

Forum: [Hack Forums](#)

Thread: [\[Delphi Chat Client\] Reverse Connection, Allow only one chat/IP, update listview](#)

Feminist - May 6, 2012, 11:46 pm

Wrote this in early 2010 or so

Code unit Main;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, WinSkinStore, WinSkinData, StdCtrls, ScktComp, ExtCtrls,
ComCtrls, Menus;

type

TForm1 = class(TForm)
SkinData: TSkinData;
SkinStore: TSkinStore;
cbListen: TCheckBox;
footer: TEdit;
ServerSocket: TServerSocket;
ConnectionsView: TListView;
updateTimer: TTimer;
commandMenu: TPopupMenu;
Disconnect1: TMenuItem;

procedure FormCreate(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure cbListenClick(Sender: TObject);
procedure ListConnections();
procedure ClearListView();
function GetListViewSelection(): Integer;
procedure ServerSocketClientConnect(Sender: TObject;
Socket: TCustomWinSocket);
procedure ServerSocketClientDisconnect(Sender: TObject;
Socket: TCustomWinSocket);
procedure updateTimerTimer(Sender: TObject);
procedure Disconnect1Click(Sender: TObject);
procedure ServerSocketClientRead(Sender: TObject;
Socket: TCustomWinSocket);
procedure ServerSocketAccept(Sender: TObject;
Socket: TCustomWinSocket);

private

{ Private declarations }

public

{ Public declarations }

end;

```
var
  Form1: TForm1;
```

implementation

```
{ $R *.dfm }
```

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  SkinData.LoadFromCollection(SkinStore, 0);
  if not SkinData.Active then SkinData.Active := true;

  footer.Text := 'Default port is 88';
  ServerSocket.Port := 88;
  if not updateTimer.Enabled then updateTimer.Enabled := true;
end;
```

```
procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  if ServerSocket.Active then ServerSocket.Active := false;
  if updateTimer.Enabled then updateTimer.Enabled := false;
end;
```

```
function TForm1.GetListViewSelection() : Integer;
var
  i: Integer;
begin
  Result := -1;
  with ConnectionsView do
  begin
    i := 0;
    while i < Items.Count do
      if Items[i].Selected then
        begin
          Result := i;
          Exit;
        end
      else
        Inc(i);
    end;
  end;
end;
```

```
procedure TForm1.ClearListView();
var
  i: Integer;
begin
  with ConnectionsView do
  begin
    for i := -1 + Items.Count downto 0 do
      if Items.Count <> 0 then
        Items.Delete(i)
    end;
  end;
```

end;

```
procedure TForm1.cbListenClick(Sender: TObject);
begin
  if cbListen.Checked then
  begin
    footer.Text := 'Listening on port 88';
    if not ServerSocket.Active then ServerSocket.Active := true;
    ClearListView;
  end

  else
  begin
    footer.Text := 'Stopped listening';
    if ServerSocket.Active then ServerSocket.Active := false;
    ClearListView;
  end;
end;
```

```
procedure TForm1.ListConnections();
var
  i: Integer;
  li: TListItem;
begin
  ConnectionsView.Items.BeginUpdate;
  ClearListView;
  for i := 0 To ServerSocket.Socket.ActiveConnections - 1 do
  begin
    li := ConnectionsView.Items.Add;
    li.Caption := IntToStr(i);
    li.SubItems.Add(ServerSocket.Socket.Connections[i].RemoteAddress);
    li.SubItems.Add(ServerSocket.Socket.Connections[i].RemoteHost);
    li.SubItems.Add('88')
  end;
  ConnectionsView.Items.EndUpdate;
  footer.Text := IntToStr(ServerSocket.Socket.ActiveConnections) + ' client(s) connected';
end;
```

```
procedure TForm1.ServerSocketClientConnect(Sender: TObject;
  Socket: TCustomWinSocket);
begin
  ListConnections;
end;
```

```
procedure TForm1.ServerSocketClientDisconnect(Sender: TObject;
  Socket: TCustomWinSocket);
begin
  ListConnections;
end;
```

```
procedure TForm1.updateTimerTimer(Sender: TObject);
begin
  if ServerSocket.Active then ListConnections;
```

end;

```
procedure TForm1.Disconnect1Click(Sender: TObject);
var
  Selection: Integer;
begin
  Selection := GetListViewSelection();
  if Selection <> -1 then
  begin
    ServerSocket.Socket.Connections[Selection].Close;
    ConnectionsView.Items.BeginUpdate;
    ConnectionsView.Items.Delete(Selection);
    ConnectionsView.Items.EndUpdate;
  end;
end;
```

```
procedure TForm1.ServerSocketClientRead(Sender: TObject;
  Socket: TCustomWinSocket);
var
  i: Integer;
  s: String;
begin
  s := Socket.ReceiveText;
  with ServerSocket.Socket do
  begin
    for i := (ActiveConnections - 1) downto 0 do
      ServerSocket.Socket.Connections[i].SendText(s);
    end;
  end;
end;
```

```
procedure TForm1.ServerSocketAccept(Sender: TObject;
  Socket: TCustomWinSocket);
var
  s: String;
  i: Integer;
  List: TListItem;
  x: Integer;
begin
  s := Socket.RemoteAddress;
  x := 0;

  for i := ConnectionsView.Items.Count - 1 downto 0 do
  begin
    List := ConnectionsView.Items.Item[i];
    if List.SubItems[0] = s then
    begin
      Inc(x);
      if x = 2 then
      begin
        Socket.SendText('SERVER: Connection closed. Reason: Duplicate Connection.');
```

```
    end;  
  end;  
end;  
end;  
  
end.
```