



NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

WEBrowser

**Graduation Project
COM400**

Student : Serhat Tunca (970179)

Supervisor : Mrs. Besime Erin

Lefkoşa - 2000

CONTENTS

1.INTRODUCTION.....	1
2.ABOUT WEBROWSER.....	3
2.1 Main Window.....	3
2.1.1 Function of Buttons.....	3
2.1.2 Main Menu.....	3
2.2 E-Mail Window.....	4
2.2.1 Function of Buttons.....	4
2.3 New E-Mail Window.....	5
2.3.1 Function of Buttons.....	5
2.4 Options Window.....	5
2.3.1 Function of Buttons.....	5
3.CODES OF PROGRAMS.....	6
3.1 Main Program (webbrowser.pas).....	6
3.1.1 Codes of webbrowser.dpr.....	6
3.1.2 Codes of uemail.pas.....	6
3.1.3 Codes of uemail.pas.....	23
3.1.4 Codes of unew.pas.....	31
3.1.5 Codes of usource.pas.....	36
3.1.6 Codes of usearch.pas.....	38
3.1.7 Codes of uoption.pas.....	40
3.1.8 Codes of uabout.pas.....	44
3.2 Installation Program (setup.pas)	45
3.2.1 Codes of setup.dpr.....	45
3.2.2 Codes of uemail.pas.....	46
3.2.3 Codes of ucancel.pas.....	53
3.3 Uninstallation Program (uninstallation.pas).....	54
3.3.1 Codes of uninstallation.dpr.....	54
3.3.2 Codes of uemail.pas.....	54
4.SCREEN LAYOUTS.....	57
5.ABSTRACT	74
6.ACKNOWLEDGEMENT.....	74
7.CONCLUSION.....	74
8.REFERENCES.....	75

1.INTRODUCTION

I decided to prepare my graduation project with Borland Delphi 4.0 programming language. Delphi is my best visual programming language i have ever know. Also Delphi has many useful components to develop an application about Internet. I used components HTML (Hyper Text Markup Language) and NMSMTP(NM Simple Mail Transfer Protocol). **Figure 1.1.**

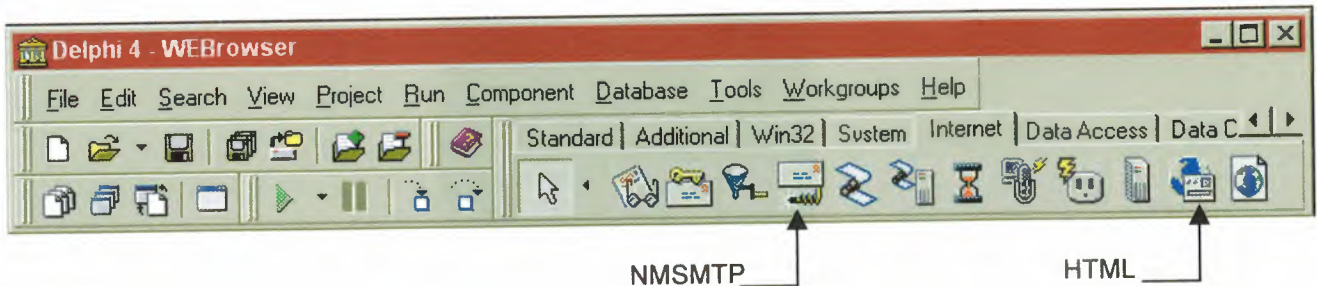


Figure 1.1

HTML : The HTML control lets you implement an HTML viewer, with or without automatic network retrieval of HTML documents. It provides parsing and layout of HTML data, as well as a scrollable view of the selected HTML page. It can also be used as a non-visual HTML parser to analyze or process HTML documents.

The HTML control supports the following features:

- Scrollable view of selected page
- Inline graphics: GIF, JPEG, BMP, XBM
- HTML version 2.X plus most NetScape 2.0 and Explorer 2.0 extensions
- Built-in document retrieval for HTTP and File URLs
- Built-in HTTP form execution
- Properties controlling the style sheet (such as fonts and colors)
- DocStream interfaces for flexible data transfer
- Events for overriding default processing
- Printing

But its not perfect, the following features are not supported by the HTML control:

- Text selection and clipboard copy
- Automatic external viewer launching
- Proxy server determination and usage
- Built-in FTP retrieval and inline FTP listings
- Basic Authorization and SSL/PCT
- Multipart document submission (file upload)

NMSMTP : The TNMSMTP component enables the sending of E-Mail via an internet mail server and the implementation of other commands specified in RFC 821. (Many of the protocols that control activity on the Internet are defined in Request for Comment (RFC) documents that are created,)

I wanted to change the Windows' standart window layout. To do that i gave BorderStyle property of the forms bsNone. **Figure 1.2.** (form is the window in construction).



Figure 1.2

The windows' main activities (Resize, move, system menu, windows title, close, minimize, maximize, etc.) are cancelled by setting BorderStyle bsNone. I had to write small helper programs to do these jobs for each window. I also tried to change the standart button style. I used circular buttons instead of rectangular. I think it made my program more interesting and attractive, it increased the size of my program. I placed too much emphasis on apprearance, because its important to do not get bored the user while he/she is connecting to Internet.

There must be an installation program to create complex directory structure of program.(**Figure 1.3**) There are buttons over seventy and four avi files in diffrent directories. The WEBrowser Installation program solves this problem by only clicking four times. The Windows folder must be determined, because the program creates shortcuts in various locations. (On Desktop, in QuickLaunch, and in Start Menu>Program Files). These locations are selectable.

Installation program ;

- Creates directory in harddisk named C. The folder name is WEBrowser. (You can change the location of the program by clicking check box under edit box. But this causes the uninstallation program not to work.)
Creates a file named <webunins.ini> at he location <C:\> and saves the information about installation, (directories locations)
- Copies the main program <WEBrowser.exe> in the folder,
- Creates directories Data,Email,Avi in folder WEBrowser,
- Copies necessary files in folders above,
- Creates shortcuts (if selected).

If the program is installed in its original folder, the uninstallation program works. It deletes all files and removes the directories. Also clears the shortcuts if they installed.

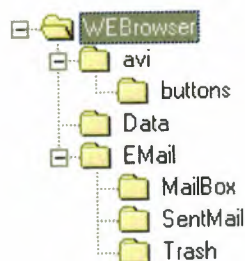
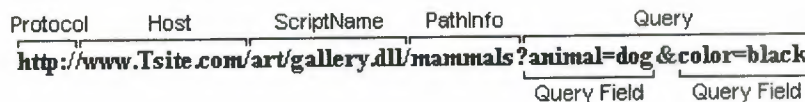


Figure 1.3

2. ABOUT WEBROWSER :

The WEBrowser program can be used to reach URLs (Uniform Resource Locators) and to send mails. It also searches entered keyword in www (world wide web) by using Yahoo.

The Uniform Resource Locator (URL) is a complete description of the location of a resource that is available over the net. It is composed of several parts that are accessed by an application. These parts are illustrated in the following figure:



2.1 Main Window

2.1.1 Function of Buttons :

In the main window there are ten buttons; (figure 2.1.1.1)



Figure 2.1.1.1

- ☐ UpArrow : Is used to change the style of window. (styles can be view in screen layouts)
- ☐ Back : Returns the last visited page.
- ☐ Next : Next page that is come.
- ☐ Stop : Stops the request.
- ☐ Refresh : Refreshes the request.
- ☐ Home : Returns the home address saved in options window.
- ☐ Send Mail : Is used to send new mail, or send saved mails.
- ☐ Search : Is used for finding desired addresses related with the entered keyword.
- ☐ Print : Prints the content of HTML page.
- ☐ Full Screen : Fills screen with browser.
- ☐ Options : Opens a window to change various properties.(fonts, homepage, mail settings).

2.1.2 Main Menu :

- ☐ File :
 - New : Opens new browser window.
 - Clear History : Deletes all visited URLs' addresses.
 - Print : Prints the content of HTML page.
 - Printer Settings : To change the present printer settings.
 - Exit : Closes browser.

- ❑ Edit :
 - Copy : Copies into clipboard.
 - Cut : Deletes and copies into clipboard.
 - Paste : Pastes from clipboard.
 - Clear : Deletes the clipboard content.

- ❑ View :
 - Full Screen : Fills the screen with the browser.
 - View Source : Displays the source code of HTML.

- ❑ E-Mails :
 - Send Mail : Opens a window to send a new mail.
 - Open Mail : Displays a window to open a saved mail.
 - Settings : Is used to hange to mail settings. (your name, email address and SMTP server)

- ❑ About :
 - About WEBrowser : Information about the program.

2.2 E-Mail Window :

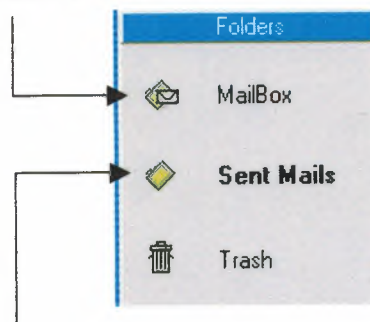
2.2.1 Function of Buttons :



- New : Opens a window to write email. Email can be saved or sent.
- Send : Sends the mail selected in the string grid.
- Print : Prints the active mail.
- Delete : If it is pressed in the MailBox or in the SentMails folder , the mail will be sent into trash. If the mail is in the trash and the user presses the delete, the email will be deleted.
- Settings : Opens the mail settings window.
- Close : Closes the window.

Folder Buttons are used to select the folder. If the folders MailBox, SentMail, or Trash has at least one mail, the picture of the button is changed. See figure below

There is a mail in MailBox



There is no mail in SentMails

2.3 New Email Window :

2.3.1 Function of Buttons :



- Send : Sends the mail to the specified address.
- Open : Opens the saved mail.
- Save : Saves the mail.
- Copy : Copies into clipboard.
- Cut : Deletes and copies into clipboard.
- Paste : Pastes from clipboard.
- Close : Closes the window.

2.4 Options Window :

2.4.1 Function of Buttons :



- Clear History : Deletes all visited URLs' addresses.
- Check : Save the information to file option.txt
- Cancel : Cancel changes.

3. CODES OF THE PROGRAMS

3.1 Main Program (WEBrowser.pas) :

3.1.1. Codes of WEBrowser.dpr :

```
program WEBrowser;

uses
  Forms,
  Umain in 'Umain.pas' {frmmain},
  Uoption in 'Uoption.pas' {frmoption},
  Uemail in 'Uemail.pas' {frmemail},
  Unew in 'Unew.pas' {frmnew},
  Usource in 'Usource.pas' {frmsource},
  Usearch in 'Usearch.pas' {frmsearch},
  Uabout in 'Uabout.pas' {frmabout};

{$R *.RES}

begin
  Application.Initialize;
  Application.CreateForm(Tfrmmain, frmmain);
  Application.CreateForm(Tfrmoption, frmoption);
  Application.CreateForm(Tfrmemail, frmemail);
  Application.CreateForm(Tfrmnew, frmnew);
  Application.CreateForm(Tfrmsource, frmsource);
  Application.CreateForm(Tfrmsearch, frmsearch);
  Application.CreateForm(Tfrmabout, frmabout);
  Application.Run;
end.
```

3.1.2. Codes of umain.pas :

```
unit umain;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ComCtrls, Menus, StdCtrls, Buttons, ExtCtrls, ToolWin, ScktComp,
  OleCtrls, isp3, ImgList, NMHTML, StdActns, ActnList, Printers;

type
  Tfrmmain = class(TForm)
    hotImageList1: TImageList;
    PrintDialog1: TPrintDialog;
    Panel: TPanel;
    imgback: TImage;
    imgnext: TImage;
    imgstop: TImage;
    imgrefresh: TImage;
    imghome: TImage;
    imgemail: TImage;
    imgsearch: TImage;
    imgprint: TImage;
```


imgchannel: TImage;
imgoption: TImage;
lblinfo: TLabel;
cmbaddress: TComboBox;
imgstyle1: TImage;
imgminimize: TImage;
imgmaximize: TImage;
imgclose: TImage;
shpleft: TShape;
shptop: TShape;
shpbottom: TShape;
shpright: TShape;
lblfile: TLabel;
lblcaption: TLabel;
Image1: TImage;
lbledit: TLabel;
lblview: TLabel;
lblmail: TLabel;
lblabout: TLabel;
lblcaption1: TLabel;
shpcaption: TShape;
Label3: TLabel;
Label5: TLabel;
Label6: TLabel;
Label8: TLabel;
Label9: TLabel;
imgclient: TImage;
Image3: TImage;
Panel1: TPanel;
popfile: TPopupMenu;
popedit: TPopupMenu;
popview: TPopupMenu;
popemail: TPopupMenu;
popabout: TPopupMenu;
New2: TMenuItem;
Copy2: TMenuItem;
Cut2: TMenuItem;
Paste2: TMenuItem;
Clear2: TMenuItem;
FullScreen2: TMenuItem;
ViewSource1: TMenuItem;
WriteEMail1: TMenuItem;
ReadMail1: TMenuItem;
Settings1: TMenuItem;
AboutWEBBrowser1: TMenuItem;
HTML1: THTML;
lblstatus: TLabel;
aniglobe: TAnimate;
anibrowsing: TAnimate;
MainMenu: TMainMenu;
File1: TMenuItem;
New1: TMenuItem;
Properties1: TMenuItem;
Print1: TMenuItem;
PrinterSetup1: TMenuItem;
Close1: TMenuItem;
Edit1: TMenuItem;
Copy1: TMenuItem;
Cut1: TMenuItem;
Paste1: TMenuItem;

```

Clear1: TMenuItem;
SelectAll1: TMenuItem;
View1: TMenuItem;
FullScreen1: TMenuItem;
Font1: TMenuItem;
Options1: TMenuItem;
Favorites1: TMenuItem;
Add1: TMenuItem;
Tools1: TMenuItem;
EMail1: TMenuItem;
SearchonWEB1: TMenuItem;
Help1: TMenuItem;
About1: TMenuItem;
anisystem: TAnimate;
imgpanel1: TImage;
imgpanel2: TImage;
ActionList1: TActionList;
Copy3: TEditCopy;
Cut3: TEditCut;
Paste3: TEditPaste;
popsystem: TPopupMenu;
Restore1: TMenuItem;
Minimize1: TMenuItem;
Maximize1: TMenuItem;
Close2: TMenuItem;
About2: TMenuItem;
Image2: TImage;
Exit1: TMenuItem;
ClearHistory1: TMenuItem;
PrinterSettings1: TMenuItem;
Print2: TMenuItem;
PrinterSetupDialog1: TPrinterSetupDialog;
ProgressBar1: TProgressBar;
procedure FormCreate(Sender: TObject);
procedure FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure cmbaddressClick(Sender: TObject);
procedure FindAddress;
procedure cmbaddressKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure HTML1BeginRetrieval(Sender: TObject);
procedure HTML1EndRetrieval(Sender: TObject);
procedure Close1Click(Sender: TObject);
procedure PrinterSetup1Click(Sender: TObject);
procedure imgbackMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure PanelMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure imgnextMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure pasif;
procedure imgstopMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure imgrefreshMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure lblinfoMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure imghomeMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure imgemailMouseMove(Sender: TObject; Shift: TShiftState; X,

```

```

Y: Integer);
procedure imgsearchMouseMove(Sender: TObject; Shift: TShiftState; X,
Y: Integer);
procedure imgprintMouseMove(Sender: TObject; Shift: TShiftState; X,
Y: Integer);
procedure imgchannelMouseMove(Sender: TObject; Shift: TShiftState; X,
Y: Integer);
procedure imgoptionMouseMove(Sender: TObject; Shift: TShiftState; X,
Y: Integer);
procedure imgstopClick(Sender: TObject);
procedure imgrefreshClick(Sender: TObject);
procedure imgprintClick(Sender: TObject);
procedure imgstyle1MouseMove(Sender: TObject; Shift: TShiftState; X,
Y: Integer);
procedure imgstyle1Click(Sender: TObject);
procedure imgcloseClick(Sender: TObject);
procedure imgminimizeClick(Sender: TObject);
procedure imgmaximizeClick(Sender: TObject);
procedure HTML1DoRequestDoc(Sender: TObject; const URL: WideString;
const Element: HTML_Element; const DocInput: DocInput;
var EnableDefault: WordBool);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure lblcaptionDragOver(Sender, Source: TObject; X, Y: Integer;
State: TDragState; var Accept: Boolean);
procedure lblcaptionDragDrop(Sender, Source: TObject; X, Y: Integer);
procedure lblcaptionMouseMove(Sender: TObject; Shift: TShiftState; X,
Y: Integer);
procedure backclick(sender:TObject);
procedure nextclick(sender:TObject);
procedure imgbackClick(Sender: TObject);
procedure imgnextClick(Sender: TObject);
procedure imgemailClick(Sender: TObject);
procedure shprightMouseMove(Sender: TObject; Shift: TShiftState; X,
Y: Integer);
procedure shpleftMouseMove(Sender: TObject; Shift: TShiftState; X,
Y: Integer);
procedure shptopMouseMove(Sender: TObject; Shift: TShiftState; X,
Y: Integer);
procedure shpbottomMouseMove(Sender: TObject; Shift: TShiftState; X,
Y: Integer);
procedure FormDeactivate(Sender: TObject);
procedure FormActivate(Sender: TObject);
procedure lblfileMouseMove(Sender: TObject; Shift: TShiftState; X,
Y: Integer);
procedure lblcaptionMouseDown(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
procedure imgoptionClick(Sender: TObject);
procedure imghomeClick(Sender: TObject);
procedure FormKeyDown(Sender: TObject; var Key: Word;
Shift: TShiftState);
procedure ViewFull;
procedure SetAnchors;
procedure imgchannelClick(Sender: TObject);
procedure Image2Click(Sender: TObject);
procedure lbleditMouseMove(Sender: TObject; Shift: TShiftState; X,
Y: Integer);
procedure lblviewMouseMove(Sender: TObject; Shift: TShiftState; X,
Y: Integer);
procedure lblmailMouseMove(Sender: TObject; Shift: TShiftState; X,
Y: Integer);

```



```

procedure laboutMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure New2Click(Sender: TObject);
procedure ViewSource1Click(Sender: TObject);
procedure ClearHistory1Click(Sender: TObject);
procedure PrinterSettings1Click(Sender: TObject);
procedure imgsearchClick(Sender: TObject);
procedure HTML1UpdateRetrieval(Sender: TObject);
procedure HTML1LayoutComplete(Sender: TObject);
procedure WriteEMail1Click(Sender: TObject);
procedure ReadMail1Click(Sender: TObject);
procedure Settings1Click(Sender: TObject);
procedure AboutWEBBrowser1Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  frmmain: Tfrmmain;

implementation

uses uemail, uoption, usource, usearch, unew, uabout;

{$R *.DFM}

var FullScr, RefreshCombo: boolean;  strnext, strback: string[1]; // to control buttons enabled or not.
    HistoryIndex, bbar, lastaktif: integer;
    HistoryList: TStringList;      filemail, filehome: textfile;

procedure Tfrmmain.FormCreate(Sender: TObject);
var s: string;
begin
  application.Icon.LoadFromFile(ExtractFilePath(application.exename) + '\avi\buttons\EARTH.ico');
  strback:='2';  strnext:='2';          // back&next buttons are disabled
  HistoryList:=TStringList.Create; HistoryIndex:=-1; // for listing NEXT and BACK
  lastaktif:=11;
  // check history.txt
  if not fileexists(ExtractFilePath(application.exename) + '\Data\history.txt') then begin
    FileCreate(ExtractFilePath(application.exename) + '\data\history.txt');
  end;
  // load visited addresses to address bar
  cmbaddress.Items.LoadFromFile(ExtractFilePath(application.exename) + '\Data\history.txt');

  // check options.txt
  AssignFile(filehome, ExtractFilePath(application.exename) + 'data\options.txt');
  if not FileExists(ExtractFilePath(application.exename) + 'data\options.txt') then begin
    Rewrite(filehome); Filemode:=1;
    // write defaults to the file
    writeln(filehome, 'www.yahoo.com');
    writeln(filehome, 'Arial');
    writeln(filehome, '10');
    flush(filehome); closefile(filehome);
  end;
  // check mail settings - mail.txt
  if not FileExists(ExtractFilePath(application.exename) + 'data\mail.txt') then begin
    AssignFile(filemail, ExtractFilePath(application.exename) + 'data\mail.txt');
    Rewrite(filemail);
  end;
end;

```



```

Filemode:=1;
writeln(filemail,'Serhat TUNCA');
writeln(filemail,'serhattunca@hotmail.com');
writeln(filemail,'mail.hotmail.com');
closefile(filemail);
end;
// check signature - sign.txt
if not FileExists(ExtractFilePath(application.exename) + 'data\sign.txt') then
  FileCreate(ExtractFilePath(application.exename) + '\data\sign.txt');

RefreshCombo:=false; FullScr:=false; bbar:=3;
aniglobe.FileName :=ExtractFilePath(application.Exename) + '\avi\globe7.avi';
anibrowsing.FileName :=ExtractFilePath(application.Exename) + '\avi\text.avi';
postmessage(Handle,WM_USER + $10000,0,0);
// load
{$I-} reset(filehome); {$I+}
if ioresult<>0 then begin
  showmessage(ExtractFilePath(application.exename) + 'data\options.txt can not be opened. ');
  exit;
end;
readln(filehome,s);  cmbaddress.Text :=s;
readln(filehome,s);  HTML1.Font.Name := s;
readln(filehome,s);  HTML1.Font.Size :=strtoint(s);
closefile(filehome);
end;

procedure Tfrmmain.FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
  pasif;
  if (x=frmmain.Width-3)and(y<frmmain.Height - 3) then begin
    frmmain.Cursor :=crSizeWE;
  end
  else if (x=frmmain.Width-3)and(y=frmmain.Height - 3) then begin
    frmmain.Cursor :=crSizeNWSE;
  end
  else if (x<frmmain.Width-3)and(y=frmmain.Height - 3) then begin
    frmmain.Cursor :=crSizeNS;
  end
  else
    frmmain.Cursor :=crDefault;
end;

procedure Tfrmmain.cmbaddressClick(Sender: TObject);
begin
  if cmbaddress.text<>'' then begin
    RefreshCombo:=true; FindAddress;
  end;
end;

procedure Tfrmmain.FindAddress;
begin
  HTML1.RequestDoc(cmbaddress.Text);
  caption :='WEBrowser ' + cmbaddress.Text;
end;

```

```

procedure Tfrmmain.cmbaddressKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  if Key=VK_return then begin
    RefreshCombo:=true;
    FindAddress;
  end;
end;

procedure Tfrmmain.HTML1BeginRetrieval(Sender: TObject);
begin
  progressbar1.Position :=0;
  if FullScr then anibrowsing.Active :=true else aniglobe.Active :=true;
end;

procedure Tfrmmain.HTML1EndRetrieval(Sender: TObject);
begin
  if FullScr then anibrowsing.Active :=false else aniglobe.Active :=false;
end;

procedure Tfrmmain.Close1Click(Sender: TObject);
begin
  close;
end;

procedure Tfrmmain.PrinterSetup1Click(Sender: TObject);
begin
  printdialog1.Execute;
end;

procedure Tfrmmain.imgstyle1MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
  pasif;
  if bbar=1 then
    imgstyle1.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\big1.bmp')
  else imgstyle1.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\small1.bmp');
  lastaktif:=0;
end;

procedure Tfrmmain.imgbackMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
  if strback="" then begin
    pasif;
    imgback.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\back1.bmp');
    lastaktif:=1;
  end;
end;

procedure Tfrmmain.PanelMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
  pasif;
end;

```

```

procedure Tfrmmain.imgnextMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
  if strnext="" then begin
    pasif;
    imgnext.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\next1.bmp');
    lastaktif:=2;
  end;
end;

procedure Tfrmmain.pasif;
begin
  case lastaktif of
    0 : begin
      if bbar =1 then
        imgstyle1.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\big.bmp') else
        imgstyle1.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\small.bmp');
      end;
    1 :
      imgback.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\back'+strback+'.bm
p');
    2 :
      imgnext.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\next'+strnext+'.bmp'
);
    3 : imgstop.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\stop.bmp');
    4 :
      imgrefresh.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\refresh.bmp');
    5 : imghome.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\home.bmp');
    6 : imgemail.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\email.bmp');
    7 : imgsearch.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\search.bmp');
    8 : imgprint.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\print.bmp');
    9 :
      imgchannel.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\channel.bmp');
    10: imgoption.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\option.bmp');
  end;
  lastaktif:=11;
end;

```

```

procedure Tfrmmain.imgstopMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
  pasif;
  imgstop.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\stop1.bmp');
  lastaktif:=3;
end;

```

```

procedure Tfrmmain.imgrefreshMouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
  pasif;
  imgrefresh.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\refresh1.bmp');
  lastaktif:=4;
end;

```

```

procedure Tfrmmain.lblinfoMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
  case lastaktif of
    0 : begin
      if bbar =1 then
        imgstyle1.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\big.bmp') else
        imgstyle1.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\small.bmp');
      end;
    1 :
      imgback.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\back'+strback+'.bm
p');
    2 :
      imgnext.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\next'+strnext+'.bmp'
);
    3 : imgstop.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\stop.bmp');
    4 :
      imgrefresh.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\refresh.bmp');
    5 : imghome.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\home.bmp');
    6 : imgemail.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\email.bmp');
    7 : imgsearch.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\search.bmp');
    8 : imgprint.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\print.bmp');
    9 :
      imgchannel.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\channel.bmp');
    10: imgoption.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\option.bmp');
  end;
  lastaktif:=11;
end;

```

```

procedure Tfrmmain.imghomeMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
  pasif;
  imghome.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\home1.bmp');
  lastaktif:=5;
end;

```

```

procedure Tfrmmain.imgemailMouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
  pasif;
  imgemail.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\email1.bmp');
  lastaktif:=6;
end;

```

```

procedure Tfrmmain.imgsearchMouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
  pasif;
  imgsearch.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\search1.bmp');
  lastaktif:=7;
end;

```

```

procedure Tfrmmain.imgprintMouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
  pasif;
  imgprint.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\print1.bmp');
  lastaktif:=8;

```



```

end;

procedure Tfrmmain.imgchannelMouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
  pasif;
  imgchannel.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\channel1.bmp');
  lastaktif:=9;
end;

procedure Tfrmmain.imgoptionMouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
  pasif;
  imgoption.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\option1.bmp');
  lastaktif:=10;
end;

procedure Tfrmmain.imgstopClick(Sender: TObject);
begin
  HTML1.Cancel('Cancel');
end;

procedure Tfrmmain.imgrefreshClick(Sender: TObject);
begin
  if cmbaddress.text<>'' then FindAddress;
end;

procedure Tfrmmain.imgprintClick(Sender: TObject);
var DonePrinting:boolean; xPage:LongInt;
begin
  Printer.BeginDoc;
  HTML1.BeginPrinting(Printer.Handle,0,0,
    printer.PageWidth ,printer.PageHeight,"");
  xPage:=1;
  repeat
    HTML1.PrintPage(Printer.Handle,xPage);
    xPage:=xPage+1;
    DonePrinting:=HTML1.IsPrintingDone[xPage];
    Printer.NewPage
  until (not DonePrinting);
  HTML1.EndPrinting;
  Printer.EndDoc;
end;

procedure Tfrmmain.imgstyle1Click(Sender: TObject);
begin
  case bbar of
    3 : begin
      lblinfo.Visible :=false;
      imgnext.Left :=imgnext.Left - 25;
      imgstop.Left :=imgstop.Left - 50;
      imgrefresh.Left :=imgrefresh.Left - 75;
      imghome.Left :=imghome.Left - 100;
      imgemail.Left :=imgemail.Left - 125;
      imgsearch.Left :=imgsearch.Left - 150;
      imgprint.Left :=imgprint.Left - 175;
      imgchannel.Left :=imgchannel.Left - 200;
      imgoption.Left :=imgoption.Left - 225;
      cmbaddress.Top :=30;
    end;
  end;
end;

```

```

cmbaddress.Width :=cmbaddress.Width - 40;
panel1.Top :=57; panel1.Height :=panel1.Height + 15;
bbar:=2;
end;
2 : begin
cmbaddress.Left :=325; cmbaddress.Top :=5;
cmbaddress.Width :=imgclient.Width - 340;
panel1.Top :=32; panel1.Height :=panel1.Height + 25;
aniglobe.Visible :=false;
bbar:=1;
end;
1 : begin
aniglobe.Visible :=true;
imgnext.Left :=imgnext.Left + 25;
imgstop.Left :=imgstop.Left + 50;
imgrefresh.Left :=imgrefresh.Left + 75;
imghome.Left :=imghome.Left + 100;
imgemail.Left :=imgemail.Left + 125;
imgsearch.Left :=imgsearch.Left + 150;
imgprint.Left :=imgprint.Left + 175;
imgchannel.Left :=imgchannel.Left + 200;
imgoption.Left :=imgoption.Left + 225;
lblinfo.Visible :=true;
cmbaddress.Left :=40; cmbaddress.Top :=45;
cmbaddress.Width :=imgclient.Width - 56;
panel1.Top :=72; panel1.Height :=panel1.Height -40;
bbar:=3;
end;
end;
end;

procedure Tfrmmain.imgcloseClick(Sender: TObject);
begin
frmmain.close;
end;

procedure Tfrmmain.imgminimizeClick(Sender: TObject);
begin
application.Minimize;
end;

procedure Tfrmmain.imgmaximizeClick(Sender: TObject);
begin
if frmmain.WindowState <> wsMaximized then begin
frmmain.WindowState :=wsMaximized;
imgmaximize.Picture.LoadFromFile(ExtractFilePath(application.exename) +
'avi\buttons\restore.bmp');
shpright.Visible :=false; shpleft.Visible :=false;
shpbottom.Visible :=false; shptop.Visible :=false;
end else begin
frmmain.WindowState :=wsNormal;
imgmaximize.Picture.LoadFromFile(ExtractFilePath(application.exename) +
'avi\buttons\maximize.bmp');
shpright.Visible :=true; shpleft.Visible :=true;
shpbottom.Visible :=true; shptop.Visible :=true;
end;
end;
end;

```

```

procedure Tfrmmain.HTML1DoRequestDoc(Sender: TObject;
const URL: WideString; const Element: HTML_Element;
const DocInput: DocInput; var EnableDefault: WordBool);
var
  NewIndex: Integer;
begin
  NewIndex := HistoryList.IndexOf(URL);
  if NewIndex = -1 then
  begin
    { Remove entries in HistoryList between last address and current address }
    if (HistoryIndex >= 0) and (HistoryIndex < HistoryList.Count - 1) then
      while HistoryList.Count > HistoryIndex do
        HistoryList.Delete(HistoryIndex);
    HistoryIndex := HistoryList.Add(URL);
  end
  else
    HistoryIndex := NewIndex;
  if HistoryList.Count > 0 then
  begin
    if (HistoryIndex < HistoryList.Count - 1) then begin
      imgnext.Picture.LoadFromFile(ExtractFilePath(application.exename) + '\avi\buttons\next.bmp');
      strnext:='';
    end else begin
      imgnext.Picture.LoadFromFile(ExtractFilePath(application.exename) + '\avi\buttons\next2.bmp');
      strnext:='2';
    end;
    if (HistoryIndex > 0) then begin
      imgback.Picture.LoadFromFile(ExtractFilePath(application.exename) + '\avi\buttons\back.bmp');
      strback:='';
    end else begin
      imgback.Picture.LoadFromFile(ExtractFilePath(application.exename) + '\avi\buttons\back2.bmp');
      strback:='2';
    end;
  end
  else
  begin
    strback:='2'; strnext:='2';
    imgnext.Picture.LoadFromFile(ExtractFilePath(application.exename) + '\avi\buttons\next2.bmp');
    imgback.Picture.LoadFromFile(ExtractFilePath(application.exename) + '\avi\buttons\back2.bmp');
  end;
  if RefreshCombo then
  begin
    RefreshCombo := False;
    NewIndex := cmbaddress.Items.IndexOf(URL);
    if NewIndex = -1 then
      cmbaddress.Items.Insert(1, URL)
    else
      cmbaddress.Items.Move(NewIndex, 1);
  end;
  cmbaddress.Text := URL;
  lblcaption.Caption := 'WEBBrowser - ' + URL;
  lblcaption1.Caption := 'WEBBrowser - ' + URL;
  lblstatus.caption := URL;
end;

procedure Tfrmmain.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  cmbaddress.Items.SaveToFile (ExtractFilePath(application.exename)+'\data\history.txt');
  HistoryList.Free;

```

```

end;

procedure Tfrmmain.lblcaptionDragOver(Sender, Source: TObject; X,
  Y: Integer; State: TDragState; var Accept: Boolean);
begin
  if State = dsDragMove then begin
    frmmain.Left :=x; frmmain.Top :=y;
  end;
end;

procedure Tfrmmain.lblcaptionDragDrop(Sender, Source: TObject; X,
  Y: Integer);
begin
  frmmain.Left :=x; frmmain.Top :=y;
end;

var oldx,oldy:integer;

procedure Tfrmmain.lblcaptionMouseDown(Sender: TObject;
  Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
  oldx:=x; oldy:=y;
end;

procedure Tfrmmain.lblcaptionMouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
  if (shift = [ssLeft])and(frmmain.WindowState <> wsMaximized) then begin
    frmmain.Left := frmmain.left + (x-oldx) ; frmmain.Top :=frmmain.Top + (y-oldy);
  end;
end;

procedure Tfrmmain.backclick(sender: TObject);
begin
  cmbaddress.Text :=(sender as TMenuItem).Caption;
  FindAddress;
end;

procedure Tfrmmain.nextclick(sender:TObject);
begin
  cmbaddress.Text :=(sender as TMenuItem).Caption;
  FindAddress;
end;

procedure Tfrmmain.imgbackClick(Sender: TObject);
begin
  if strback="" then begin
    cmbaddress.Text :=HistoryList[HistoryIndex - 1];
    FindAddress;
  end;
end;

procedure Tfrmmain.imgnextClick(Sender: TObject);
begin
  if strnext="" then begin
    cmbaddress.Text :=HistoryList[HistoryIndex + 1];
    FindAddress;
  end;
end;

```



```

procedure Tfrmmain.imgemailClick(Sender: TObject);
begin
  frmemail.show;
end;

procedure Tfrmmain.shprightMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
  if (shift=[ssLeft]) then begin
    frmmain.width:=x+frmmain.Width ;
    // if (frmmain.Width < 400) then shift:=[ssRight];
  end;
end;

procedure Tfrmmain.shpleftMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
  if (shift=[ssLeft]){and(frmmain.Width > 100)} then begin
    frmmain.width:=frmmain.Width - x ;
    frmmain.left := frmmain.left + x ;
  end;
end;

procedure Tfrmmain.shptopMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
  if (shift=[ssLeft]){and(frmmain.Height > 100)} then begin
    frmmain.height :=frmmain.height - y ;
    frmmain.top := frmmain.top + y ;
  end;
end;

procedure Tfrmmain.shpbottomMouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
  if (shift=[ssLeft]){and(frmmain.Height > 100)} then begin
    frmmain.Height:=y+frmmain.Height ;
  end;
end;

procedure Tfrmmain.FormDeactivate(Sender: TObject);
begin
  shpcaption.Brush.Color :=clgray;
  lblcaption.Font.Color :=clsilver;
  anisystem.Active :=false;
end;

procedure Tfrmmain.FormActivate(Sender: TObject);
begin
  shpcaption.Brush.Color :=$00FF8000;
  lblcaption.Font.Color :=clYellow;
  anisystem.Active :=true;
end;

```

```

procedure Tfrmmain.lblfileMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
  popfile.Popup(x+frmmain.Left+lblfile.Left ,y+frmmain.Top + 30 );
end;

procedure Tfrmmain.imgoptionClick(Sender: TObject);
begin
  frmoption.show;
end;

procedure Tfrmmain.imghomeClick(Sender: TObject);
var s:string;
begin
  pasif;
  AssignFile(filehome,ExtractFilePath(application.exename)+'data\options.txt');
  {$I-} reset(filehome); {$I+}
  if ioresult<>0 then begin
    showmessage(ExtractFilePath(application.exename)+'data\options.txt cannot be opened. ');
    exit;
  end;
  readln(filehome,s);  closefile(filehome);
  cmbaddress.Text :=s;  RefreshCombo:=true;
  FindAddress;
end;

procedure Tfrmmain.FormKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  if key=VK_F12 then ViewFull;
end;

procedure Tfrmmain.ViewFull;
begin
  if not FullScr then begin
    anibrowsing.Active :=aniglobe.Active;
    lblstatus.Visible :=false;  progressbar1.Visible :=false;
    if frmmain.WindowState <> wsMaximized then imgmaximize.OnClick (self); // max the window
    shpcaption.Visible :=false; // hide window caption
    lblcaption.Visible :=false;
    lblcaption1.Visible :=false;
    imgpanel1.Visible :=true; // show system buttons
    imgpanel2.Visible :=true;
    lblfile.Visible :=false; // hide main menu
    lbledit.Visible :=false;
    lblview.Visible :=false;
    lblmail.Visible :=false;
    lblabout.Visible :=false;
    lblinfo.Visible :=false; // hide buttons caption
    aniglobe.Visible :=false; // change avi
    anibrowsing.Visible :=true;
    panel.Align :=alClient; // set panel as client
    panel1.Top :=30; // set HTML size and place
    panel1.Height :=imgclient.Height - 31;
    cmbaddress.Left :=325; // move address bar to top
    cmbaddress.Top :=5; cmbaddress.Width :=300;
    if bbar=3 then begin
      imgnext.Left :=imgnext.Left - 25;  imgstop.Left :=imgstop.Left - 50;
      imgrefresh.Left :=imgrefresh.Left - 75;  imghome.Left :=imghome.Left - 100;
    end;
  end;
end;

```

```

imgemail.Left :=imgemail.Left - 125;
imgsearch.Left :=imgsearch.Left - 150;
imgprint.Left :=imgprint.Left - 175;
imgchannel.Left :=imgchannel.Left - 200;
imgoption.Left :=imgoption.Left - 225;
lblinfo.Visible :=true; // show buttons caption
end;
FullScr:=true;
end else begin
aniglobe.Active :=anibrowsing.Active;
lblstatus.Visible :=true; progressbar1.Visible :=true;
imgmaximize.OnClick (self); // restore window
shpcaption.Visible :=true; // show window caption
lblcaption.Visible :=true;
lblcaption1.Visible :=true;
imgpanel1.Visible :=false; // hide system buttons
imgpanel2.Visible :=false;
lblfile.Visible :=true; // show main menu
bledit.Visible :=true;
lblview.Visible :=true;
lblmail.Visible :=true;
lblabout.Visible :=true;
aniglobe.Visible :=true; // change avi
anibrowsing.Visible :=false;
panel.Align :=alNone; // set panel to its original size
panel.Top :=48; panel.Left :=7;
panel.Width := imgclient.width-7; panel.Height :=imgclient.Height - 50;
panel1.Top :=72; // set HTML size and place
panel1.Height :=imgclient.Height - 143;
cmbaddress.Left :=40; // move address bar to back
cmbaddress.Top :=45; cmbaddress.Width :=frmmain.width - 71;
if bbar=3 then begin
imgnext.Left :=imgnext.Left + 25; // put buttons to their original place
imgstop.Left :=imgstop.Left + 50;
imgrefresh.Left :=imgrefresh.Left + 75;
imghome.Left :=imghome.Left + 100;
imgemail.Left :=imgemail.Left + 125;
imgsearch.Left :=imgsearch.Left + 150;
imgprint.Left :=imgprint.Left + 175;
imgchannel.Left :=imgchannel.Left + 200;
imgoption.Left :=imgoption.Left + 225;
end;
SetAnchors; // Refresh all object anchors
FullScr:=false;
end;
end;

procedure Tfrmmain.SetAnchors;
begin
panel.Anchors :=[akLeft,akTop,akRight,akBottom];
cmbaddress.Anchors :=[akLeft,akTop,akRight];
aniglobe.Anchors :=[akTop,akRight];
panel1.Anchors :=[akLeft,akTop,akRight,akBottom];
end;

procedure Tfrmmain.imgchannelClick(Sender: TObject);
begin
pasif;
ViewFull;
end;

```

```

procedure Tfrmmain.Image2Click(Sender: TObject);
begin
  popsystem.Popup(frmmain.Left + 10,frmmain.Top + 10);
end;

procedure Tfrmmain.IbleditMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
  popedit.Popup(x+frmmain.Left+Ibledit.Left ,y+frmmain.Top + 30);
end;

procedure Tfrmmain.IblviewMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
  popview.Popup(x+frmmain.Left+Iblview.Left ,y+frmmain.Top + 30 );
end;

procedure Tfrmmain.IblmailMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
  popemail.Popup(x+frmmain.Left+Iblmail.Left ,y+frmmain.Top + 30);
end;

procedure Tfrmmain.IblaboutMouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
  popabout.Popup(x+frmmain.Left+Iblabout.Left ,y+frmmain.Top + 30);
end;

procedure Tfrmmain.New2Click(Sender: TObject);
begin
  WinExec(PChar(application.exename),SW_SHOW);
end;

procedure Tfrmmain.ViewSource1Click(Sender: TObject);
begin
  frmsource.show;
end;

procedure Tfrmmain.ClearHistory1Click(Sender: TObject);
begin
  cmbaddress.Items.Clear;
  cmbaddress.Items.Add(' ');
  strback:='2'; // imgback picture = back2
  strnext:='2'; // imgback picture = back2
  HistoryList.Clear;
end;

procedure Tfrmmain.PrinterSettings1Click(Sender: TObject);
begin
  printersetupdialog1.Execute;
end;

procedure Tfrmmain.imgsearchClick(Sender: TObject);
begin
  frmsearch.show;
end;

```



```

procedure Tfrmmain.HTML1UpdateRetrieval(Sender: TObject);
var receivedbytes,totalbytes,percent:integer;
begin
  TotalBytes :=HTML1.RetrieveBytesTotal;
  ReceivedBytes :=HTML1.RetrieveBytesDone;
  if ReceivedBytes>0 then begin
    Percent := Round((TotalBytes / ReceivedBytes) * 100);
    progressbar1.Position :=Percent;
  end;
end;

procedure Tfrmmain.HTML1LayoutComplete(Sender: TObject);
begin
  lblstatus.Caption :='Done...';
end;

procedure Tfrmmain.WriteEMail1Click(Sender: TObject);
begin
  frmnew.show;
end;

procedure Tfrmmain.ReadMail1Click(Sender: TObject);
begin
  frmemail.Show;
end;

procedure Tfrmmain.Settings1Click(Sender: TObject);
begin
  frmoption.show;
end;

procedure Tfrmmain.AboutWEBrowser1Click(Sender: TObject);
begin
  frmabout.show;
end;

end.

```

3.1.3. Codes of uemail.pas :

```

unit uemail;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  Menus, ComCtrls, ExtCtrls, Grids, StdCtrls, ImgList, Psock, NMsmtp,
  ToolWin, Buttons, Printers;

type
  Tfrmemail = class(TForm)
    Panel4: TPanel;
    memomail: TMemo;
    Panel5: TPanel;
    imgmaximize: TImage;
    imgminimize: TImage;
    imgclose: TImage;

```

```

Image2: TImage;
smtp1: TNMSMTP;
lblcaption: TLabel;
lblcaption1: TLabel;
shpcaption: TShape;
ToolBar1: TToolBar;
btnnew: TToolButton;
shpleft: TShape;
shptop: TShape;
shpright: TShape;
shpbottom: TShape;
ImageList2: TImageList;
Panel2: TPanel;
Panel7: TPanel;
Splitter2: TSplitter;
Panel8: TPanel;
Panel1: TPanel;
gridfolder: TStringGrid;
pnltile: TPanel;
Splitter1: TSplitter;
btnprint: TToolButton;
ToolButton5: TToolButton;
btndelete: TToolButton;
ToolButton7: TToolButton;
btnclose: TToolButton;
ImageList3: TImageList;
btnmailbox: TSpeedButton;
btnsent: TSpeedButton;
btntrash: TSpeedButton;
lblmailbox: TLabel;
lblsent: TLabel;
lbltrash: TLabel;
anisystem: TAnimate;
btnset: TToolButton;
btnsend: TToolButton;
memo2: TMemo;
memostatus: TMemo;
procedure imgcloseClick(Sender: TObject);
procedure lblcaptionMouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
procedure lblcaptionMouseMove(Sender: TObject; Shift: TShiftState; X,
    Y: Integer);
procedure shpleftMouseMove(Sender: TObject; Shift: TShiftState; X,
    Y: Integer);
procedure shptopMouseMove(Sender: TObject; Shift: TShiftState; X,
    Y: Integer);
procedure shprightMouseMove(Sender: TObject; Shift: TShiftState; X,
    Y: Integer);
procedure shpbottomMouseMove(Sender: TObject; Shift: TShiftState; X,
    Y: Integer);
procedure FormActivate(Sender: TObject);
procedure FormDeactivate(Sender: TObject);
procedure btnmailboxClick(Sender: TObject);
procedure btnsentClick(Sender: TObject);
procedure btntrashClick(Sender: TObject);
procedure TxtToMail(fxt,folder:string);
procedure ClearGrid;
procedure btnnewClick(Sender: TObject);
procedure imgmaximizeClick(Sender: TObject);
procedure imgminimizeClick(Sender: TObject);

```

```

procedure btnprintClick(Sender: TObject);
procedure btnsetClick(Sender: TObject);
procedure btndeleteClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure gridfolderSelectCell(Sender: TObject; ACol, ARow: Integer;
  var CanSelect: Boolean);
procedure btnsendClick(Sender: TObject);
procedure smtp1Connect(Sender: TObject);
procedure smtp1ConnectionFailed(Sender: TObject);
procedure smtp1Disconnect(Sender: TObject);
procedure smtp1Success(Sender: TObject);
procedure smtp1SendStart(Sender: TObject);
procedure smtp1Failure(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  frmemail: Tfrmemail;

implementation

uses unew, uoption;

{$R *.DFM}

procedure Tfrmemail.imgcloseClick(Sender: TObject);
begin
  frmemail.close;
end;

var oldx,oldy:integer;

// Move Window
procedure Tfrmemail.lblcaptionMouseDown(Sender: TObject;
  Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
  oldx:=x;  oldy:=y;
end;

procedure Tfrmemail.lblcaptionMouseMove(Sender: TObject;
  Shift: TShiftState; X, Y: Integer);
begin
  if (shift = [ssLeft])and(frmemail.WindowState <> wsMaximized) then begin
    frmemail.Left := frmemail.left + (x-oldx) ;  frmemail.Top :=frmemail.Top + (y-oldy);
  end;
end;

// Resize Window - Left
procedure Tfrmemail.shpleftMouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
  if (shift=[ssLeft]){and(frmemail.Width > 100)} then begin
    frmemail.width:=frmemail.Width - x ;
    frmemail.left := frmemail.left + x ;
  end;
end;

```

```

// Resize Window - Top
procedure Tfrmemail.shptopMouseMove(Sender: TObject; Shift: TShiftState; X,
Y: Integer);
begin
if (shift=[ssLeft]){and(frmemail.Height > 100) }then begin
frmemail.height :=frmemail.height - y ;
frmemail.top := frmemail.top + y ;
end;
end;

// Resize Window - Right
procedure Tfrmemail.shprightMouseMove(Sender: TObject; Shift: TShiftState;
X, Y: Integer);
begin
if (shift=[ssLeft]) then begin
frmemail.width:=x+frmemail.Width ;
// if (frmemail.Width < 400) then shift:=[ssRight];
end;
end;

// Resize Window - Bottom
procedure Tfrmemail.shpbottomMouseMove(Sender: TObject; Shift: TShiftState;
X, Y: Integer);
begin
if (shift=[ssLeft]){and(frmemail.Height > 100) }then begin
frmemail.Height:=y+frmemail.Height ;
end;
end;

var emails:TSearchRec; fmail:TextFile; idxgrid:integer; bdel:boolean;
xfolder:string[8];

// Active Window Caption and write grids headers
procedure Tfrmemail.FormActivate(Sender: TObject);
begin
shpcaption.Brush.Color :=$00FF8000;
lblcaption.Font.Color :=clYellow;
anisystem.Active :=true;
gridfolder.Cells[0,0] :='To';
gridfolder.Cells[1,0] :='Subject';
gridfolder.Cells[2,0] :=' ';
gridfolder.Cells[3,0] :='Size';
gridfolder.Cells[4,0] :='File Name';
idxgrid:=1;
if not bdel then begin
btntrash.OnClick(self);
btnsent.OnClick(self);
btnmailbox.OnClick (self);
end;
bdel:=false;
end;

// Deactive window caption
procedure Tfrmemail.FormDeactivate(Sender: TObject);
begin
shpcaption.Brush.Color :=clgray;
lblcaption.Font.Color :=clsilver;
anisystem.Active :=false;
end;

```



```

// Look to Mails in MailBox
procedure Tfrmemail.btnmailboxClick(Sender: TObject);
begin
  xfolder:='MailBox';
  lblmailbox.Font.Style := lblmailbox.Font.Style + [fsBold];
  lblsent.Font.Style := [];
  lbltrash.Font.Style := [];
  lblcaption.Caption := 'Mail Box - WEBrowser';
  lblcaption1.Caption := 'Mail Box - WEBrowser';
  ClearGrid;
  if findfirst(ExtractFilePath(application.exename)+'\email\MailBox\*.txt',faAnyFile,emails)=0 then begin
    TxtToMail(emails.Name,'MailBox');
    gridfolder.RowCount :=gridfolder.RowCount - 1;
  end;
  while findnext(emails)=0 do begin
    TxtToMail(emails.Name,'MailBox');
  end;
  pnltitle.Caption := inttostr(idxgrid-1) + ' mail(s) listed.';
  if idxgrid > 1 then begin
    btnmailbox.Glyph.LoadFromFile(ExtractFilePath(application.exename)+'\avi\buttons\folder1.bmp');
    btndelete.Enabled :=true;  btnprint.Enabled :=true;
  end else begin
    btnmailbox.Glyph.LoadFromFile(ExtractFilePath(application.exename)+'\avi\buttons\folder.bmp');
    btndelete.Enabled :=false;  btnprint.Enabled :=false;
  end;
  // load mail text into memomail
  if gridfolder.cells[4,gridfolder.Row] <>" then
    memomail.Lines.LoadFromFile(
      ExtractFilePath(application.exename)+'\email\'+xfolder+'\''+gridfolder.cells[4,gridfolder.Row]);
end;

// Look to Sent Mails
procedure Tfrmemail.btnsentClick(Sender: TObject);
begin
  xfolder:='SentMail';
  lblmailbox.Font.Style := [];
  lblsent.Font.Style := lblsent.Font.Style + [fsBold];
  lbltrash.Font.Style := [];
  lblcaption.Caption := 'Sent Mail - WEBrowser';
  lblcaption1.Caption := 'Sent Mail - WEBrowser';
  ClearGrid;
  if findfirst(ExtractFilePath(application.exename)+'\email\SentMail\*.txt',faAnyFile,emails)=0 then begin
    TxtToMail(emails.Name,'SentMail');
    gridfolder.RowCount :=gridfolder.RowCount - 1;
  end;
  while findnext(emails)=0 do begin
    TxtToMail(emails.Name,'SentMail');
  end;
  pnltitle.Caption := inttostr(idxgrid-1) + ' mail(s) listed.';
  if idxgrid > 1 then begin
    btnsent.Glyph.LoadFromFile(ExtractFilePath(application.exename)+'\avi\buttons\folder1.bmp');
    btndelete.Enabled :=true;  btnprint.Enabled :=true;
  end else begin
    btnsent.Glyph.LoadFromFile(ExtractFilePath(application.exename)+'\avi\buttons\folder.bmp');
    btndelete.Enabled :=false;  btnprint.Enabled :=false;
  end;
  // load mail text into memomail
  if gridfolder.cells[4,gridfolder.Row] <>" then

```

```

memomail.Lines.LoadFromFile(
  ExtractFilePath(application.exename)+'\email\' + xfolder + '\' + gridfolder.cells[4,gridfolder.Row]);
end;

// Look to Deleted Mails
procedure Tfrmemail.btntrashClick(Sender: TObject);
begin
  xfolder:='Trash';
  lblmailbox.Font.Style := [];
  lblsent.Font.Style := [];
  lbltrash.Font.Style := lbltrash.Font.Style + [fsBold];
  lblcaption.Caption := 'Trash - WEBrowser';
  lblcaption1.Caption := 'Trash - WEBrowser';
  ClearGrid;
  if findfirst(ExtractFilePath(application.exename)+'\email\Trash\*.txt',faAnyFile,emails)=0 then begin
    TxtToMail(emails.Name,'Trash');
    gridfolder.RowCount := gridfolder.RowCount - 1;
  end;
  while findnext(emails)=0 do begin
    TxtToMail(emails.Name,'Trash');
  end;
  pnltitle.Caption := inttostr(idxgrid-1) + ' mail(s) listed.';
  if idxgrid > 1 then begin
    btntrash.Glyph.LoadFromFile(ExtractFilePath(application.exename)+'\avi\buttons\trashful.bmp');
    btndelete.Enabled := true;  btnprint.Enabled := true;
  end else begin
    btntrash.Glyph.LoadFromFile(ExtractFilePath(application.exename)+'\avi\buttons\trash.bmp');
    btndelete.Enabled := false;  btnprint.Enabled := false;
  end;
  // load mail text into memomail
  if gridfolder.cells[4,gridfolder.Row] <> '' then
    memomail.Lines.LoadFromFile(
      ExtractFilePath(application.exename)+'\email\' + xfolder + '\' + gridfolder.cells[4,gridfolder.Row]);
end;

procedure Tfrmemail.TxtToMail(fbtxt, folder:string);
var frow:String; {i:integer; }xmail:TSearchRec;
begin
  gridfolder.RowCount := gridfolder.RowCount + 1;
  AssignFile(fmail,ExtractFilePath(application.exename)+'\email\' + folder + '\' + fbtxt);
  Reset(fmail);
  Readln(fmail,frow); gridfolder.Cells[0, idxgrid] := frow;
  Readln(fmail,frow); gridfolder.Cells[1, idxgrid] := frow;
  CloseFile(fmail);

  xmail.Name := ExtractFilePath(application.exename)+'email\' + folder + '\' + fbtxt;
  gridfolder.Cells[3,idxgrid] := inttostr(emails.size) + ' bytes';
  gridfolder.Cells[4,idxgrid] := emails.Name;
  memomail.Lines.LoadFromFile(ExtractFilePath(application.exename)+'\email\' + folder + '\' + fbtxt);
  idxgrid := idxgrid + 1;
end;

procedure Tfrmemail.ClearGrid;
var i:integer;
begin
  for i:=1 to gridfolder.RowCount do begin
    gridfolder.Cells[0,i] := '';
    gridfolder.Cells[1,i] := '';
    gridfolder.Cells[2,i] := '';

```

```

gridfolder.Cells[3,i] := "";
gridfolder.Cells[4,i] := "";
end;
memomail.Lines.Clear;
gridfolder.RowCount := 2;   idxgrid:=1;
end;

```

```

procedure Tfrmemail.btnnewClick(Sender: TObject);
begin
  frmnew.show;
end;

```

```

procedure Tfrmemail.imgmaximizeClick(Sender: TObject);
begin
  if frmemail.WindowState <> wsMaximized then begin
    frmemail.WindowState :=wsMaximized;
    imgmaximize.Picture.LoadFromFile(ExtractFilePath(application.exename) +
    '\avi\buttons\restore.bmp');
    shpright.Visible :=false;  shpleft.Visible :=false;
    shpbottom.Visible :=false; shptop.Visible :=false;
  end else begin
    frmemail.WindowState :=wsNormal;
    imgmaximize.Picture.LoadFromFile(ExtractFilePath(application.exename) +
    '\avi\buttons\maximize.bmp');
    shpright.Visible :=true;  shpleft.Visible :=true;
    shpbottom.Visible :=true; shptop.Visible :=true;
  end;
end;

```

```

procedure Tfrmemail.imgminimizeClick(Sender: TObject);
begin
  frmemail.WindowState :=wsMinimized;
end;

```

```

procedure Tfrmemail.btnprintClick(Sender: TObject);
var i,j:integer;
begin
  Printer.BeginDoc; j:=50;
  Printer.Canvas.TextOut(100,200,'To : ' + gridfolder.cells[0,gridfolder.row]);
  Printer.Canvas.TextOut(100,200 + j,'Subject : ' + gridfolder.cells[1,gridfolder.row]);
  for i:=0 to memomail.Lines.Count -1 do begin
    printer.Canvas.TextOut(100,200 + (i+3)*j,memomail.lines[i]);
  end;
  Printer.EndDoc;
end;

```

```

procedure Tfrmemail.btnsetClick(Sender: TObject);
begin
  frmoption.show;
end;

```

```

procedure Tfrmemail.btndeleteClick(Sender: TObject);
begin
  bdel:=true;
  if pos('Mail Box',lblcaption.Caption)=1 then begin
    if messagedlg('Are you sure to send this mail into trash?',mtWarning,[mbYes, mbNo], 0)=mrYes then
      begin
        MoveFile(PChar(ExtractFilePath(application.exename)+'email\mailbox\'+
        gridfolder.cells[4,gridfolder.row]),

```



```

        PChar(ExtractFilePath(application.exename)+'email\trash\' + gridfolder.cells[4,gridfolder.row]));
    btnmailbox.Click;
end;
end else if pos('Sent Mail',lblcaption.Caption)=1 then begin
    if messagedlg('Are you sure to send this mail into trash?',mtWarning,[mbYes, mbNo], 0)=mrYes then
    begin
        MoveFile(PChar(ExtractFilePath(application.exename)+'email\sentmail\' +
gridfolder.cells[4,gridfolder.row]),
            PChar(ExtractFilePath(application.exename)+'email\trash\' + gridfolder.cells[4,gridfolder.row]));
        btnsent.Click;
    end;
end else begin
    if messagedlg('Are you sure to delete this mail?',mtWarning,[mbYes, mbNo], 0)=mrYes then begin
        DeleteFile(PChar(ExtractFilePath(application.exename)+'email\trash\' +
gridfolder.cells[4,gridfolder.row]));
        btntrash.Click;
    end;
end;
end;

procedure Tfrmemail.FormCreate(Sender: TObject);
begin
    bdel:=false;
end;

procedure Tfrmemail.gridfolderSelectCell(Sender: TObject; ACol,
    ARow: Integer; var CanSelect: Boolean);
begin
    if gridfolder.cells[4,ARow] <>"" then
        memomail.Lines.LoadFromFile(
            ExtractFilePath(application.exename)+'\email\' + xfolder + '\'+gridfolder.cells[4,ARow]);
end;

procedure Tfrmemail.btnsendClick(Sender: TObject);
var xfile:TextFile; s:string[100];
begin
    // get mails' content
    AssignFile(xfile,(ExtractFilePath(application.exename) + 'EMail\' +
        xfolder + '\'+ gridfolder.Cells[4,gridfolder.row]));
    reset(xfile);
    readln(xfile,s);
    smtp1.PostMessage.ToAddress.Clear;
    smtp1.PostMessage.ToAddress.Add(s);
    readln(xfile,s);
    smtp1.PostMessage.Subject :=s;
    smtp1.PostMessage.Body.SetText(memomail.lines.gettext);
    closefile(xfile);

    AssignFile(xfile,(ExtractFilePath(application.exename) + '\data\mail.txt'));
    reset(xfile);
    readln(xfile,s); smtp1.UserID :=s;
    readln(xfile,s); smtp1.UserID :=smtp1.UserID + ' <' + s + '>';
    smtp1.PostMessage.FromAddress :=smtp1.UserID;
    readln(xfile,s); smtp1.Host :=s;
    smtp1.Connect;
    smtp1.SendMail ;
    smtp1.Disconnect;
end;

```



```

procedure Tfrmemail.smtp1Connect(Sender: TObject);
begin
    memostatus.Lines.Add('Connected...');
end;

procedure Tfrmemail.smtp1ConnectionFailed(Sender: TObject);
begin
    memostatus.Lines.Add('Connection Failed...');
end;

procedure Tfrmemail.smtp1Disconnect(Sender: TObject);
begin
    memostatus.Lines.Add('Disconnected...');
end;

procedure Tfrmemail.smtp1Success(Sender: TObject);
begin
    memostatus.Lines.Add('Mail Sent...');
end;

procedure Tfrmemail.smtp1SendStart(Sender: TObject);
begin
    memostatus.Lines.Add('Sending...');
end;

procedure Tfrmemail.smtp1Failure(Sender: TObject);
begin
    memostatus.Lines.Add('Mail cannot be sent...');
end;

end.

```

3.1.4. Codes of unew.pas :

```

unit unew;

interface

uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
    StdCtrls, ExtCtrls, ImgList, ComCtrls, ToolWin, StdActns, ActnList,
    Psock, NMsmtp;

type
    Tfrmnew = class(TForm)
        Image2: TImage;
        shpcaption: TShape;
        lblcaption: TLabel;
        lblcaption1: TLabel;
        shptop: TShape;
        shpleft: TShape;
        shpbottom: TShape;
        shpright: TShape;
        ActionList1: TActionList;
        Copy1: TEditCopy;
        Cut1: TEditCut;
        Paste1: TEditPaste;
        Panel1: TPanel;
        edtto: TEdit;
    end;

```

```

edtsystem: TEdit;
Label1: TLabel;
lbto: TLabel;
memo1: TMemo;
ToolBar2: TToolBar;
btnsend: TToolButton;
ToolButton2: TToolButton;
ImageList1: TImageList;
ImageList2: TImageList;
btncopy: TToolButton;
btncut: TToolButton;
btnpaste: TToolButton;
ToolButton6: TToolButton;
btnclose: TToolButton;
imgminimize: TImage;
imgmaximize: TImage;
imgclose: TImage;
anisystem: TAnimate;
btnsave: TToolButton;
SaveDialog1: TSaveDialog;
btnopen: TToolButton;
ToolButton3: TToolButton;
OpenDialog1: TOpenDialog;
smtp1: TNMSMTP;
Memo2: TMemo;
memostatus: TMemo;
procedure lbcaptionMouseDown(Sender: TObject;
  Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
procedure lbcaptionMouseMove(Sender: TObject;
  Shift: TShiftState; X, Y: Integer);
procedure shpleftMouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
procedure shptopMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure shprightMouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
procedure shpbottomMouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
procedure imgmaximizeClick(Sender: TObject);
procedure imgminimizeClick(Sender: TObject);
procedure FormActivate(Sender: TObject);
procedure FormDeactivate(Sender: TObject);
procedure imgcloseClick(Sender: TObject);
procedure btnsaveClick(Sender: TObject);
procedure btnopenClick(Sender: TObject);
procedure edttoEnter(Sender: TObject);
procedure btnsendClick(Sender: TObject);
procedure smtp1Connect(Sender: TObject);
procedure smtp1Success(Sender: TObject);
procedure smtp1Failure(Sender: TObject);
procedure smtp1Disconnect(Sender: TObject);
procedure memo1KeyPress(Sender: TObject; var Key: Char);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  frmnew: Tfrmnew;

```

implementation

```
{ $R *.DFM }
var oldx,oldy:integer;

// Move Window
procedure Tfrmnew.lblcaptionMouseDown(Sender: TObject;
  Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
  oldx:=x;  oldy:=y;
end;

procedure Tfrmnew.lblcaptionMouseMove(Sender: TObject;
  Shift: TShiftState; X, Y: Integer);
begin
  if (shift = [ssLeft])and(frmnew.WindowState <> wsMaximized) then begin
    frmnew.Left := frmnew.left + (x-oldx) ;  frmnew.Top :=frmnew.Top + (y-oldy);
  end;
end;

// Resize Window - Left
procedure Tfrmnew.shpleftMouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
  if (shift=[ssLeft]){and(frmnew.Width > 100)} then begin
    frmnew.width:=frmnew.Width - x ;
    frmnew.left := frmnew.left + x ;
  end;
end;

// Resize Window - Top
procedure Tfrmnew.shptopMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
  if (shift=[ssLeft]){and(frmnew.Height > 100) }then begin
    frmnew.height :=frmnew.height - y ;
    frmnew.top := frmnew.top + y ;
  end;
end;

// Resize Window - Right
procedure Tfrmnew.shprightMouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
  if (shift=[ssLeft]) then begin
    frmnew.width:=x+frmnew.Width ;
  // if (frmnew.Width < 400) then shift:=[ssRight];
  end;
end;

// Resize Window - Bottom
procedure Tfrmnew.shpbottomMouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
  if (shift=[ssLeft]){and(frmnew.Height > 100) }then begin
    frmnew.Height:=y+frmnew.Height ;
  end;
end;
```

```

procedure Tfrmnew.imgmaximizeClick(Sender: TObject);
begin
  if frmnew.WindowState <> wsMaximized then begin
    frmnew.WindowState :=wsMaximized;
    imgmaximize.Picture.LoadFromFile(ExtractFilePath(application.exename) +
'avi\buttons\restore.bmp');
    shpright.Visible :=false; shpleft.Visible :=false;
    shpbottom.Visible :=false; shptop.Visible :=false;
  end else begin
    frmnew.WindowState :=wsNormal;
    imgmaximize.Picture.LoadFromFile(ExtractFilePath(application.exename) +
'avi\buttons\maximize.bmp');
    shpright.Visible :=true; shpleft.Visible :=true;
    shpbottom.Visible :=true; shptop.Visible :=true;
  end;
end;

procedure Tfrmnew.imgminimizeClick(Sender: TObject);
begin
  frmnew.WindowState :=wsMinimized;
end;

procedure Tfrmnew.FormActivate(Sender: TObject);
begin
  anisystem.Active :=true;
  memostatus.Lines.Clear;
  shpcaption.Brush.Color :=$00FF8000;
  lblcaption.Font.Color :=clYellow;
  edtto.Text :="; edtsubject.Text :="; memo1.Lines.Clear;
end;

procedure Tfrmnew.FormDeactivate(Sender: TObject);
begin
  anisystem.Active :=false;
  shpcaption.Brush.Color :=clGray;
  lblcaption.Font.Color :=clSilver;
end;

procedure Tfrmnew.imgcloseClick(Sender: TObject);
begin
  frmnew.Close;
end;

procedure Tfrmnew.btnsaveClick(Sender: TObject);
begin
  SaveDialog1.InitialDir :=ExtractFilePath(application.exename)+'email\mailbox';
  SaveDialog1.Filter :='*.*.txt';
  if SaveDialog1.Execute then begin
    memo1.Lines.Insert(0,edtto.text);
    memo1.Lines.Insert(1,edtsubject.text);
    memo1.Lines.SaveToFile (SaveDialog1.filename + '.txt');
    frmnew.Close;
  end;
end;

```



```

procedure Tfrmnew.btnopenClick(Sender: TObject);
begin
  OpenFileDialog1.InitialDir := ExtractFilePath(application.exename)+'email\mailbox';
  OpenFileDialog1.Filter := '*.txt';
  if OpenFileDialog1.Execute then begin
    // AssignFile(filemail, OpenFileDialog1.FileName);
    // reset(filemail);
    // readln(filemail,s); edtto.Text :=s;
    // readln(filemail,s); edtsubject.Text :=s;
    // closefile(filemail);
    memo1.Lines.LoadFromFile (OpenDialog1.filename);
    edtto.Text :=memo1.Lines[0];
    edtsubject.Text :=memo1.Lines[1];
    memo1.Lines.Delete(0); memo1.Lines.Delete(1);
  end;
end;

procedure Tfrmnew.edttoEnter(Sender: TObject);
begin
  if (edtto.Text <>")and(memo1.Text <> ") then
    btnsend.Enabled :=true
  else
    btnsend.Enabled :=false;
end;

var filex:TextFile;

procedure Tfrmnew.btnsendClick(Sender: TObject);
var i:integer; s:string[100];
begin
  AssignFile(filex,ExtractFilePath(application.exename)+'\data\mail.txt');
  reset(filex);
  readln(filex,s); smtp1.UserID :=s;
  readln(filex,s); smtp1.UserID := smtp1.UserID + ' <' + s + '>';
  readln(filex,s); smtp1.Host :=s;
  memo2.Lines.LoadFromFile(ExtractFilePath(application.exename)+'\data\sign.txt');
  memo1.Lines.Add(memo2.text);
  smtp1.Connect;
  // add information to mail
  smtp1.PostMessage.FromAddress :=memo2.Lines[0] + ' <' + memo2.Lines[1] + '>';
  smtp1.PostMessage.ToAddress.Clear;
  smtp1.PostMessage.ToAddress.Add(edtto.text);
  smtp1.PostMessage.Subject :=edtsubject.Text;
  smtp1.PostMessage.Body.SetText(memo1.lines.gettext);
  smtp1.SendMail ;
  smtp1.Disconnect;
end;

procedure Tfrmnew.smtp1Connect(Sender: TObject);
begin
  memostatus.Lines.Add('Connected to host...');
end;

procedure Tfrmnew.smtp1Success(Sender: TObject);
begin
  memostatus.Lines.Add('Mail Sent...');
  savedialog1.InitialDir := ExtractFilePath(application.exename) + '\email\sentmail';
  savedialog1.Filter := '*.txt';
  if savedialog1.Execute then begin

```

```

memo1.Lines.Insert(0,edtto.text);
memo1.Lines.Insert(1,edtsubject.text);
memo1.Lines.SaveToFile(savedialog1.Files.GetNamePath +
    savedialog1.filename + '.txt');

end;
end;

procedure Tfrmnew.smtp1Failure(Sender: TObject);
begin
    memostatus.Lines.Add('Cannot Sent...');
end;

procedure Tfrmnew.smtp1Disconnect(Sender: TObject);
begin
    memostatus.Lines.Add('Disconnected from host...');
end;

procedure Tfrmnew.memo1KeyPress(Sender: TObject; var Key: Char);
begin
    if memo1.Text <> '' then btnsend.Enabled :=true else btnsend.Enabled :=false;
end;

end.

```

3.1.5. Codes of usource.pas :

```

unit usource;

interface

uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
    ExtCtrls, StdCtrls, ComCtrls, Menus, StdActns, ActnList;

type
    Tfrmsource = class(TForm)
        lblcaption1: TLabel;
        lblcaption: TLabel;
        shpcaption: TShape;
        imgclient: TImage;
        imgclose: TImage;
        anisystem: TAnimate;
        Panel1: TPanel;
        imgstop: TImage;
        Memo1: TMemo;
        ActionList1: TActionList;
        PopupMenu1: TPopupMenu;
        Copy1: TEditCopy;
        Cut1: TEditCut;
        Paste1: TEditPaste;
        Copy2: TMenuItem;
        Cut2: TMenuItem;
        Paste2: TMenuItem;
        procedure imgstopMouseMove(Sender: TObject; Shift: TShiftState; X,
            Y: Integer);
        procedure Panel1MouseMove(Sender: TObject; Shift: TShiftState; X,
            Y: Integer);
        procedure lblcaptionMouseDown(Sender: TObject; Button: TMouseButton;
            Shift: TShiftState; X, Y: Integer);
        procedure lblcaptionMouseMove(Sender: TObject; Shift: TShiftState; X,

```

```

    Y: Integer);
    procedure FormActivate(Sender: TObject);
    procedure imgstopClick(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    frmsource: Tfrmsource;

implementation

uses umain;

{$R *.DFM}

procedure Tfrmsource.imgstopMouseMove(Sender: TObject; Shift: TShiftState;
    X, Y: Integer);
begin
    imgstop.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\stop1.bmp');
end;

procedure Tfrmsource.Panel1MouseMove(Sender: TObject; Shift: TShiftState;
    X, Y: Integer);
begin
    imgstop.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\stop.bmp');
end;

var oldx,oldy:integer;

procedure Tfrmsource.lblcaptionMouseDown(Sender: TObject;
    Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
    oldx:=x;  oldy:=y;
end;

procedure Tfrmsource.lblcaptionMouseMove(Sender: TObject;
    Shift: TShiftState; X, Y: Integer);
begin
    if (shift = [ssLeft])and(frmsource.WindowState <> wsMaximized) then begin
        frmsource.Left := frmsource.left + (x-oldx) ;  frmsource.Top :=frmsource.Top + (y-oldy);
    end;
end;

procedure Tfrmsource.FormActivate(Sender: TObject);
begin
    memo1.Lines.Add(AdjustLineBreaks(frmmain.HTML1.SourceText));
end;

procedure Tfrmsource.imgstopClick(Sender: TObject);
begin
    frmsource.Close;
end;

end.

```

3.1.6 Codes of usearch.pas :

```
unit usearch;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ExtCtrls, ComCtrls, StdCtrls;

type
  Tfrmsearch = class(TForm)
    shpcaption: TShape;
    lblcaption: TLabel;
    lblcaption1: TLabel;
    anisystem: TAnimate;
    imgclient: TImage;
    imgclose: TImage;
    Panel1: TPanel;
    Label1: TLabel;
    imgsearch: TImage;
    imgstop: TImage;
    cmbsearch: TComboBox;
    procedure imgsearchMouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure imgstopMouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure imgstopClick(Sender: TObject);
    procedure imgsearchClick(Sender: TObject);
    procedure Panel1MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure FormActivate(Sender: TObject);
    procedure FormDeactivate(Sender: TObject);
    procedure lblcaptionMouseDown(Sender: TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure lblcaptionMouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure cmbsearchKeyPress(Sender: TObject; var Key: Char);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmsearch: Tfrmsearch;

implementation

uses umain;

{$R *.DFM}

procedure Tfrmsearch.imgsearchMouseMove(Sender: TObject;
  Shift: TShiftState; X, Y: Integer);
begin
  imgstop.Picture.LoadFromFile(ExtractFilePath(application.exename) + 'avi\buttons\stop.bmp');
  imgsearch.Picture.LoadFromFile(ExtractFilePath(application.exename) + 'avi\buttons\search1.bmp');
```



```

end;

procedure Tfrmsearch.imgstopMouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
  imgstop.Picture.LoadFromFile(ExtractFilePath(application.exename) + '\avi\buttons\stop1.bmp');
  imgsearch.Picture.LoadFromFile(ExtractFilePath(application.exename) + '\avi\buttons\search.bmp');
end;

procedure Tfrmsearch.imgstopClick(Sender: TObject);
begin
  frmsearch.Close;
end;

procedure Tfrmsearch.imgsearchClick(Sender: TObject);
var s:string;
begin
  if cmbsearch.Items.IndexOf(cmbsearch.text)=-1 then
    cmbsearch.Items.Add(cmbsearch.text);
    { Convert spaces to zeroes }
    s:=cmbsearch.Text;
    while Pos(' ', s) > 0 do
      s[Pos(' ', s)] := '+';

  frmmain.cmbaddress.Text := 'http://search.yahoo.com/bin/search?p=' +
    s + '&n=100&h=s&b=1';
  frmmain.FindAddress;  frmsearch.Close;
end;

procedure Tfrmsearch.Panel1MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
  imgstop.Picture.LoadFromFile(ExtractFilePath(application.exename) + '\avi\buttons\stop.bmp');
  imgsearch.Picture.LoadFromFile(ExtractFilePath(application.exename) + '\avi\buttons\search.bmp');
end;

procedure Tfrmsearch.FormActivate(Sender: TObject);
begin
  anisystem.Active :=true;
end;

procedure Tfrmsearch.FormDeactivate(Sender: TObject);
begin
  anisystem.Active :=true;
end;

var oldy,oldx:integer;

procedure Tfrmsearch.IblcaptionMouseDown(Sender: TObject;
  Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
  oldx:=x;  oldy:=y;
end;

procedure Tfrmsearch.IblcaptionMouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
  if (shift = [ssLeft])and(frmsearch.WindowState <> wsMaximized) then begin
    frmsearch.Left := frmsearch.left + (x-oldx) ;  frmsearch.Top :=frmsearch.Top + (y-oldy);
  end;

```

end;

```
procedure Tfrmsearch.cmbsearchKeyPress(Sender: TObject; var Key: Char);
begin
  if key=#13 then imgsearch.OnClick(self);
end;
```

end.

3.1.7 Codes of uoption.pas :

unit uoption;

interface

uses

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
StdCtrls, ComCtrls, ExtCtrls, umain;

type

```
Tfrmoption = class(TForm)
  Image1: TImage;
  lblcaption: TLabel;
  shpcaption: TShape;
  lblcaption1: TLabel;
  imgclose: TImage;
  Panel1: TPanel;
  Image2: TImage;
  Label1: TLabel;
  edthome: TEdit;
  imgok: TImage;
  imgcancel: TImage;
  imghistory: TImage;
  lblhistory: TLabel;
  GroupBox1: TGroupBox;
  anisystem: TAnimate;
  Label3: TLabel;
  Label5: TLabel;
  Label6: TLabel;
  Label10: TLabel;
  edtname: TEdit;
  edtaddress: TEdit;
  edtSMTP: TEdit;
  GroupBox2: TGroupBox;
  Label2: TLabel;
  cmbfont: TComboBox;
  Label4: TLabel;
  edtsize: TEdit;
  UpDown1: TUpDown;
  memosign: TMemo;
  procedure FormActivate(Sender: TObject);
  procedure imgcloseClick(Sender: TObject);
  procedure lblcaption1MouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
  procedure lblcaption1MouseMove(Sender: TObject; Shift: TShiftState; X,
    Y: Integer);
  procedure imgokMouseMove(Sender: TObject; Shift: TShiftState; X,
    Y: Integer);
  procedure imgcancelMouseMove(Sender: TObject; Shift: TShiftState; X,
```

```

    Y: Integer);
procedure Panel1MouseMove(Sender: TObject; Shift: TShiftState; X,
    Y: Integer);
procedure imgokClick(Sender: TObject);
procedure imgcancelClick(Sender: TObject);
procedure imghistoryMouseMove(Sender: TObject; Shift: TShiftState; X,
    Y: Integer);
procedure imghistoryClick(Sender: TObject);
procedure FormDeactivate(Sender: TObject);
procedure UpDown1Click(Sender: TObject; Button: TUDBtnType);
private
    { Private declarations }
public
    { Public declarations }
end;

{type optionrec=record
    homepage:string[100];
    ratio:string[3];
    day:string[3];
    fontname:string[20];
    fontsize:string[2];
end;}

var
    frmoption: Tfrmoption;

implementation

{$R *.DFM}

var filemail,filehome:TextFile; oldx,oldy:integer;
    s:string;

procedure Tfrmoption.FormActivate(Sender: TObject);
var i:integer;
begin
    shpcaption.Brush.Color :=$00FF8000;
    lblcaption.Font.Color :=clYellow;
    anisystem.Active :=true;
    lblhistory.Left :=549; lblhistory.Caption :='Clean History';
    // load options from file options.txt
    AssignFile(filehome,ExtractFilePath(application.exename) + 'data\options.txt');
    if not FileExists(ExtractFilePath(application.exename) + 'data\options.txt') then begin
        Rewrite(filehome);
        FileMode:=1;
        writeln(filehome,'www.yahoo.com');
        writeln(filehome,'Arial');
        writeln(filehome,'10');
        closefile(filehome);
    end;
    // load available font into fonts combobox
    for i:=0 to screen.fonts.Count - 1 do
        cmbfont.Items.Add(screen.fonts.Strings[i]);
    AssignFile(filehome,ExtractFilePath(application.exename) + 'data\options.txt');
    {$I-} reset(filehome); {$I+}
    if ioresult<>0 then begin
        showmessage(ExtractFilePath(application.exename) + 'data\options.txt can not be opened. ');
        exit;
    end;
end;

```

```

readln(filehome,s); edthome.Text :=s;
readln(filehome,s); cmbfont.ItemIndex :=cmbfont.items.IndexOf(s);
readln(filehome,s); edtsize.Text :=s;
closefile(filehome);
// load mail settings from file mail.txt
AssignFile(filemail,ExtractFilePath(application.exename) + 'data\mail.txt');
if not FileExists(ExtractFilePath(application.exename) + 'data\mail.txt') then begin
  Rewrite(filemail);
  Filemode:=1;
  writeln(filemail,'Serhat TUNCA');
  writeln(filemail,'serhattunca@hotmail.com');
  writeln(filemail,'mail.hotmail.com');
  closefile(filemail);
end;
{$I-} reset(filemail); {$I+}
if ioresult<>0 then begin
  showmessage(ExtractFilePath(application.exename) + 'data\mail.txt can not be opened.');
```

```

  exit;
end;
readln(filemail,s); edtname.Text :=s;
readln(filemail,s); edtaddress.Text :=s;
readln(filemail,s); edtSMTP.Text :=s;
closefile(filemail);
if not FileExists(ExtractFilePath(application.exename) + '\data\sign.txt') then
  FileCreate(ExtractFilePath(application.exename) + '\data\sign.txt');
memosign.Lines.LoadFromFile(ExtractFilePath(application.exename) + '\data\sign.txt');
end;

procedure Tfrmoption.imgcloseClick(Sender: TObject);
begin
  frmoption.Close;
end;

procedure Tfrmoption.lblcaption1MouseDown(Sender: TObject;
  Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
  oldx:=x; oldy:=y;
end;

procedure Tfrmoption.lblcaption1MouseMove(Sender: TObject;
  Shift: TShiftState; X, Y: Integer);
begin
  if (shift = [ssLeft]) then begin
    frmoption.Left := frmoption.left + (x-oldx) ; frmoption.Top :=frmoption.Top + (y-oldy);
  end;
end;

procedure Tfrmoption.imgokMouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
  imgok.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\ok1.bmp');
  imgcancel.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\stop.bmp');
  imghistory.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\button1.bmp');
end;

```



```

procedure Tfrmoption.imgcancelMouseMove(Sender: TObject;
  Shift: TShiftState; X, Y: Integer);
begin
  imgok.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\ok.bmp');
  imgcancel.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\stop1.bmp');
  imghistory.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\button1.bmp');
end;

procedure Tfrmoption.Panel1MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
  imgok.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\ok.bmp');
  imgcancel.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\stop.bmp');
  imghistory.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\button1.bmp');
end;

procedure Tfrmoption.imgokClick(Sender: TObject);
begin
  // save options into options.txt
  AssignFile(filehome, ExtractFilePath(application.exename) + '\data\options.txt');
  rewrite(filehome); filemode:=1;
  s:=edthome.text;          writeln(filehome,s);
  s:=cmbfont.Items[cmbfont.itemindex]; writeln(filehome,s);
  s:=edtsize.Text;          writeln(filehome,s);
  flush(filehome);
  closefile(filehome);
  // update the HTMLs font name and size
  frmmain.HTML1.Font.Name :=cmbfont.Items[cmbfont.itemindex];
  frmmain.HTML1.Font.Size :=strtoint(edtsize.text);
  // save mail settings into mail.txt
  AssignFile(filemail, ExtractFilePath(application.exename) + '\data\mail.txt');
  rewrite(filemail); filemode:=1;
  s:=edtname.text;          writeln(filemail,s);
  s:=edtaddress.text;        writeln(filemail,s);
  s:=edtSMTP.text;          writeln(filemail,s);
  flush(filemail);
  closefile(filemail);
  frmoption.close;
  if not FileExists((application.exename) + '\data\sign.txt') then
    FileCreate((application.exename) + '\data\sign.txt');
  memosign.Lines.SaveToFile((application.exename) + '\data\sign.txt');
end;

procedure Tfrmoption.imgcancelClick(Sender: TObject);
begin
  frmoption.Close;
end;

procedure Tfrmoption.imghistoryMouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
  imghistory.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\button2.bmp');
end;

procedure Tfrmoption.imghistoryClick(Sender: TObject);
begin
  frmmain.cmbaddress.Items.Clear; frmmain.cmbaddress.Items.Add(' ');
  lblhistory.Left :=555; lblhistory.Caption :='CLEANED';
  imghistory.Picture.LoadFromFile(ExtractFilePath(application.exename)+'avi\buttons\button1.bmp');
end;

```

```

end;

procedure Tfrmoption.FormDeactivate(Sender: TObject);
begin
  anisystem.Active := false; frmoption.Close;
end;

// controlling the up n down
procedure Tfrmoption.UpDown1Click(Sender: TObject; Button: TUDBtnType);
begin
  if Button=btNext then begin
    edtsize.text := inttostr( strtoint(edtsize.text) + 1 );
  end else begin
    edtsize.text := inttostr( strtoint(edtsize.text) - 1 );
  end;
end;

end.

```

3.1.8 Codes of uabout.pas :

```

unit uabout;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ExtCtrls, ComCtrls, StdCtrls;

type
  Tfrmabout = class(TForm)
    imgclient: TImage;
    shpleft: TShape;
    shptop: TShape;
    shpright: TShape;
    shpbottom: TShape;
    lblcaption: TLabel;
    shpcaption: TShape;
    Label1: TLabel;
    anisystem: TAnimate;
    imgclose: TImage;
    Panel1: TPanel;
    aniglobe: TAnimate;
    Memo1: TMemo;
    Image1: TImage;
    procedure FormActivate(Sender: TObject);
    procedure Panel1MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure Image1MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure FormDeactivate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var

```

```

frmabout: Tfrmabout;

implementation

{$R *.DFM}

procedure Tfrmabout.FormActivate(Sender: TObject);
begin
  image1.Picture.LoadFromFile(ExtractFilePath(application.exename) +
    '\avi\buttons\ok.bmp');
  aniglobe.Active := true;
end;

procedure Tfrmabout.Panel1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
  image1.Picture.LoadFromFile(ExtractFilePath(application.exename) +
    '\avi\buttons\ok.bmp');
end;

procedure Tfrmabout.Image1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
  image1.Picture.LoadFromFile(ExtractFilePath(application.exename) +
    '\avi\buttons\ok1.bmp');
end;

procedure Tfrmabout.FormDeactivate(Sender: TObject);
begin
  frmabout.Close;
end;

end.

```

3.2 Installation Program (Setup.pas) :

3.2.1 Codes of Setup.dpr :

```

program Setup;

uses
  Forms,
  umain in 'umain.pas' {frmsetup},
  ucancel in 'ucancel.pas' {frmcancel};

{$R *.RES}

begin
  Application.Initialize;
  Application.CreateForm(Tfrmsetup, frmsetup);
  Application.CreateForm(Tfrmcancel, frmcancel);
  Application.Run;
end.

```

3.2.2 Codes of umain.pas :

```
unit umain;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ExtCtrls, StdCtrls, ComCtrls, FileCtrl;

type
  Tfrmsetup = class(TForm)
    label1: TLabel;
    btnback: TButton;
    btnnext: TButton;
    btncancel: TButton;
    Panel1: TPanel;
    Memo1: TMemo;
    Animate1: TAnimate;
    Animate2: TAnimate;
    lblname: TLabel;
    labelcourse: TLabel;
    OpenFileDialog1: TOpenDialog;
    Memo2: TMemo;
    Panel2: TPanel;
    lblfolder: TLabel;
    edtdir: TEdit;
    lblwindows: TLabel;
    edtwindir: TEdit;
    Memo3: TMemo;
    label2: TLabel;
    label3: TLabel;
    Panel3: TPanel;
    chkdesktop: TCheckBox;
    chkquick: TCheckBox;
    Label4: TLabel;
    Memo4: TMemo;
    Label5: TLabel;
    Panel4: TPanel;
    Label6: TLabel;
    ProgressBar1: TProgressBar;
    chkstart: TCheckBox;
    chkread: TCheckBox;
    procedure btncancelClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure btnnextClick(Sender: TObject);
    procedure hideX(x:integer);
    procedure showX(x:integer);
    procedure btnbackClick(Sender: TObject);
    procedure install;
    procedure chkreadClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmsetup: Tfrmsetup;
```


implementation

uses ucancel, Unit1;

{ \$R *.DFM }

var idx:integer;

```
procedure Tfrmsetup.btncancelClick(Sender: TObject);
begin
  if btncancel.caption='&Finish' then application.terminate
  else frmcancel.show;
end;
```

```
procedure Tfrmsetup.FormCreate(Sender: TObject);
begin
  idx:=1;
end;
```

```
procedure Tfrmsetup.btnnextClick(Sender: TObject);
begin
  if idx>4 then exit;
  hideX(idx); idx:=idx+1; showX(idx);
  if idx=5 then install;
end;
```

```
procedure Tfrmsetup.btnbackClick(Sender: TObject);
begin
  if idx<2 then exit;
  hideX(idx); idx:=idx-1; showX(idx);
end;
```

```
procedure Tfrmsetup.hideX(x:integer);
begin
  case x of
    1 : begin
        label1.Visible :=false; memo1.Visible :=false;
        end;
    2 : begin
        label2.Visible :=false;
        memo2.Visible :=false;    panel2.Visible :=false;
        end;
    3 : begin
        label3.Visible :=false;
        memo3.Visible :=false;    panel3.Visible :=false;
        end;
    4 : begin
        label4.Visible :=false;
        memo4.Visible :=false;
        btnnext.Caption :='&Install';
        end;
  end;
end;
```

```
procedure Tfrmsetup.showX(x:integer);
begin
  case x of
    1 : begin
        label1.Visible :=true; memo1.Visible :=true;
```

```

    btnback.enabled :=false;
end;
2 : begin
    label2.Visible :=true;
    memo2.Visible :=true;    panel2.Visible :=true;
    btnback.enabled :=true;
end;
3 : begin
    label3.Visible :=true;
    memo3.Visible :=true;    panel3.Visible :=true;
    btnnext.Caption := '&Next >';
end;
4 : begin
    label4.Visible :=true;
    memo4.Visible :=true;
    btnnext.Caption := '&Install';
end;
end;
end;

procedure Tfrmsetup.install;
begin
    label6.Visible :=true;    panel4.Visible :=true;
    progressbar1.Position :=0;
    label6.Caption := 'Creating Directories...';
    // create directory structure
    if not DirectoryExists(edtdir.Text) then mkdir(edtdir.Text);
    if not DirectoryExists(edtdir.Text+'avi') then mkdir(edtdir.Text+'avi');
    if not DirectoryExists(edtdir.Text+'avi\buttons') then mkdir(edtdir.Text+'avi\buttons');
    if not DirectoryExists(edtdir.Text+'data') then mkdir(edtdir.Text+'data');
    if not DirectoryExists(edtdir.Text+'email') then mkdir(edtdir.Text+'email');
    if not DirectoryExists(edtdir.Text+'email\mailbox') then mkdir(edtdir.Text+'email\mailbox');
    if not DirectoryExists(edtdir.Text+'email\sentmail') then mkdir(edtdir.Text+'email\sentmail');
    if not DirectoryExists(edtdir.Text+'email\trash') then mkdir(edtdir.Text+'email\trash');
    progressbar1.Position :=progressbar1.Position + 10;
    // structure created...

    // start to copy files into destination folder
    // first the main program
    label6.Caption := 'Copying Main Program...';
    CopyFile(PChar(ExtractFilePath(application.exename) + 'webbrowser.exe'),
        PChar(edtdir.Text+'webbrowser.exe'),false);
    progressbar1.Position :=progressbar1.Position + 10;
    // the bitmaps into avi
    label6.Caption := 'Copying Bitmaps...';
    CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\abort.bmp'),
        PChar(edtdir.Text+'avi\abort.bmp'),false);
    progressbar1.Position :=progressbar1.Position + 1;
    CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\abort1.bmp'),
        PChar(edtdir.Text+'avi\abort1.bmp'),false);
    progressbar1.Position :=progressbar1.Position + 1;
    CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\copy.bmp'),
        PChar(edtdir.Text+'avi\copy.bmp'),false);
    progressbar1.Position :=progressbar1.Position + 1;
    CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\copy1.bmp'),
        PChar(edtdir.Text+'avi\copy1.bmp'),false);
    progressbar1.Position :=progressbar1.Position + 1;
    CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\cut.bmp'),
        PChar(edtdir.Text+'avi\cut.bmp'),false);
    progressbar1.Position :=progressbar1.Position + 1;

```

```

CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\cut1.bmp'),
PChar(edtdir.Text+'avi\cut1.bmp'),false);
progressbar1.Position :=progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\erase.bmp'),
PChar(edtdir.Text+'avi\erase.bmp'),false);
progressbar1.Position :=progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\erase1.bmp'),
PChar(edtdir.Text+'avi\erase1.bmp'),false);
progressbar1.Position :=progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\mailtext.bmp'),
PChar(edtdir.Text+'avi\mailtext.bmp'),false);
progressbar1.Position :=progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\mailtext1.bmp'),
PChar(edtdir.Text+'avi\mailtext1.bmp'),false);
progressbar1.Position :=progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\paste.bmp'),
PChar(edtdir.Text+'avi\paste.bmp'),false);
progressbar1.Position :=progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\paste1.bmp'),
PChar(edtdir.Text+'avi\paste1.bmp'),false);
progressbar1.Position :=progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\print.bmp'),
PChar(edtdir.Text+'avi\print.bmp'),false);
progressbar1.Position :=progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\print1.bmp'),
PChar(edtdir.Text+'avi\print1.bmp'),false);
progressbar1.Position :=progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\send.bmp'),
PChar(edtdir.Text+'avi\send.bmp'),false);
progressbar1.Position :=progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\send1.bmp'),
PChar(edtdir.Text+'avi\send1.bmp'),false);
progressbar1.Position :=progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\abort.bmp'),
PChar(edtdir.Text+'avi\buttons\abort.bmp'),false);
progressbar1.Position :=progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\base.bmp'),
PChar(edtdir.Text+'avi\buttons\base.bmp'),false);
progressbar1.Position :=progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\button2.bmp'),
PChar(edtdir.Text+'avi\buttons\button2.bmp'),false);
progressbar1.Position :=progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\button1.bmp'),
PChar(edtdir.Text+'avi\buttons\button1.bmp'),false);
progressbar1.Position :=progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\folder.bmp'),
PChar(edtdir.Text+'avi\buttons\folder.bmp'),false);
progressbar1.Position :=progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\maximize.bmp'),
PChar(edtdir.Text+'avi\buttons\maximize.bmp'),false);
progressbar1.Position :=progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\ok.bmp'),
PChar(edtdir.Text+'avi\buttons\ok.bmp'),false);
progressbar1.Position :=progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\print.bmp'),
PChar(edtdir.Text+'avi\buttons\print.bmp'),false);
progressbar1.Position :=progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\print1.bmp'),
PChar(edtdir.Text+'avi\buttons\print1.bmp'),false);
progressbar1.Position :=progressbar1.Position + 1;

```



```
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\search.bmp'),
PChar(edtdir.Text + '\avi\buttons\search.bmp'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\search1.bmp'),
PChar(edtdir.Text + '\avi\buttons\search1.bmp'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\trash.bmp'),
PChar(edtdir.Text + '\avi\buttons\trash.bmp'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\trashful.bmp'),
PChar(edtdir.Text + '\avi\buttons\trashful.bmp'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\refresh.bmp'),
PChar(edtdir.Text + '\avi\buttons\refresh.bmp'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\refresh1.bmp'),
PChar(edtdir.Text + '\avi\buttons\refresh1.bmp'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\ok1.bmp'),
PChar(edtdir.Text + '\avi\buttons\ok1.bmp'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\minimize.bmp'),
PChar(edtdir.Text + '\avi\buttons\minimize.bmp'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\tools.bmp'),
PChar(edtdir.Text + '\avi\buttons\tools.bmp'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\tools1.bmp'),
PChar(edtdir.Text + '\avi\buttons\tools1.bmp'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\folder1.bmp'),
PChar(edtdir.Text + '\avi\buttons\folder1.bmp'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\back.bmp'),
PChar(edtdir.Text + '\avi\buttons\back.bmp'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\back1.bmp'),
PChar(edtdir.Text + '\avi\buttons\back1.bmp'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\back2.bmp'),
PChar(edtdir.Text + '\avi\buttons\back2.bmp'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\big.bmp'),
PChar(edtdir.Text + '\avi\buttons\big.bmp'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\big1.bmp'),
PChar(edtdir.Text + '\avi\buttons\big1.bmp'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\channel.bmp'),
PChar(edtdir.Text + '\avi\buttons\channel.bmp'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\channel1.bmp'),
PChar(edtdir.Text + '\avi\buttons\channel1.bmp'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\next.bmp'),
PChar(edtdir.Text + '\avi\buttons\next.bmp'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\next1.bmp'),
PChar(edtdir.Text + '\avi\buttons\next1.bmp'), false);
progressbar1.Position := progressbar1.Position + 1;
```



```

CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\next2.bmp'),
    PChar(edtdir.Text + '\avi\buttons\next2.bmp'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\option.bmp'),
    PChar(edtdir.Text + '\avi\buttons\option.bmp'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\option1.bmp'),
    PChar(edtdir.Text + '\avi\buttons\option1.bmp'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\small.bmp'),
    PChar(edtdir.Text + '\avi\buttons\small.bmp'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\small1.bmp'),
    PChar(edtdir.Text + '\avi\buttons\small1.bmp'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\winclose.bmp'),
    PChar(edtdir.Text + '\avi\buttons\winclose.bmp'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\floppy.bmp'),
    PChar(edtdir.Text + '\avi\buttons\floppy.bmp'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\floppy1.bmp'),
    PChar(edtdir.Text + '\avi\buttons\floppy1.bmp'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\fileopen1.bmp'),
    PChar(edtdir.Text + '\avi\buttons\fileopen1.bmp'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\fileopen.bmp'),
    PChar(edtdir.Text + '\avi\buttons\fileopen.bmp'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\stop.bmp'),
    PChar(edtdir.Text + '\avi\buttons\stop.bmp'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\stop1.bmp'),
    PChar(edtdir.Text + '\avi\buttons\stop1.bmp'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\home.bmp'),
    PChar(edtdir.Text + '\avi\buttons\home.bmp'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\home1.bmp'),
    PChar(edtdir.Text + '\avi\buttons\home1.bmp'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\restore.bmp'),
    PChar(edtdir.Text + '\avi\buttons\restore.bmp'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\erase.bmp'),
    PChar(edtdir.Text + '\avi\buttons\erase.bmp'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\close.bmp'),
    PChar(edtdir.Text + '\avi\buttons\close.bmp'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\earth.ico'),
    PChar(edtdir.Text + '\avi\buttons\earth.ico'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\email.bmp'),
    PChar(edtdir.Text + '\avi\buttons\email.bmp'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\buttons\email1.bmp'),
    PChar(edtdir.Text + '\avi\buttons\email1.bmp'), false);

```

```

// the avis into avi
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\globe7.avi'),
    PChar(edtdir.Text + '\avi\globe7.avi'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\web.avi'),
    PChar(edtdir.Text + '\avi\web.avi'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'avi\text.avi'),
    PChar(edtdir.Text + '\avi\text.avi'), false);
progressbar1.Position := progressbar1.Position + 1;

// copy data files
CopyFile(PChar(ExtractFilePath(application.exename) + 'data\history.txt'),
    PChar(edtdir.Text + '\data\history.txt'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'data\mail.txt'),
    PChar(edtdir.Text + '\data\mail.txt'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'data\option.txt'),
    PChar(edtdir.Text + '\data\option.txt'), false);
progressbar1.Position := progressbar1.Position + 1;
CopyFile(PChar(ExtractFilePath(application.exename) + 'data\sign.txt'),
    PChar(edtdir.Text + '\data\sign.txt'), false);
progressbar1.Position := progressbar1.Position + 1;

progressbar1.Position := 100;

// make shortcut icons
if (chkdesktop.Checked) and (edtdir.Text = 'c:\program files\webbrowser') then begin
    label6.Caption := 'Creating Shortcut on Desktop...';
    CopyFile(PChar(ExtractFilePath(application.exename) + '\webbrowser.Ink'),
        PChar(edtwindir.text + '\Desktop\webbrowser.Ink'), false);
    progressbar1.Position := progressbar1.Position + 1;
end;
if (chkquick.Checked) and (edtdir.Text = 'c:\program files\webbrowser') then begin
    label6.Caption := 'Creating Shortcut on Quick Launch...';
    CopyFile(PChar(ExtractFilePath(application.exename) + '\webbrowser.Ink'),
        PChar(edtwindir.text + '\Application Data\Microsoft\Internet Explorer\Quick
Launch\webbrowser.Ink'), false);
    progressbar1.Position := progressbar1.Position + 10;
end;
if (chkstart.Checked) and (edtdir.Text = 'c:\program files\webbrowser') then begin
    label6.Caption := 'Creating Shortcut on Start Menu...';
    if DirectoryExists(edtwindir.text + '\Start Menu\Programs') then begin
        mkdir(edtwindir.text + '\Start Menu\Programs\WEBrowser');
        CopyFile(PChar(ExtractFilePath(application.exename) + '\webbrowser.Ink'),
            PChar(edtwindir.text + '\Start Menu\Programs\WEBrowser\webbrowser.Ink'), false);
    end
    else if DirectoryExists(edtwindir.text + '\Start Menu\Programlar') then begin
        if not DirectoryExists(edtwindir.text + '\Start Menu\Programlar\WEBrowser') then
            mkdir(edtwindir.text + '\Start Menu\Programlar\WEBrowser');
        CopyFile(PChar(ExtractFilePath(application.exename) + '\webbrowser.Ink'),
            PChar(edtwindir.text + '\Start Menu\Programlar\WEBrowser\webbrowser.Ink'), false);
    end;
    progressbar1.Position := progressbar1.Position + 10;
end;
memo1.Lines.Clear;
memo1.Lines[0] := edtwindir.text;
memo1.Lines.Add(edtdir.text);

```

```
if FileExists('c:\webunins.ini') then DeleteFile('c:\webunins.ini');
memo1.Lines.SaveToFile('c:\webunins.ini');
```

```
label6.Caption := 'Installation completed.';
btnback.Visible := false; btnnext.Visible := false;
btncancel.Caption := '&Finish';
end;
```

```
procedure Tfrmsetup.chkreadClick(Sender: TObject);
begin
  edtdir.ReadOnly := chkread.Checked ;
  if chkread.Checked then edtdir.Color := clSilver
  else edtdir.Color := clWhite;
end;
```

```
end.
```

3.2.3 Codes of ucancel.pas :

```
unit ucancel;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
StdCtrls;
```

```
type
```

```
Tfrmcancel = class(TForm)
  Label3: TLabel;
  Label1: TLabel;
  Label2: TLabel;
  Label4: TLabel;
  Label5: TLabel;
  Label6: TLabel;
  btnresume: TButton;
  btnexit: TButton;
  procedure btnresumeClick(Sender: TObject);
  procedure btnexitClick(Sender: TObject);
private
public
end;
```

```
var
```

```
frmcancel: Tfrmcancel;
```

```
implementation
```

```
{ $R *.DFM }
```

```
procedure Tfrmcancel.btnresumeClick(Sender: TObject);
begin
  frmcancel.close;
end;
```

```
procedure Tfrmcancel.btnexitClick(Sender: TObject);
begin
  frmcancel.close; application.Terminate;
end;
```


end.

3.3 Uninstallation Program (Uninstall.pas) :

3.3.1 Codes of Uninstall.dpr :

```
program uninstall;

uses
  Forms,
  Umain in 'Umain.pas' {frmmain};

{$R *.RES}

begin
  Application.Initialize;
  Application.CreateForm(Tfrmmain, frmmain);
  Application.Run;
end.
```

3.3.2 Codes of Umain.dpr :

```
unit Umain;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, ExtCtrls, FileCtrl, ComCtrls;

type
  Tfrmmain = class(TForm)
    Memo1: TMemo;
    Image1: TImage;
    Panel1: TPanel;
    btnfinish: TButton;
    btncancel: TButton;
    Panel2: TPanel;
    chkdata: TCheckBox;
    chkemail: TCheckBox;
    chkbitmap: TCheckBox;
    chkavi: TCheckBox;
    chkmain: TCheckBox;
    ProgressBar1: TProgressBar;
    Label1: TLabel;
    chkshort: TCheckBox;
    procedure btncancelClick(Sender: TObject);
    procedure btnfinishClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
```



```

frmmain: Tfrmmain;

implementation

{$R *.DFM}

procedure Tfrmmain.btncancelClick(Sender: TObject);
begin
  if btncancel.Caption = '&Close' then frmmain.close;
  if messagedlg('Are you sure you want to cancel the uninstallation?',
    mtWarning,[mbYes,mbNo],0) = mrYes then
    frmmain.Close;
end;

procedure Tfrmmain.btnfinishClick(Sender: TObject);
var xfile:TSearchRec; fileweb:TextFile; swin,s:string;
begin
  panel2.Visible :=true;
  progressbar1.Position :=0;
  if FileExists('c:\webunins.ini') then begin
    AssignFile(fileweb,'c:\webunins.ini'); reset(fileweb);
    readln(fileweb,swin); readln(fileweb,s);
    CloseFile(fileweb);
    // remove data
    if DirectoryExists(s + '\data') then begin
      findfirst(s + '\data\*. *',faAnyFile,xfile); findnext(xfile);
      while (findnext(xfile)=0) do
        deletefile(s + '\data\' + xfile.Name);
      if Removedir(s + '\data') then chkdata.Checked :=true;
    end;
    progressbar1.Position :=progressbar1.Position + 20;

    // remove emails
    if DirectoryExists(s + '\email') then begin
      findfirst(s + '\email\*. *',faAnyFile,xfile); findnext(xfile);
      while (findnext(xfile)=0) do
        deletefile(s + '\email\' + xfile.Name);

      if DirectoryExists(s + '\email\sentmail') then begin
        findfirst(s + '\email\sentmail\*. *',faAnyFile,xfile); findnext(xfile);
        while (findnext(xfile)=0) do
          deletefile(s + '\email\sentmail\' + xfile.Name);
        RemoveDir(s + '\email\sentmail');
      end;
      progressbar1.Position :=progressbar1.Position + 20;

      if DirectoryExists(s + '\email\trash') then begin
        findfirst(s + '\email\trash\*. *',faAnyFile,xfile); findnext(xfile);
        while (findnext(xfile)=0) do
          deletefile(s + '\email\trash\' + xfile.Name);
        Removedir(s + '\email\trash');
      end;
      progressbar1.Position :=progressbar1.Position + 20;

      if DirectoryExists(s + '\email\mailbox') then begin
        findfirst(s + '\email\mailbox\*. *',faAnyFile,xfile); findnext(xfile);
        while (findnext(xfile)=0) do
          deletefile(s + '\email\mailbox\' + xfile.Name);
        Removedir(s + '\email\mailbox');
      end;
    end;
  end;
end;

```

```

if Removedir(s + '\email') then chkemail.Checked :=true;
end;
progressbar1.Position :=progressbar1.Position + 20;

// remove buttons
if DirectoryExists(s + '\avi\buttons') then begin
  findfirst(s + '\avi\buttons\*. *',faAnyFile,xfile); findnext(xfile);
  while (findnext(xfile)=0) do
    deletefile(s + '\avi\buttons\' + xfile.Name);
  if Removedir(s + '\avi\buttons') then chkbitmap.Checked :=true;
end;
progressbar1.Position :=progressbar1.Position + 20;

// remove avis
if DirectoryExists(s + '\avi') then begin
  findfirst(s + '\avi\*. *',faAnyFile,xfile); findnext(xfile);
  while (findnext(xfile)=0) do
    deletefile(s + '\avi\' + xfile.Name);
  if Removedir(s + '\avi') then chkavi.Checked :=true;
end;
progressbar1.Position :=progressbar1.Position + 20;

// remove main program
if DirectoryExists(s) then begin
  findfirst(s + '\*. *',faAnyFile,xfile); findnext(xfile);
  while (findnext(xfile)=0) do
    deletefile(s + '\ ' + xfile.Name);
  if Removedir(s) then chkmain.Checked :=true;
end;
progressbar1.Position :=progressbar1.Position + 20;
end;
if DirectoryExists(swin) then begin
  deletefile(swin + '\Desktop\webbrowser.lnk');
  deletefile(swin + '\Application Data\Microsoft\Internet Explorer\Quick Launch\webbrowser.lnk');
  chkshort.Checked :=true;
end;
Label1.Visible :=true;
btnfinish.Visible :=false; btncancel.Caption :='&Close';
end;

end.

```

4.SCREEN LAYOUTS

- ❑ WEBBrowser Main Window
 - ❑ Style 1 (Default)
 - ❑ Style 2
 - ❑ Style 3
 - ❑ Full Screen

- ❑ WEBBrowser E-Mail Window
 - ❑ New E-Mail Window

- ❑ WEBBrowser Options Window

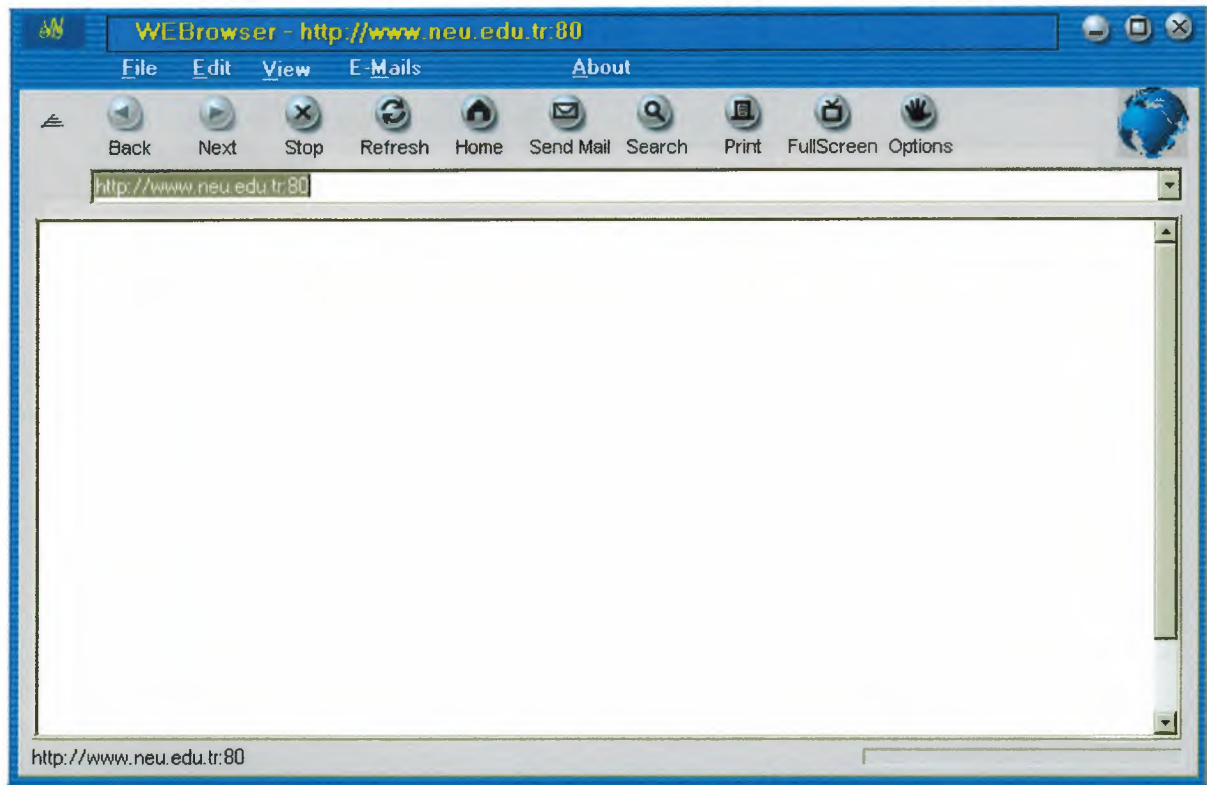
- ❑ WEBBrowser Search Window

- ❑ WEBBrowser About Window

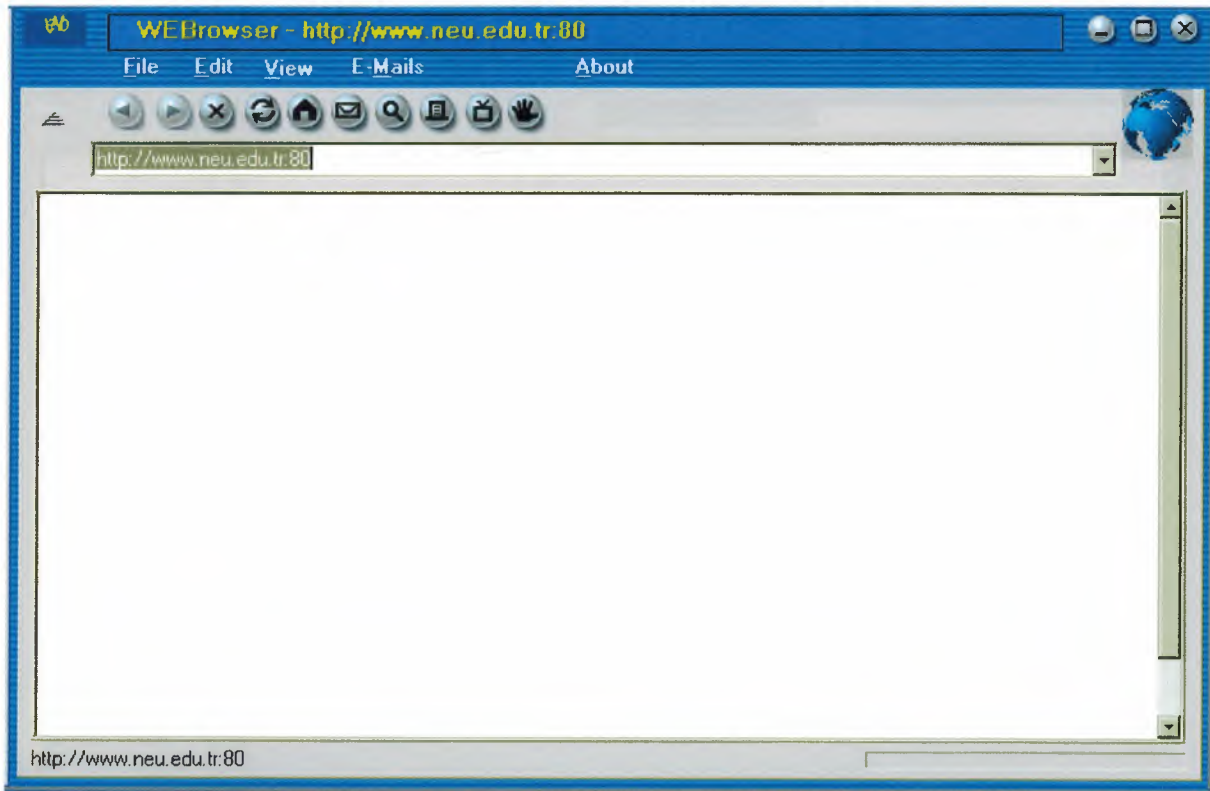
- ❑ WEBBrowser Install Window
 - ❑ Step 1
 - ❑ Step 2
 - ❑ Step 3
 - ❑ Step 4
 - ❑ Step 5

- ❑ WEBBrowser Uninstall Window
 - ❑ Step 1
 - ❑ Step 2

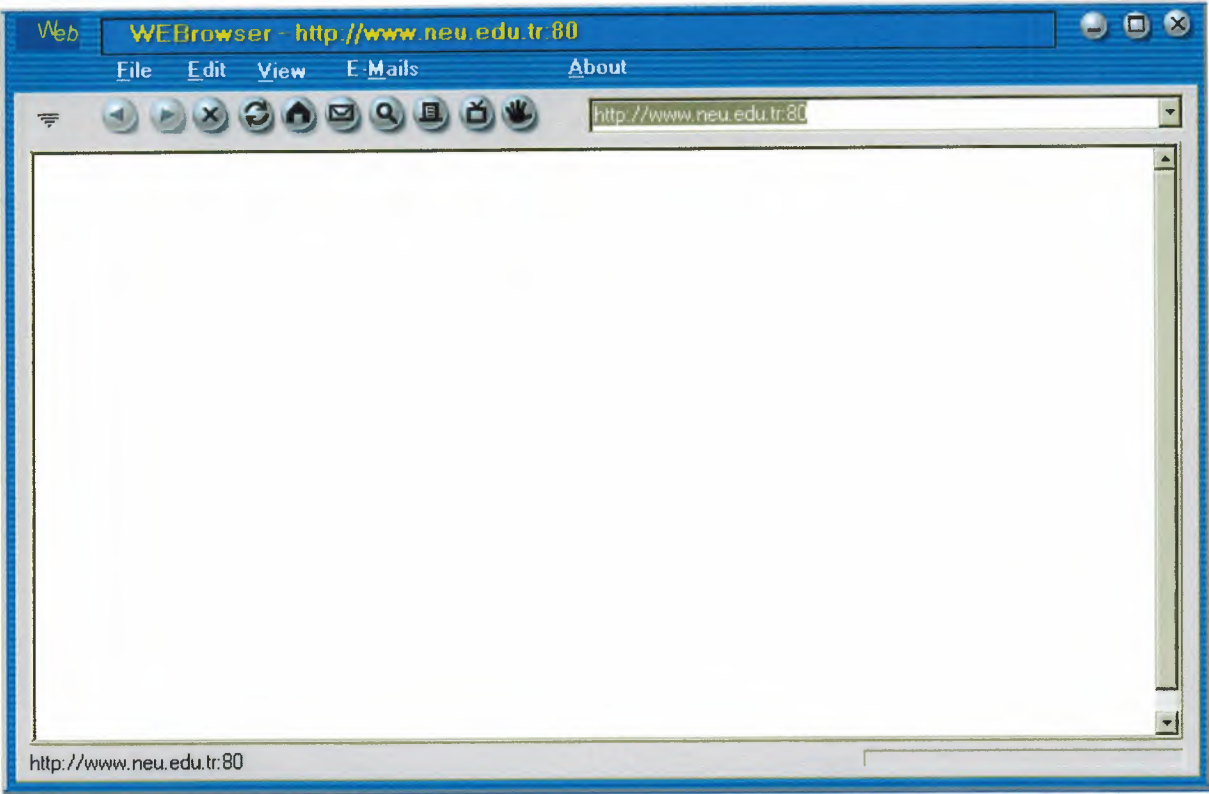
WEBrowser Main Window – Style 1 (Default)



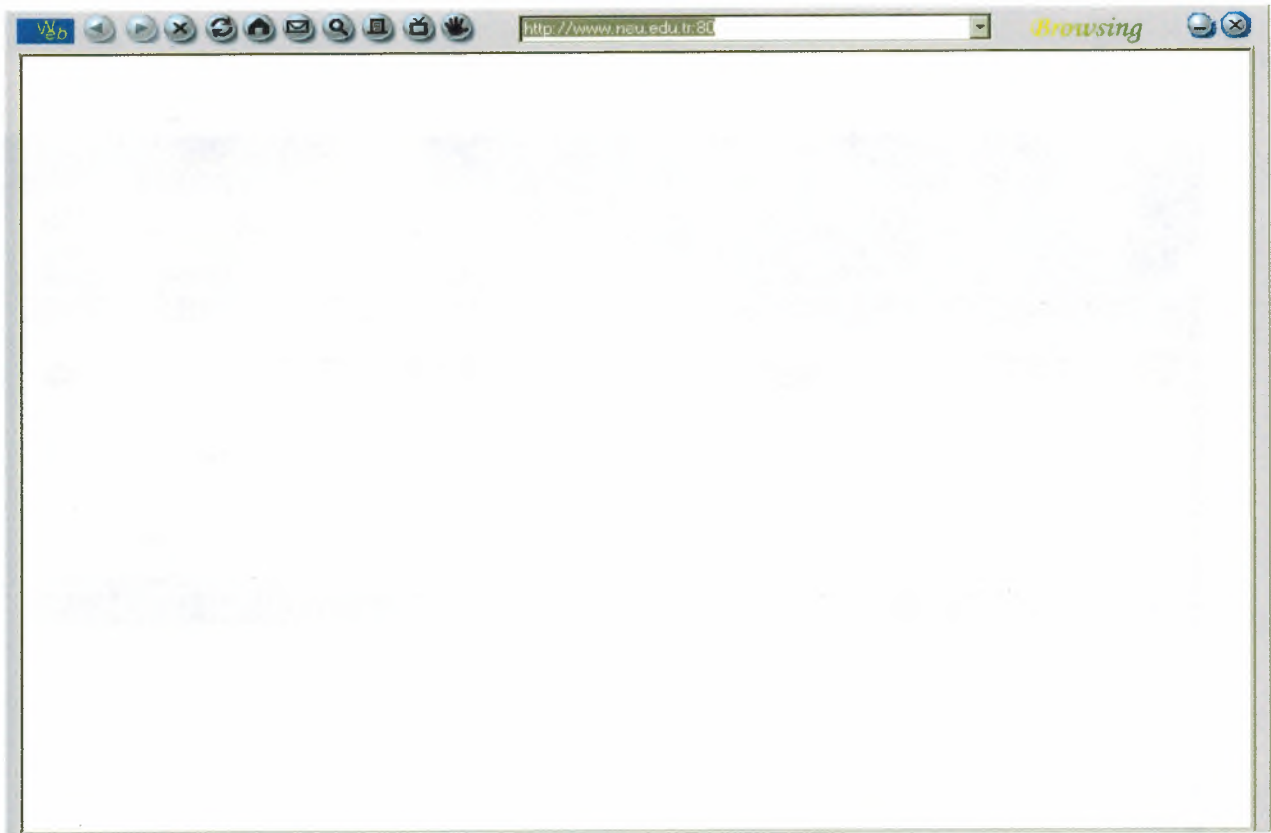
WEBrowser Main Window – Style 2



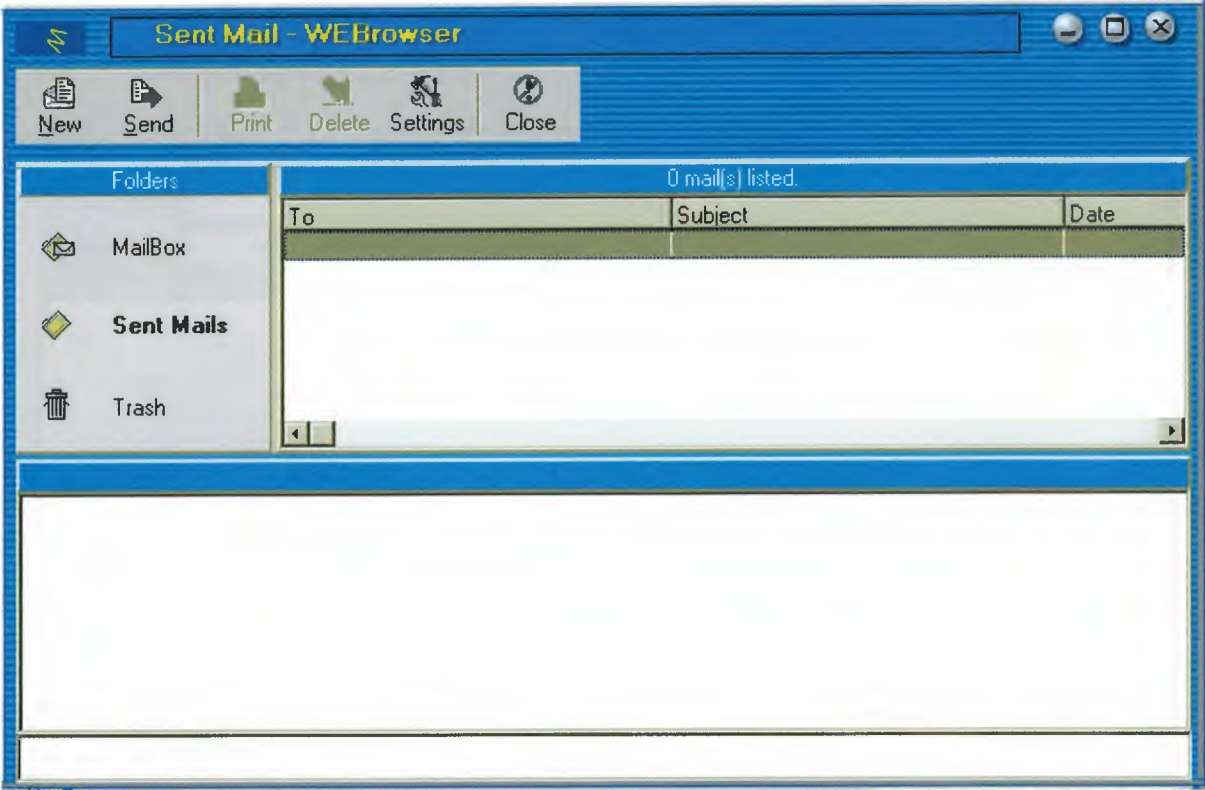
WEBrowser Main Window – Style 3



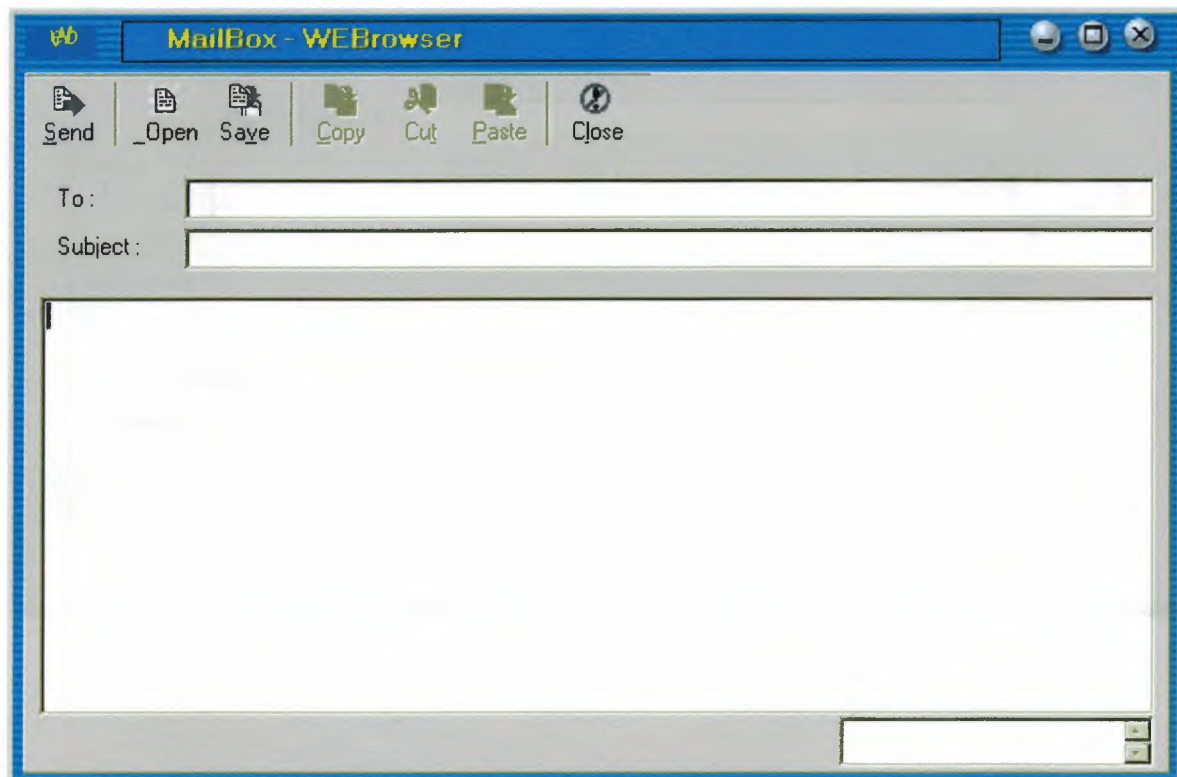
WEBrowser Main Window – Full Screen



WEBrowser E-mail Window



WEBrowser New E-mail Window



WEBrowser Options Window

Web: **WEBrowser - Options**

HomePage

Default

Font Name Font Size

Mail Settings

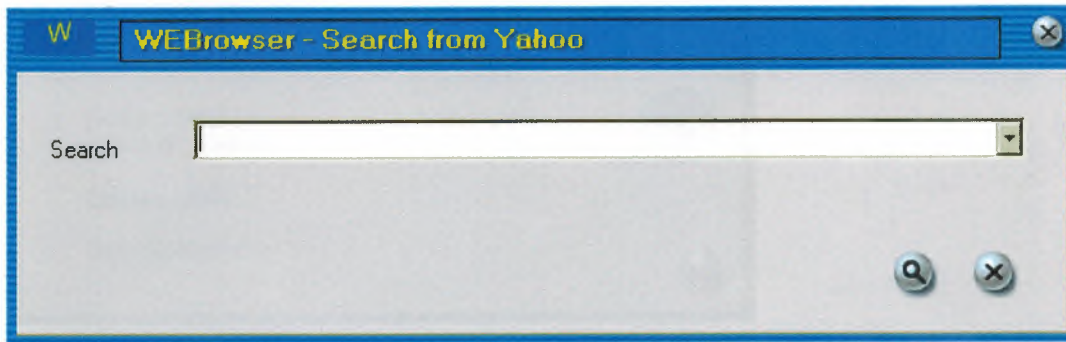
Your Name

Your Email Address

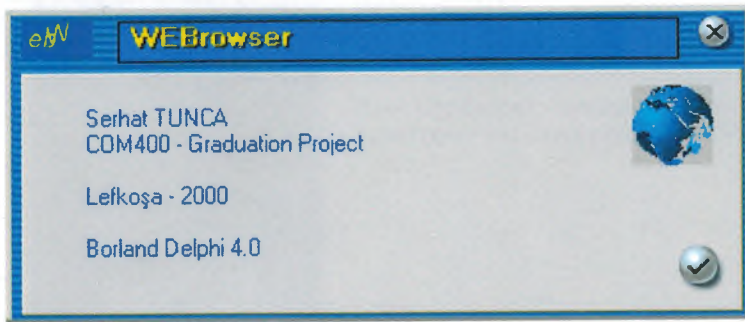
Signature

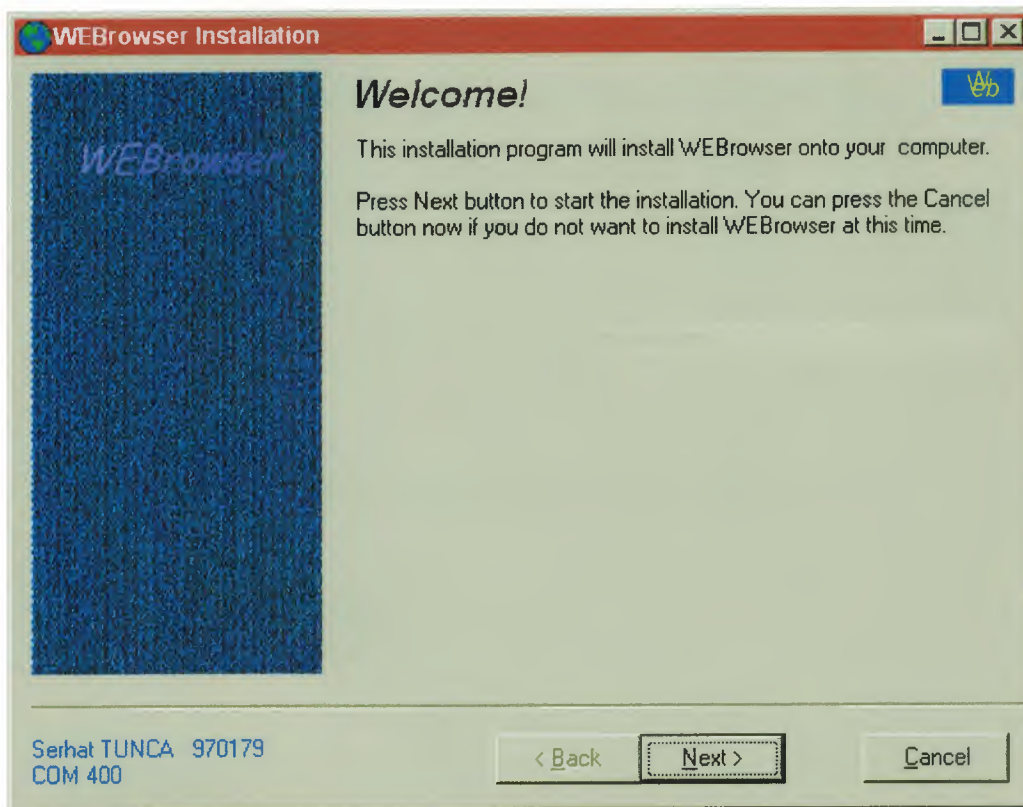
SMTP

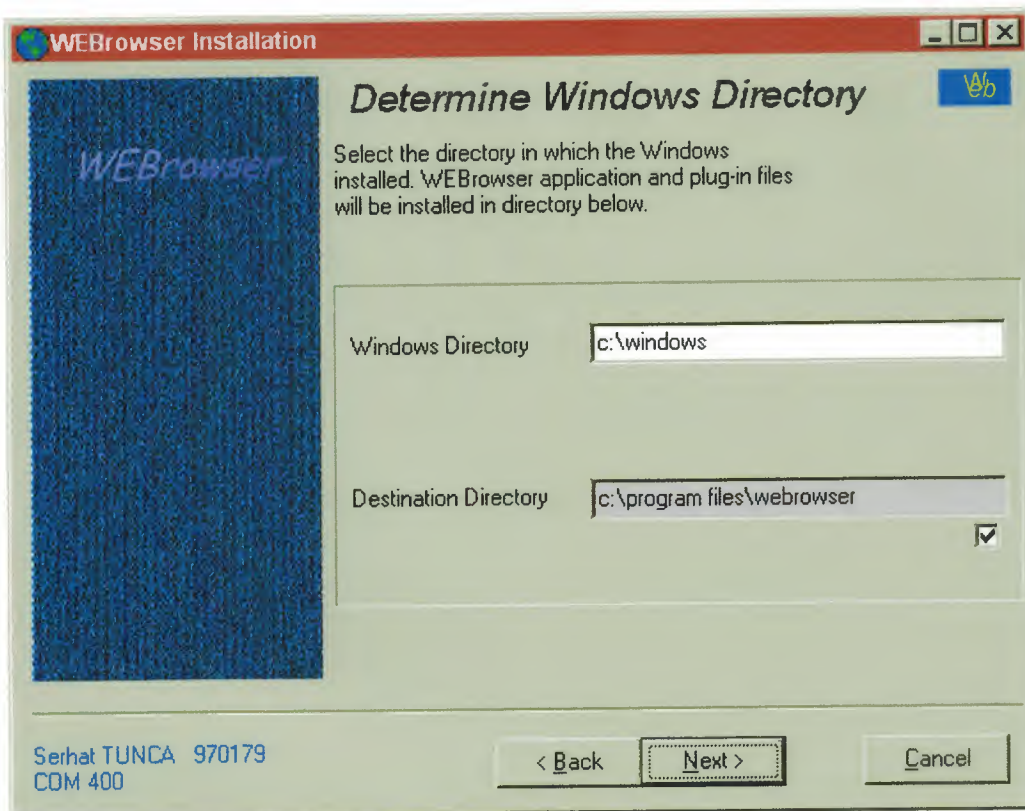
WEBrowser Search Window

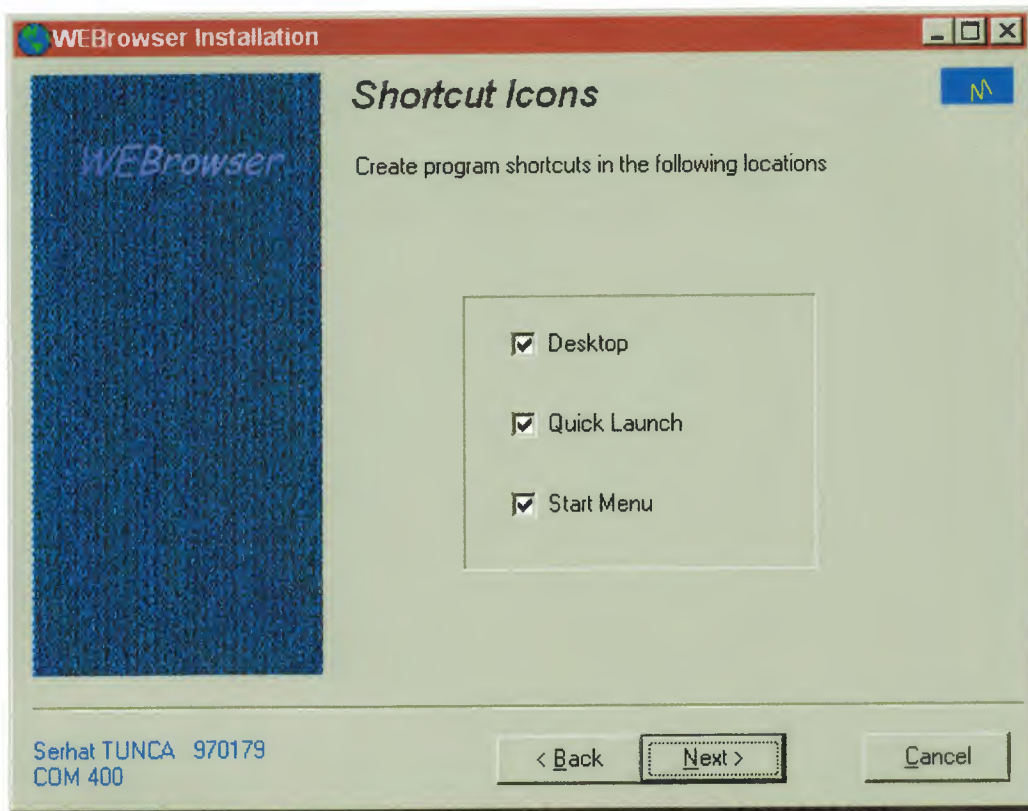


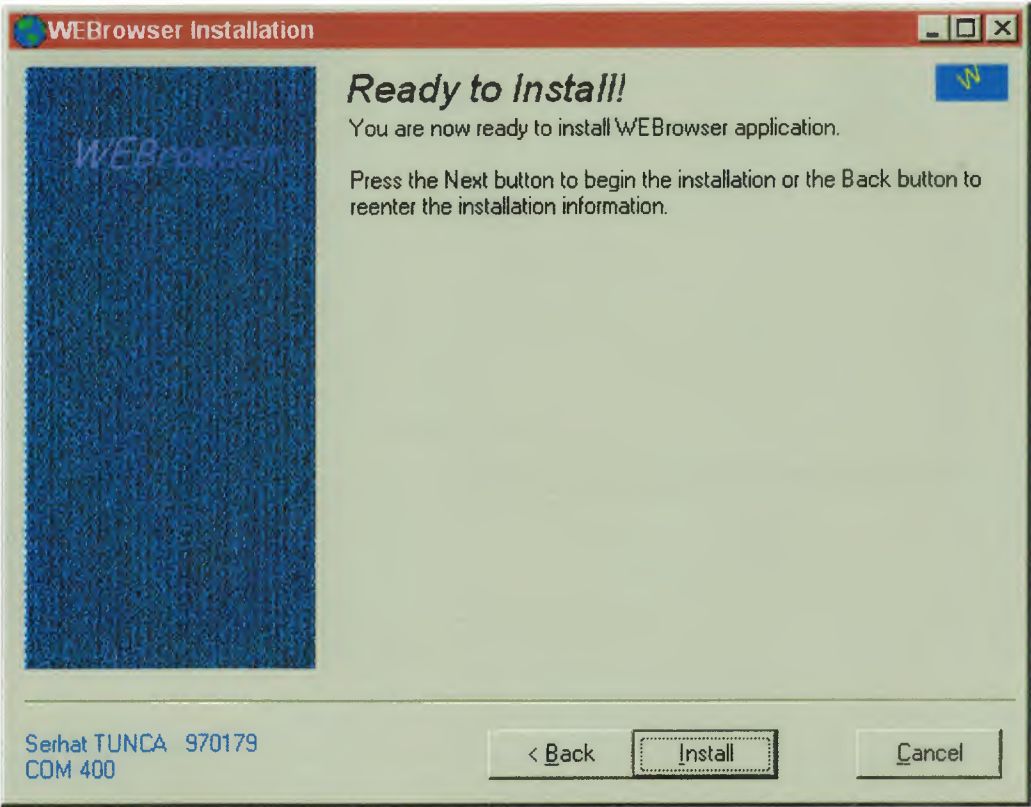
WEBrowser Search Window

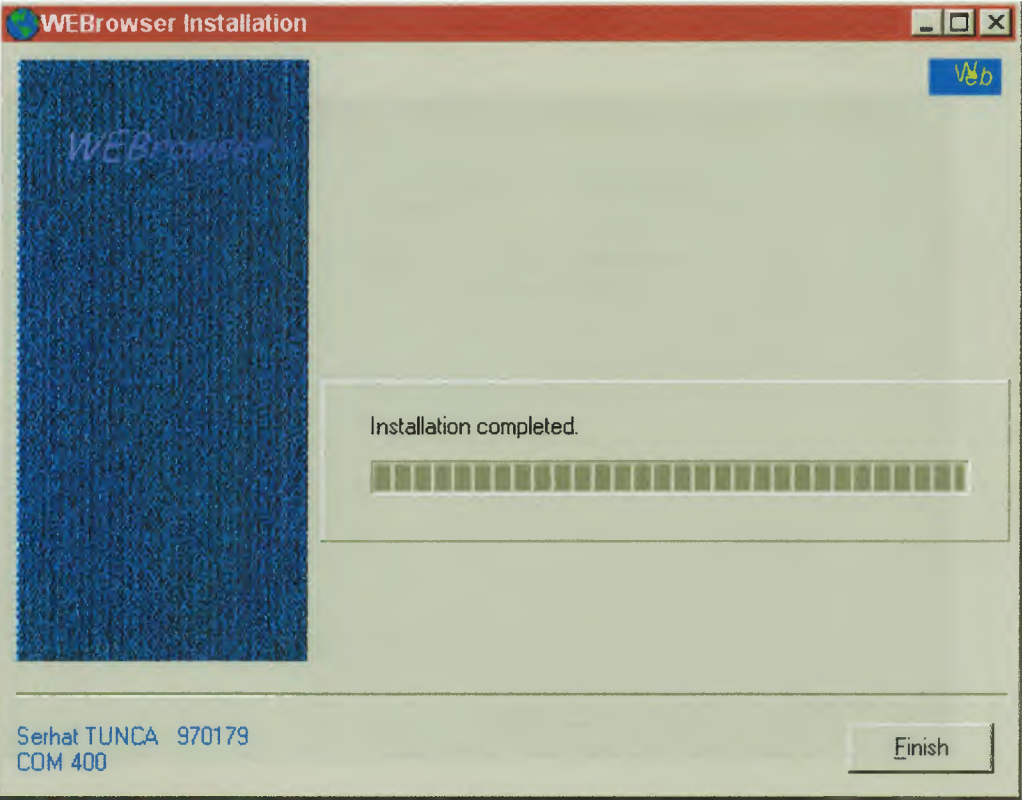


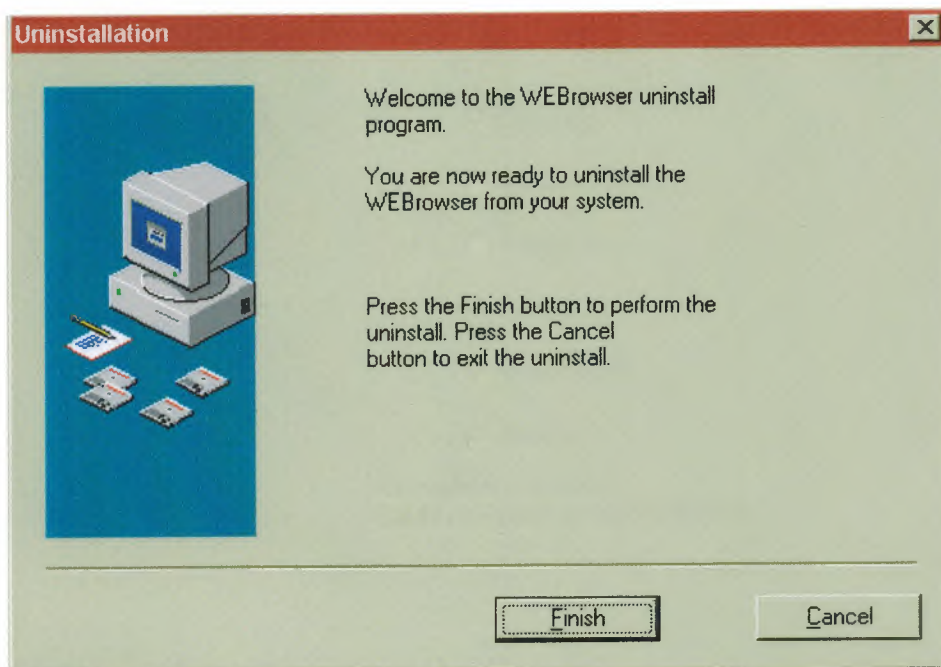


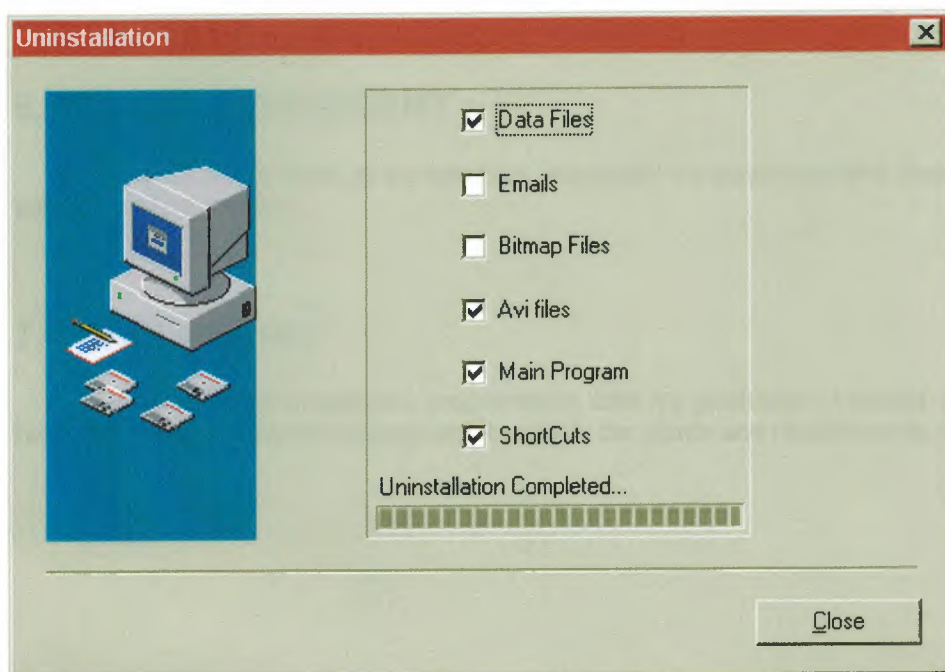












5.ABSTRACT :

It is hard to develop an internet browsing program which is more efficient than known programs such as Internet Explorer, Netscape in a short time period. But i recognized that these programs have boring screen layouts. Because of that i gave emphasis on screen layouts.

Also these programs do not have any direct searching window. I put this window in my browser to make searching easier. This searching technique can be developed by increasing the number of search engines. The entered keyword can be searched by five or more search engines to make search more efficient.

6.ACKNOWLEDGEMENT :

I would like to thank all my teachers, especially my supervisor Mrs. Besime ERIN and my advisor Mr.Tayseer Bey.

7.CONCLUSION :

I want to work on software programming after my graduation. I should learn completely at least one visual programming language to satisfy the wants and requirements of the customers.

8.REFERENCES :

- ❑ Mastering Delphi 4 – Marco Cantu Sybex 1999
- ❑ Bitmaps of buttons from <http://www.lycos.com>
- ❑ Animations are created by using Paint Shop Pro 6 (Animation Shop 2)