

C# Network Programming- Target 12 – (Chapter2, Lecture 3)

Writing by FADI Abdel-Qader/fadi822000@yahoo.com

My Space: <http://spaces.msn.com/members/csharp2005>

بسم الله الرحمن الرحيم

الدرس الثاني عشر: Asynchronous Sockets Programming :

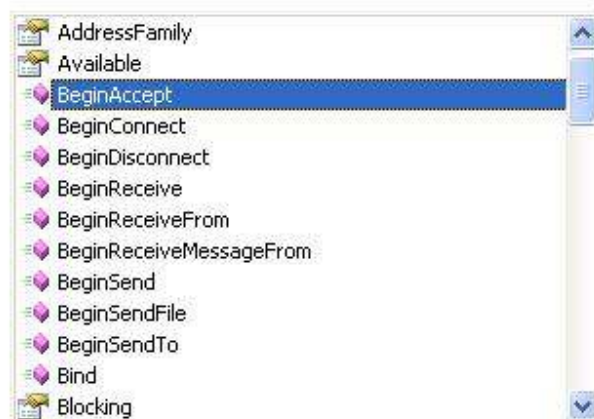
سوف نتحدث في هذا الدرس عن استخدام ال Asynchronous Socket بشكل أكثر تفصيلا عما تحدثنا به سابقا وسوف نطبق مجموعة من الأمثلة العملية على استخدام الاتصال الغير متزامن في برمجيات الشبكات ...

من المعروف أن الاتصال المتزامن مهم جدا في البرمجيات التي تحتاج إلى العمل في الزمن الحقيقي حيث لا يسمح باستخدام الاتصال لأمر آخر إلى بعد انتهاء العملية الجارية واستخدامه مهم جدا في العمليات التي تتطلب مثل هذه الأمور لا كن لا ينصح أبدا استخدامه في حالة إذا كانت الجهة المستقبلة للبيانات تستخدم Slow Connection كاعتماد الشبكة على ال Dialup لربط الجهازين المرسل مع المستقبل أو في حالة إذا كان هنالك مجموعة كبيرة من المستخدمين تستخدم السيرفر حيث يمنع الأسلوب المتزامن بقية المستخدمين على الشبكة من إجراء عملية الإرسال في حالة كون ال Server يستقبل بيانات من جهاز آخر ، وفي هذه الحالة ينصح باستخدام الاتصال الغير المتزامن إذ يعتبر مهم جدا في حالة إذا أردنا من البرنامج القيام بعدة مهام وعلى نفس ال Thread وباستخدام نفس ال Connection ، أو كما ذكرنا سابقا في حالة إذا كان الاتصال بطيء نسبيا أو انه يوجد عدد مستخدمين يستخدمون نفس ال Server ..

أولا Asynchronous Socket Class and its members :

تدعم الدوت نيت الاتصال غير المتزامن بمجموعة من ال methods الموجودة ضمن ال Socket Class والتي يتم استدعائها من ال System.Net.Socket Namespaces وقد ميزت الدوت نيت هذه الميثودس بوجود ال Begin في بداية أسم الميثود، ولكل ال Begin Method يوجد ال End Method مقابلة لها والتي تستخدم لإرجاع ال callback result عند انتهاء ال Method من التنفيذ وهي كما يلي:

```
Socket MySocket = new Socket(AddressFamily  
MySocket.b
```



1- BeginAccept و تستخدم لقبول ال Client Request وإسناده إلى ال Object AsyncCallback وباستخدام هذه الطريقة سوف يتمكن ال Server من استقبال عدد من ال Clients Requests في نفس الوقت وبدون الحاجة لانتظار الانتهاء من العملية الجارية حيث يتم في كل مرة استدعاء الميثود باستخدام ال AsyncCallback Delegate وتستخدم كما يلي كما يلي:

```
m_mainSocket = new Socket(AddressFamily.InterNetwork,
SocketType.Stream, ProtocolType.Tcp);
IPEndPoint ipLocal = new IPEndPoint (IPAddress.Any, 5020);
```

```
m_mainSocket.Bind (ipLocal);
m_mainSocket.Listen (10);
m_mainSocket.BeginAccept (new AsyncCallback (Client_request_method), null);
```

حيث سيتم إضافة ال Client Request في Callback Reference منفصل عن السابق وهنا لابد من إنشاء method لاستقبال ال Client Request وإنهاء ال Client Accepted Object باستخدام الميثود EndAccept :

```
public void Client_request_method(IAsyncResult ar)
{
    Socket listener = (Socket)ar.AsyncState;
    Myclient = listener.EndAccept(ar);
    Myclient.Send(/* data to be send */);
    listener.BeginAccept(new AsyncCallback(Client_request_method), listener);
    Console.WriteLine("Socket connected to {0}", client.RemoteEndPoint.ToString());
}
```

في Dot Net 2005 أصبحت ال BeginAccept Method تأخذ عدة أشكال كما يلي:

الشكل الأول في الدوت نيت 2003 و 2005 وتأخذ ال AsyncCallback Delegate و ال State Object لإرجاع معلومات عن حالة ال Request في ال Socket وكما يلي:

```
MySocket.BeginAccept(AsyncCallback , object state)
```

الشكل الثاني في الدوت نيت 2005 حيث يمكنك فيه تحديد حجم البيانات المستلمة

```
MySocket.BeginAccept(int Data_Receive_Size , AsyncCallback , object state)
```

الشكل الثالث في الدوت نيت 2005 حيث يمكن فيه تحديد ال Accepted Socket

```
MySocket.BeginAccept(Socket accept_Socket , int Data_Receive_Size ,
AsyncCallback , object state)
```

-2 BeginConnect وتستخدم لبدأ Asynchronous Connection على ال Socket ورقم البورت المحدد حيث يسند لها ال IPEndPoint وال Asynchronous Callback وال State Object وكما يلي:

```
MySocket.BeginConnect(EndPoint IP, AsyncCallback Result, object state)
```

وتستخدم كما يلي كمثال:

```
Socket MySocket = new Socket (AddressFamily.InterNetwork,
SocketType.Stream, ProtocolType.Tcp);
IPEndPoint ipend = new IPEndPoint(IPAddress.Parse("192.168.1.101"), 5020);
```

```
MySocket.BeginConnect(ipend, new AsyncCallback(Connected), MySocket);
```

في ال Connected Method يتم تحديد ال Socket Callback كما يلي:

```
public static void Connected(IAsyncResult iar)
{
    Socket sock = (Socket)iar.AsyncState;
    try
    {
        sock.EndConnect(iar);
    }
    catch (SocketException)
    {
        Console.WriteLine("Unable to connect to host");
    }
}
```

-3 BeginReceive وتستخدم لإستقبال بيانات من ال Client وتخزينها في Byte Array والصيغة العامة لها كما يلي:

```
MySocket.BeginReceive(Byte[] buffer, int offset, SocketFlags, AsyncCallback,
object sate)
```

ويستخدم كما يلي كمثال:

```
byte[] data = new byte[1024];
MySocket.BeginReceive(data, 0, data.Length, SocketFlags.None, new
AsyncCallback(ReceivedData), MySocket);
```

```
void ReceivedData(IAsyncResult iar)
{
    Socket remote = (Socket)iar.AsyncState;
    int recv = remote.EndReceive(iar);
    string receivedData = Encoding.ASCII.GetString(data, 0, recv);
    Console.WriteLine(receivedData);
}
```

كما تستخدم الميثود BeginReceiveFrom لإستقبال البيانات من موقع محدد باستخدام ال UDP حيث يضاف إلى التركيب السابق ال IPEndPoint Refrance Object .

4- BeginSend وتستخدم لإرسال بيانات إلى الطرف المستقبل عبر ال Asynchronous Socket والصيغة العامة لها كما يلي:

```
MySocket.BeginSend (Byte[] buffer,int offset, SocketFlags,AsyncCallback, object  
sate)
```

وتستخدم كما يلي كمثال:

```
private static void SendData(IAsyncResult iar)  
{  
    Socket server = (Socket)iar.AsyncState;  
    int sent = server.EndSend(iar);  
}
```

```
byte[] data = Encoding.ASCII.GetBytes("Hello Word");  
MySocket.BeginSend(data, 0, data.Length, SocketFlags.None,  
new AsyncCallback(SendData), MySocket);
```

كما تستخدم الميثود BeginSendto لإرسال البيانات إلى Remote Host محدد باستخدام ال UDP حيث يضاف إلى التركيب السابق ال IPEndPoint Refrance Object .

5- كما تم إضافة مجموعة من الميثود الجديدة في الدوت نيت 2005 وهي:
BegonDiconnect لإنهاء الاتصال و **BeginSendFile** لإرسال ملف و ال **BeginReceiveMessageFrom** والتي تستخدم لإستقبال عدد محدد من البيانات وتخزينها في مكان محدد في ال Bufer ..

تأخذ ال **BeginSendFile** التركيب التالي:

```
MySocket.BeginSendFile(string filename,AsyncCallback Asyn,object state)
```

وال **BeginReceiveMessageFrom** التركيب التالي:

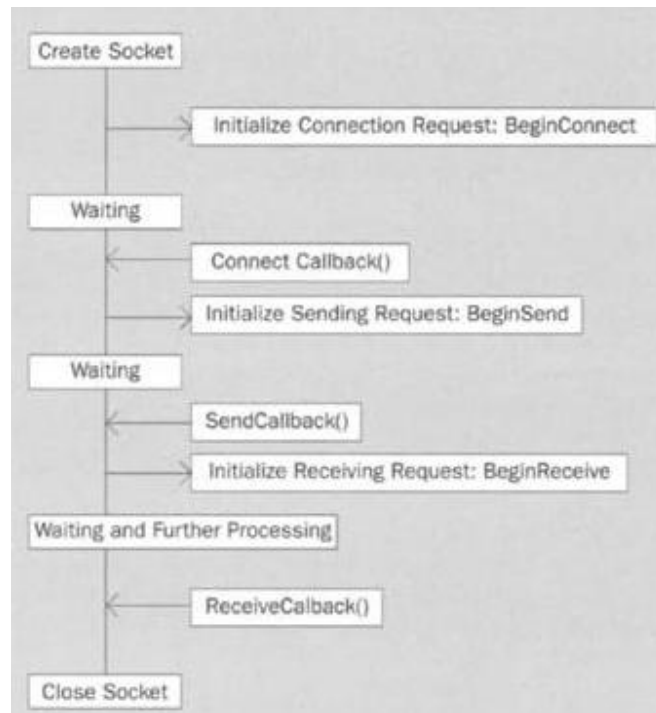
```
MySocket.BeginReceiveMessageFrom(byte Buffer ,int offset,int size,SocketFlags  
sf,ref EndPoint,AsyncCallback ascb,object state)
```

وال **BegonDiconnect** التركيب التالي:

```
MySocket.BeginDisconnect(bool reuseSocket,AsyncCallback ascb,object state)
```

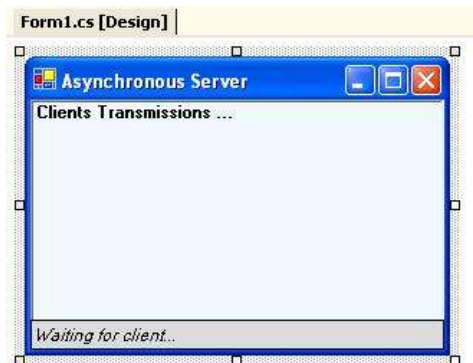
ثانياً: تطبيقات ال Asynchronous Socket في الدوت نت :

تمر عملية الاتصال الغير متزامن بمجموعة من المراحل تبدأ بإنشاء ال Socket Object في ال Server Side بعد ذلك يتم تعريف ال BeginConnect لبداً ال Asynchronous Connection على ال Socket حيث يتم إسناد ال Object ال IP ال Asynchronous Method وال Callback ال State Object لها وتبدأ في هذه الحالة عملية الاتصال بال Socket ، وبعد ذلك تمرر إلى ال BeginAccept لقبول ال Client Request حيث يتم قبول الطلب ويرسل ال Acknowledgement إلى ال Client ليعلمه فيها بقبول الجلسة وإمكانية البدء للإرسال و يستطيع ال Client بعد الموافقة على الجلسة البدء بالإرسال باستخدام الميثود ال BeginReceive ويستقبل ال Server الرسالة من ال Client باستخدام الميثود ال BeginSend وكما ذكرنا سابقاً فإن لكل عملية ال Begin تقابلها الميثود ال End للاستعداد لإجراء عملية أخرى على نفس ال Thread في البرنامج وهو ما ميز الاتصال الغير متزامن عن الاتصال المتزامن.



وبناء على المفاهيم السابقة سوف نقوم الآن بإنشاء برنامج Client/Server Chatting يعتمد على ال Asynchronous Socket لإرسال واستقبال البيانات .

ولبدء قم بإنشاء مشروع جديد كما في الشكل التالي:



سوف نستخدم ال Namespaces التالية:

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

في ال Global Declaration (أي بعد تعريف ال Main Class) قم بإضافة التعاريف التالية:

```
public class Form1 : System.Windows.Forms.Form  
{  
    Socket server = new Socket(AddressFamily.InterNetwork, SocketType.Stream,  
        ProtocolType.Tcp);  
    IPEndPoint iep = new IPEndPoint(IPAddress.Any, 5020);  
    private byte[] data = new byte[1024];  
    private int size = 1024;
```

في ال Form Load قم بإضافة الكود التالي حيث سنعرف Connection يعتمد على ال TCP ويعمل على البورت 5020 ثم تعريف عملية قبول الاتصال باستخدام ال BeginAccept :

```
private void Form1_Load(object sender, System.EventArgs e)  
{  
    server = new Socket(AddressFamily.InterNetwork, SocketType.Stream,  
        ProtocolType.Tcp);  
    IPEndPoint iep = new IPEndPoint(IPAddress.Any, 5020);  
    server.Bind(iep);  
    server.Listen(5);  
    server.BeginAccept(new AsyncCallback(AcceptConn), server);  
}
```

ثم إنشاء Accept Callback Method والذي سيتم فيه إنهاء ال Accepted Request باستخدام ال EndAccept Method وبعد ذلك إرسال Acknowledgement إلى ال Client تخبره فيها بقبول الطلب وترسل باستخدام ال BeginSend Method كما يلي:

```
void AcceptConn(IAsyncResult iar)  
{  
    Socket oldserver = (Socket)iar.AsyncState;  
    Socket client = oldserver.EndAccept(iar);  
    conStatus.Text = "Connected to: " + client.RemoteEndPoint.ToString();  
    string stringData = "Welcome to my server";  
    byte[] message1 = Encoding.ASCII.GetBytes(stringData);  
    client.BeginSend(message1, 0, message1.Length, SocketFlags.None, new  
        AsyncCallback(SendData), client);  
}
```

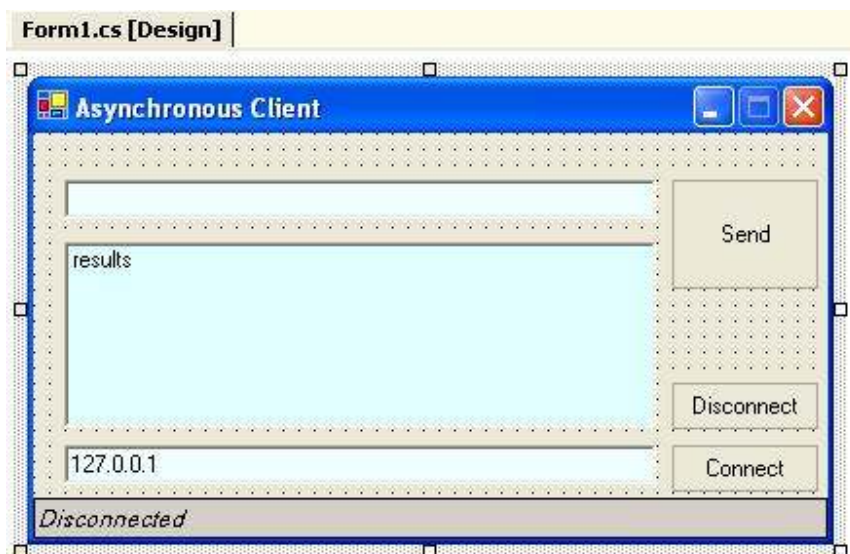
ثم إنشاء Send Callback method لإنهاء ال BeginSend وكما يلي:

```
void SendData(IAsyncResult iar)
{
    Socket client = (Socket)iar.AsyncState;
    int sent = client.EndSend(iar);
    client.BeginReceive(data, 0, size, SocketFlags.None, new
    AsyncCallback(ReceiveData), client);
}
```

ثم إنشاء Receive Callback method لإنهاء ال BeginReceive وكما يلي:

```
void ReceiveData(IAsyncResult iar)
{
    Socket client = (Socket)iar.AsyncState;
    int rcv = client.EndReceive(iar);
    if (rcv == 0)
    {
        client.Close();
        conStatus.Text = "Waiting for client...";
        server.BeginAccept(new AsyncCallback(AcceptConn), server);
        return;
    }
    string receivedData = Encoding.ASCII.GetString(data, 0, rcv);
    results.Items.Add(receivedData);
    byte[] message2 = Encoding.ASCII.GetBytes(receivedData);
    client.BeginSend(message2, 0, message2.Length, SocketFlags.None, new
    AsyncCallback(SendData), client);
}
```

وهنا قد تم الانتهاء من برنامج ال Server والآن سوف نقوم بإنشاء برنامج ال Client وللبداء قم بإنشاء مشروع جديد كما في الشكل التالي:



سوف نستخدم ال Namespaces التالية:

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

في ال Global Declaration (أي بعد تعريف ال Main Class) قم بإضافة التعاريف التالية:

```
public class Form1 : System.Windows.Forms.Form  
{  
    private Socket client;  
    private byte[] data = new byte[1024];  
    private int size = 1024;
```

في ال Connect Button قم بكتابة الكود التالي:

```
conStatus.Text = "Connecting...";  
Socket newsock = new Socket(AddressFamily.InterNetwork,  
    SocketType.Stream, ProtocolType.Tcp);  
IPEndPoint iep = new IPEndPoint(IPAddress.Parse(textBox1.Text), 5020);  
newsock.BeginConnect(iep, new AsyncCallback(Connected), newsock);
```

ثم قم بإنشاء Callback Connect method كما يلي:

```
void Connected(IAsyncResult iar)  
{  
    client = (Socket)iar.AsyncState;  
    try  
    {  
        client.EndConnect(iar);  
        conStatus.Text = "Connected to: " + client.RemoteEndPoint.ToString();  
        client.BeginReceive(data, 0, size, SocketFlags.None, new  
            AsyncCallback(ReceiveData), client);  
    }  
    catch (SocketException)  
    {  
        conStatus.Text = "Error connecting";  
    }  
}
```

ثم إنشاء Receive Callback method لإنهاء ال BeginReceive وكما يلي:

```
void ReceiveData(IAsyncResult iar)  
{  
    Socket remote = (Socket)iar.AsyncState;  
    int recv = remote.EndReceive(iar);  
    string stringData = Encoding.ASCII.GetString(data, 0, recv);  
    results.Items.Add(stringData);  
}
```

ثم إضافة الكود التالي في ال Send Button :

```
try
{
byte[] message = Encoding.ASCII.GetBytes(newText.Text);
newText.Clear();
client.BeginSend(message, 0, message.Length, SocketFlags.None, new
AsyncCallback(SendData), client);
newText.Focus();
}
catch(Exception ex){MessageBox.Show(ex.Message);}
```

ثم إنشاء Send Callback method لإنهاء ال BeginSend وكما يلي:

```
void SendData(IAsyncResult iar)
{
try
{
Socket remote = (Socket)iar.AsyncState;
int sent = remote.EndSend(iar);
remote.BeginReceive(data, 0, size, SocketFlags.None, new
AsyncCallback(ReceiveData), remote);
}
catch(Exception ex){MessageBox.Show(ex.Message);}
}
```

ثم إنشاء Receive Callback method لإنهاء ال BeginReceive وكما يلي:

```
void ReceiveData(IAsyncResult iar)
{
try
{
Socket remote = (Socket)iar.AsyncState;
int recv = remote.EndReceive(iar);
string stringData = Encoding.ASCII.GetString(data, 0, recv);
results.Items.Add(stringData);
}
catch(Exception ex){MessageBox.Show(ex.Message);}
}
```

وكما لاحظنا فإن برنامج ال Client لا يختلف كثيرا عن برنامج ال Server حيث نعرف في ال Server ال Socket Connection وال BeginAccept Method أما في ال Client فنعرف ال Socket Connection وال BeginConnect Method وتبقى عملية الإرسال والاستقبال هي نفسها في ال Server وال Client ...