

Re: DataReceived type event for tcpclient network stream?

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2008-02/msg02312.html>

- *From:* "Peter Duniho" <NpOeStPeAdM@xxxxxxxxxxxxxxxxxxxx>
 - *Date:* Sun, 17 Feb 2008 16:27:34 -0800
-

On Sun, 17 Feb 2008 03:27:15 -0800, JamesB <jamesb@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote:

I have a service and a client forms app – I want to be able to have the service send data back to the client app intermittently, and I've established a connection between the two using tcpclient.

What I can't see is any sort of event I can use that will be triggered when data is received at the client. Is there such a thing?

Or should I create a separate "listening" thread that polls the connection for data and then triggers whatever I need to happen? Or is there a better way?

IMHO, the "best" way is to use the async API, which provides something sort of like events, albeit with a bit more complexity. It works in a very similar way though, at least when you are using the callback aspect of it.

In particular, the various classes — TcpClient, NetworkStream, Socket — include methods that start with the word "Begin" or "End". These methods all represent an asynchronous version of a synchronous method. You start by calling the "Begin" method, and when the operation completes, you are notified via some mechanism at which point you call the "End" method, providing your own code with the results of the operation.

The standard .NET async model provides three different ways to know that an operation is completed: IAsyncResult.IsCompleted property, which you can poll (generally the wrong thing to do); IAsyncResult.AsyncWaitHandle property, which you can wait on (better than polling, but blocks a thread and so doesn't usually wind up better than just dedicating a thread using the synchronous method); and a callback delegate, which you provide when you call the "Begin" method, and which is called when then operation has actually completed.

The last technique is IMHO the best, and it's pretty much the same behavior that an event would provide. The main differences being that you pass the delegate to a method (the "Begin" method) rather than adding it to an event, and that because of that you can only have one delegate "subscribed" to the "event" at a time.

Pete

.