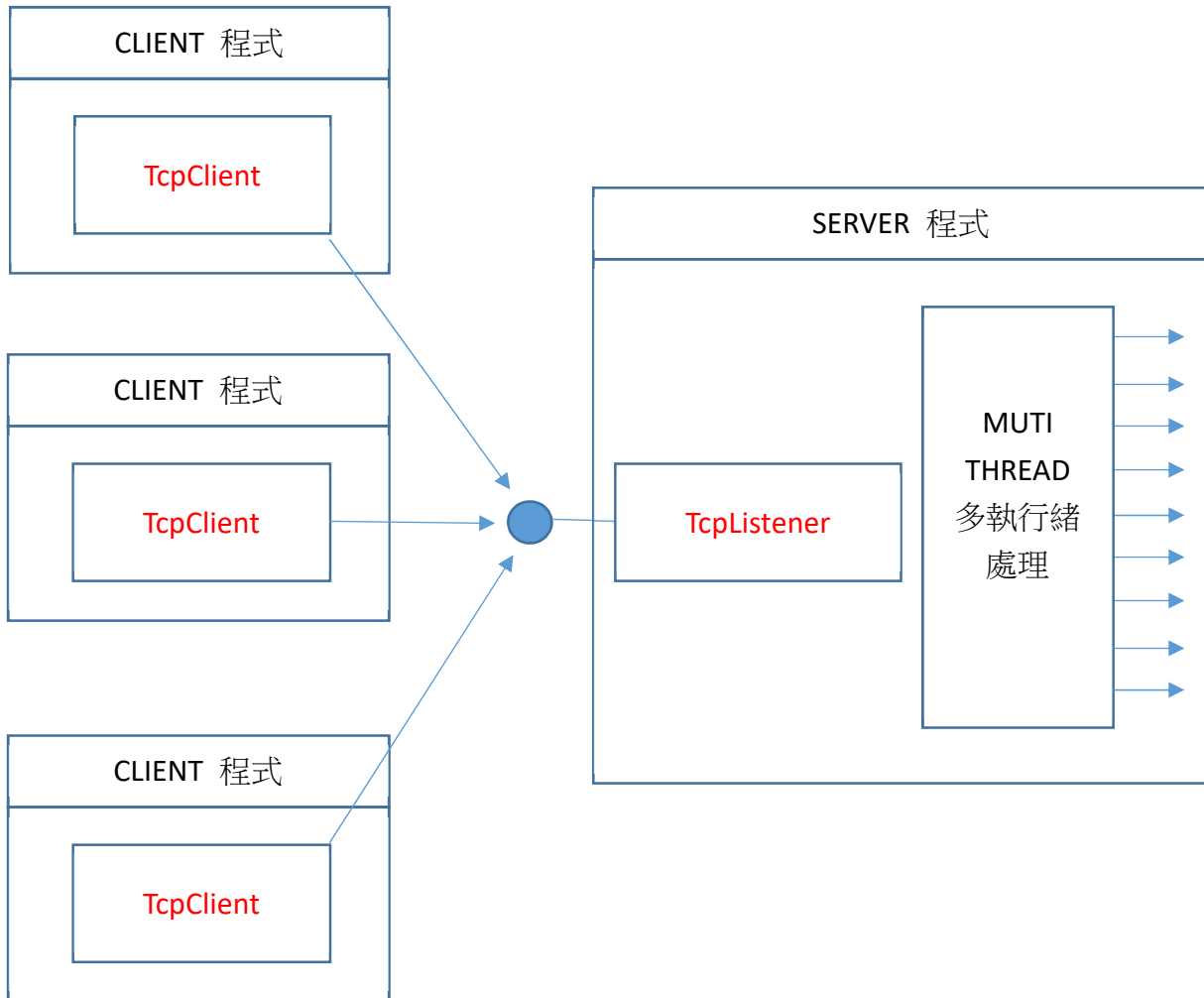
```
GET_FILE("1234", "", "C:\\POSSVR\\BIN\\POS\\*.\"", "c:\\_temp");
```

把 SERVER 端的 C:\\POSSVR\\BIN\\POS*.\" 抓回到 CLIENT 端的 "c:_temp\\" 目錄下

通訊元件架構設計



範例一 Visual Studio 2017 C# (使用內建 TcpClient / TcpListener)



[CLIENT 端]

兩個公有變數

```
public partial class Form1 : Form
{
    TcpClient MyTCPClient1;
    NetworkStream MyStream1;
    .....
    .....
}
```

//使用例 : SND_STR("1234", "POS_MSG", "HELLO", ""); //向 SERVER 送出一個 "HELLO"

```
String SND_STR(String command_id2, String command_id3, String command_text,
String command_text2)
{
    String result = String.Empty;
    MemoLog("●SND_STR(" + command_id2 + "," + command_id3 + "," +
command_text + "," + command_text2 + ") 開始");

    try
    {
        MyTCPClient1 = new TcpClient(textBox2.Text, 6501); //textBox2.Text 為 IP
        MyStream1 = MyTCPClient1.GetStream(); // 取得client stream
        MyStream1.ReadTimeout = 1000;

        String tmp = "1" + "\n" + //1:SND_STR 2:GET_STR 3:SND_FILE 4:GET_FILE
            command_id2 + "\n" +
            command_id3 + "\n" +
            command_text + "\n" +
            command_text2 + "\n";

        Byte[] data = System.Text.Encoding.Unicode.GetBytes(tmp);
        Byte[] data2 = new byte[1024];
        data.CopyTo(data2, 0); //保持 1024 長度
        MyStream1.Write(data2, 0, data2.Length);
    }
    catch (Exception ex)
    {
        MemoLog(ex.Message);
    }
    finally
    {
        if (MyStream1 != null) MyStream1.Close();
        if (MyTCPClient1 != null) MyTCPClient1.Close();
        MemoLog("○SND_STR() 結束");
    }

    return (result);
}
```



```

//使用例 : GET_STR("1234", "SRV_TIME", "", ""); //取得後台時間

String GET_STR(String command_id2, String command_id3, String command_text,
String command_text2)
{
    String result = String.Empty;
    MemoLog("●GET_STR(" + command_id2 + "," + command_id3 + "," +
command_text + "," + command_text2 + ") 開始");

    try
    {
        MyTCPClient1 = new TcpClient(textBox2.Text, 6501);
        MyStream1 = MyTCPClient1.GetStream(); // 取得client stream
        MyStream1.ReadTimeout = 1000;

        //發送字串之處理
        String tmp = "2" + "\n" + //1:SEND_STR 2:GET_STR 3:SEND_FILE 4:GET_FILE
            command_id2 + "\n" +
            command_id3 + "\n" +
            command_text + "\n" +
            command_text2 + "\n";

        //將 byte[] 寫入 MyStream1
        Byte[] data = System.Text.Encoding.Unicode.GetBytes(tmp);
        Byte[] data2 = new byte[1024];
        data.CopyTo(data2, 0); //保持 1024 長度
        MyStream1.Write(data2, 0, data2.Length);

        //讀回資料
        int count = 0;
        byte[] readBuffer = new byte[1024];
        if ((count = MyStream1.Read(readBuffer, 0, readBuffer.Length)) != 0)
        {
            result = Encoding.Unicode.GetString(readBuffer, 0, count);
        }
        MemoLog("接收到回傳字串 : '" + result + "'");
    }
    catch (Exception ex)
    {
        MemoLog(ex.Message);
    }
    finally
    {
        if (MyStream1 != null) MyStream1.Close();
        if (MyTCPClient1 != null) MyTCPClient1.Close();
        MemoLog("○GET_STR() 結束");
    }

    return (result);
}

```

```

String SND_FILE(String command_id2, String command_id3, String command_text,
String command_text2)
{
    String result = String.Empty;
    MemoLog("●SND_FILE(" + command_id2 + "," + command_id3 + "," +
command_text + "," + command_text2 + ") 開始");

    FileStream fstream = null; //using System.IO
    int filesize;
    int readcount;
    int fileoffset;

    try
    {
        fstream = new FileStream(command_text, FileMode.Open, FileAccess.Read);
        filesize = (int)fstream.Length;
        fileoffset = 0;

        MyTCPClient1 = new TcpClient(textBox2.Text, 6501);
        MyStream1 = MyTCPClient1.GetStream(); // 取得client stream
        MyStream1.ReadTimeout = 1000;
        //發送字串之處理
        String tmp = "3" + "\n" + //1:SND_STR 2:GET_STR 3:SND_FILE 4:GET_FILE
            command_id2 + "\n" +
            filesize.ToString() + "\n" +
            command_text + "\n" +
            command_text2+ "\n";

        //將 byte[] 寫入 MyStream1
        Byte[] data = System.Text.Encoding.Unicode.GetBytes(tmp);
        Byte[] data2 = new byte[1024];
        data.CopyTo(data2, 0); //保持 1024 長度
        MyStream1.Write(data2, 0, data2.Length);
        //將檔案 stream 寫入 MyStream1
        fstream.Position = 0;
        while (fstream.Position < filesize) {
            if ((filesize - fstream.Position) >= data2.Length)
                readcount = data2.Length;
            else
                readcount = filesize - (int)fstream.Position;

            fstream.Read(data2, 0, readcount);
            MyStream1.Write(data2, 0, readcount);
        }
        fstream.Close();
    }
    catch (Exception ex)
    {
        MemoLog(ex.Message);
    }
    finally
    {
        if (fstream != null) fstream.Close();
        if (MyStream1 != null) MyStream1.Close();
        if (MyTCPClient1 != null) MyTCPClient1.Close();
        MemoLog("○SND_FILE() 結束");
    }
    return (result);
}

```

```

String GET_FILE(String command_id2, String command_id3, String command_text,
String command_text2)
{
    String result = String.Empty;
    MemoLog("●GET_FILE(" + command_id2 + "," + command_id3 + "," +
command_text + "," + command_text2 + ") 開始");
    FileStream fstream = null; //using System.IO
    String filename = Path.GetFileName(command_text);
    int filesize=0;
    int readcount=0;
    int fileoffset=0;
    String get_file_path="";
    byte[] readBuffer = new byte[1024];
    int count = 0;

    try
    {
        MyTCPClient1 = new TcpClient(textBox2.Text, 6501);
        MyStream1 = MyTCPClient1.GetStream(); // 取得client stream
        MyStream1.ReadTimeout = 1000;

        //發送字串之處理
        String tmp = "4" + "\n" + //1:SEND_STR 2:GET_STR 3:SEND_FILE 4:GET_FILE
                    command_id2 + "\n" +
                    command_id3 + "\n" +
                    command_text + "\n" +
                    command_text2 + "\n";

        //將 byte[] 寫入 MyStream1
        Byte[] data = System.Text.Encoding.Unicode.GetBytes(tmp);
        Byte[] data2 = new byte[1024];
        data.CopyTo(data2, 0); //保持 1024 長度
        MyStream1.Write(data2, 0, data2.Length);

        //讀回
        if ((count = MyStream1.Read(readBuffer, 0, readBuffer.Length)) != 0)
        {
            tmp = Encoding.Unicode.GetString(readBuffer, 0, count);
            string[] lines = tmp.Split(new[] { "\r\n", "\r", "\n" },
StringSplitOptions.None);

            filename = lines[0];
            filesize = Int32.Parse(lines[1]);
        }

        //假如 Client 端已存在該檔案要先刪除
        get_file_path = command_text2;
        ForceDirectories(get_file_path);
        filename = get_file_path + "\\\" + Path.GetFileName(filename);
        if (File.Exists(filename)) File.Delete(filename);
    }
}

```

程式碼接下頁

```

//接收檔案
fstream = new FileStream(filename, FileMode.Create, FileAccess.Write);
while (filesize > fstream.Length)
{
    if ((filesize - fstream.Length) > readBuffer.Length)
        readcount = readBuffer.Length;
    else
        readcount = filesize - (int)fstream.Length;

    MyStream1.Read(readBuffer, 0, readcount);
    fstream.Write(readBuffer, 0, readcount);
}
fstream.Close();

//解壓縮檔案
if (Path.GetExtension(filename) == ".ZIP" ||
Path.GetExtension(filename) == ".zip")
{
    ZipFile.ExtractToDirectory(filename, command_text2);
    File.Delete(filename);
}

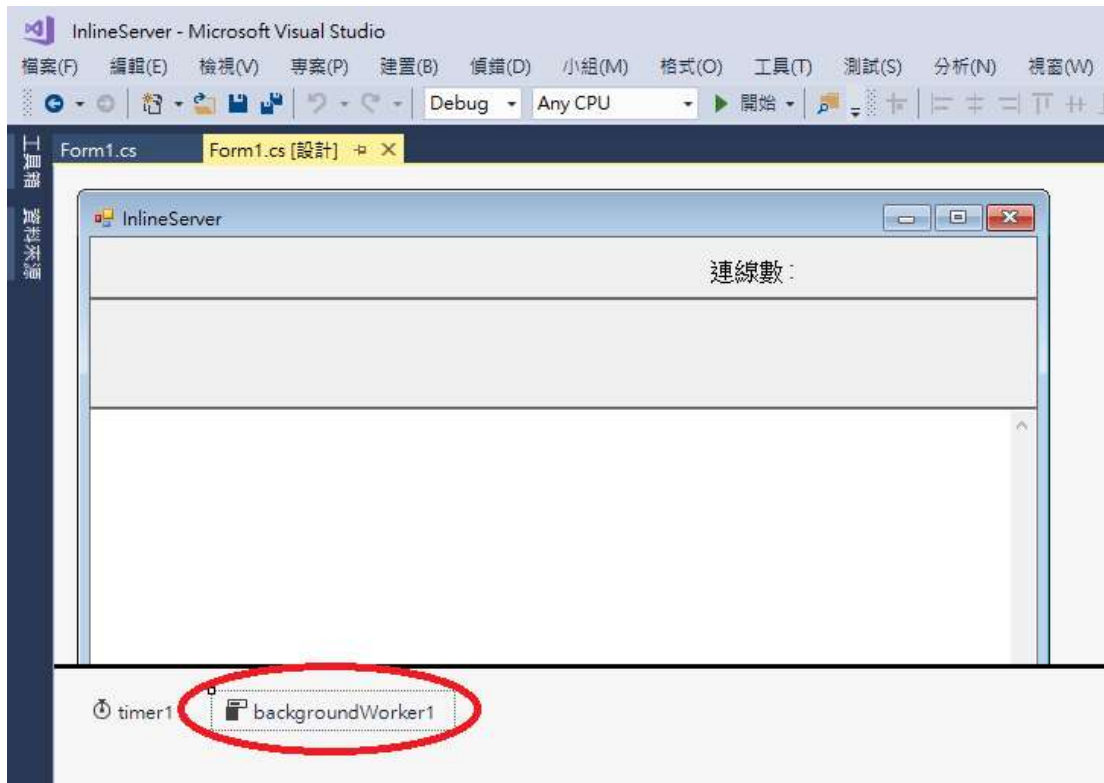
}
catch (Exception ex)
{
    MemoLog(ex.Message);
}
finally
{
    if (fstream != null) fstream.Close();
    if (MyStream1 != null) MyStream1.Close();
    if (MyTCPClient1 != null) MyTCPClient1.Close();
    MemoLog("◦GET_FILE() 結束");
}

return (result);
}

```

[SERVER 端]

1. 在 C# IDE 中, TcpClient / TcpListener 並非以元件(Control) 形式存在, 而是要自己去把該物件 new 出來
2. C# IDE 中有個特別的元件(Control) 叫做 backgroundWorker, 用來執行背景工作的, 我也是第一次看到這種元件(Control), 感覺它就是一個 THREAD, 以元件(Control) 形式存在, 讓你可以拖拉這個元件到設計畫面上, 然後就有一個 THREAD 可用



3. backgroundWorker 元件的使用方式

- (1) 呼叫 `backgroundWorker1.RunWorkerAsync()` =>
會觸發執行 `backgroundWorker1_DoWork()` 事件
- (2) 呼叫 `backgroundWorker1.ReportProgress()` =>
會觸發執行 `backgroundWorker1_ProgressChanged()` 事件
- (3) 在 SERVER 端程式中, 因啟動 TcpListener 監聽動作會 hold 住整個程式, 所以不能把監聽動作放在主執行緒中; 我把 TcpListener 監聽元件放在 `backgroundWorker1_DoWork()` 事件中, 然後在 `Form1_Load` 時呼叫 `backgroundWorker1.RunWorkerAsync()`, 就會觸發 `backgroundWorker1_DoWork()` 事件並在背景啟動 TcpListener 監聽動作
- (4) 在監聽過程所產生的 THREAD 若要更新主 Form GUI 時, 則透過呼叫 `backgroundWorker1.ReportProgress()`, 然後引發 `backgroundWorker1_ProgressChanged()` 事件, 在該事件中處理 GUI 相關動作

主執行緒

Form1_Load 時呼叫 backgroundWorker1.RunWorkerAsync()



backgroundWorker1

backgroundWorker1_DoWork()

```
{  
    建立 TcpListener 及相關監聽動作程序  
    若有監聽到 CLIENT 連入, 則為每一個 CLIENT 再建立一條執行緒  
}
```



follow 每一個 CLIENT 的自建執行緒 (THREAD)

TCPServer1Execute()

```
{  
    每一個 CLIENT 的對應處理  
}
```

Form Load 前的公用變數宣告, 及 Form Load 中的動作

```
int connection_count = 0; //client 連線數
private TcpListener _Listener; //宣告 Server 端 TcpListener 物件

private void Form1_Load(object sender, EventArgs e)
{
    backgroundWorker1.RunWorkerAsync(); //啟動背景工作元件
}
```

Form Closed 中的動作

```
private void Form1_FormClosed(object sender, FormClosedEventArgs e)
{
    if (backgroundWorker1.WorkerSupportsCancellation == true)
    {
        backgroundWorker1.CancelAsync();
    }

    if (_Listener != null)
    {
        _Listener.Stop();
    }
}
```

GUI 動作在此事件中處理

```
private void backgroundWorker1_ProgressChanged(object sender,
ProgressChangedEventArgs e)
{
    //e.ProgressPercentage 為 backgroundWorker1.ReportProgress() 傳入的參數
    label1.Text = "連線數 : " + e.ProgressPercentage.ToString();
}
```

在 `backgroundWorker1` 背景工作元件的 `DoWork()` 中 `new` 出 `_Listener` 物件，並在無窮迴圈中一直處理監聽動作，只要監聽到有 `Client` 連線進來，就馬上 `new` 出一條名為 `SckSAcceptTd` 的 `Thread` 與其對應

```
private void backgroundWorker1_DoWork(object sender, DoWorkEventArgs e)
{
    try
    {
        //若 TCP Listener 正在工作，則停止
        if (_Listener != null)
            _Listener.Stop();

        //初始化 TcpListener
        _Listener = new TcpListener(IPAddress.Any, 6501);

        //啟動 listener
        _Listener.Start();

        //===== 持續接受監聽 socket client 的連線 ===== (start)
        while (true)
        {
            //監聽到來自 socket client 的連線要求
            TcpClient socket4Client = _Listener.AcceptTcpClient();

            //調用ProgressChanged改變form元件數值 否則無法跨執行緒執行(刷新UI動作)
            connection_count++;
            backgroundWorker1.ReportProgress(connection_count);

            //可以傳入參數的 thread
            Thread SckSAcceptTd = new Thread(new
            ParameterizedThreadStart(TCPServer1Execute));
            SckSAcceptTd.IsBackground = true;

            //開始執行 SckSAcceptTd 這個執行緒，並把 TcpClient 傳入
            SckSAcceptTd.Start(socket4Client);

        }
        //===== 持續接受監聽 socket client 的連線 ===== (end)
    }
    catch (Exception exp)
    {
        //sMessages.Add(exp.ToString());
    }
}
```


每個 SckSAcceptTd 執行緒內的處理動作 (對應到每條 Client)

```
private void TCPServer1Execute(object AContext)
{
    TcpClient _TcpClient = (TcpClient)AContext;
    int _ClientNo = 0;
    string client_ip =
((IPEndPoint)_TcpClient.Client.RemoteEndPoint).Address.ToString();

    //server & client 已經連線完成
    while (_TcpClient.Connected)
    {
        //取得網路串流物件，取得來自 socket client 的訊息
        NetworkStream netStream = _TcpClient.GetStream();
        byte[] readBuffer = new byte[1024];
        int count = 0;

        string clientRequest = "";
        int command_id = 0;
        string command_id2 = "";
        string command_id3 = "";
        string command_text = "";
        string command_text2 = "";
        string log_text = "";

        FileStream fstream = null;
        string filename;
        int filesize;
        int readcount;
        int file_count;
        String tmp;

        try
        {
            if ((count = netStream.Read(readBuffer, 0,
readBuffer.Length)) != 0)
            {
                clientRequest = Encoding.Unicode.GetString(readBuffer, 0,
count);

                string[] lines = clientRequest.Split(new[] { "\r\n", "\r",
"\n" }, StringSplitOptions.None);

                command_id = Int32.Parse(lines[0]); //Command_Id =>
1:SND_STR 2:GET_STR 3:SND_FILE 4:GET_FILE
                command_id2 = lines[1];
                command_id3 = lines[2];
                command_text = lines[3];
                command_text2 = lines[4];
            }
        }
    }
}
```

程式碼接下頁

```
        switch (command_id)
        {

            case 1: //收到 SND_STR
                    // (如後說明)
                    break;

            case 2: //收到 GET_STR
                    // (如後說明)
                    break;

            case 3: //收到 SND_FILE
                    // (如後說明)
                    break;

            case 4: //收到 GET_FILE
                    // (如後說明)
                    break;

        }
    }
    catch (Exception ex)
    {
        sMessages.Add(" >> " + "From client(" + _ClientNo + ") =>
ex=" + ex.Message);
    }
    finally
    {
        netStream.Close();
        _TcpClient.Close();
    }

    //調用ProgressChanged改變form元件數值 否則無法跨執行緒執行(刷新UI動作)
    connection_count--;
    backgroundWorker1.ReportProgress(connection_count);

    break;
}
}
```

補充說明：針對四大基本通訊功能的處理說明

```
case 1: //收到 SND_STR
    log_text = client_ip + "/" + command_id2 + " (SND_STR): command_id3=" +
    command_id3 + ", command_text=" + command_text + ", command_text2=" +
    command_text2;
    sMessages.Add(log_text);

    //-----
    //POS_CONNECT
    //-----
    if (command_id3 == "POS_CONNECT")
    {
        log_text = "POS_CONNECT(" + command_id2 + ")";
        sMessages.Add(log_text);
    }

    //-----
    //POS_MSG
    //-----
    if (command_id3 == "POS_MSG")
    {
        log_text = command_text + "(" + command_id2 + ")";
        sMessages.Add(log_text);
    }

break;
```

```

case 2: //收到 GET_STR
    log_text = client_ip + "/" + command_id2 + " (GET_STR): command_id3=" +
command_id3 + ", command_text=" + command_text + ", command_text2=" +
command_text2;
    sMessages.Add(log_text);

    //-----
    //取得 Server 時間
    //-----
    if (command_id3 == "SRV_TIME")
    {
        Byte[] data = System.Text.Encoding.Unicode.GetBytes("主機時間 SERVER
TIME IS " + FormatDateTime("hh:nn:ss", DateTime.Now));
        netStream.Write(data, 0, data.Length);
    }

    //-----
    //發票取號
    //-----
    if (command_id3 == "GetInvoRollStr")
    {
        //在 Muti-Thread 中保證同一時間只能有一個執行動作的機制
        //參考 [筆記]C# 鎖定-使用lock、Monitor.Enter、Monitor.TryEnter的小範例
        //https://dotblogs.com.tw/noncoder/2018/06/30/lock-monitor
        //在使用lock時，要小心鎖死的問題，而使用Monitor提供了TryEnter方法可以傳入等待逾
時時間
        Monitor.TryEnter(this, 6000);
        try
        {
            ; ;
        }
        finally
        {
            Monitor.Exit(this);
        }
    }
break;

```

```
case 3: //收到 SND_FILE
    log_text = client_ip + "/" + command_id2 + " (SND_FILE): command_id3=" +
command_id3 + ", command_text=" + command_text + ", command_text2=" +
command_text2;
    sMessages.Add(log_text);

    filename = command_text2;
    filesize = Convert.ToInt32(command_id3);

    if (File.Exists(filename))
    {
        File.Delete(filename);
    }

    //netStream->fstream
    fstream = new FileStream(filename, FileMode.Create, FileAccess.Write);
    while (filesize > fstream.Length)
    {
        if ((filesize - fstream.Length) > readBuffer.Length)
            readcount = readBuffer.Length;
        else
            readcount = filesize - (int)fstream.Length;

        netStream.Read(readBuffer, 0, readcount);
        fstream.Write(readBuffer, 0, readcount);
    }
    fstream.Close();

break;
```

```

case 4: //收到 GET_FILE
    log_text = client_ip + "/" + command_id2 + " (GET_FILE): command_id3=" +
command_id3 + ", command_text=" + command_text + ", command_text2=" +
command_text2;
    sMessages.Add(log_text);

    //不支援 command_text 中帶 "/S" (子目錄) 的處理
    file_count = 0;
    filename = "";
    foreach (string fname in
Directory.GetFiles(Path.GetDirectoryName(command_text),
Path.GetFileName((command_text))))
    {
        file_count++;
    }

    if (file_count == 0)
    {
        filename = "";
    }
    else if (file_count == 1)
    {
        ; ;
    }
    else //檔案超過 1 個
    {
        //壓縮目錄中的檔案
        //若要使用ZipFile類別，您必須加入參考System.IO.Compression.FileSystem 組件
        //並 using System.IO.Compression;
        //整個目錄中的檔案都會壓縮，不支援過濾遮罩(如 *.PAS)
        filename = FormatDateTime("hhnnsszzz", DateTime.Now) + ".zip";
        ZipFile.CreateFromDirectory(Path.GetDirectoryName(command_text),
filename);
    }

    if (filename != "")
    {
        fstream = new FileStream(filename, FileMode.Open, FileAccess.Read);
        filesize = (int)fstream.Length;

        tmp = filename + "\n" +
            filesize.ToString() + "\n";

        //將 byte[] 寫入 MyStream1
        Byte[] data = System.Text.Encoding.Unicode.GetBytes(tmp);
        Byte[] data2 = new byte[1024];
        data.CopyTo(data2, 0); //保持 1024 長度
        netStream.Write(data2, 0, data2.Length);
    }
}

```

程式碼接下頁

```
//fstream->netStream
fstream.Position = 0;
while (fstream.Position < filesize)
{
    if ((filesize - fstream.Position) >= data2.Length)
        readcount = data2.Length;
    else
        readcount = filesize - (int)fstream.Position;

    fstream.Read(data2, 0, readcount);

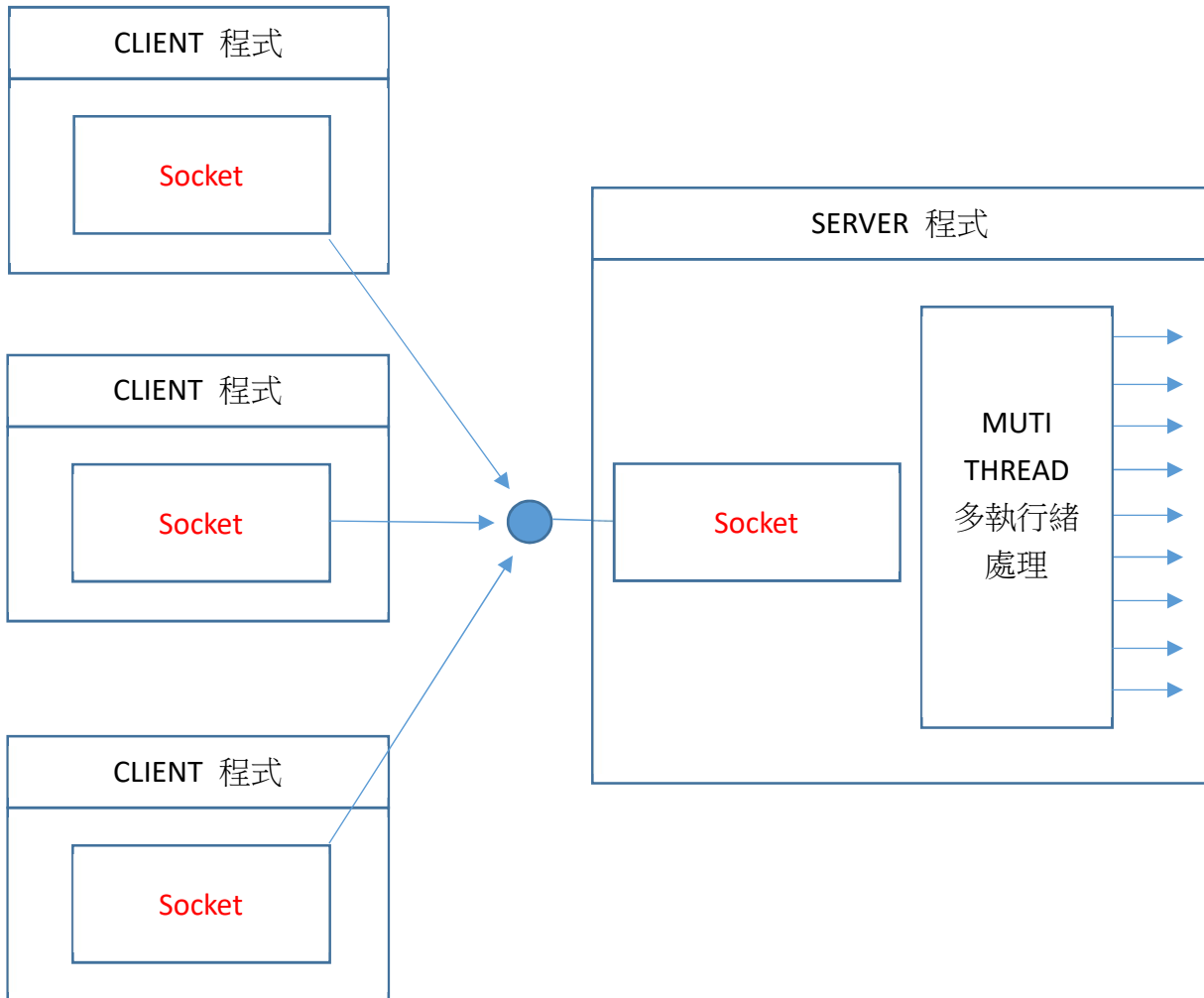
    netStream.Write(data2, 0, readcount);
}
fstream.Close();

}
else
{
    //回傳空白檔案資訊
    tmp = "" + "\n" +
        "0" + "\n";

    //將 byte[] 寫入 MyStream1
    Byte[] data = System.Text.Encoding.Unicode.GetBytes(tmp);
    Byte[] data2 = new byte[1024];
    data.CopyTo(data2, 0); //保持 1024 長度
    netStream.Write(data2, 0, data2.Length);

}
break;
```

範例二 Visual Studio 2017 C# (使用內建 Socket)



通訊改用 Socket 物件 (原使用 TcpClient & TcpListener) 以便做更廣應用測試
發覺修改很簡單, 很快就改好, 以下為修改前後比較

1.CLIENT 端

```
//MyTCPClient1 = new TcpClient(textBox2.Text, 6501); // textBox2.Text 內容為 IP 位置
//MyStream1 = MyTCPClient1.GetStream(); // 取得 client stream
MyClientSocket = new Socket(AddressFamily.InterNetwork, SocketType.Stream,
ProtocolType.Tcp);
MyClientSocket.Connect(textBox2.Text, 6501);
MyStream = new NetworkStream(MyClientSocket);
```

2.SERVER 端

```
//_Listener = new TcpListener(IPAddress.Any, 6501);
//_Listener.Start();
MyServerSocket = new Socket(AddressFamily.InterNetwork, SocketType.Stream,
ProtocolType.Tcp);
MyServerSocket.Bind(new IPEndPoint(IPAddress.Any, 6501));
MyServerSocket.Listen(10);
```

執行緒中...

```
//TcpClient socket4Client = _Listener.AcceptTcpClient();
Socket socket4Client = MyServerSocket.Accept();
```

```
//TcpClient _TcpClient = (TcpClient)AContext;
Socket MyClientSocket = (Socket)AContext;
```

```
//NetworkStream netStream = _TcpClient.GetStream();
NetworkStream netStream = new NetworkStream(MyClientSocket);
```

[CLIENT 端]

兩個公有變數

```
public partial class Form1 : Form
{
    Socket MyClientSocket;
    NetworkStream MyStream;
    .....
    .....
}
```

```
String SND_STR(String command_id2, String command_id3, String command_text,
String command_text2)
{
    String result = String.Empty;
    MemoLog("●SND_STR(" + command_id2 + "," + command_id3 + "," +
command_text + "," + command_text2 + ") 開始");

    try
    {
        MyClientSocket = new Socket(AddressFamily.InterNetwork,
SocketType.Stream, ProtocolType.Tcp);
        MyClientSocket.Connect(textBox2.Text, 6501); //textBox2.Text 為 IP
        MyStream = new NetworkStream(MyClientSocket);

        String tmp = "1" + "\n" + //1:SND_STR 2:GET_STR 3:SND_FILE 4:GET_FILE
            command_id2 + "\n" +
            command_id3 + "\n" +
            command_text + "\n" +
            command_text2 + "\n";

        Byte[] data = System.Text.Encoding.Unicode.GetBytes(tmp);
        Byte[] data2 = new byte[1024];
        data.CopyTo(data2, 0); //保持 1024 長度
        MyStream.Write(data2, 0, data2.Length);
    }
    catch (Exception ex)
    {
        MemoLog(ex.Message);
    }
    finally
    {
        if (MyStream != null) MyStream.Close();
        if (MyClientSocket != null) MyClientSocket.Close();
        MemoLog("○SND_STR() 結束");
    }

    return (result);
}
```

```

String GET_STR(String command_id2, String command_id3, String command_text,
String command_text2)
{
    String result = String.Empty;
    MemoLog("●GET_STR(" + command_id2 + "," + command_id3 + "," +
command_text + "," + command_text2 + ") 開始");

    try
    {
        MyClientSocket = new Socket(AddressFamily.InterNetwork,
SocketType.Stream, ProtocolType.Tcp);
        MyClientSocket.Connect(textBox2.Text, 6501);
        MyStream = new NetworkStream(MyClientSocket);

        //發送字串之處理
        String tmp = "2" + "\n" + //1:SND_STR 2:GET_STR 3:SND_FILE 4:GET_FILE
            command_id2 + "\n" +
            command_id3 + "\n" +
            command_text + "\n" +
            command_text2 + "\n";

        //將 byte[] 寫入 MyStream1
        Byte[] data = System.Text.Encoding.Unicode.GetBytes(tmp);
        Byte[] data2 = new byte[1024];
        data.CopyTo(data2, 0); //保持 1024 長度
        MyStream.Write(data2, 0, data2.Length);

        //讀回資料
        int count = 0;
        byte[] readBuffer = new byte[1024];
        if ((count = this.MyStream.Read(readBuffer, 0, readBuffer.Length)) !=
0)
        {
            result = Encoding.Unicode.GetString(readBuffer, 0, count);
        }
        MemoLog("接收到回傳字串 : '" + result + "'");
    }
    catch (Exception ex)
    {
        MemoLog(ex.Message);
    }
    finally
    {
        if (MyStream != null) MyStream.Close();
        if (MyClientSocket != null) MyClientSocket.Close();
        MemoLog("○GET_STR() 結束");
    }

    return (result);
}

```

```

String SND_FILE(String command_id2, String command_id3, String command_text,
String command_text2)
{
    String result = String.Empty;
    MemoLog("●SND_FILE(" + command_id2 + "," + command_id3 + "," +
command_text + "," + command_text2 + ") 開始");

    FileStream fstream = null; //using System.IO
    int filesize;
    int readcount;
    int fileoffset;
    try
    {
        fstream = new FileStream(command_text, FileMode.Open, FileAccess.Read);
        filesize = (int)fstream.Length;
        fileoffset = 0;

        MyClientSocket = new Socket(AddressFamily.InterNetwork,
SocketType.Stream, ProtocolType.Tcp);
        MyClientSocket.Connect(textBox2.Text, 6501);
        MyStream = new NetworkStream(MyClientSocket);

        //發送字串之處理
        String tmp = "3" + "\n" + //1:SND_STR 2:GET_STR 3:SND_FILE 4:GET_FILE
            command_id2 + "\n" +
            filesize.ToString() + "\n" +
            command_text + "\n" +
            command_text2+ "\n";

        //將 byte[] 寫入 MyStream1
        Byte[] data = System.Text.Encoding.Unicode.GetBytes(tmp);
        Byte[] data2 = new byte[1024];
        data.CopyTo(data2, 0); //保持 1024 長度
        MyStream.Write(data2, 0, data2.Length);
        //將檔案 stream 寫入 MyStream1
        fstream.Position = 0;
        while (fstream.Position < filesize) {
            if ((filesize - fstream.Position) >= data2.Length)
                readcount = data2.Length;
            else
                readcount = filesize - (int)fstream.Position;

            fstream.Read(data2, 0, readcount);
            MyStream.Write(data2, 0, readcount);
        }
        fstream.Close();
    }
    catch (Exception ex)
    {
        MemoLog(ex.Message);
    }
    finally
    {
        if (fstream != null) fstream.Close();
        if (MyStream != null) MyStream.Close();
        if (MyClientSocket != null) MyClientSocket.Close();
        MemoLog("○SND_FILE() 結束");
    }

    return (result);
}

```

```

String GET_FILE(String command_id2, String command_id3, String command_text,
String command_text2)
{
    String result = String.Empty;
    MemoLog("●GET_FILE(" + command_id2 + "," + command_id3 + "," +
command_text + "," + command_text2 + ") 開始");

    FileStream fstream = null; //using System.IO
    String filename = Path.GetFileName(command_text);
    int filesize=0;
    int readcount=0;
    int fileoffset=0;
    String get_file_path="";
    byte[] readBuffer = new byte[1024];
    int count = 0;

    try
    {
        MyClientSocket = new Socket(AddressFamily.InterNetwork,
SocketType.Stream, ProtocolType.Tcp);
        MyClientSocket.Connect(textBox2.Text, 6501);
        MyStream = new NetworkStream(MyClientSocket);

        //發送字串之處理
        String tmp = "4" + "\n" + //1:SNД_STR 2:GET_STR 3:SNД_FILE 4:GET_FILE
            command_id2 + "\n" +
            command_id3 + "\n" +
            command_text + "\n" +
            command_text2 + "\n";

        //將 byte[] 寫入 MyStream1
        Byte[] data = System.Text.Encoding.Unicode.GetBytes(tmp);
        Byte[] data2 = new byte[1024];
        data.CopyTo(data2, 0); //保持 1024 長度
        MyStream.Write(data2, 0, data2.Length);
        //讀回
        if ((count = MyStream.Read(readBuffer, 0, readBuffer.Length)) != 0)
        {
            tmp = Encoding.Unicode.GetString(readBuffer, 0, count);
            string[] lines = tmp.Split(new[] { "\r\n", "\r", "\n" },
StringSplitOptions.None);

            filename = lines[0];
            filesize = Int32.Parse(lines[1]);

        }
        //假如 Client 端已存在該檔案要先刪除
        get_file_path = command_text2;
        ForceDirectories(get_file_path);
        filename = get_file_path + "\\\" + Path.GetFileName(filename);
        if (File.Exists(filename)) File.Delete(filename);
    }
}

```

程式碼接下頁

```
//接收檔案
fstream = new FileStream(filename, FileMode.Create, FileAccess.Write);
while (filesize > fstream.Length)
{
    if ((filesize - fstream.Length) > readBuffer.Length)
        readcount = readBuffer.Length;
    else
        readcount = filesize - (int)fstream.Length;

    MyStream.Read(readBuffer, 0, readcount);
    fstream.Write(readBuffer, 0, readcount);
}
fstream.Close();

//解壓縮檔案
if (Path.GetExtension(filename) == ".ZIP" || Path.GetExtension(filename)
== ".zip")
{
    ZipFile.ExtractToDirectory(filename, command_text2);
    File.Delete(filename);
}
}
catch (Exception ex)
{
    MemoLog(ex.Message);
}
finally
{
    if (fstream != null) fstream.Close();
    if (MyStream != null) MyStream.Close();
    if (MyClientSocket != null) MyClientSocket.Close();
    MemoLog("oGET_FILE() 結束");
}

return (result);
}
```

[SERVER 端]

同範例一，無論使用 TcpClient / TcpListener 或是使用 Socket 物件，都要使用背景工作元件 backgroundWorker

Form Load 前的公用變數宣告，及 Form Load 中的動作

```
int connection_count = 0; //client 連線數
Socket MyServerSocket; //宣告 Server 端 TcpListener 物件

private void Form1_Load(object sender, EventArgs e)
{
    backgroundWorker1.RunWorkerAsync(); //啟動背景工作元件
}
```

Form Closed 中的動作

```
private void Form1_FormClosed(object sender, FormClosedEventArgs e)
{
    if (backgroundWorker1.WorkerSupportsCancellation == true)
    {
        backgroundWorker1.CancelAsync();
    }

    if (MyServerSocket != null)
    {
        MyServerSocket.Close();
        MyServerSocket = null;
    }
}
```

GUI 動作在此事件中處理

```
private void backgroundWorker1_ProgressChanged(object sender,
ProgressChangedEventArgs e)
{
    //e.ProgressPercentage 為 backgroundWorker1.ReportProgress() 傳入的參數
    label1.Text = "連線數 : " + e.ProgressPercentage.ToString();
}
```

在 `backgroundWorker1` 背景工作元件的 `DoWork()` 中 `new` 出 `_Listener` 物件, 並在無窮迴圈中一直處理監聽動作, 只要監聽到有 `Client` 連線進來, 就馬上 `new` 出一條名為 `SckSAcceptTd` 的 `Thread` 與其對應

```
private void backgroundWorker1_DoWork(object sender, DoWorkEventArgs e)
{
    try
    {
        MyServerSocket = new Socket(AddressFamily.InterNetwork,
        SocketType.Stream, ProtocolType.Tcp);
        MyServerSocket.Bind(new IPEndPoint(IPAddress.Any, 6501));
        MyServerSocket.Listen(10);

        //===== 持續接受監聽 socket client 的連線 ===== (start)
        while (true)
        {
            //監聽到來自 socket client 的連線要求
            Socket socket4Client = MyServerSocket.Accept();

            //調用ProgressChanged改變form元件數值 否則無法跨執行緒執行(刷新UI動作)
            connection_count++;
            backgroundWorker1.ReportProgress(connection_count);

            //可以傳入參數的 thread
            Thread SckSAcceptTd = new Thread(new
            ParameterizedThreadStart(TCPServer1Execute));
            SckSAcceptTd.IsBackground = true;
            SckSAcceptTd.Start(socket4Client); // 開始執行 SckSAcceptTd 這個執行
            緒, 並把 Socket 傳入

        }
        //===== 持續接受監聽 socket client 的連線 ===== (end)
    }
    catch (Exception exp)
    {
        //sMessages.Add(exp.ToString());
    }
}
```


每個 SckSAcceptTd 執行緒內的處理動作 (對應到每條 Client)

```
private void TCPServer1Execute(object AContext)
{
    Socket MyClientSocket = (Socket)AContext;
    int _ClientNo = 0;
    string client_ip =
((IPEndPoint)MyClientSocket.RemoteEndPoint).Address.ToString();

    while (MyClientSocket.Connected)
    {
        //取得網路串流物件，取得來自 socket client 的訊息
        //NetworkStream netStream = _TcpClient.GetStream();
        NetworkStream netStream = new NetworkStream(MyClientSocket);
        byte[] readBuffer = new byte[1024];
        int count = 0;

        string clientRequest = "";
        int command_id = 0;
        string command_id2 = "";
        string command_id3 = "";
        string command_text = "";
        string command_text2 = "";
        string log_text = "";

        FileStream fstream = null;
        string filename;
        int filesize;
        int readcount;
        int file_count;
        String tmp;
```



程式碼接下頁

```

        try
        {
            if ((count = netStream.Read(readBuffer, 0, readBuffer.Length)) != 0)
            {
                clientRequest = Encoding.Unicode.GetString(readBuffer, 0, count);
                string[] lines = clientRequest.Split(new[] { "\r\n", "\r", "\n" },
StringSplitOptions.None);

                command_id = Int32.Parse(lines[0]); //Command_Id => 1:SND_STR
2:GET_STR  3:SND_FILE  4:GET_FILE
                command_id2 = lines[1];
                command_id3 = lines[2];
                command_text = lines[3];
                command_text2 = lines[4];
            }

            switch (command_id)
            {
                case 1: //收到 SND_STR
                    log_text = client_ip + "/" + command_id2 + " (SND_STR):
command_id3=" + command_id3 + ", command_text=" + command_text + ", command_text2="
+ command_text2;

                    sMessages.Add(log_text);

                    //-----
                    //POS_CONNECT
                    //-----
                    if (command_id3 == "POS_CONNECT")
                    {
                        log_text = "POS_CONNECT(" + command_id2 + ")";
                        sMessages.Add(log_text);
                    }
                    //-----
                    //POS_MSG
                    //-----
                    if (command_id3 == "POS_MSG")
                    {
                        log_text = command_text + "(" + command_id2 + ")";
                        sMessages.Add(log_text);
                    }
                    break;
            }
        }
    }
}

```

程式碼接下頁

```

        case 2: //收到 GET_STR
            log_text = client_ip + "/" + command_id2 + " (GET_STR):
command_id3=" + command_id3 + ", command_text=" + command_text + ", command_text2="
+ command_text2;

            sMessages.Add(log_text);

            //-----
            //取得 Server 時間
            //-----
            if (command_id3 == "SRV_TIME")
            {
                Byte[] data = System.Text.Encoding.Unicode.GetBytes("
主機時間 SERVER TIME IS " + FormatDateTime("hh:nn:ss", DateTime.Now));
                netStream.Write(data, 0, data.Length);
            }

            //-----
            //發票取號
            //-----
            if (command_id3 == "GetInvoRollStr")
            {
                //參考 [筆記]C# 鎖定-使用 lock、Monitor.Enter、
                //https://dotblogs.com.tw/noncoder/2018/06/30/lock-
                //在使用 lock 時, 要小心鎖死的問題, 而使用 Monitor
                //提供了 TryEnter 方法可以傳入等待逾時時間
                Monitor.TryEnter(this, 6000);
                try
                {
                    ;;
                }
                finally
                {
                    Monitor.Exit(this);
                }
            }

            break;

```

Monitor.TryEnter 的小範例

monitor

提供了 TryEnter 方法可以傳入等待逾時時間

程式碼接下頁

```
case 3://收到 SND_FILE
    log_text = client_ip + "/" + command_id2 + " (SND_FILE):
command_id3=" + command_id3 + ", command_text=" + command_text + ", command_text2="
+ command_text2;

    sMessages.Add(log_text);

    filename = command_text2;
    filesize = Convert.ToInt32(command_id3);

    if (File.Exists(filename))
    {
        File.Delete(filename);
    }

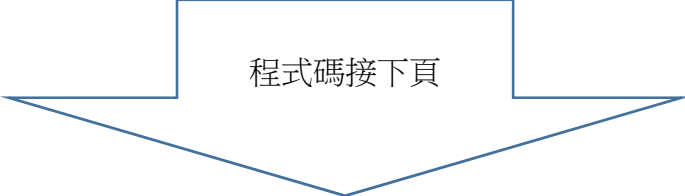
    //netStream->fstream
    fstream = new FileStream(filename, FileMode.Create,
FileAccess.Write);

    while (filesize > fstream.Length)
    {
        if ((filesize - fstream.Length) > readBuffer.Length)
            readcount = readBuffer.Length;
        else
            readcount = filesize - (int)fstream.Length;

        netStream.Read(readBuffer, 0, readcount);
        fstream.Write(readBuffer, 0, readcount);

    }
    fstream.Close();

    break;
```



程式碼接下頁

```

        case 4: //收到 GET_FILE
            log_text = client_ip + "/" + command_id2 + " (GET_FILE):
command_id3=" + command_id3 + ", command_text=" + command_text + ", command_text2="
+ command_text2;

            sMessages.Add(log_text);

            //不支援 command_text 中帶 "/S" (子目錄) 的處理
            file_count = 0;
            filename = "";
            foreach (string fname in
Directory.GetFiles(Path.GetDirectoryName(command_text),
Path.GetFileName((command_text))))
            {
                file_count++;
                filename = fname;
            }
            //SocketServer.sMessages.Add("file_count="+
file_count.ToString());

            if (file_count == 0)
            {
                filename = "";
            }
            else if (file_count == 1)
            {
                ;;
            }
            else //檔案超過 1 個
            {
                //壓縮目錄中的檔案
                //若要使用 ZipFile 類別，您必須加入參考
System.IO.Compression.FileSystem 組件
                //並 using System.IO.Compression;
                //整個目錄中的檔案都會壓縮，不支援過濾遮罩(如
*.PAS)

                filename = FormatDateTime("hhnnsszzz",
DateTime.Now) + ".zip";

                ZipFile.CreateFromDirectory(Path.GetDirectoryName(command_text), filename);
            }

```

程式碼接下頁

```
if (filename != "")
{
    FileStream fstream = new FileStream(filename, FileMode.Open,
FileAccess.Read);

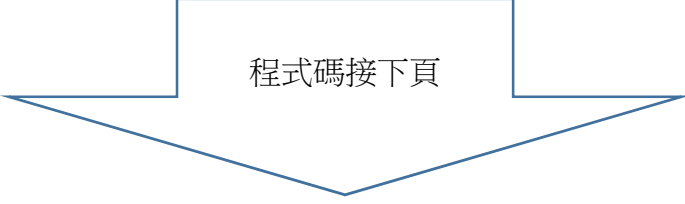
    filesize = (int)fstream.Length;

    tmp = filename + "\n" +
        filesize.ToString() + "\n";

    //將 byte[] 寫入 MyStream1
    Byte[] data =
System.Text.Encoding.Unicode.GetBytes(tmp);
    Byte[] data2 = new byte[1024];
    data.CopyTo(data2, 0); //保持 1024 長度
    netStream.Write(data2, 0, data2.Length);

    //fstream->netStream
    fstream.Position = 0;
    while (fstream.Position < filesize)
    {
        if ((filesize - fstream.Position) >= data2.Length)
            readcount = data2.Length;
        else
            readcount = filesize - (int)fstream.Position;

        fstream.Read(data2, 0, readcount);
        netStream.Write(data2, 0, readcount);
    }
    fstream.Close();
}
}
```



程式碼接下頁

```

else
{
    //回傳空白檔案資訊
    tmp = "" + "\n" +
        "0" + "\n";

    //將 byte[] 寫入 MyStream1
    Byte[] data =
System.Text.Encoding.Unicode.GetBytes(tmp);
    Byte[] data2 = new byte[1024];
    data.CopyTo(data2, 0); //保持 1024 長度
    netStream.Write(data2, 0, data2.Length);
}

break;

}
}
catch (Exception ex)
{
    sMessages.Add(">> " + "From client(" + _ClientNo + ") => ex=" +
ex.Message);
}
finally
{
    netStream.Close();
    MyClientSocket.Close();
}

//調用 ProgressChanged 改變 form 元件數值 否則無法跨執行緒執行(刷
新 UI 動作)
connection_count--;
backgroundWorker1.ReportProgress(connection_count);

break;
}
}
}

```

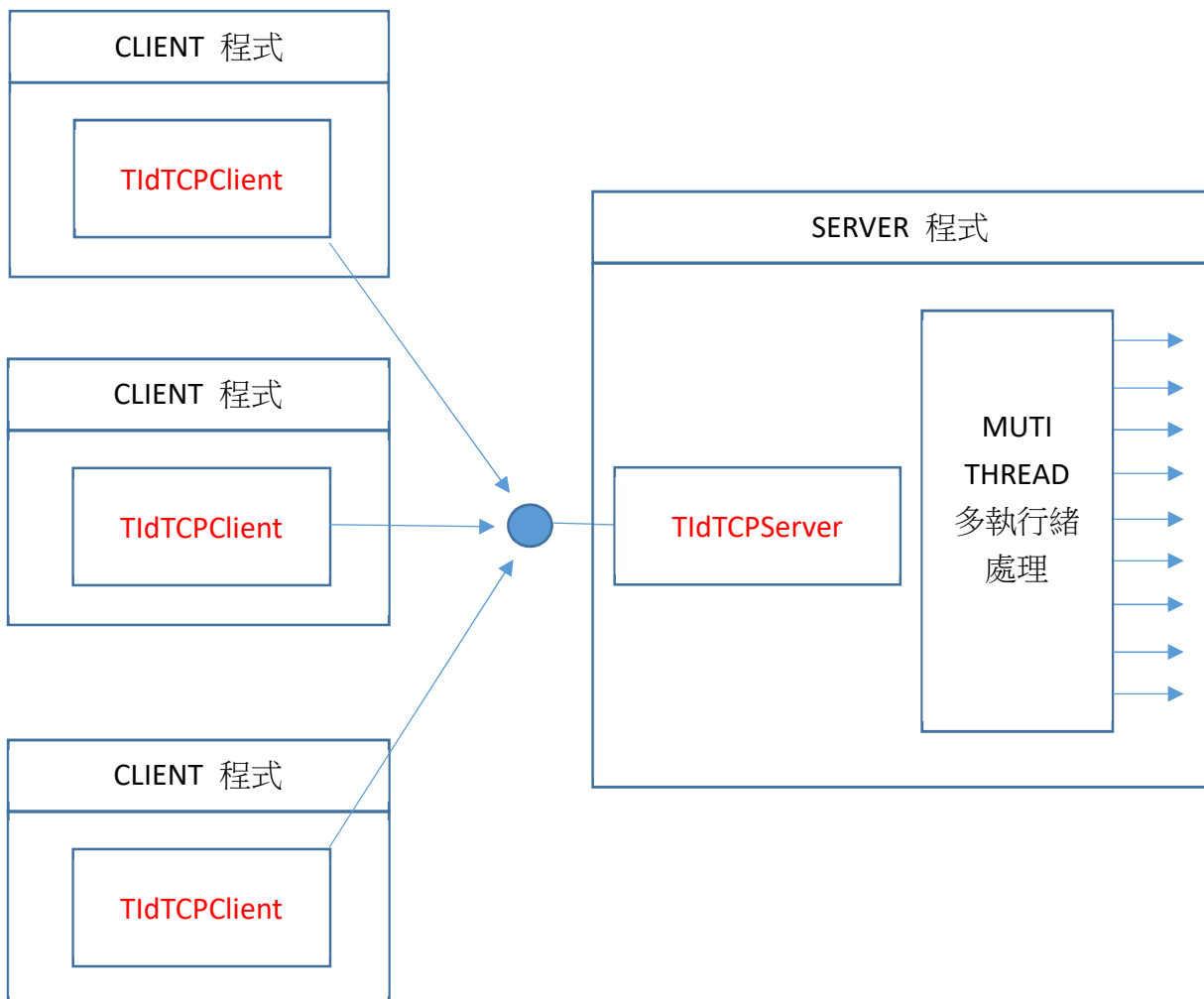
範例三 Lazarus 2.0 (使用外掛 INDY 元件)

Lazarus 使用 FREE PASCAL 語言, 具有免費以及跨平台特性(有 WIN32/WIN64/Linux/MAC ... 各種版本); 短小精幹安裝快速, 讓我有回到以前 “信手拈來, 就可以寫支小程式的手感”, 雖然是開源(OPEN SOURCE)產品, 但可開發商用執行檔; 它的語法及 IDE 開發環境跟 Delphi 有 95% 相容性, 因為以前用過 Delphi, 所以對我來說很快就上手

Lazarus 的下載網頁在 <https://www.lazarus-ide.org/>
目前最新版本為 2.0

Lazarus 中並無內建 INDY TCP/IP 通訊元件, 可到 <http://indy.fulgan.com/ZIP/> 免費下載, 詳細安裝過程請參考附錄

INDY TCP/IP 通訊元件很多東西都已幫你包裝好, 不用自己去創建 THREAD, 直接在 **TIdTCPServer** 元件(Control) 的事件中就可以處理每個 CLIENT 的 REQUEST, 程式碼相對也變得簡單許多



INDY TCP/IP 通訊元件有獨立寫出數值及寫出字串的方法, SERVER 接收端只要照順序接收, 都能完整還原原來資料順序, 不會攪在一起, 所以資料檔頭部分不用組成像前面介紹中, C# 中固定 1024 bytes 的寫法

[CLIENT 端]

```
IdTCPClient1.IOHandler.Write(LongInt(1),true); // 1:SND_STR 2:GET_STR 3:SND_FILE 4:GET_FILE
IdTCPClient1.IOHandler.WriteLine(Edit0.Text); //command_id2
IdTCPClient1.IOHandler.WriteLine('POS_CONNECT'); //command_id3
IdTCPClient1.IOHandler.WriteLine(""); //command_text
IdTCPClient1.IOHandler.WriteLine(""); //command_text2
```

[SERVER 端] 照順序回收資料檔頭

```
command_id := AContext.Connection.IOHandler.ReadLongInt(true);
command_id2 := AContext.Connection.IOHandler.ReadLn;
command_id3 := AContext.Connection.IOHandler.ReadLn;
command_text := AContext.Connection.IOHandler.ReadLn;
command_text2 := AContext.Connection.IOHandler.ReadLn;
```

```
switch (command_id)
```

```
{
    case 1 : //處理 SND_STR
        break;
    case 2 : //處理 GET_STR
        break;
    case 3 : //處理 SND_FILE
        break;
    case 4 : //處理 GET_FILE
        break;
```

```
}
```

本文前面 C# 中 SND_FILE 資料的設計方式

每段固定長度 1024 bytes					
command_id 固定為 "3"+ "\n"	command_id2 不定長字串+ "\n"	command_id3 不定長字串+ "\n"	command_text 不定長字串+ "\n"	command_text2 不定長字串+ "\n"	剩餘空字 串補足 1024 bytes
1024 bytes 的檔案串流(Stream)資料 (第一段)					
1024 bytes 的檔案串流(Stream)資料 (第二段)					
.....					
1024 bytes 的檔案串流(Stream)資料 (第 N 段)					
檔案尾段不足 1024 bytes 的剩餘串流(Stream)資料 + 補足 1024 bytes 的空資料					

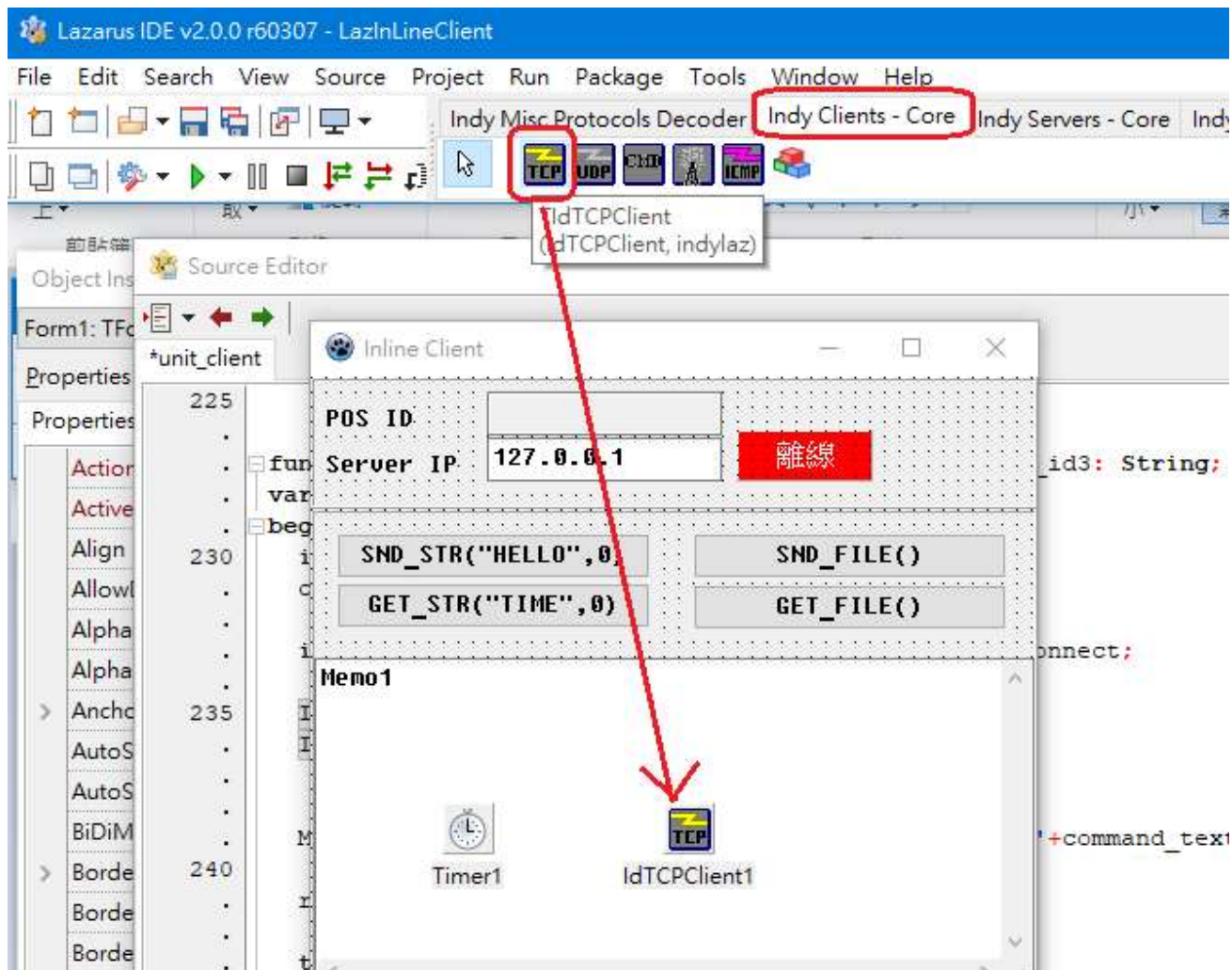
本文使用 INDY 元件 SND_FILE 資料的設計方式 (字串不用自己加 "\n")

每段固定長度 1024 bytes					
command_id 固定為 "3"	command_id2 不定長字串	command_id3 不定長字串	command_text 不定長字串	command_text2 不定長字串	
1024 bytes 的檔案串流(Stream)資料 (第一段)					
1024 bytes 的檔案串流(Stream)資料 (第二段)					
.....					
1024 bytes 的檔案串流(Stream)資料 (第 N 段)					
檔案尾段不足 1024 bytes 的剩餘串流(Stream)資料 + 補足 1024 bytes 的空					

不用補足 1024 bytes

[CLIENT 端]

INDY 是以元件(Component)型式存在, 所以要開發 CLIENT 端程式時, 先拉一個 TIdClient 元件下來, 然後就可以開始寫程式



```

//使用例 : SND_STR(Edit0.Text, 'POS_MSG', 'HELLO', '');

procedure TForm1.SND_STR(command_id2: String; command_id3: String;
command_text: String; command_text2: String);
begin

    if (IdTCPClient1.Connected) then IdTCPClient1.Disconnect;

    IdTCPClient1.Host := Edit1.Text; //ip 位置
    IdTCPClient1.Port := 6501;
    IdTCPClient1.ReadTimeout:=1000;

    MemoLog('●SND_STR('+command_id2+', '+command_id3+', '+command_text+',
'+command_text2+') 開始');

    try
        IdTCPClient1.Connect;
    except
        MemoLog('連結失敗');
        connection_busy:=false;
        MemoLog('○SND_STR() 結束');
        exit;
    end;

    try
        IdTCPClient1.IOHandler.Write(LongInt(1), true); //Command_id => 1:SND_STR
2:GET_STR 3:SND_FILE 4:GET_FILE
        IdTCPClient1.IOHandler.WriteLine(command_id2); //command_id2
        IdTCPClient1.IOHandler.WriteLine(command_id3); //command_id3
        IdTCPClient1.IOHandler.WriteLine(command_text); //command_text
        IdTCPClient1.IOHandler.WriteLine(command_text2); //command_text2
    finally
        IdTCPClient1.Disconnect;
        connection_busy:=false;
        MemoLog('○SND_STR() 結束');
    end;

end;

```

```

//使用例 : GET_STR(Edit0.Text, 'SRV_TIME', '', ''); //取得後台時間字串

function TForm1.GET_STR(command_id2: String; command_id3: String;
command_text: String; command_text2: String): String;
var r:String;
begin
  if (IdTCPClient1.Connected) then IdTCPClient1.Disconnect;

  IdTCPClient1.Host := Edit1.Text; //ip 位置
  IdTCPClient1.Port := 6501;
  IdTCPClient1.ReadTimeout:=1000;

  MemoLog('●GET_STR('+command_id2+', '+command_id3+', '+command_text+',
'+command_text2+') 開始');

  r:='';

  try
    IdTCPClient1.Connect;
  except
    MemoLog('連結失敗');
    connection_busy:=false;
    MemoLog('○GET_STR() 結束');
    result:=r;
    exit;
  end;

  try
    IdTCPClient1.IOHandler.Write(LongInt(2), true); //Command_id => 1:SND_STR
2:GET_STR 3:SND_FILE 4:GET_FILE
    IdTCPClient1.IOHandler.WriteLine(command_id2); //command_id2
    IdTCPClient1.IOHandler.WriteLine(command_id3); //command_id3
    IdTCPClient1.IOHandler.WriteLine(command_text); //command_text
    IdTCPClient1.IOHandler.WriteLine(command_text2); //command_text2

    r:=IdTCPClient1.IOHandler.ReadLn;
    MemoLog('接收到回傳字串 : '+r+'');
  finally
    IdTCPClient1.Disconnect;
    connection_busy:=false;
    MemoLog('○GET_STR() 結束');
  end;

  result:=r;
end;

```

```

//使用例 : SND_FILE(Edit0.Text, '', sour_file, dest_file);

function TForm1.SND_FILE(command_id2: String; command_id3: String;
command_text: String; command_text2: String): boolean;
var buff:array[0..1024] of byte;
    buff2:array of Byte; //TBytes
    buff3:TIdBytes; //uses idGlobal
    fstream:TFileStream;
    filesize:integer;
    filename:String;
    readcount:integer;
    rv:boolean;
begin

    rv:=false;

    if (IdTCPClient1.Connected) then IdTCPClient1.Disconnect;

    IdTCPClient1.Host := Edit1.Text;
    IdTCPClient1.Port := 6501;
    IdTCPClient1.ReadTimeout:=30000;

MemoLog('●SND_FILE('+command_id2+', '+command_id3+', '+command_text+', '+command_
text2+') 開始');

    try
        IdTCPClient1.Connect;
    except
        MemoLog('連結失敗');
        connection_busy:=false;
        MemoLog('○SND_FILE() 結束');
        exit;
    end;

    try
        fstream:=TFileStream.Create(command_text, fmShareDenyNone);
        filename:=ExtractFileName(command_text);
        filesize:=fstream.Size;

        IdTCPClient1.IOHandler.Write(LongInt(3), true); //Command_Id => 1:SND_STR
2:GET_STR 3:SND_FILE 4:GET_FILE
        IdTCPClient1.IOHandler.WriteLn(command_id2); //command_id2
        IdTCPClient1.IOHandler.WriteLn(IntToStr(filesize)); //command_id3
        IdTCPClient1.IOHandler.WriteLn(command_text); //command_text
        IdTCPClient1.IOHandler.WriteLn(command_text2); //command_text2

        MemoLog('發送檔案:'+command_text);

```

程式碼接下頁

```
fstream.Position:=0;
While (fstream.Position < filesize) do
begin
    if ((filesize - fstream.Position) >= 1024) then //PASCAL 不能用
SIZEOF(buff) ??
        readcount := 1024
    else
        readcount := filesize - fstream.Position;

    fstream.ReadBuffer(buff, readcount);
    SetLength(buff3,readcount);
    Move(buff[0],buff3[0],readcount);
    IdTCPClient1.Socket.Write(buff3, readcount);

end;

MemoLog('發送檔案完成 ('+IntToStr(filesize)+' bytes)');

rv:=true;

finally
    fstream.Free;
    IdTCPClient1.Disconnect;
    connection_busy:=false;
    MemoLog('oSND_FILE() 結束');
end;

result:=rv;

end;
```

```

//使用例 : GET_FILE(Edit0.Text, '', sour_file_path, dest_file_path);

function TForm1.GET_FILE(command_id2: String; command_id3: String;
command_text: String; command_text2: String): boolean;
var buff:array[0..1024] of byte;
    buff3:TIdBytes; //uses idGlobal
    fstream:TFileStream;
    filesize:longint;
    filename:String;
    readcount:integer;
    get_file_path:String;
    TmpList: TStringList;
    TmpList2: TStringList;
    i: integer;
    err_flag: integer; //2015/10/07
    rv:boolean;
    UnZipper: TUnZipper;
begin

    rv:=false;

    if (IdTCPClient1.Connected) then IdTCPClient1.Disconnect;

    IdTCPClient1.Host := Edit1.Text;
    IdTCPClient1.Port := 6501;
    IdTCPClient1.ReadTimeout:=30000;

MemoLog('●GET_FILE('+command_id2+', '+command_id3+', '+command_text+', '+command_t
ext2+') 開始');

    try
        IdTCPClient1.Connect;
    except
        MemoLog('連結失敗');
        connection_busy:=false;
        MemoLog('○GET_FILE 結束');
        exit;
    end;

    try
        IdTCPClient1.IOHandler.Write(LongInt(4), true); //Command_Id => 1:SND_STR
2:GET_STR 3:SND_FILE 4:GET_FILE
        IdTCPClient1.IOHandler.WriteLine(command_id2); //command_id2
        IdTCPClient1.IOHandler.WriteLine(command_id3); //command_id3
        IdTCPClient1.IOHandler.WriteLine(command_text); //command_text
        IdTCPClient1.IOHandler.WriteLine(command_text2); //command_text2

        //Sleep(5000);

        filename:=IdTCPClient1.IOHandler.ReadLn; //讀回檔名列
        filesize:=IdTCPClient1.IOHandler.ReadLongInt(true); //讀回檔案 size 列

```

程式碼接下頁


```

//假如 Client 端已存在該檔案要先刪除
get_file_path:=command_text2;
ForceDirectories(get_file_path);
filename:=get_file_path+'\''+ExtractFileName(filename);
if (FileExists(filename)) then DeleteFile(filename);

fstream:=TFileStream.Create(filename, fmCreate); //接收端檔名路徑
MemoLog('接收檔案:'+filename);

while (filesize > fstream.Size) do begin
  if ((filesize - fstream.Size) > 1024) then begin
    readcount := 1024
  end else begin
    readcount := filesize - fstream.Size;
  end;

  IdTCPClient1.Socket.ReadBytes(buff3, readcount, false);
  Move(buff3[0],buff[0],readcount);
  fstream.WriteBuffer(buff, readcount);
  Application.ProcessMessages();

end;

fstream.Free;

//-----
//壓縮檔案參考 http://wiki.freepascal.org/paszlib
//-----
if (UpperCase(ExtractFileExt(filename))='.ZIP') then begin
  UnZipper := TUnZipper.Create;
  try
    UnZipper.FileName := filename;
    UnZipper.OutputPath := get_file_path;
    UnZipper.Examine;
    UnZipper.UnZipAllFiles;
  finally
    UnZipper.Free;
  end;
  DeleteFile(filename);
end;

finally
  IdTCPClient1.Disconnect;
  connection_busy:=false;
  MemoLog('oGET_FILE 結束');
end;

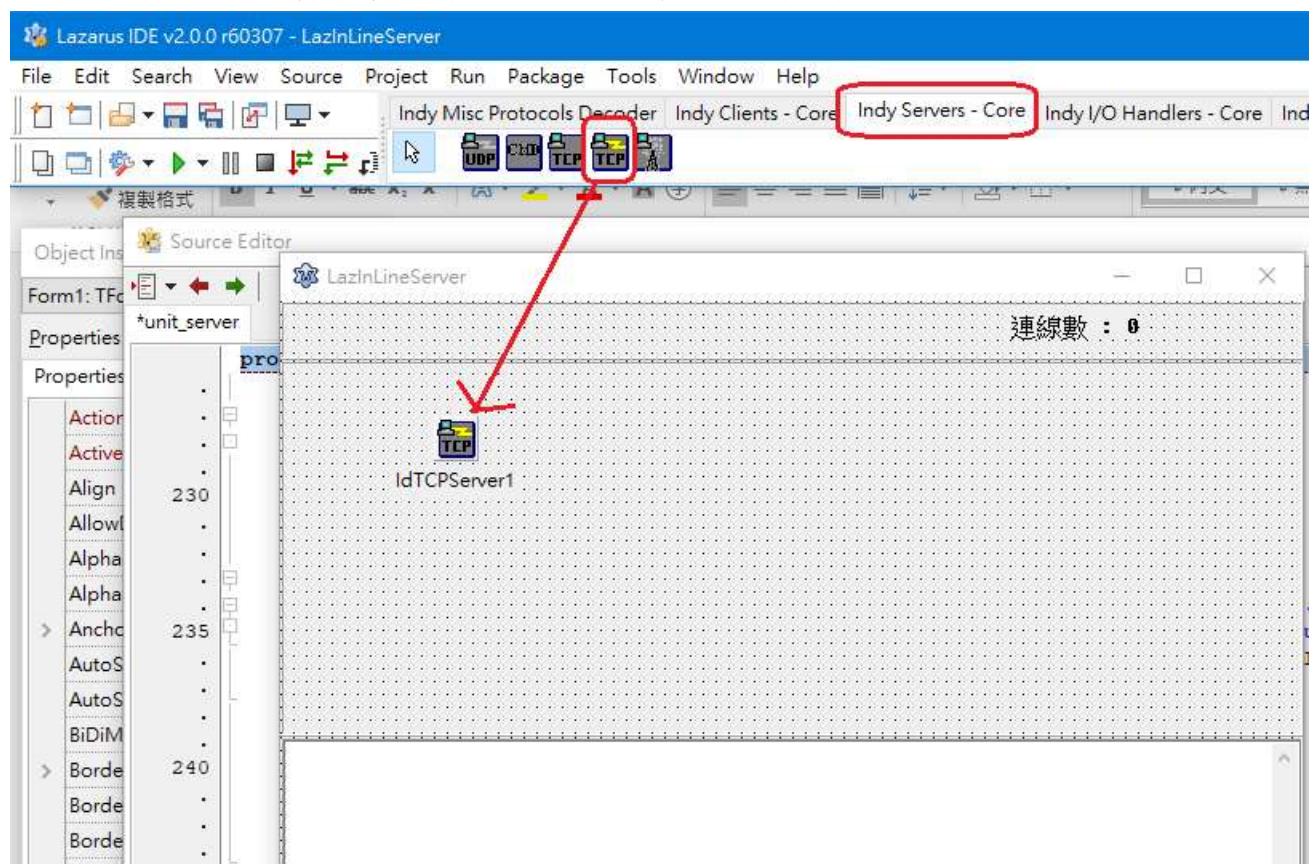
result:=rv;

end;

```

[SERVER 端]

拉一個 TIdServer 元件下來, 然後就可以開始寫 Server 程式



在 FormActive 時啟動 IdTCPServer, 以及建立臨界區變數

```
uses
  Classes, SysUtils, Forms, Controls, Graphics, Dialogs, ExtCtrls, StdCtrls,
  IdTCPServer, IdIPWatch, IdCustomTCPServer, IdContext, idsync, idGlobal,
  Zipper,
  Lazutf8;

.....
.....

var
  Form1: TForm1;
  run_once: boolean;
  connection_busy: boolean;
  MyCs:TRTLCriticalSection; //臨界區
  connection_count: integer;
  log_text:String;

implementation

{$R *.lfm}

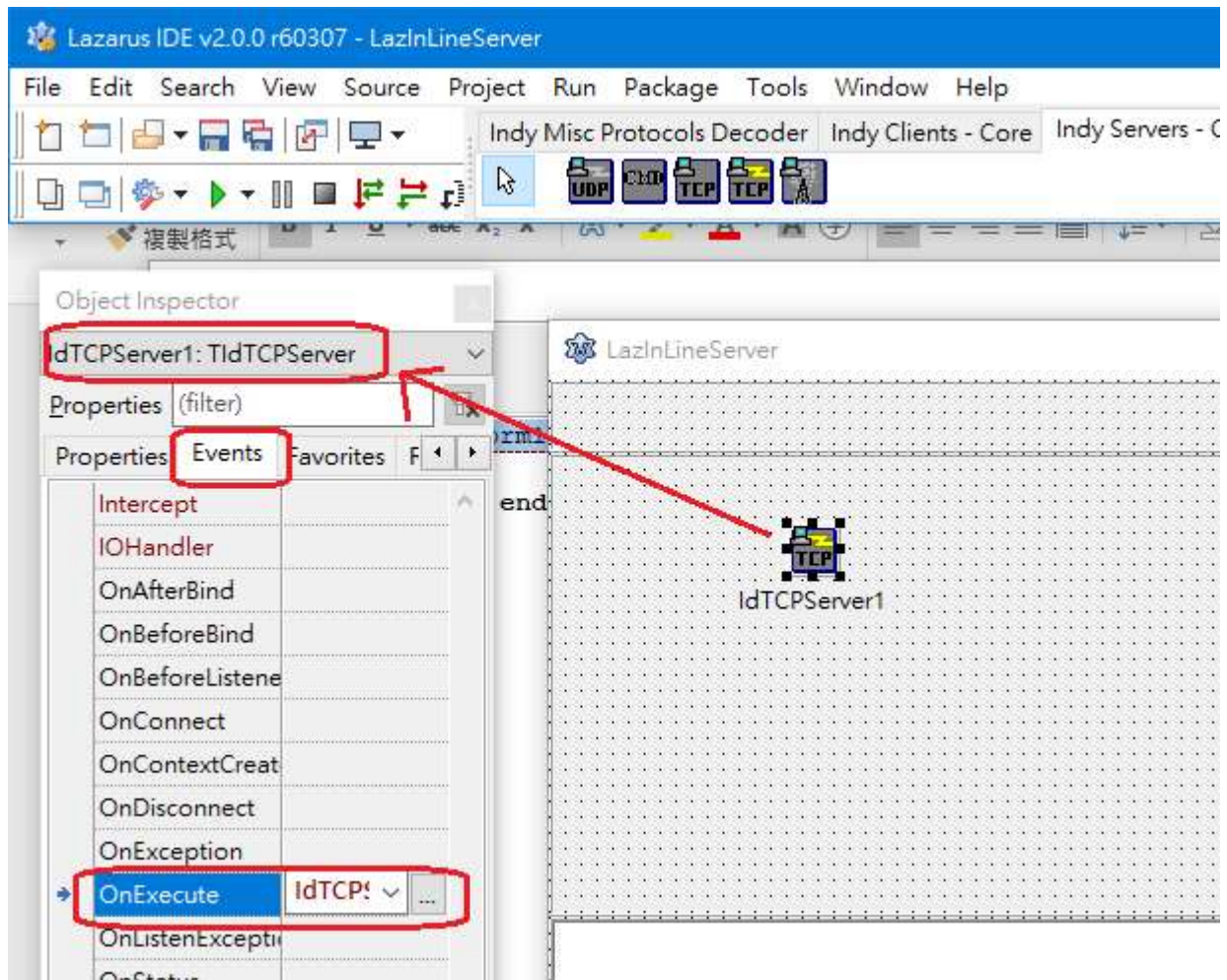
{ TForm1 }

procedure TForm1.FormCreate(Sender: TObject);
begin
  connection_count:=0;
  InitCriticalSection(MyCs); //初始化臨界區
end;

procedure TForm1.FormActivate(Sender: TObject);
begin
  IdTCPServer1.DefaultPort:= 6501;
  IdTCPServer1.Active:=true;
end;

procedure TForm1.FormClose(Sender: TObject; var CloseAction: TCloseAction);
begin
  IdTCPServer1.Active:=false;
  DoneCriticalSection(MyCs); //刪除臨界區
end;
```

當有 CLIENT 連上來時，我們不用自己寫 THREAD 來處理連線工作，直接在 IdTCPServer OnExecute 事件中撰寫連線處理程序即可



```
procedure TForm1.IdTCPServer1Execute(AContext: TIdContext);
var command_id:integer;
    command_id2: String;
    command_id3: String;
    command_text: String;
    command_text2: String;
    client_ip:String;
    result_str: String;
    MySync: TIdSync;

    buff:array[0..1023] of byte;
    buff2:array of Byte;
    buff3:TIdBytes; //uses idGlobal

    fstream:TFileStream;
    filesize:longint;
    filename:String;
    readcount:integer;

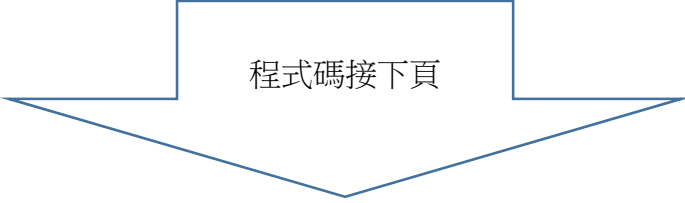
    TmpList:TStringList;
    TmpList2:TStringList;
    TmpList3:TStringList;

    OurZipper: TZipper;
    i: integer;
    err_flag: integer;
begin

    client_ip:=AContext.Connection.Socket.Binding.PeerIP; //同
    AContext.Binding.PeerIP;

    MySync:=TIdSync.Create();
    MySync.SynchronizeMethod(@IncrConnectioncount); //uses idsync; indy 10 要用這
    種寫法計算連線數

    try
        command_id:=AContext.Connection.IOHandler.ReadLongInt(true); //Command_Id
=> 1:SND_STR 2:GET_STR 3:SND_FILE 4:GET_FILE
        command_id2 :=AContext.Connection.IOHandler.ReadLn; //pos id
        command_id3 :=AContext.Connection.IOHandler.ReadLn; //func id
        command_text:=AContext.Connection.IOHandler.ReadLn; //參數一
        command_text2:=AContext.Connection.IOHandler.ReadLn; //參數二
```



程式碼接下頁

```
case command_id of
  1 : begin //收到 SND_STR
        log_text:=client_ip+'/'+ command_id2 +' (SND_STR):
command_id3='+command_id3+', command_text="'+command_text+'",
command_text2="'+command_text2;
        MySync.SynchronizeMethod(@AddLogEntry);

        //-----
        //POS_CONNECT
        //-----
        if (command_id3='POS_CONNECT') then begin
            log_text:='POS_CONNECT('+command_id2+')';
            MySync.SynchronizeMethod(@AddLogEntry);
        end;

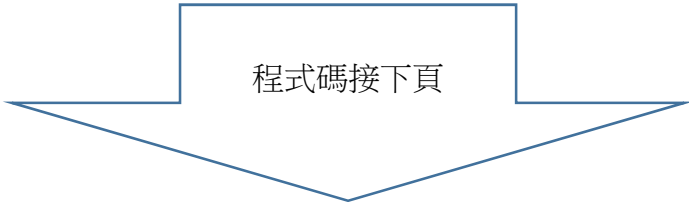
        //-----
        //pos message
        //-----
        if (command_id3='POS_MSG') then begin
            log_text:=command_text+'('+command_id2+')';
            MySync.SynchronizeMethod(@AddLogEntry);
        end;

    end;

  2 : begin //收到 GET_STR

        log_text:=client_ip+'/'+ command_id2 +' (GET_STR):
command_id3='+command_id3+', command_text="'+command_text+'",
command_text2="'+command_text2;
        MySync.SynchronizeMethod(@AddLogEntry);

        //-----
        //取得 Server 時間
        //-----
        if (command_id3='SRV_TIME') then begin
            AContext.Connection.IOHandler.WriteLine('SERVER TIME IS
'+FormatDateTime('hh:nn:ss', Now));
        end;
    end;
```



程式碼接下頁

```

//-----
//發票取號
//-----
if (command_id3='GetInvoRollStr') then begin
    result_str:='';
    EnterCriticalSection(MyCs); //進入臨界區(為了保證同一時間只有一個
Thread 取號)
    try
        //MySync.SynchronizeMethod(@CloseInvoPool);
        //result_str:=GetInvoRollStr(command_text, command_text2);
        ///MySync.SynchronizeMethod(@ShowInvoPool);
    finally
        AContext.Connection.IOHandler.WriteLine(result_str);
        LeaveCriticalSection(MyCs); //離開臨界區
    end;
end;

end;

3 : begin //收到 SND_FILE

    log_text:=client_ip+'/' + command_id2 + ' (SND_FILE):
command_id3='+command_id3+', command_text="'+command_text+'",
command_text2="'+command_text2';
    MySync.SynchronizeMethod(@AddLogEntry);

    filename:=command_text2;
    filesize:=StrToInt(command_id3);

    if (FileExists(filename)) then DeleteFile(filename);

    fstream:=TFileStream.Create(filename, fmCreate);
    while (filesize>fstream.Size) do begin
        if ((filesize-fstream.Size) >= 1024) then
            readcount := 1024
        else
            readcount := filesize - fstream.Size;

        //SetLength(buff3,readcount); //這裡可以不用加
        AContext.Connection.Socket.ReadBytes(buff3, readcount, false);

        Move(buff3[0],buff[0],readcount);
        fstream.WriteBuffer(buff, readcount);

    end;
    fstream.Free;

    log_text:=client_ip+'/' +command_id2+' (SND_FILE): SERVER 接收檔案完成
('+IntToStr(filesize)+' bytes)';
    MySync.SynchronizeMethod(@AddLogEntry);

end;

```

程式碼接下頁

```

4 : begin //收到 GET_FILE

    log_text:=client_ip+'/'+ command_id2 +' (GET_FILE):
command_id3='+command_id3+', command_text="'+command_text+"",
command_text2="'+command_text2;
    MySync.SynchronizeMethod(@AddLogEntry);

    TmpList:=TStringList.Create;
    _GetFileList2(TmpList, command_text);

    if (TmpList.Count=0) then begin
        filename:='';
    end else if (TmpList.Count=1) then begin
        filename:=TmpList.Strings[0];
    end else begin
        //壓縮檔案參考 http://wiki.freepascal.org/paszlib
        OurZipper := TZipper.Create;
        try
            filename:=FormatDateTime('hhnnsszzz', Now)+'.zip';
            OurZipper.FileName := filename;
            for i := 0 to TmpList.Count-1 do begin
                //OurZipper.Entries.AddFileEntry(TmpList.Strings[i],
TmpList.Strings[i]);
                //OurZipper.Entries.AddFileEntry(TmpList.Strings[i],
wincptoutf8(TmpList.Strings[i]));
                OurZipper.Entries.AddFileEntry(TmpList.Strings[i],
ExtractFileName(TmpList.Strings[i]));
            end;
            OurZipper.ZipAllFiles;
        finally
            OurZipper.Free;
        end;

    end;

    TmpList.Free;

    err_flag:=0;
    if (filename<>'') then begin

        fstream:=TFileStream.Create(filename, fmShareDenyWrite);
        filesize:=fstream.Size;

```

程式碼接下頁


```

        //回傳檔案資訊
AContext.Connection.IOHandler.WriteLine(ExtractFileName(filename));
        AContext.Connection.IOHandler.Write(filesize, true);

        log_text:='filename='+filename;
        MySync.SynchronizeMethod(@AddLogEntry);
        log_text:='filesize='+IntToStr(filesize);
        MySync.SynchronizeMethod(@AddLogEntry);

        //fstream.Position:=0;
        While (fstream.Position < filesize) do begin
            if ((filesize - fstream.Position) >= 1024) then
                readcount := 1024
            else
                readcount := filesize - fstream.Position;

            fstream.ReadBuffer(buff, readcount);
            SetLength(buff3, readcount);
            Move(buff[0], buff3[0], readcount);
            AContext.Connection.IOHandler.Write(buff3, readcount);

            //log_text:=client_ip+'/'+command_id2+' (GET_FILE):
send_bytes = '+IntToStr(readcount);
            //MySync.SynchronizeMethod(@AddLogEntry);

            end;
            fstream.Free;

        end else begin
            //回傳空白檔案資訊
            AContext.Connection.IOHandler.WriteLine('');
            AContext.Connection.IOHandler.Write(0, true);
        end;

        if (err_flag=1) then
            log_text:='GET_FILE 中斷('+client_ip+'/'+command_id2+')'
        else
            log_text:='GET_FILE 全部完成('+client_ip+'/'+command_id2+')';
        MySync.SynchronizeMethod(@AddLogEntry);

        end;
    end;

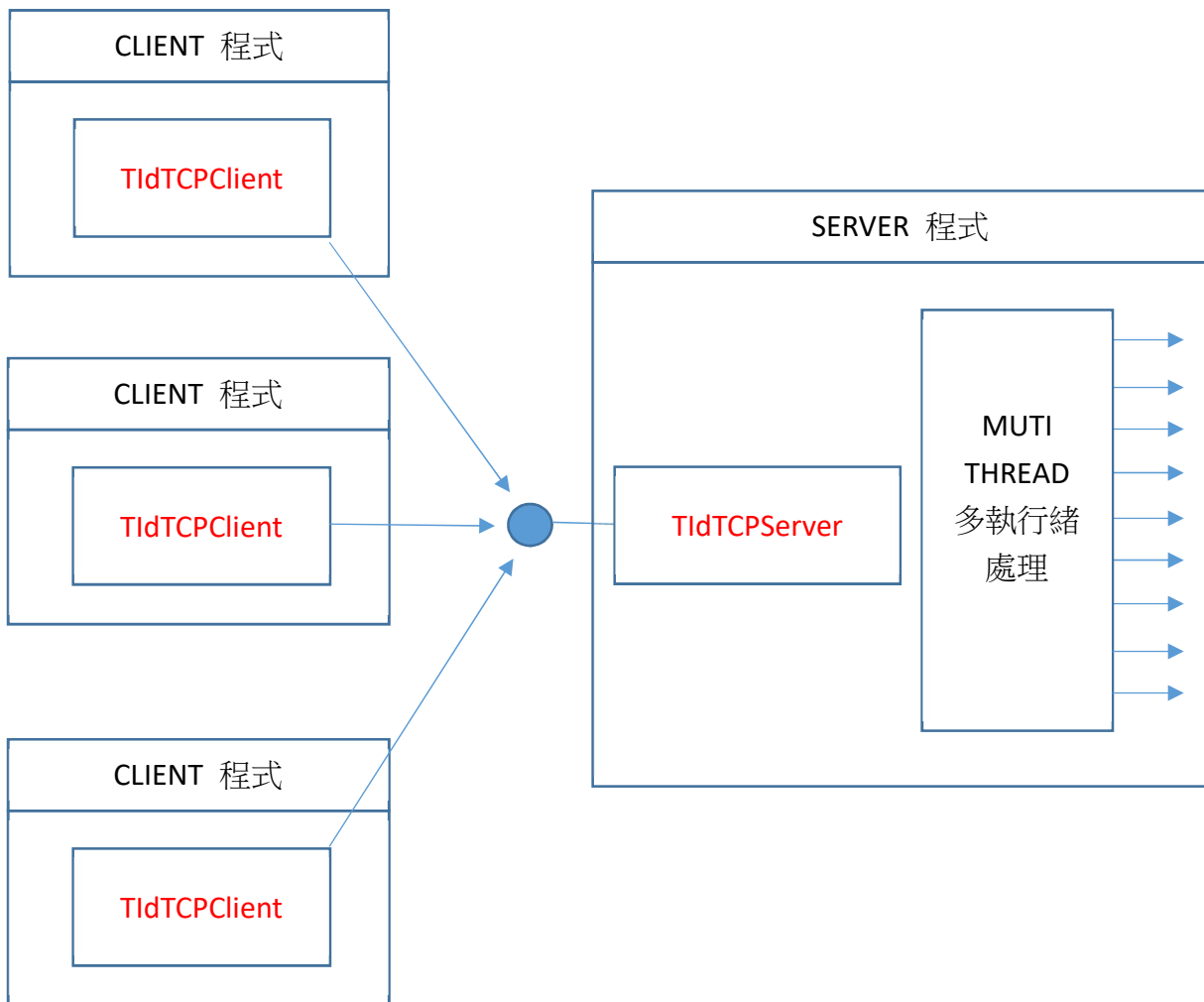
finally
    AContext.Connection.Disconnect();

    MySync.SynchronizeMethod(@DecrConnectioncount); //uses idsync; indy 10 要用
    這種寫法計算連線數
    MySync.Free;
    end;
end;

```

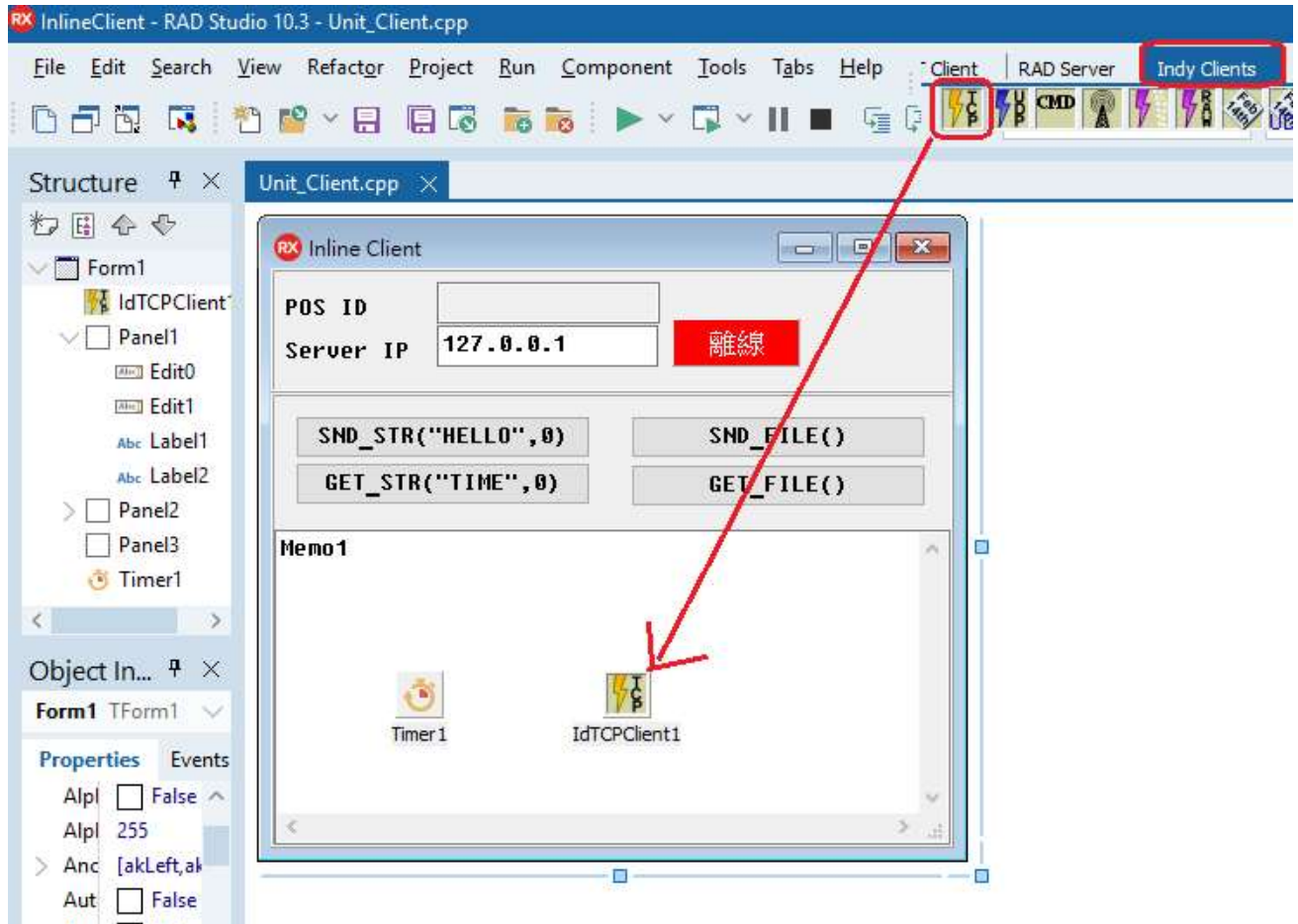
範例四 RAD Studio 10.3.1 Rio C++ Builder (使用內建 INDY 元件)

INDY TCP/IP 通訊元件很多東西都已幫你包裝好, 不用自己去創建 THREAD, 直接在 SERVER 元件(Control) 的事件中就可以處理每個 CLIENT 的 REQUEST, 程式碼相對也變得簡單許多; 在 RAD Studio 10.3.1 Rio C++ Builder 中, INDY TCP/IP 通訊元件是內建就有的元件 (TIdTCPClient / TIdTCPServer)



[CLIENT 端]

在 RAD Studio 10.3.1 Rio C++ Builder 中 INDY TCP/IP 通訊元件的使用方法完全同範例三 Lazarus 2.0 的使用方式，只是語法換成 C++



```

void __fastcall TForm1::SND_STR(String command_id2, String command_id3, String
command_text, String command_text2)
{
    if (IdTCPClient1->Connected){
        IdTCPClient1->Disconnect();
    }

    IdTCPClient1->Host = Edit1->Text;
    IdTCPClient1->Port = 6501;
    IdTCPClient1->ReadTimeout=1000;

    MemoLog("●SND_STR("+command_id2+", "+command_id3+", "+command_text+", "+command_t
ext2+") 開始");

    try{
        IdTCPClient1->Connect();
    }
    catch(...){
        MemoLog("連結失敗");
        connection_busy=false;
        MemoLog("○SND_STR() 結束");
        return;
    }

    try{
        IdTCPClient1->IOHandler->Write(LongInt(1),true); //Command_id => 1:SND_STR
2:GET_STR 3:SND_FILE 4:GET_FILE
        IdTCPClient1->IOHandler->WriteLn(command_id2); //command_id2
        IdTCPClient1->IOHandler->WriteLn(command_id3); //command_id3
        IdTCPClient1->IOHandler->WriteLn(command_text); //command_text
        IdTCPClient1->IOHandler->WriteLn(command_text2); //command_text2
    }
    __finally
    {
        IdTCPClient1->Disconnect();
        connection_busy=false;
        MemoLog("○SND_STR() 結束");
    }
}

```

```

String __fastcall TForm1::GET_STR(String command_id2, String command_id3,
String command_text, String command_text2)
{
    if (IdTCPClient1->Connected){
        IdTCPClient1->Disconnect();
    }

    IdTCPClient1->Host = Edit1->Text;
    IdTCPClient1->Port = 6501;
    IdTCPClient1->ReadTimeout=1000;

    MemoLog("●GET_STR("+command_id2+", "+command_id3+", "+command_text+", "+command_text2+") 開始");

    String rv="";

    try{
        IdTCPClient1->Connect();
    }
    catch(...){
        MemoLog("連結失敗");
        connection_busy=false;
        MemoLog("○GET_STR() 結束");
        return(rv);
    }

    try{
        IdTCPClient1->IOHandler->Write(LongInt(2),true); //Command_id => 1:SND_STR
2:GET_STR 3:SND_FILE 4:GET_FILE
        IdTCPClient1->IOHandler->WriteLn(command_id2); //command_id2
        IdTCPClient1->IOHandler->WriteLn(command_id3); //command_id3
        IdTCPClient1->IOHandler->WriteLn(command_text); //command_text
        IdTCPClient1->IOHandler->WriteLn(command_text2); //command_text2

        rv=IdTCPClient1->IOHandler->ReadLn();
        MemoLog("接收到回傳字串 : '"+rv+"'");

    }
    __finally
    {
        IdTCPClient1->Disconnect();
        connection_busy=false;
        MemoLog("○GET_STR() 結束");
    }

    return(rv);
}

```

```

String __fastcall TForm1::SND_FILE(String command_id2, String command_id3,
String command_text, String command_text2)
{
    if (IdTCPClient1->Connected){
        IdTCPClient1->Disconnect();
        if (IdTCPClient1->IOHandler!=NULL){
            IdTCPClient1->IOHandler->InputBuffer->Clear();
        }
    }

    IdTCPClient1->Host = Edit1->Text;
    IdTCPClient1->Port = 6501;
    IdTCPClient1->ReadTimeout=30000;

MemoLog ("●SND_FILE("+command_id2+", "+command_id3+", "+command_text+", "+command_t
ext2+") 開始");

    String rv="";

    try{
        IdTCPClient1->Connect();
    }
    catch(...){
        MemoLog ("連結失敗");
        connection_busy=false;
        MemoLog ("○SND_FILE() 結束");
        return(rv);
    }

    TFileStream *fstream = new TFileStream(command_text, fmShareDenyNone);
    String filename=ExtractFileName(command_text);
    LongInt filesize=fstream->Size;
    int readcount;
    byte buff[1024];
    TBytes buff2;
    TIdBytes buff3; //uses idGlobal

    try{

        IdTCPClient1->IOHandler->Write(LongInt(3),true); //Command_id => 1:SND_STR
2:GET_STR 3:SND_FILE 4:GET_FILE
        IdTCPClient1->IOHandler->WriteLn(command_id2); //command_id2
        IdTCPClient1->IOHandler->WriteLn(IntToStr(filesize)); //command_id3
        IdTCPClient1->IOHandler->WriteLn(command_text); //command_text
        IdTCPClient1->IOHandler->WriteLn(command_text2); //command_text2

        MemoLog ("發送檔案:"+command_text);

```

程式碼接下頁

```
fstream->Position=0;
while (fstream->Position < filesize){

    if ((filesize - fstream->Position) >= 1024){
        readcount = 1024;
    }else{
        readcount = filesize - fstream->Position;
    }

    buff3.set_length(readcount);
    fstream->ReadBuffer(buff3, readcount);

    IdTCPClient1->Socket->Write(buff3, readcount);

}

MemoLog("發送檔案完成 (" + IntToStr(filesize) + " bytes)");

rv="1"; //ok

}
__finally
{
    delete fstream;
    IdTCPClient1->Disconnect();
    connection_busy=false;
    MemoLog("oSND_FILE() 結束");
}

return(rv);

}
```

```

String __fastcall TForm1::GET_FILE(String command_id2, String command_id3,
String command_text, String command_text2)
{
    if (IdTCPClient1->Connected){
        IdTCPClient1->Disconnect();
        if (IdTCPClient1->IOHandler!=NULL){
            IdTCPClient1->IOHandler->InputBuffer->Clear();
        }
    }

    IdTCPClient1->Host = Edit1->Text;
    IdTCPClient1->Port = 6501;
    IdTCPClient1->ReadTimeout=30000;

MemoLog("●GET_FILE("+command_id2+", "+command_id3+", "+command_text+", "+command_t
ext2+") 開始");

    String rv="";

    try{
        IdTCPClient1->Connect();
    }
    catch(...){
        MemoLog("連結失敗");
        connection_busy=false;
        MemoLog("○GET_FILE() 結束");
        return(rv);
    }

    TFileStream *fstream;
    String filename=ExtractFileName(command_text);
    LongInt filesize;
    int readcount;
    byte buff[1024];
    TBytes buff2;
    TIdBytes buff3; //uses idGlobal
    String get_file_path;

    try{

        IdTCPClient1->IOHandler->Write(LongInt(4),true); //Command_id =>
1:SND_STR 2:GET_STR 3:SND_FILE 4:GET_FILE
        IdTCPClient1->IOHandler->WriteLn(command_id2); //command_id2
        IdTCPClient1->IOHandler->WriteLn(command_id3); //command_id3
        IdTCPClient1->IOHandler->WriteLn(command_text); //command_text
        IdTCPClient1->IOHandler->WriteLn(command_text2); //command_text2

        filename=IdTCPClient1->IOHandler->ReadLn(); //讀回檔名列
        filesize=IdTCPClient1->IOHandler->ReadLongInt(true); //讀回檔案 size 列

```

程式碼接下頁


```

//假如 Client 端已存在該檔案要先刪除
get_file_path=command_text2;
ForceDirectories(get_file_path);
filename=get_file_path+ "\\\" + ExtractFileName(filename);
if (FileExists(filename)) DeleteFile(filename);

//接收檔案
fstream = new TFileStream(filename, fmCreate); //接收端檔名路徑
MemoLog("接收檔案:"+filename);

while (filesize > fstream->Size){
    if ((filesize - fstream->Size) > 1024){
        readcount = 1024;
    }else{
        readcount = filesize - fstream->Size;
    }

    IdTCPClient1->Socket->ReadBytes(buff3, readcount, false);
    //Move(buff3[0],buff[0],readcount);
    fstream->WriteBuffer(buff3, readcount);
    Application->ProcessMessages();
}
delete fstream;

//解壓縮檔案
if (UpperCase(ExtractFileExt(filename))==".ZIP"){
    TZipFile * zip = new TZipFile(); // #include <System.Zip.hpp>
    zip->Open(filename, TZipMode::zmRead);

    ForceDirectories(command_text2);
    zip->ExtractAll(command_text2);

    zip->Close();
    delete zip;
    DeleteFile(filename);
}

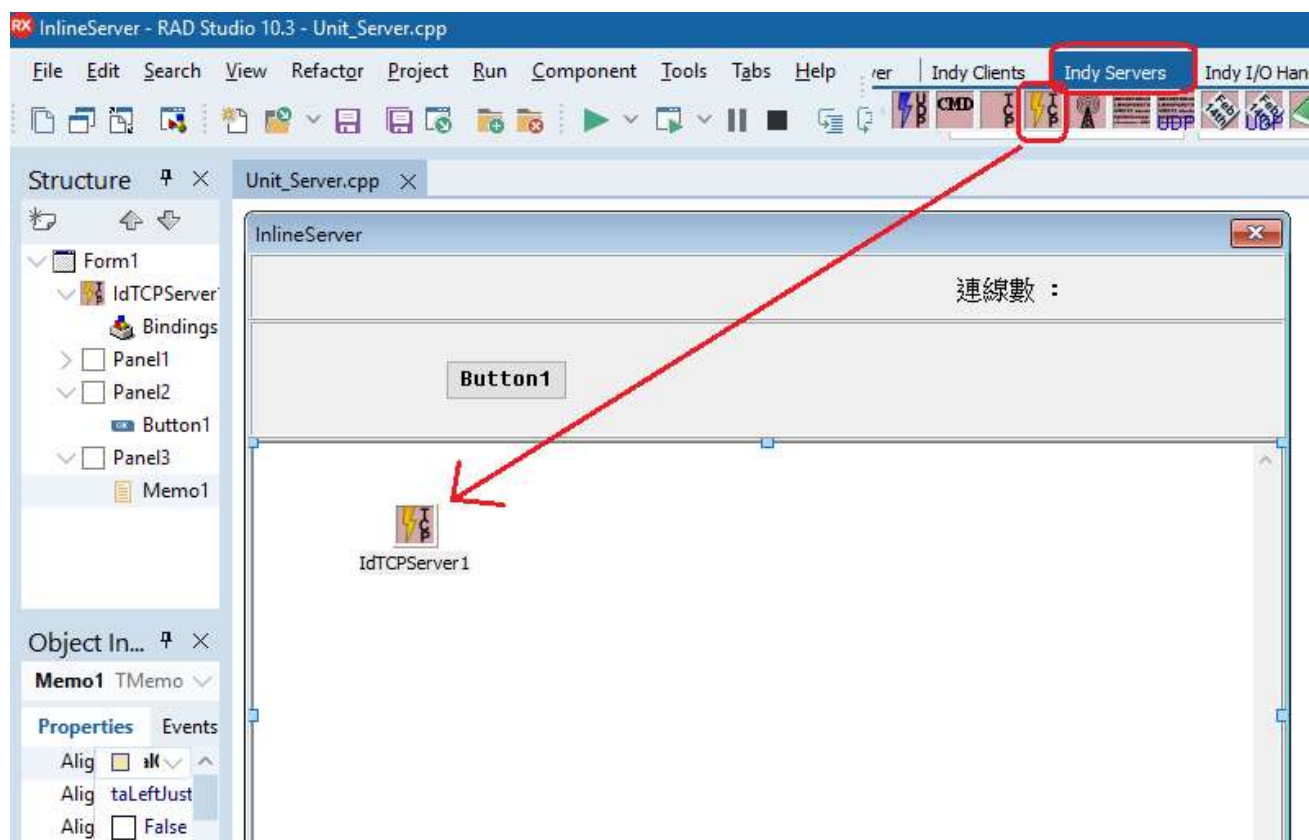
}
__finally{
    IdTCPClient1->Disconnect();
    connection_busy=false;
    MemoLog("oGET_FILE() 結束");
}

return(rv);
}

```

[SERVER 端]

拉一個 TIdServer 元件下來, 然後就可以開始寫 Server 程式



在 FormActive 時啟動 IdTCPServer，以及建立臨界區變數

```
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
bool run_once;
bool connection_busy;
TCriticalSection *MyCs; //臨界區

int connection_count;
String log_text;

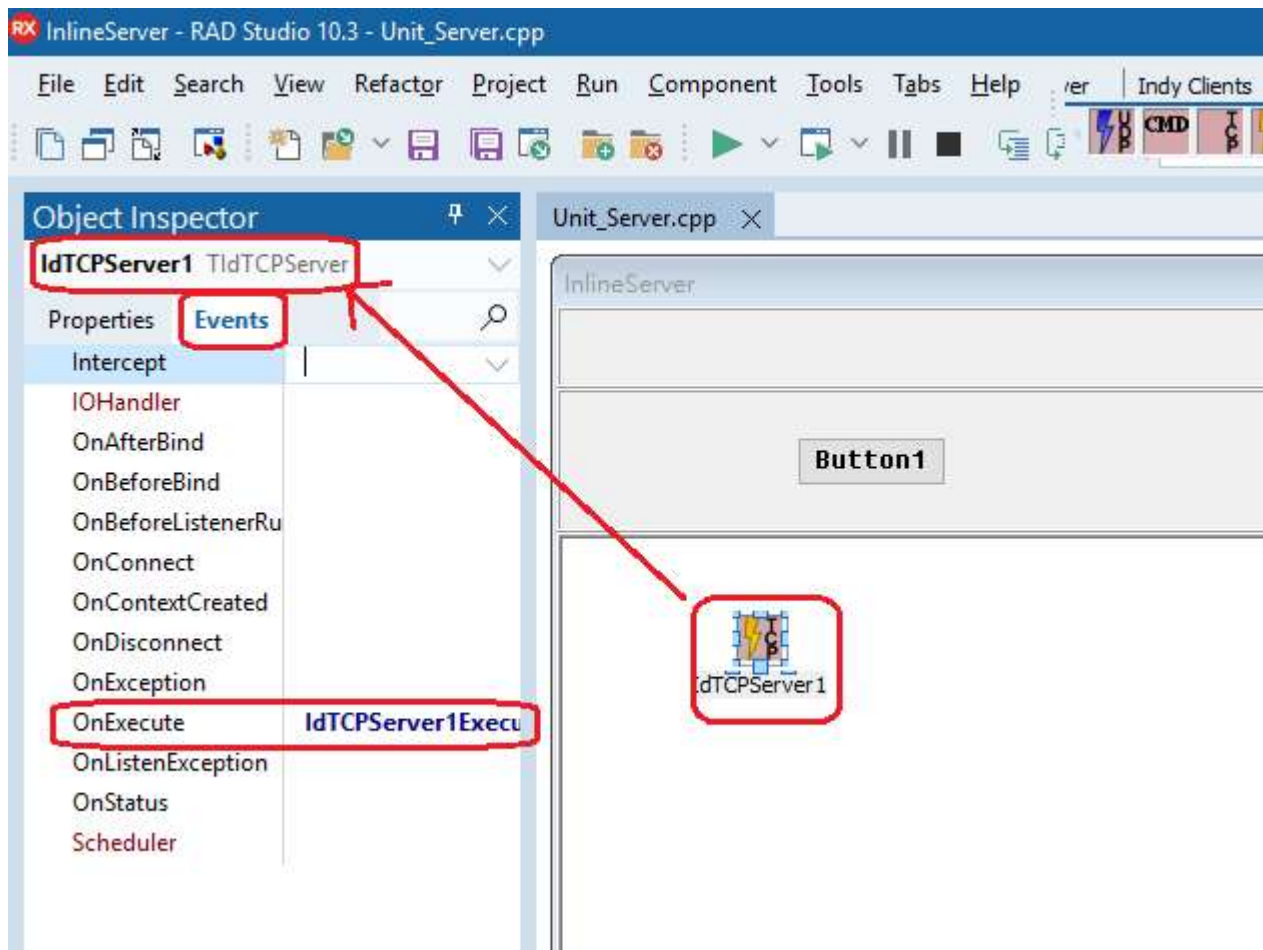
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TForm1::FormCreate(TObject *Sender)
{
    run_once=false;
    connection_busy=false;
    connection_count=0;

    //InitCriticalSection(MyCs); //初始化臨界區
    MyCs = new TCriticalSection();
}
//-----
void __fastcall TForm1::FormActivate(TObject *Sender)
{
    IdTCPServer1->DefaultPort= 6501;
    IdTCPServer1->Active=true;
}
//-----
void __fastcall TForm1::FormClose(TObject *Sender, TCloseAction &Action)
{
    try
    {
        TList *Lst = IdTCPServer1->Contexts->LockList();
        for(int i = 0; i < Lst->Count; i++)
        {
            TIdContext *AContext = reinterpret_cast<TIdContext *>(Lst->Items[i]);
            AContext->Connection->Disconnect();
        }
    }
    __finally
    {
        IdTCPServer1->Contexts->UnlockList();
    }

    IdTCPServer1->Active = false;

    //DoneCriticalSection(MyCs); //刪除臨界區
    delete MyCs;
}
}
```

當有 CLIENT 連上來時，我們不用自己寫 THREAD 來處理連線工作，直接在 IdTCPServer OnExecute 事件中撰寫連線處理程序即可



```

void __fastcall TForm1::IdTCPServer1Execute(TIdContext *AContext)
{
    int command_id;
    String command_id2;
    String command_id3;
    String command_text;
    String command_text2;
    String client_ip;
    String result_str;
    TIdSync *MySync; // #include <Idsync.hpp>

    TFileStream *fstream;
    String filename;
    LongInt filesize;
    int readcount;
    byte buff[1024];
    TIdBytes buff3; //uses idGlobal
    TStringList *TmpList;
    int err_flag;
    TZipFile *zip; // #include <System.Zip.hpp>

    client_ip=AContext->Connection->Socket->Binding->PeerIP; //同
    AContext.Binding.PeerIP;

    //更新連線數
    TIdSync::SynchronizeMethod(&IncrConnectioncount); // #include <Idsync.hpp>

    try
    {
        command_id = AContext->Connection->IOHandler->ReadLongInt(true);
        //Command_Id => 1:SND_STR 2:GET_STR 3:SND_FILE 4:GET_FILE
        command_id2 = AContext->Connection->IOHandler->ReadLn(); //pos id
        command_id3 = AContext->Connection->IOHandler->ReadLn(); //func id
        command_text= AContext->Connection->IOHandler->ReadLn(); //參數一
        command_text2 = AContext->Connection->IOHandler->ReadLn(); //參數二

        switch (command_id) {
            case 1 : //收到 SND_STR
                log_text=client_ip+"/"+ command_id2 +" (SND_STR):
command_id3="+command_id3+", command_text="+command_text+",
command_text2="+command_text2;
                TIdSync::SynchronizeMethod (&AddLogEntry);

                //-----
                //POS_CONNECT
                //-----
                if (command_id3=="POS_CONNECT") {
                    log_text="POS_CONNECT("+command_id2+)";
                    TIdSync::SynchronizeMethod (&AddLogEntry);
                }
            }
    }
}

```

程式碼接下頁

```

//-----
//POS_MSG
//-----
if (command_id3=="POS_MSG"){
    log_text=command_text+"("+command_id2+")";
    TIdSync::SynchronizeMethod(&AddLogEntry);
}

break;

case 2 : //收到 GET_STR
    log_text=client_ip+"/"+command_id2+" (GET_STR):
command_id3="+command_id3+", command_text="+command_text+",
command_text2="+command_text2;
    TIdSync::SynchronizeMethod(&AddLogEntry);

//-----
//取得 Server 時間
//-----
if (command_id3=="SRV_TIME"){
    AContext->Connection->IOHandler->WriteLn("SERVER TIME IS
"+FormatDateTime("hh:nn:ss", Now()));
}

//-----
//發票取號
//-----
if (command_id3=="GetInvoRollStr"){
    result_str="";
    MyCs->Acquire(); //進入臨界區(為了保證同一時間只有一個 Thread
取號)
    try{
        //MySync.SynchronizeMethod(@CloseInvoPool); //remark
on 2018/06/22 //2018/06/26 打開
        //result_str:=GetInvoRollStr(command_text,
command_text2); //參數 : 機號, 卷數
        ///MySync.SynchronizeMethod(@ShowInvoPool); //remark
on 2018/06/22
    }
    __finally{
        AContext->Connection->IOHandler->WriteLn(result_str);
        MyCs->Release(); //離開臨界區
    }
}

break;

```

程式碼接下頁

```
case 3 : //SND_FILE

        log_text=client_ip+"/"+command_id2 +" (SND_FILE):
command_id3="+command_id3+", command_text="+command_text+",
command_text2="+command_text2;
        TIdSync::SynchronizeMethod(&AddLogEntry);

        filename=command_text2;
        filesize=StrToInt(command_id3);

        if (FileExists(filename)){
            DeleteFile(filename);
        }

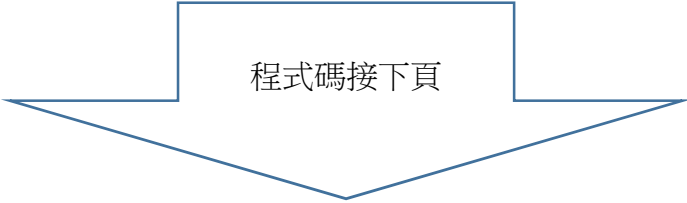
        fstream = new TFileStream(filename, fmCreate);
        while (filesize>fstream->Size){
            if ((filesize-fstream->Size) >= 1024){
                readcount = 1024;
            }else{
                readcount = filesize - fstream->Size;
            }

            AContext->Connection->Socket->ReadBytes(buff3, readcount,
false);

            fstream->WriteBuffer(buff3, readcount);
        }
        delete fstream;

        log_text=client_ip+"/"+command_id2+" (SND_FILE): SERVER 接收檔
案完成 (" +IntToStr(filesize)+" bytes)";
        TIdSync::SynchronizeMethod(&AddLogEntry);

        break;
```



程式碼接下頁

```
case 4 : //收到 GET_FILE

        log_text=client_ip+"/"+ command_id2 +" (GET_FILE):
command_id3="+command_id3+", command_text="+command_text+",
command_text2="+command_text2;
        TIdSync::SynchronizeMethod(&AddLogEntry);

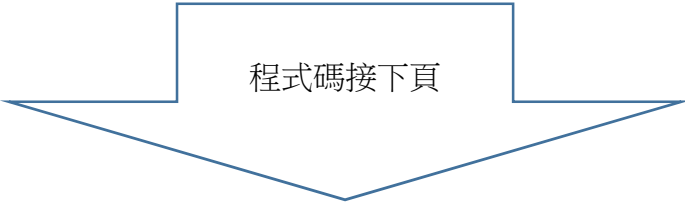
        TmpList = new TStringList;
        _GetFileList2(TmpList, command_text);

        if (TmpList->Count==0){
            filename="";
        }else if (TmpList->Count==1){
            filename=TmpList->Strings[0];
        }else{
            //壓縮檔案
            zip = new TZipFile(); // #include <System.Zip.hpp>
            filename=FormatDateTime("hhnnsszzz", Now())+".zip";

            //壓縮整個目錄
            //zip->ZipDirectoryContents(filename, command_text);

            //一個一個撈出來 Add 也可
            zip->Open(filename, TZipMode::zmWrite);
            for (int i=0; i <= TmpList->Count-1 ; i++) {
                zip->Add(TmpList->Strings[i], ExtractFileName(TmpList-
>Strings[i]));
            }
            zip->Close();
            delete zip;
        }
        delete TmpList;

        err_flag=0;
        if (filename!=""){
            fstream = new TFileStream(filename, fmShareDenyNone);
            filesize=fstream->Size;
        }
    }
```



程式碼接下頁


```

        //回傳檔案資訊
        AContext->Connection->IOHandler-
>WriteLn(ExtractFileName(filename));
        AContext->Connection->IOHandler->Write(filesize, true);

        log_text="filename="+filename;
        TIdSync::SynchronizeMethod(&AddLogEntry);
        log_text="filesize="+IntToStr(filesize);
        TIdSync::SynchronizeMethod(&AddLogEntry);

        fstream->Position=0;
        while (fstream->Position < filesize){
            if ((filesize - fstream->Position) >= 1024){
                readcount = 1024;
            }else{
                readcount = filesize - fstream->Position;
            }

            buff3.set_length(readcount);
            fstream->ReadBuffer(buff3, readcount);
            AContext->Connection->IOHandler->Write(buff3,
readcount);

            Application->ProcessMessages();
        }
        delete fstream;

    }else{
        //回傳空白檔案資訊
        AContext->Connection->IOHandler->WriteLn("");
        AContext->Connection->IOHandler->Write(0, true);
    }

    if (err_flag==1){
        log_text="GET_FILE 中斷("+client_ip+"/"+command_id2+)";
    }else{
        log_text="GET_FILE 全部完成
("+client_ip+"/"+command_id2+)";
    }

    TIdSync::SynchronizeMethod(&AddLogEntry);

    break;

    default:
        ;
    }

}
__finally
{
    AContext->Connection->Disconnect();

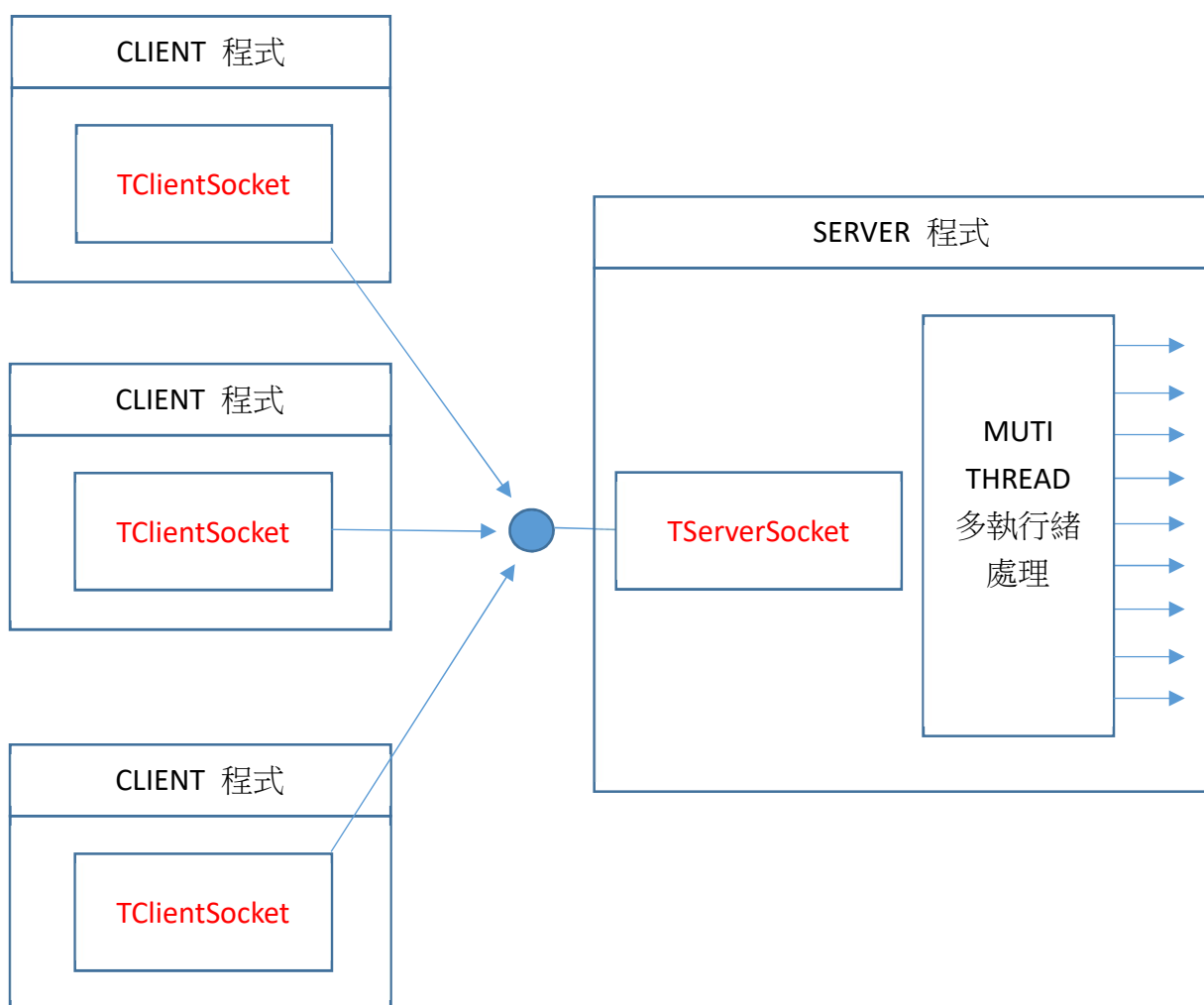
    //更新連線數
    TIdSync::SynchronizeMethod(&DecrConnectioncount); // #include <Idsync.hpp>
}
}

```

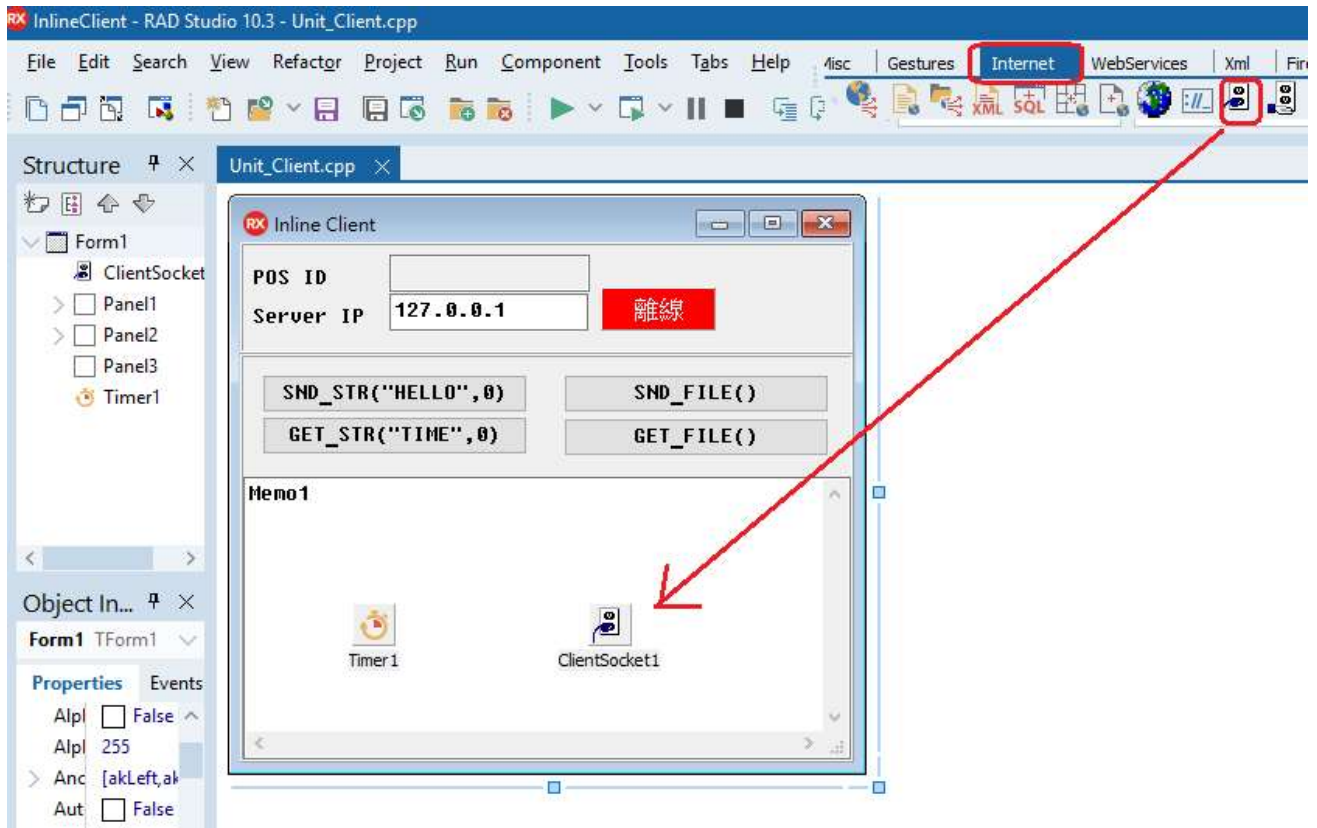
範例五 RAD Studio 10.3.1 Rio C++ Builder (使用 TClientSocket / TServerSocket 元件)

TClientSocket / TServerSocket 兩個元件是為相容於舊版 C++ Builder 而存在, Default 並未放在元件盤上, 需要自行手動掛上 ; 在 RAD Studio 10.3.1 Rio C++ Builder 的 IDE 中, 主選單 -> Project -> Option... -> 選到 Packages 然後按下 “Add” 按鈕

找到 C:\Program Files (x86)\Embarcadero\Studio\20.0\bin\dclsockets260.bpl 後 [開啟] (加入) 然後按下 [SAVE], 回到 IDE, 就會發現元件盤上 “Internet” 頁籤內多了兩個元件 TClientSocket & TServerSocket ; 不過使用上沒有 INDY 元件那麼方便, 要自己手動建立 THREAD 來處理每個 CLIENT 的連線 REQUEST



[CLIENT 端]



關於使用 ClientSocket1 收送資料的方式：

1. ClientSocket1->Socket->SendText()
多批 SendText() 資料到 Server 端時，資料會攪在一起, Server 端無法分辨
2. ClientSocket1->Socket->SendBuf(buff, 1024);
傳送固定長度, 但處理 SND_FILE() 時, 接收端收不到資料(不穩定)
3. help 建議 [TWinSocketStream](#) to read or write information over a blocking socket connection

```

void __fastcall TForm1::SND_STR(String command_id2, String command_id3, String
command_text, String command_text2)
{
    if (ClientSocket1->Active){
        ClientSocket1->Close();
    }

    ClientSocket1->Address = Edit1->Text;
    ClientSocket1->Port = 6501;
    ClientSocket1->ClientType = ctBlocking;

MemoLog ("●SND_STR("+command_id2+", "+command_id3+", "+command_text+", "+command_t
ext2+") 開始");

    try{
        ClientSocket1->Open();
    }
    catch(...){
        MemoLog ("連結失敗");
        connection_busy=false;
        MemoLog ("○SND_STR() 結束");
        return;
    }

    try{

        int TIMEOUT=2000;

        TWInSocketStream *netStream = new TWInSocketStream(ClientSocket1->Socket,
TIMEOUT);
        byte buff[1024];

        memset(buff, 0, 1024); //要加這個, 不然對方可能收到最後一碼有垃圾資料

        String tmp = "1\r\n" +
            command_id2 + "\r\n" +
            command_id3 + "\r\n" +
            command_text + "\r\n" +
            command_text2 + "\r\n";

        memcpy(&buff[0], &tmp.BytesOf()[0], tmp.Length());

        netStream->Write(buff, 1024);

        delete netStream;

    }
    __finally
    {
        ClientSocket1->Close();
        connection_busy=false;
        MemoLog ("○SND_STR() 結束");
    }
}

```

```

String __fastcall TForm1::GET_STR(String command_id2, String command_id3,
String command_text, String command_text2)
{
    if (ClientSocket1->Active){
        ClientSocket1->Close();
    }
    ClientSocket1->Address = Edit1->Text;
    ClientSocket1->Port = 6501;
    ClientSocket1->ClientType = ctBlocking;
    MemoLog("●GET_STR("+command_id2+", "+command_id3+", "+command_text+", "+command_text2+") 開始");

    String rv="";
    try{
        ClientSocket1->Open();
    }
    catch(...){
        MemoLog("連結失敗");
        connection_busy=false;
        MemoLog("○GET_STR() 結束");
        return(rv);
    }
    try{
        int TIMEOUT=2000;
        TWinSocketStream *netStream = new TWinSocketStream(ClientSocket1->Socket,
TIMEOUT);
        byte buff[1024];
        memset(buff, 0, 1024); //要加這個，不然對方可能收到最後一碼有垃圾資料
        String tmp = "2\r\n" +
                    command_id2 + "\r\n" +
                    command_id3 + "\r\n" +
                    command_text + "\r\n" +
                    command_text2 + "\r\n";
        memcpy(&buff[0], &tmp.BytesOf()[0], tmp.Length());
        netStream->Write(buff, 1024);

        //讀回
        byte read_buff[1024];
        int byte_length=sizeof(read_buff) / sizeof(read_buff[0]); //c++ 沒有取 byte
陣列長度的函式，只能這樣計算
        //ShowMessage(IntToStr(byte_length));
        memset(read_buff, 0, 1024); // initialize the buffer
        if (netStream->WaitForData(TIMEOUT)){
            if (netStream->Read(read_buff, byte_length) !=0){
                rv=StringOf(read_buff);
                MemoLog("接收到回傳字串 : '"+ rv + "'"); //奇怪，最後一個 "" 顯示不出來
            }
        }
        delete netStream;
    }
    __finally
    {
        ClientSocket1->Close();
        connection_busy=false;
        MemoLog("○GET_STR() 結束");
    }
    return(rv);
}

```

```

String __fastcall TForm1::SND_FILE(String command_id2, String command_id3,
String command_text, String command_text2)
{
    if (ClientSocket1->Active){
        ClientSocket1->Close();
    }

    ClientSocket1->Address = Edit1->Text;
    ClientSocket1->Port = 6501;
    ClientSocket1->ClientType = ctBlocking;

MemoLog ("●SND_FILE("+command_id2+", "+command_id3+", "+command_text+", "+command_t
ext2+") 開始");
    String rv="";

    try{
        ClientSocket1->Open();
    }
    catch(...){
        MemoLog ("連結失敗");
        connection_busy=false;
        MemoLog ("○SND_FILE() 結束");
        return(rv);
    }

    TFileStream *fstream = new TFileStream(command_text, fmShareDenyNone);
    String filename=ExtractFileName(command_text);
    LongInt filesize=fstream->Size;
    int readcount;
    byte buff[1024];
    TBytes buff2;
    //TIdBytes buff3; //uses idGlobal

    try{

        int TIMEOUT=2000;

        TWinSocketStream *netStream = new TWinSocketStream(ClientSocket1->Socket,
TIMEOUT);
        byte buff[1024];
        memset(buff, 0, 1024); //要加這個, 不然對方可能收到最後一碼有垃圾資料

        String tmp = "3\r\n" +
            command_id2 + "\r\n" +
            IntToStr(filesize) + "\r\n" +
            command_text + "\r\n" +
            command_text2 + "\r\n";

        memcpy(&buff[0], &tmp.BytesOf()[0], tmp.Length());
        netStream->Write(buff, 1024);

```

程式碼接下頁

```
MemoLog("發送檔案:"+command_text);

fstream->Position=0;
while (fstream->Position < filesize){

    if ((filesize - fstream->Position) >= 1024){
        readcount = 1024;
    }else{
        readcount = filesize - fstream->Position;
    }

    buff2.set_length(readcount);
    fstream->ReadBuffer(buff2, readcount);
    netStream->Write(buff2, readcount);
}
delete netStream;

MemoLog("發送檔案完成 (" +IntToStr(filesize)+" bytes)");

rv="1"; //ok
}
__finally
{
    delete fstream;
    ClientSocket1->Close();
    connection_busy=false;
    MemoLog("oSND_FILE() 結束");
}
return(rv);
}
```

```
String __fastcall TForm1::GET_FILE(String command_id2, String command_id3,
String command_text, String command_text2)
{
    if (ClientSocket1->Active){
        ClientSocket1->Close();
    }

    ClientSocket1->Address = Edit1->Text;
    ClientSocket1->Port = 6501;
    ClientSocket1->ClientType = ctBlocking;

    MemoLog("●GET_FILE("+command_id2+", "+command_id3+", "+command_text+", "+command_
text2+") 開始");

    String rv="";

    try{
        ClientSocket1->Open();
    }
    catch(...){
        MemoLog("連結失敗");
        connection_busy=false;
        MemoLog("○GET_FILE() 結束");
        return(rv);
    }

    TFileStream *fstream;
    String filename=ExtractFileName(command_text);
    LongInt filesize;
    int readcount;
    byte buff[1024];
    TBytes buff2;
    //TIdBytes buff3; //uses idGlobal
    String get_file_path;

    try{
        int TIMEOUT=2000;

        TwinSocketStream *netStream = new TwinSocketStream(ClientSocket1->Socket,
TIMEOUT);
        byte buff[1024];

        memset(buff, 0, 1024); //要加這個, 不然對方可能收到最後一碼有垃圾資料
```



程式碼接下頁


```
String tmp = "4\r\n" +
    command_id2 + "\r\n" +
    command_id3 + "\r\n" +
    command_text + "\r\n" +
    command_text2 + "\r\n";

memcpy(&buff[0], &tmp.BytesOf()[0], tmp.Length());
netStream->Write(buff, 1024);

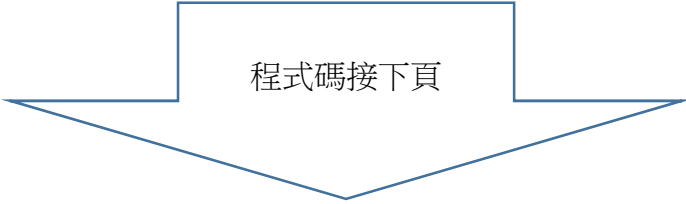
memset(buff, 0, 1024); // initialize the buffer
if (netStream->WaitForData(TIMEOUT))
{
    if (netStream->Read(buff, 1024) == 1024)
    {
        //拆分字串
        TStringList *lstLine = new TStringList;
        lstLine->Text = StringOf(buff);
        filename=lstLine->Strings[0]; //讀回檔名列
        filesize=StrToInt(lstLine->Strings[1]); //讀回檔案 size 列
        delete lstLine;
    }
}

//假如 Client 端已存在該檔案要先刪除
get_file_path=command_text2;
ForceDirectories(get_file_path);
filename=get_file_path+ "\\\" + ExtractFileName(filename);
if (FileExists(filename)) DeleteFile(filename);

//接收檔案
fstream = new TFileStream(filename, fmCreate); //接收端檔名路徑
MemoLog("接收檔案:"+filename);

while (filesize > fstream->Size){
    if ((filesize - fstream->Size) > 1024){
        readcount = 1024;
    }else{
        readcount = filesize - fstream->Size;
    }

    if (netStream->WaitForData(TIMEOUT)){
        buff2.set_length(readcount);
        if (netStream->Read(buff2, readcount) == readcount){
            fstream->WriteBuffer(buff2, readcount);
        }
    }
}
delete fstream;
```



程式碼接下頁

```
//解壓縮檔案
if (UpperCase(ExtractFileExt(filename))==".ZIP"){
    TZipFile * zip = new TZipFile(); // #include <System.Zip.hpp>
    zip->Open(filename, TZipMode::zmRead);

    ForceDirectories(command_text2);
    zip->ExtractAll(command_text2);

    zip->Close();
    delete zip;
    DeleteFile(filename);
}

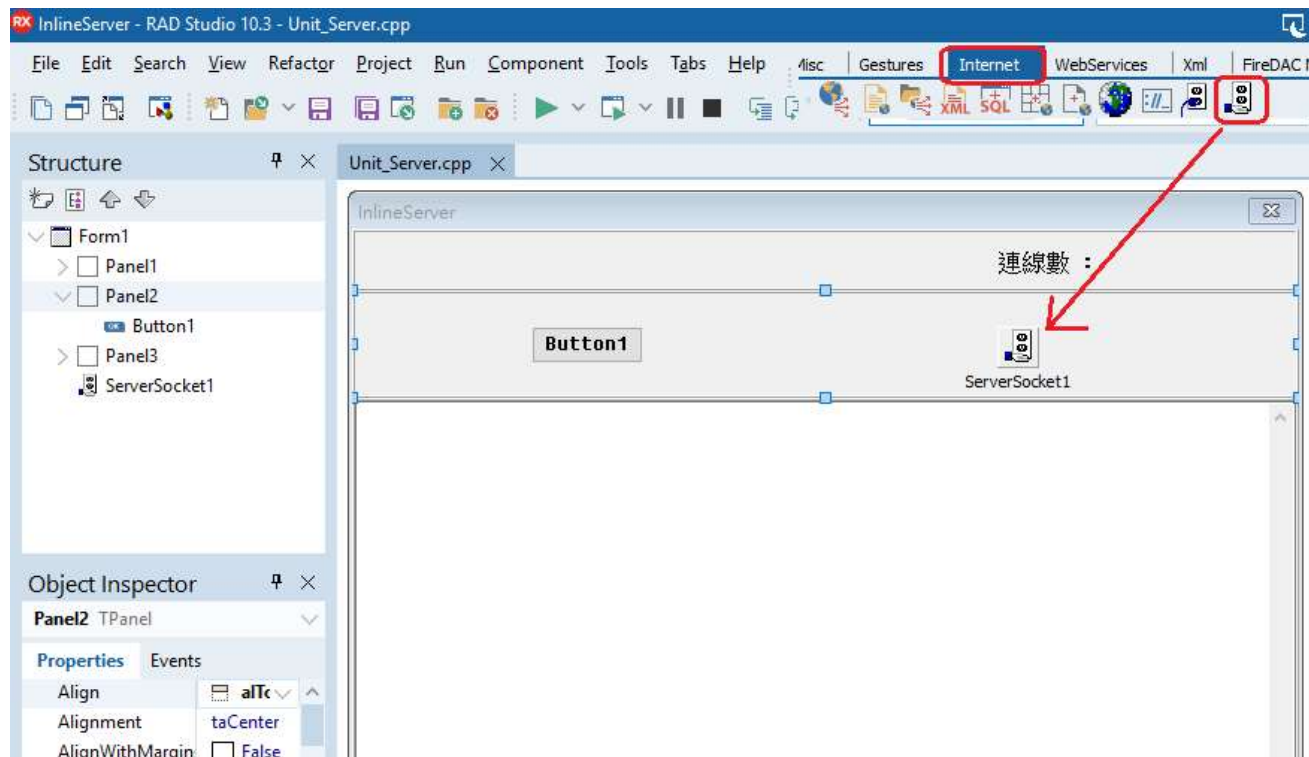
}

__finally{
    ClientSocket1->Close();
    connection_busy=false;
    MemoLog("oGET_FILE() 結束");
}

return(rv);
}
```

[SERVER 端]

拉一個 ServerSocket 元件下來, 然後就可以開始寫 Server 程式, 不過必須幫它手動建立一個 SrvThread 執行緒類別, 有些麻煩, 測試好久才成功



在 *.H 檔中建立 SrvThread 類別

```
//-----  
#ifndef Unit_ServerH  
#define Unit_ServerH  
//-----  
#include <System.Classes.hpp>  
#include <Vcl.Controls.hpp>  
#include <Vcl.StdCtrls.hpp>  
#include <Vcl.Forms.hpp>  
#include <Vcl.ExtCtrls.hpp>  
#include <System.Zip.hpp>  
#include <System.Win.ScktComp.hpp>  
//-----  
class TForm1 : public TForm  
{  
    __published:      // IDE-managed Components  
        TServerSocket *ServerSocket1;  
        void __fastcall FormCreate(TObject *Sender);  
        void __fastcall FormActivate(TObject *Sender);  
        void __fastcall FormClose(TObject *Sender, TCloseAction &Action);  
        void __fastcall ServerSocket1GetThread(TObject *Sender,  
TServerClientWinSocket *ClientSocket, TServerClientThread *&SocketThread);  
private:      // User declarations  
public:      // User declarations  
    __fastcall TForm1(TComponent* Owner);  
    void __fastcall AddLogEntry();  
    void __fastcall IncrConnectioncount();  
    void __fastcall DecrConnectioncount();  
    void __fastcall _GetFileList2(TStrings *AStrings, String AFindFile);  
};  
  
//BCB ServerSocket 多线程示例  
//https://blog.csdn.net/u014438664/article/details/44735073  
class SrvThread : public TServerClientThread  
{  
protected:  
    //重载 TServerClientThread 的 ClientExecute 而不是 TThread 的 Execute  
    void __fastcall ClientExecute(void);  
public:  
    __fastcall SrvThread(bool CreateSuspended, TServerClientWinSocket*  
ASocket);  
};  
  
//-----  
extern PACKAGE TForm1 *Form1;  
//-----  
#endif
```

在 `FormActive` 時啟動 `ServerSocket`, 建立臨界區變數, 以及名為 `SocketThread` 的 `SrvThread` 類別

```
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
bool run_once;
bool connection_busy;

TCriticalSection *MyCs; //臨界區

int connection_count;
String log_text;

SrvThread *SocketThread;

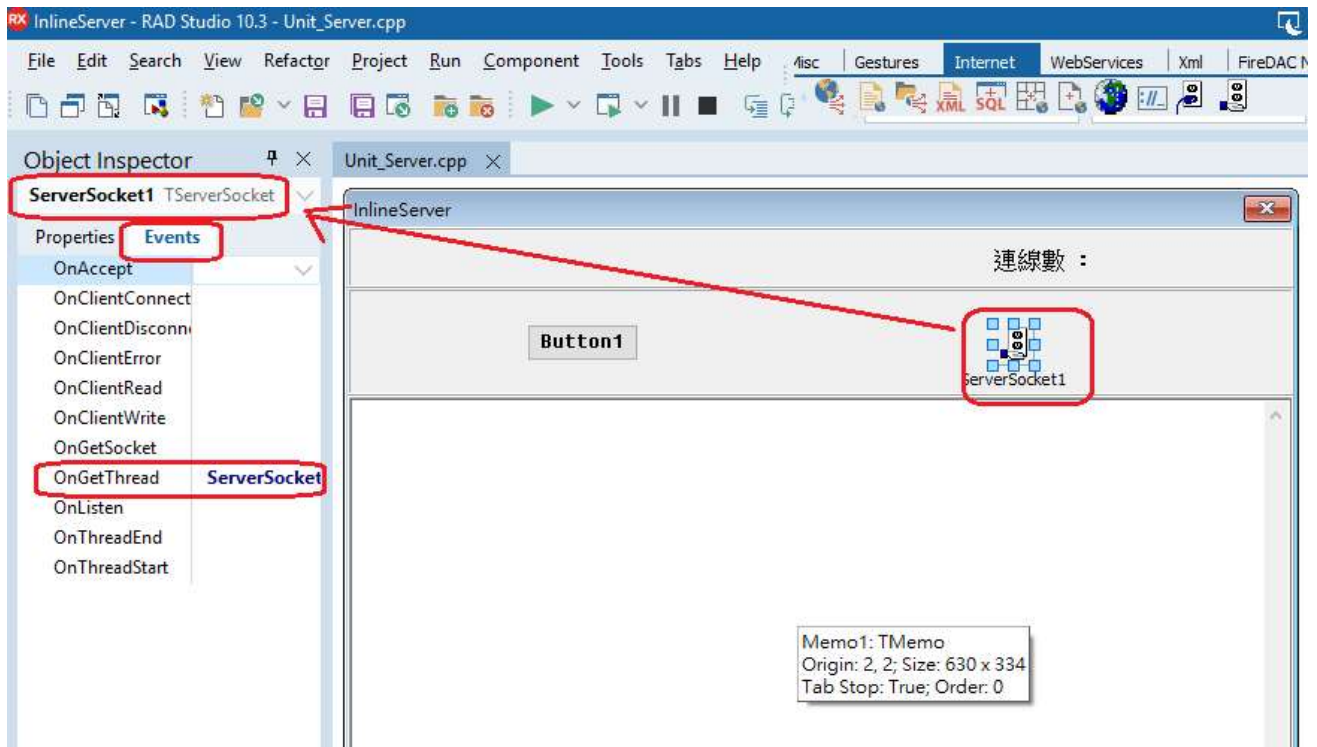
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TForm1::FormCreate(TObject *Sender)
{
    run_once=false;
    connection_busy=false;
    connection_count=0;

    //初始化臨界區
    MyCs = new TCriticalSection();
}
//-----
void __fastcall TForm1::FormActivate(TObject *Sender)
{
    ServerSocket1->Port = 6501;
    ServerSocket1->ServerType = stThreadBlocking;
    ServerSocket1->Open();
}
//-----
void __fastcall TForm1::FormClose(TObject *Sender, TCloseAction &Action)
{
    if (SocketThread != NULL) SocketThread->Terminate();

    ServerSocket1->Close();

    //刪除臨界區
    delete MyCs;
}
```

當有 CLIENT 連上來時，會觸發 ServerSocket 的 OnGetThread 事件，但事件裡面還要再 new 出一個 THREAD，而不是直接在 OnGetThread 事件內處理 CLIENT，有些麻煩



在 OnGetThread 事件內 new 出一個 THREAD

```
void __fastcall TForm1::ServerSocket1GetThread(TObject *Sender,
TServerClientWinSocket *ClientSocket,
    TServerClientThread *&SocketThread)
{
    //new 出另一個 THREAD
    SocketThread = new SrvThread(false, ClientSocket);
}
```

```

//-----
//拿掉會錯誤
__fastcall SrvThread::SrvThread(bool CreateSuspended, TServerClientWinSocket*
ASocket)
    : TServerClientThread(CreateSuspended, ASocket)
{
}

//-----
void __fastcall SrvThread::ClientExecute()
{
    int command_id=0;
    String command_id2;
    String command_id3;
    String command_text;
    String command_text2;
    String client_ip;
    String result_str;
    //TIdSync *MySync; // #include <Idsync.hpp>

    TFileStream *fstream;
    String filename;
    LongInt filesize;
    int readcount;
    byte buff[1024];
    TBytes buff2;
    //TIdBytes buff3; //uses idGlobal
    TStringList *TmpList;
    int err_flag;
    TZipFile *zip; // #include <System.Zip.hpp>
    TWinSocketStream *netStream; //2019/04/19

    client_ip=ClientSocket->RemoteAddress;

    Synchronize(Form1->IncrConnectioncount);

    try
    {
        int TIMEOUT=60000;
        netStream = new TWinSocketStream(ClientSocket, TIMEOUT);

        memset(buff, 0, 1024); // initialize the buffer

        // give the client 60 seconds to start writing
        if (netStream->WaitForData(TIMEOUT))
        {
            if (netStream->Read(buff, 1024) == 1024)
            {

```

程式碼接下頁

```

//拆分字串
TStringList *lstLine = new TStringList;
lstLine->Clear();
lstLine->Text = StringOf(buff);
command_id = StrToInt(lstLine->Strings[0]); //Command_Id =>
1:SND_STR 2:GET_STR 3:SND_FILE 4:GET_FILE
command_id2 = lstLine->Strings[1]; //pos id
command_id3 = lstLine->Strings[2]; //func id
command_text= lstLine->Strings[3]; //參數一
command_text2 = lstLine->Strings[4]; //參數二
delete lstLine;

//debug;
log_text="" +command_text2+"";
Synchronize(Form1->AddLogEntry);
}
}

switch (command_id) {
case 1 : //收到 SND_STR
log_text=client_ip+"/" + command_id2 + " (SND_STR):
command_id3="+command_id3+", command_text="+command_text+",
command_text2="+command_text2;
Synchronize (Form1->AddLogEntry);

//-----
//POS_CONNECT
//-----
if (command_id3=="POS_CONNECT"){
log_text="POS_CONNECT("+command_id2+"";
Synchronize (Form1->AddLogEntry);
}

//-----
//POS_MSG
//-----
if (command_id3=="POS_MSG"){
log_text=command_text+"("+command_id2+"";
Synchronize (Form1->AddLogEntry);
}

break;

```

程式碼接下頁


```

        case 2 : //收到 GET_STR
            log_text=client_ip+"/"+ command_id2 +" (GET_STR):
command_id3="+command_id3+", command_text="+command_text+",
command_text2="+command_text2;
            Synchronize (Form1->AddLogEntry);

            //-----
            //取得 Server 時間
            //-----
            if (command_id3=="SRV_TIME"){

                String str="主機時間 SERVER TIME IS
"+FormatDateTime ("hh:nn:ss", Now());

                netStream->Write(str.BytesOf(), ByteLength(str));
            }

            //-----
            //發票取號
            //-----
            if (command_id3=="GetInvoRollStr"){
                result_str="";
                //進入臨界區(為了保證同一時間只有一個 Thread 取號)
                MyCs->Acquire();
                try{
                    //MySync.SynchronizeMethod(@CloseInvoPool); //remark
on 2018/06/22 //2018/06/26 打開
                    //result_str:=GetInvoRollStr(command_text,
command_text2); //參數 : 機號, 卷數
                    ///MySync.SynchronizeMethod(@ShowInvoPool); //remark
on 2018/06/22
                }
                __finally{
                    ClientSocket->SendText(result_str);
                    //離開臨界區
                    MyCs->Release();
                }
            }

            break;

```

程式碼接下頁

```
case 3 : //SND_FILE

        log_text=client_ip+"/"+command_id2+" (SND_FILE):
command_id3="+command_id3+", command_text="+command_text+",
command_text2="+command_text2;
        Synchronize (Form1->AddLogEntry);

        filename=command_text2;
        filesize=StrToInt (command_id3);

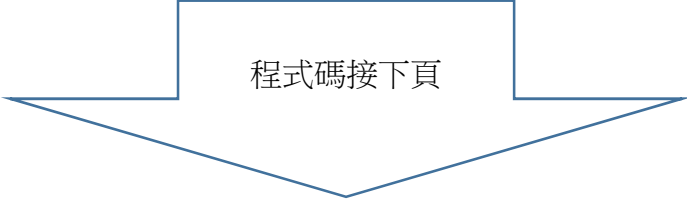
        if (FileExists (filename)) {
            DeleteFile (filename);
        }

        fstream = new TFileStream (filename, fmCreate);
        while (filesize>fstream->Size) {
            if ((filesize-fstream->Size) >= 1024) {
                readcount = 1024;
            } else {
                readcount = filesize - fstream->Size;
            }

            if (netStream->WaitForData (TIMEOUT)) {
                buff2.set_length (readcount);
                if (netStream->Read (buff2, readcount) == readcount) {
                    fstream->WriteBuffer (buff2, readcount);
                }
            }
        }
        delete fstream;

        log_text=client_ip+"/"+command_id2+" (SND_FILE): SERVER 接收檔
案完成 (" +IntToStr (filesize)+" bytes)";
        Synchronize (Form1->AddLogEntry);

        break;
```



程式碼接下頁

```
case 4 : //收到 GET_FILE

        log_text=client_ip+"/"+ command_id2 +" (GET_FILE):
command_id3="+command_id3+", command_text="+command_text+",
command_text2="+command_text2;
        Synchronize(Form1->AddLogEntry);

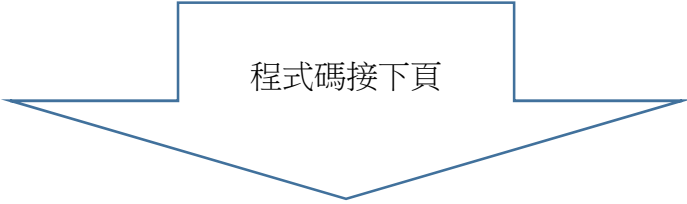
        TmpList = new TStringList;
        Form1->_GetFileList2(TmpList, command_text);

        if (TmpList->Count==0){
            filename="";
        }else if (TmpList->Count==1){
            filename=TmpList->Strings[0];
        }else{
            //壓縮檔案
            zip = new TZipFile(); // #include <System.Zip.hpp>
            filename=FormatDateTime("hhnnsszzz", Now())+".zip";

            //壓縮整個目錄
            //zip->ZipDirectoryContents(filename, command_text);

            //一個一個撈出來 Add 也可
            zip->Open(filename, TZipMode::zmWrite);
            for (int i=0; i <= TmpList->Count-1 ; i++) {
                zip->Add(TmpList->Strings[i], ExtractFileName(TmpList-
>Strings[i]));
            }
            zip->Close();
            delete zip;
        }
        delete TmpList;

        err_flag=0;
        if (filename!=""){
            fstream = new TFileStream(filename, fmShareDenyNone);
            //fmOpenRead(會獨佔), fmShareDenyWrite, fmShareDenyNone
            filesize=fstream->Size;
```



程式碼接下頁

```
//回傳檔案資訊
String tmp = ExtractFileName(filename)+ "\r\n" +
            IntToStr(filesize)+ "\r\n";

memset(buff, 0, 1024); // initialize the buffer
memcpy(&buff[0], &tmp.BytesOf()[0], tmp.Length());

netStream->Write(buff, 1024);

log_text="filename="+filename;
Synchronize(Form1->AddLogEntry);
log_text="filesize="+IntToStr(filesize);
Synchronize(Form1->AddLogEntry);

//fstream -> netStream
fstream->Position=0;
while (fstream->Position < filesize){
    if ((filesize - fstream->Position) >= 1024){
        readcount = 1024;
    }else{
        readcount = filesize - fstream->Position;
    }

    buff2.set_length(readcount);
    fstream->ReadBuffer(buff2, readcount);
    netStream->Write(buff2, readcount);

}
delete fstream;

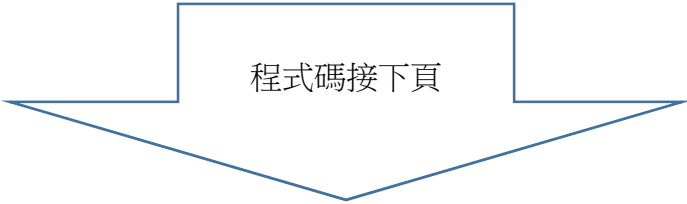
}else{

//回傳空白檔案資訊
String tmp = "\r\n0\r\n";

memset(buff, 0, 1024); // initialize the buffer
memcpy(&buff[0], &tmp.BytesOf()[0], tmp.Length());

netStream->Write(buff, 1024);

}
```



程式碼接下頁

```
        if (err_flag==1){
            log_text="GET_FILE 中斷("+client_ip+"/"+command_id2+)";
        }else{
            log_text="GET_FILE 全部完成
("+client_ip+"/"+command_id2+)";
        }

        Synchronize (Form1->AddLogEntry);

        break;

    default:
        ;
    }

}
finally
{
    delete netStream;

    ClientSocket->Close();

    Synchronize (Form1->DecrConnectioncount);
}

}
```