

```

1  unit ClientUnit;
2
3  interface
4
5  uses  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
6        StdCtrls, ExtCtrls, ScktComp, ExtDlgs;
7
8  type
9      TClientForm = class(TForm)
10         ClientSocket: TClientSocket;
11         Timer: TTimer;
12         Image1: TImage;
13         ConnectButton: TButton;
14         GetPictureButton: TButton;
15         DisconnectButton: TButton;
16         PictureList: TListBox;
17         GetListButton: TButton;
18         ClearListButton: TButton;
19         procedure ConnectButtonClick(Sender: TObject);
20         procedure GetPictureButtonClick(Sender: TObject);
21         procedure DisconnectButtonClick(Sender: TObject);
22         procedure ClientSocketRead(Sender: TObject; Socket: TCustomWinSocket);
23         procedure TimerTimer(Sender: TObject);
24         procedure FormCreate(Sender: TObject);
25         procedure GetListButtonClick(Sender: TObject);
26         procedure ClearListButtonClick(Sender: TObject);
27     private
28         { Private declarations }
29     public
30         { Public declarations }
31     end;
32
33 var ClientForm: TClientForm;
34     StartGetData: Boolean;
35     BmpFile: File;
36
37 implementation
38
39 {$R *.DFM}
40
41 procedure TClientForm.ConnectButtonClick(Sender: TObject);
42 begin
43     ClientSocket.Open;
44 end;
45
46 procedure TClientForm.GetPictureButtonClick(Sender: TObject);
47 begin
48     StartGetData:= True;
49     ClientSocket.Socket.SendText(PictureList.Items[PictureList.ItemIndex]);
50 end;
51
52 procedure TClientForm.GetListButtonClick(Sender: TObject);
53 begin
54     StartGetData:= False;
55     ClientSocket.Socket.SendText('*');
56 end;
57
58 procedure TClientForm.DisconnectButtonClick(Sender: TObject);
59 begin
60     ClientSocket.Close;
61     Close;
62 end;
63
64 procedure TClientForm.ClearListButtonClick(Sender: TObject);
65 begin
66     PictureList.Clear;
67 end;
68
69

```

```

69 procedure TClientForm.ClientSocketRead(Sender: TObject; Socket: TCustomWinSocket);
70 var Buf: PChar;
71     Length: Integer;
72 begin
73     If PictureList.Items.Count = 0 Then Begin
74         PictureList.Items.SetText(PChar(Socket.ReceiveText));
75     End
76     Else Begin
77         Timer.Enabled:= False;
78         Length:= Socket.ReceiveLength;
79         GetMem(Buf, Length);
80         Socket.ReceiveBuf(Buf^, Length);
81         AssignFile(BmpFile, 'tmp.bmp');
82         If StartGetData Then Begin
83             Rewrite(BmpFile, 1);
84             StartGetData:= False;
85         End
86         Else Begin
87             Reset(BmpFile, 1);
88             Seek(BmpFile, FileSize(BmpFile));
89         End;
90         BlockWrite(BmpFile, Buf^, Length);
91         CloseFile(BmpFile);
92         FreeMem (Buf, Length);
93         Timer.Enabled:= True;
94     End;
95 end;
96
97 procedure TClientForm.TimerTimer(Sender: TObject);
98 begin
99     Timer.Enabled:= False;
100     Image1.Picture.LoadFromFile('tmp.bmp');
101     Erase(BmpFile);
102 end;
103
104 procedure TClientForm.FormCreate(Sender: TObject);
105 begin
106     Timer.Interval:= 100;
107     Timer.Enabled:= False;
108 end;
109
110 end.
111

```

```

111 unit ServerUnit;
112
113 interface
114
115 uses
116     Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
117     StdCtrls, ScktComp, ExtDlgs;
118
119 type
120     TServerForm = class(TForm)
121         ServerSocket1: TServerSocket;
122         ExitButton: TButton;
123         CountLabel: TLabel;
124         Counter: TLabel;
125         FileNameLabel: TLabel;
126         PictureList: TListBox;
127         procedure ExitButtonClick(Sender: TObject);
128         procedure ServerSocket1ClientConnect(Sender: TObject; Socket:
129 TCustomWinSocket);
130         procedure ServerSocket1ClientDisconnect(Sender: TObject;
131 Socket: TCustomWinSocket);
132         procedure ServerSocket1ClientRead(Sender: TObject; Socket: TCustomWinSocket);
133     private
134         { Private declarations }
135     public
136         { Public declarations }
137     end;
138 var ServerForm: TServerForm;
139
140 implementation
141
142 {$R *.DFM}
143
144 procedure TServerForm.ExitButtonClick(Sender: TObject);
145 begin
146     If (ServerSocket1.Socket.ActiveConnections = 0) or
147         (MessageDlg('There exist active connections. Disconnect?',
148             mtWarning, [mbYes, mbNo], 0) = mrYes)
149     Then Begin
150         ServerSocket1.Close;
151         Close;
152     End;
153 end;
154
155 procedure TServerForm.ServerSocket1ClientConnect(Sender: TObject;
156     Socket: TCustomWinSocket);
157 begin
158     With Counter Do Begin
159         Caption:= IntToStr(ServerSocket1.Socket.ActiveConnections);
160     End;
161 end;
162
163 procedure TServerForm.ServerSocket1ClientDisconnect(Sender: TObject;
164     Socket: TCustomWinSocket);
165 begin
166     With Counter Do Begin
167         Caption:= IntToStr(ServerSocket1.Socket.ActiveConnections-1);
168     End;
169 end;
170
171

```

```

171 procedure TServerForm.ServerSocket1ClientRead(Sender: TObject;
172     Socket: TCustomWinSocket);
173 Var SStream: TStringStream;
174     FStream: TFileStream;
175     s: String;
176 begin
177     s:= Socket.ReceiveText;
178     If s='*' Then Begin
179         SStream:= TStringStream.Create(PictureList.Items.Text);
180         Socket.SendStream(SStream);
181     End
182     Else Begin
183         FStream:= TFileStream.Create(s, fmOpenRead);
184         Socket.SendStream(FStream);
185     End;
186 end;
187
188 end.

```