

TCP Sample

Installation

The sample was written in Visual Studio 2002.

You need to unpack the provided archive and open the *TCP 1.0 > Project.sln* solution file. Then rebuild the solution.

Press **F5** to run both client and server in Visual Studio.

Configuration

Server IP address is specified in the *Client > App.config* file. You should directly modify *Client > Bin > Debug > Client.exe.config* if you need to change the IP address without recompiling the solution.

Event approach

The second approach from the article available via this link

<http://www.genuinechannels.com/Content.aspx?id=27&type=1> has been used to implement the sending of events.

GTCP Sample

Installation

The sample was written in Visual Studio 2002.

You need to unpack the provided archive and open the *GTCP 1.0 > Project.sln* solution file. Then rebuild the solution.

Press **F5** to run both client and server in Visual Studio.

Configuration

Server IP address is specified in the *Client > App.config* file. You should directly modify *Client > Bin > Debug > Client.exe.config* if you need to change the IP address without recompiling the solution.

Design

Known layer contains only communication interfaces.

Server implements the Event Provider interface and binds business object to known URI. Server uses Client Session to keep track of the specific client and its nickname.

Client implements event receiver interface and subscribes the listener. Client builds local transparent proxy pointed to the known URI to access server's business object. Client's code uses channel events to determine the restart of the server. If you close the server and start it again, clients re-subscribe to the chat event and continue receiving chat messages.

You can find more information about the used approach here:

<http://www.genuinechannels.com/Content.aspx?id=28&type=1>

Uncomment a line containing "GlobalLoggerContainer.Logger" in *Client > ChatClient.cs* and *Server > ChatServer.cs* to enable logging into a file. You can use Find in Files (CTRL-SHIFT-F) feature in Visual Studio to find the entrance.

Event approach

Broadcast Engine was used to implement the server's event. This resulted in simpler source-code on the server side. You can find more information about Broadcast Engine in Programming Guide and in articles dedicated to events:

<http://www.genuinechannels.com/Content.aspx?id=27&type=1>

<http://www.genuinechannels.com/Content.aspx?id=73&type=1>

GHTTP Sample

Installation

The sample was written in Visual Studio 2002.
You need to unpack the provided archive, open *Internet Information Services Console*, create virtual directory with **GHTTP 1.0 > GHTTPChatSample** alias and direct it to *GHTTPChatSample* directory. After that, open *GHTTP 1.0 > Project.sln* solution file and rebuild the solution.
Press **F5** to run both client and server in Visual Studio.

Configuration

URL to the server is specified in the *Client > App.config* file. You should modify directly *Client > Bin > Debug > Client.exe.config* file if you need to change the server address without recompiling the solution.
Please refer to *Programming Guide > Channels > GHTTP > GHTTP Transport Context Parameters* section if you need to specify proxy settings.

Design

GHTTP sample has absolutely the same design as GTCP sample. *Default.aspx* page outputs all events and chat information. You can periodically click *Refresh* in your browser to see the refreshed content.

Known layer contains only communication interfaces.

Server implements event provider interface and binds business object to the known URI.

Client implements event receiver interface and subscribes the listener. Client builds local transparent proxy pointed to the known URI to access server's business object. Client's code uses channel events to determine the restart of the server. If you close the server and start it again, clients re-subscribe to the chat event and continue receiving chat messages.

You can find more information about the used approach here:
<http://www.genuinechannels.com/Content.aspx?id=28&type=1>

Uncomment a line containing "GlobalLoggerContainer.Logger" in *Client > ChatClient.cs* and *GHTTPChatSample > Global.asax.cs* to enable logging into a file. You can use Find in Files (CTRL-SHIFT-F) feature in Visual Studio to find the entrance.

Event approach

Broadcast Engine was used to implement the server's event. This resulted in simpler source-code at the server side. You can find more information about Broadcast Engine in Programming Guide and in articles dedicated to events:

<http://www.genuinechannels.com/Content.aspx?id=27&type=1>
<http://www.genuinechannels.com/Content.aspx?id=73&type=1>