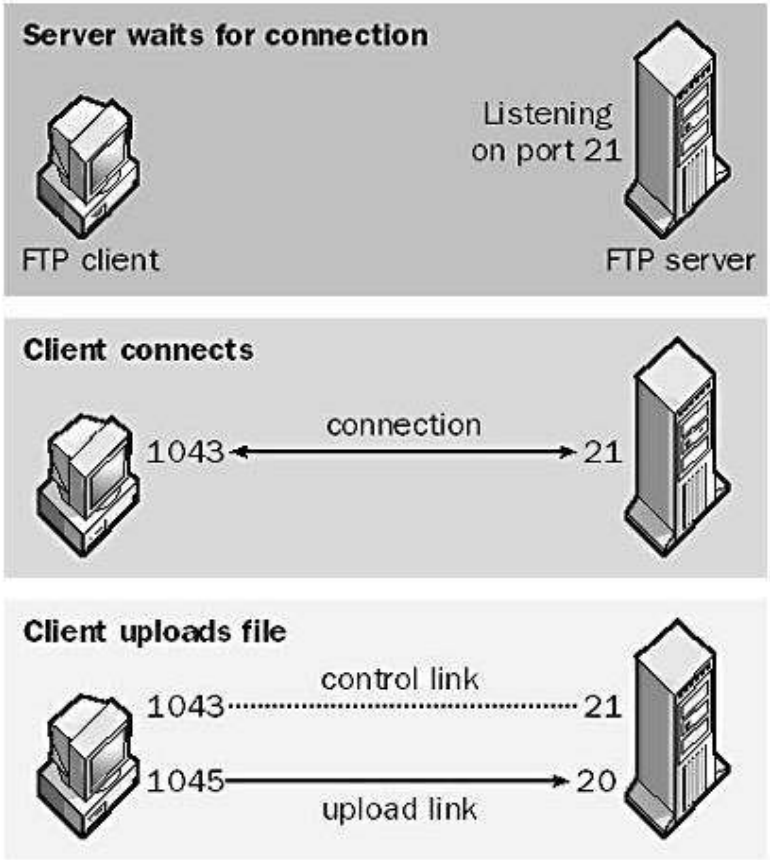
Writing by FADI Abdel-Qader/fadi822000@yahoo.com
My Space: http://spaces.msn.com/members/csharp2005

بسم الله الرحمن الرحيم

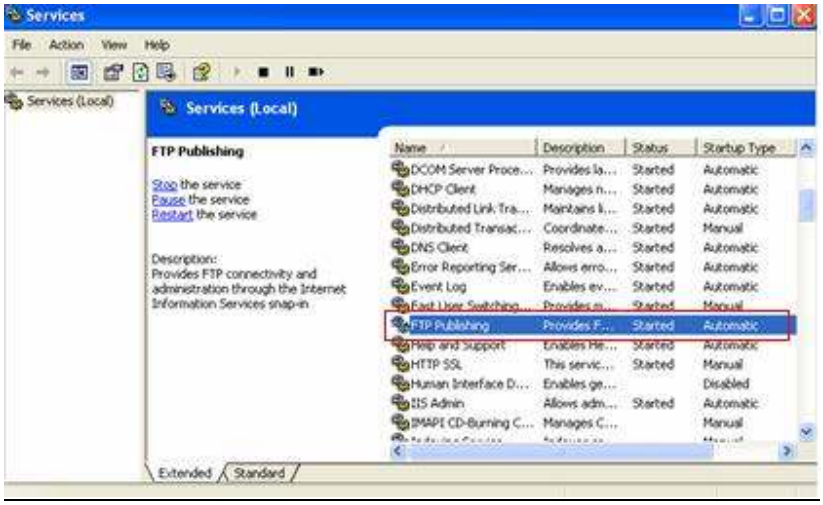
الدرس التاسع FTP – File Transfer Protocol Programming :

سوف نبدأ هنا بشرح برتوكول أخر من برتوكولات ال Application Layer وهو برتوكول ال FTP والذي يستخدم بشكل أساسي في عملية تنزيل downloading و رفع uploading الملفات من و إلى ال FTP Server وكالعادة في اغلب برمجيات الشبكات و التي تعتمد على وجود Client/Server حيث يقوم السيرفر بتصنت على البورت المخصص لل FTP وهو البورت 21 باستخدام ال TCP Connection Oriented Protocolحيث يبقى السيرفر بوضع الانتظار لورود طلب من ال Client بإنشاء Session معه وبعد إجراء عمليات التحقق Authentication والتأكد من الصلاحيات يتم الموافقة على البدء بالجلسة حيث يتم تحديد رقم البورت والذي سوف يتم استقبال البيانات من خلاله ويتم الإرسال إلى جهاز الزبون عبر البورت 20 في السيرفر وتتضح هذه العملية كما في الشكل التالي :

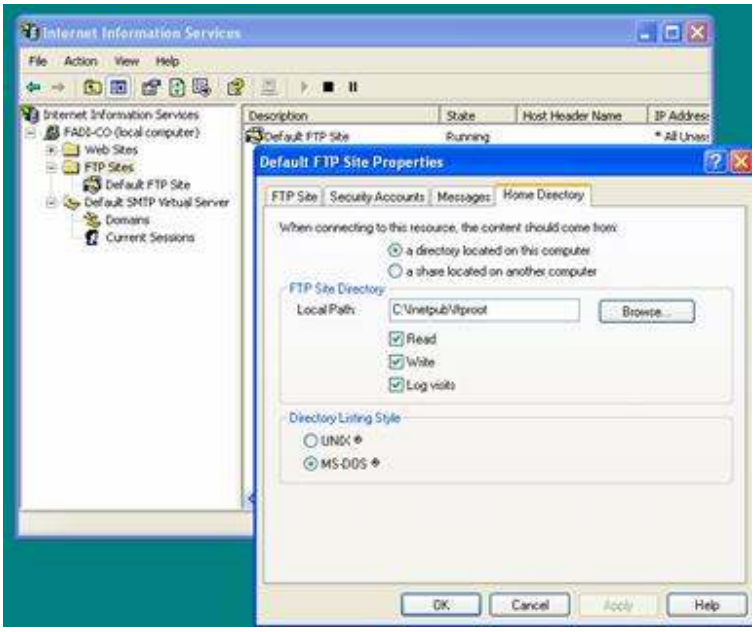


ملاحظة: لتفعيل خدمة ال FTP لديك بحيث يعمل جهازك ك FTP Server يجب أولا التأكد من أن ال FTP Services مثبتة لديك مع ال IIS و كما يظهر في الشكل التالي :

ومن ثم التأكد من تفعيلها ب Services من Control Panel ثم Administrative Tools ثم Services وكما يظهر في الشكل التالي :

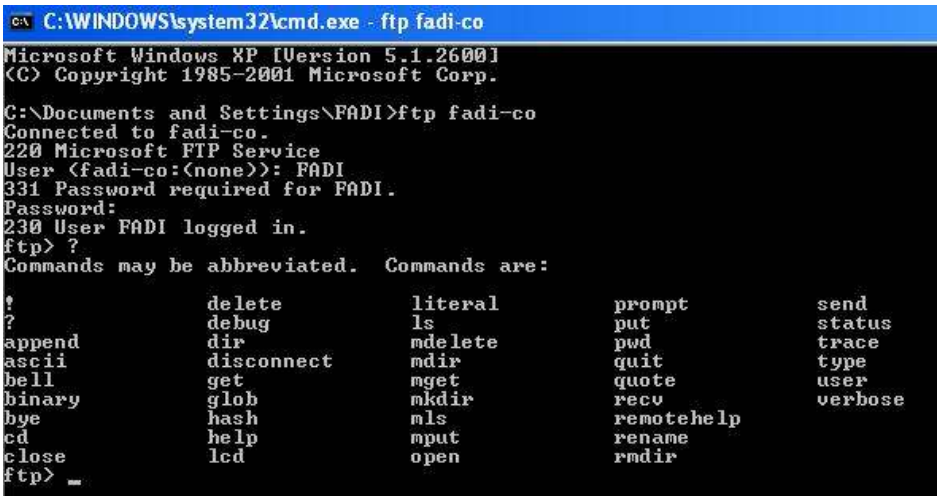


ثم التأكد منه في ال IIS بحيث يظهر كما في الشكل التالي :



أولا : FTP Commands

تشبه عملية الاتصال و الاستخدام لل FTP عملية ال Telnet إلى حد كبير حيث يدعم برتوكول ال FTP مجموعة من الأوامر والتي يتم من خلالها عملية التخاطب مع السيرفر أو مع ال Remote Host وتضح هذه العملية كما في الشكل التالي :

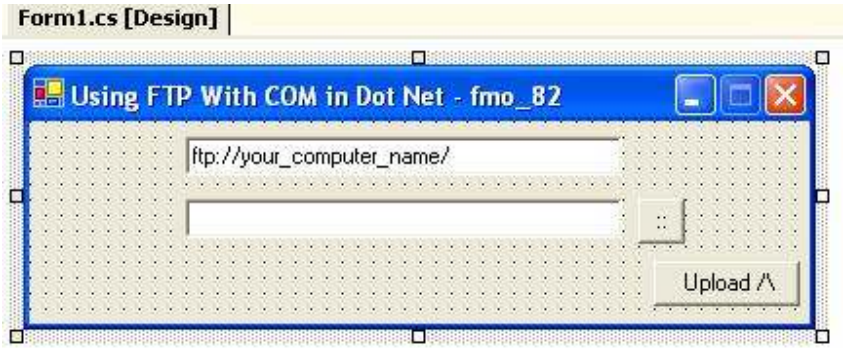


وهنا شرح لأهم ال FTP Commands:

USER <username> & PASS <password>	مطلوبة لعملية التحقق لإنشاء الجلسة
RECV أو RETR <*filename*>	ويستخدم لتنزيل ملف من السيرفر بعد تحديد اسم الملف
SEND أو STOR <*filename*>	ويستخدم لرفع الملف إلى السيرفر بعد تحديد اسم الملف
TYPE <*type indicator*>	لتحديد طبيعة أو هيئة البيانات التي يتم نقلها وكما يلي : -a ASCII –e EBCDIC – I for Binary Data - L<Byte Size> والذي سيتم نقله
PASV	لتحديد نوع الجلسة سواء Passive أو Active إذ انه في حالة ال Passive يتم تفعيل الاتصال فقط في حالة ورود أو رفع أي ملف من و إلى السيرفر .
Status أو STAT	لفحص حالة الاتصال و uploading & Downloading
Delete , cd , mkdir , rename …	وهي كما هو متعارف عليه في التعامل مع الملفات و المجلدات في نظام ال DOS
Close أو QUIT	لإنهاء الجلسة مع ال Remote Host

ثانيا : التعامل مع ال FTP في الدوت نيت باستخدام COM Components :

تدعم الدوت نيت استخدام ال FTP عبر ITC – Internet Transfer Control وهو جزء من ال COM Components Controls وللبدء قم بإنشاء New Windows Application كما في الشكل التالي :



ثم قم بإضافة النيم سبيسس التالية :

```
using System.IO;
using System.Reflection;
```

ثم إضافة الكود التالي إلى ال Upload Button :

```
private void button1_Click(object sender, System.EventArgs e)
        {
        FileInfo thisFile = new FileInfo(tbFile.Text);
        Type ITC;
        object[] parameter= new object[2];
        object ITCObject;
        ITC = Type.GetTypeFromProgID("InetCtls.Inet");
        ITCObject = Activator.CreateInstance(ITC);
        parameter[0] = (string)tbServer.Text;
        parameter[1] = (string)"PUT " + thisFile.FullName + " /" +
        thisFile.Name;
```

```
ITC.InvokeMember("execute", BindingFlags.InvokeMethod,null, ITCObject,
parameter);
      }
```

تم في البداية تعريف ال ITC من خلال ال Type Class والموجود ضمن النيم سبيس System.Reflection ثم عرفنا Array من النوع Object وذلك لاستخدامها في تمرير اسم الملف و ال FTP Server إلى الميثود InvokeMember والموجودة ضمن ال ITC Control Object ...

سوف تجد الملف الذي سيتم رفعه في المجلد :

C:\Inetpub\ftproot

ثالثا : التعامل مع ال FTP في الدوت نت باستخدام ال Web Class :

يمكن برمجة ال FTP باستخدام web Class والموجودة ضمن النيم سبيسس System.Net وتشبه عملية التعامل معه كما في التعامل مع ال WebRequest و ال webResponse Classes و التي تعاملنا معها في برمجة ال HTTP حيث يمكننا الاستفادة منها لتعامل مع ال FTP Protocol وهي كما يلي :

- WebClient إذ تم دعم dot net Framework 2 استخدام الكلاس WebClient والذي يدعم التعامل مع ال FTP والذي يتم استدعائه من النيم سبيسس System.Net ويتم تعريفه كما يلي :

```
using System;
using System.Net;

namespace Web_Client
{
    class Program
    {
        public static void Main(string[] args)
        {
            string filename = "ftp://ms.com/files/dotnetfx.exe";
            WebClient client = new WebClient();
            client.DownloadFile(filename, "dotnetfx.exe");

        }

    }
}
```

- FtpRequestCreator ويستخدم لتسجيل وبدأ العمل مع ال FTP ويعرف كما يلي :

```
using System;
using System.Net;

namespace FTP
{
    public class FtpRequestCreator : IWebRequestCreate
    {
        public FtpRequestCreator()
        {
        }

        public System.Net.WebRequest Create(System.Uri uri)
        {
            return new FtpWebRequest(uri);
```

```
        }
    }
}
```

- FtpWebRequest ويستخدم لعمل download or upload a file on an FTP server ويتم تعريفها كما يلي :

```
using System;
using System.Net;

namespace FTP
{
    public class FtpWebRequest : WebRequest
    {
        private string username = "Fadi";
        internal string password = "fff";
        private Uri uri;
        private bool binaryMode = true;
        private string method = "GET";

        internal FtpWebRequest(Uri uri)
        {
            this.uri = uri;
        }

        public string Username
        {
            get { return username; }
            set { username = value; }
        }

        public string Password
        {
            set { password = value; }
        }

        public bool BinaryMode
        {
            get { return binaryMode; }
            set { binaryMode = value; }
        }

        public override System.Uri RequestUri
        {
            get { return uri; }
        }

        public override string Method
        {
            get { return method; }
            set { method = value; }
        }

        public override System.Net.WebResponse GetResponse()
        {
            FtpWebResponse response = new FtpWebResponse(this);
            return response;
        }
    }
```

```
}
```

- FtpWebResponse ويستخدم لعملية الرد من قبل السيرفر ويتم تعريفها كما يلي:

```
using System;
using System.IO;
using System.Net;
using System.Net.Sockets;

namespace FTP
{
    public class FtpWebResponse : WebResponse
    {
        private FtpWebRequest request;
        private FtpClient client;

        internal FtpWebResponse(FtpWebRequest request)
        {
            this.request = request;
        }
    }
}
```

- FtpWebStream ويستخدم لتعريف ال Stream والذي سوف يستخدم لعملية النقل ويعرف بشكل مبدئي كما يلي :

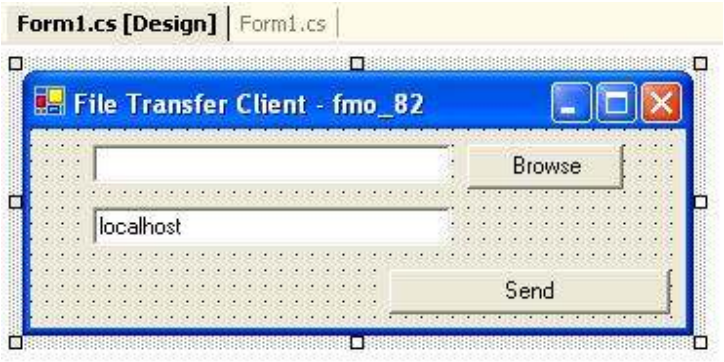
```
using System;
using System.IO;
using System.Net.Sockets;

namespace FTP
{
    internal class FtpWebStream : Stream
    {
        private FtpWebResponse response;
        private NetworkStream dataStream;

 public FtpWebStream(NetworkStream dataStream, FtpWebResponse response)
        {
            this.dataStream = dataStream;
            this.response = response;
        }
    }
}
```

رابعا : مثال تطبيقي لرفع ملف من جهاز Client إلى جهاز Server باستخدام ال Stream وال Socket:

في هذا الجزء سوف نقوم بإنشاء برنامجين Client / Server وبتعامل مع ال Stream Library سوف نقوم بتحويل الملف إلى Byte Array و إرساله عبر ال Stream باستخدام ال Socket و TCP Connection ، ولبرمجة الجزء الخاص بالإرسال أو ال Client قم بإنشاء مشروع جديد كما في الشكل التالي :



سوف نستخدم النيم سبيسس التالية :

```
using System.IO;
using System.Net;
using System.Net.Sockets;
using System.Text;
```

في ال Send Button قم بكتابة الكود التالي :

```
try
{
Stream fileStream = File.OpenRead(textBox1.Text);
// Alocate memory space for the file
byte[] fileBuffer = new byte[fileStream.Length];
fileStream.Read(fileBuffer, 0, (int)fileStream.Length);
// Open a TCP Connection and send the data
TcpClient clientSocket = new TcpClient(textBox2.Text,8880);
NetworkStream networkStream = clientSocket.GetStream();
networkStream.Write(fileBuffer,0,fileBuffer.GetLength(0));
networkStream.Close();
}
catch (Exception ex){MessageBox.Show(ex.Message);}
```

قمنا في البداية بقراءة الملف الذي نود إرساله وتخزينه ب Stream Object وحتى نستطيع إرساله عبر ال Socket لابد من تحويله إلى مصفوفة من النوع Byte وقمنا بتسميته ب fileBuffer ثم تعبئته باستخدام الميثود Read والموجودة ضمن fileStream وبعد ذلك قمنا بإنشاء اتصال مع ال Server باستخدام ال TCP Connection حيث تم إرسال محتويات ال fileBuffer إلى ال Server باستخدام NetworkStream Class ...

ولبرمجة جزء Server وهو المسئول عن استقبال الملف وتخزينه قم بإنشاء مشروع جديد كما يظهر في الشكل التالي :

سوف نستخدم النيم سبيسس التالية :

```
using System.Threading;
using System.Net;
using System.Net.Sockets;
using System.Text;
using System.IO;
```

ثم قم بكتابة ميثود جديدة وليكن اسمها handlerThread وكما يلي :

```
public void handlerThread()
{
Socket handlerSocket = (Socket)alSockets[alSockets.Count-1];
NetworkStream networkStream = new
                        NetworkStream(handlerSocket);
int thisRead=0;
int blockSize=1024;
Byte[] dataByte = new Byte[blockSize];
lock(this)
      {
// Only one process can access
// the same file at any given time
Stream fileStream = File.OpenWrite(@"c:\upload");

while(true)
            {
thisRead=networkStream.Read(dataByte,0,blockSize);
fileStream.Write(dataByte,0,thisRead);
if (thisRead==0) break;
fileStream.Close();
            }
                  lbConnections.Items.Add("File Written");
                  handlerSocket = null;
            }
```

ثم قم بكتابة ميثود أخرى جديدة وذلك لفتح TCP Connection على البورت 8880 وهو افتراضي والتصنت عليها وليكن اسمها listenerThread وكما يلي :

```
public void listenerThread()
{
      TcpListener tcpListener = new TcpListener(8880);
      tcpListener.Start();
      while(true)
      {
      Socket handlerSocket = tcpListener.AcceptSocket();
      if (handlerSocket.Connected)
            {
      lbConnections.Items.Add(handlerSocket.RemoteEndPoint.ToString() +
" connected.");
      lock (this)
            {
            alSockets.Add(handlerSocket);
            }
ThreadStart thdstHandler = new
                  ThreadStart(handlerThread);
Thread thdHandler = new Thread(thdstHandler);
thdHandler.Start();
            }
      }
}
```

ثم قم بإضافة الكود التالي إلى حدث بدأ تشغيل البرنامج Form Load :

```
private void Form1_Load(object sender, System.EventArgs e)
{
IPHostEntry IPHost = Dns.GetHostByName(Dns.GetHostName());

lbConnections.Text = "My IP address is " +
      IPHost.AddressList[0].ToString();

alSockets = new ArrayList();

Thread thdListener = new Thread(new ThreadStart(listenerThread));
thdListener.Start();

}
```

باستخدام ال Thread تم تنفيذ ال listenerThread Method والتي قمنا فيها بتعريف ال tcpListener وتفعيله على البورت 8880 حيث سيتم قبول أي طلب يأتي من ال Client على هذا البورت وبعد ذلك استدعاء الميثود handlerThread والتي سيتم فيها استقبال ال Stream Data وتخزينها في Byte Array ثم قراءتها وتخزينها في المكان المحدد وباستخدام ال fileStream.Write حيث مررنا له ال Stream والذي يحتوي على اسم الملف thisRead وال dataByte Array ...