

INVIO DI POSTA TRAMITE DELPHI

L'invio di posta tramite delphi può avvenire in 2 modi differenti, o sfruttando il client di posta che è presente nel computer, o invece utilizzando componenti che lo stesso delphi mette a disposizione. Poniamo di avere il seguente form che usa tutti e due i metodi, il primo associato ad un pulsante, il secondo in automatico all'apertura del programma (FIGURA 1).

The screenshot shows a Windows-style application window titled "DOMINI". It contains several sections of form fields:

- Top Section:** Fields for "Rag. Sociale" (pippo), "Nome" (poppoloni), "Cognome" (poppoloni), "Sito" (www.poppoloni.it), "Scadenza" (07/07/2009), "Canone Annuale" (3234), and "Attivazione" (05/06/2005). There are buttons for "Invio Email", "Rinnova", and "Cerca". A spinner for "ANNI" is set to 1.
- Options Section:** Two groups of radio buttons: "Gestione DNS" (SI selected) and "Gestione Dominio" (SI selected).
- GESTIONE POSTA Section:** A sub-section with a "SI" radio button selected, and fields for "Indirizzo Server di Posta" (poppo.poppo.it), "Numero Caselle di Posta" (3), and "Password Amministratore di Posta" (34567898).
- SERVER WEB Section:** A sub-section with a "SI" radio button selected, fields for "Indirizzo Server" (sdad) and "Indirizzo IP" (assad), and a large text area containing "asdda".
- Contact Information Section:** Fields for "Telefono" (0696869868), "E-mail" (PIPP0@PIPP0.IT), "Telefonino" (33987654), and "Fax" (675443344).
- Bottom Section:** A row of navigation buttons (back, forward, home, etc.) and an "Esci" button.

FIGURA 1

Il codice da inserire all'interno del pulsante Invio Email è il seguente:

```
procedure TForm1.BTNemailClick(Sender: TObject);  
var  
stringa : string;  
stringa2: string;
```

```

begin
// invio una e-mail per avvertire direttamente il cliente
try
if form1.controllaemail(email.Text) = true then
begin
stringa2 := ' Attenzione il suo sito < ' + Sito.Text + ' > scadrà/è scaduto in data : ';
stringa := 'mailto:' + email.Text + '?Subject=Rinnovo del suo sito &Body= ' + stringa2 +
scadenza.text;
shellexecute (handle, 'open',pchar(stringa), "", "", sw_show);
end
else
showmessage('L"indirizzo e-mail non e" valido [4]');

except
showmessage('Invio e-mail fallito [5]');
end;
end;

function tform1.ControllaEmail(email:string) : boolean;
var
l,P,lung,lung2,contatore:integer;
stringa,carattere:string;
begin // 1
lung := Length(email);
stringa := email;
for l := 1 to lung do //2
begin // 2
carattere := LeftStr(stringa, 1);
if carattere = '@' then //3
begin //3
stringa := RightStr(stringa,(lung-(l+1)));
lung2 := Length(stringa);
if lung2 <= 1 then //7
begin //7
result := false;
exit;
end; //7
for P := 1 to lung2 do //8
begin //8
contatore := lung2 -P;
carattere := LeftStr(stringa, 1);
if carattere = '.' then //4b
begin //4
result := true;
exit;
end //4
else //4b
stringa := RightStr(stringa,(contatore));
if stringa = "" then //5
begin //5
result := false;

```

```

    exit;
    end; // 5
end; //4b
    end; //3
    stringa := RightStr(stringa,(lung-1));
    if stringa = " then //6
    begin //6
    result := false;
    exit;
    end; // 6
    end; // 2
    result := false; // aggiunta perchè mi dà errore
    exit;
end; // 1

```

La stringa principale è quella rossa, che viene eseguita dall'istruzione SHELLEXECUTE che prende in considerazione il programma di posta installato di default. Importante da ricordare e che per poter utilizzare tale routine/classe e la routine che controlla l'e-mail, è necessario inserire nelle USES → **shellapi** e anche → **strutils**.

Ecco come si presenta la finestra di invio della e-mail (FIGURA 2):

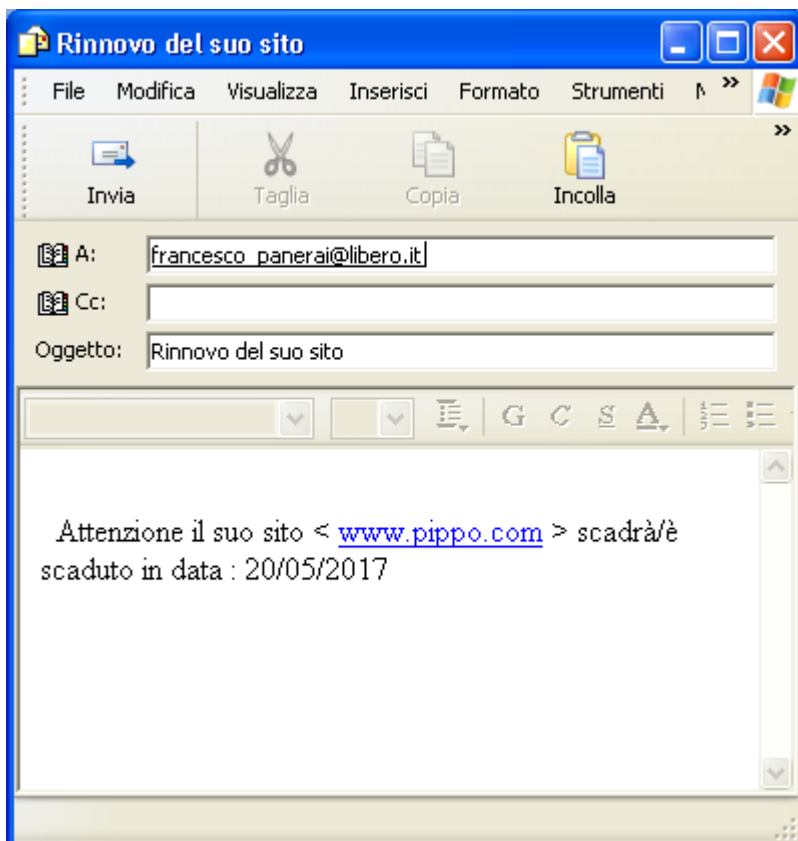


FIGURA 2

Il secondo metodo è un po' più complesso. Per prima cosa bisogna fare un form che dia la possibilità di salvare i dati di collegamento della posta POP3 e SMTP in un file ini. Il form dovrebbe essere come il seguente (FIGURA 3).

FIGURA 3

Con il seguente codice.....

```
unit setup;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, ExtCtrls, IdBaseComponent, IdMessage, IdComponent,  
IdTCPConnection, IdTCPClient, IdMessageClient, IdSMTP;
```

```
type
```

```
TEmailsetup = class(TForm)  
    Label1: TLabel;  
    Label2: TLabel;  
    Label3: TLabel;  
    POP3: TEdit;  
    PORT: TEdit;  
    Label4: TLabel;  
    ACCOUNT: TEdit;  
    Label5: TLabel;  
    PASSWORD: TEdit;  
    Label6: TLabel;  
    Email: TEdit;  
    Image1: TImage;  
    Label7: TLabel;  
    Label8: TLabel;  
    smtp: TEdit;  
    Label9: TLabel;
```

```

PortSmtplib: TEdit;
Label10: TLabel;
Label11: TLabel;
Accountsmtplib: TEdit;
Label12: TLabel;
passwordsmtplib: TEdit;
BTNok: TButton;
typesmtplib: TComboBox;
procedure BTNokClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
private
  { Private declarations }
function leggidaticonessione: boolean;
public
  { Public declarations }

end;

var
  Emailsetup: TEmailsetup;

implementation

{$R *.dfm}

uses dom1, IniFiles;

procedure TEmailsetup.BTNokClick(Sender: TObject);
var
  MailIni: TIniFile;
begin
  MailIni := TIniFile.Create(ExtractFilePath(ParamStr(0)) + 'Mail.ini');
  with MailIni do
  begin
    WriteString('Pop3', 'ServerName', POP3.Text);
    dom1.Pop3ServerName := POP3.Text;

    WriteString('Pop3', 'ServerPort', port.Text);
    Pop3ServerPort := StrToIntDef(port.Text, 110);

    WriteString('Pop3', 'ServerUser', Account.Text);
    Pop3ServerUser := Account.Text;

    WriteString('Pop3', 'ServerPassword', Password.Text);
    Pop3ServerPassword := Password.Text;

    WriteString('Smtplib', 'ServerName', SMTP.Text);
    SmtplibServerName := SMTP.Text;

    WriteString('Smtplib', 'ServerPort', Portsmtplib.Text);
    SmtplibServerPort := StrToIntDef(portsmtplib.Text, 110);
  end;
end;

```

```

WriteString('Smtplib', 'ServerUser', Accountsmtplib.Text);
SmtplibServerUser := Accountsmtplib.Text;

WriteString('Smtplib', 'ServerPassword', Passwordsmtplib.Text);
SmtplibServerPassword := Passwordsmtplib.Text;

WriteString('Email', 'PersonalEmail', Email.Text);
UserEmail := Email.Text;

WriteInteger('Smtplib', 'SMTPAuthenticationType', Typesmtplib.ItemIndex);
SmtplibAuthType := typesmtplib.ItemIndex;

end;
MailIni.Free;
Close;
end;

function TEmailSetup.Leggidaticonnessione: boolean;
var
  MailIni: TIniFile;
begin
  try
    MailIni := TIniFile.Create(ExtractFilePath(ParamStr(0)) + 'Mail.ini');
    with MailIni do
      begin
        Pop3ServerName := ReadString('Pop3', 'ServerName', '');
        Pop3ServerPort := StrToInt(ReadString('Pop3', 'ServerPort', '110'));
        Pop3ServerUser := ReadString('Pop3', 'ServerUser', '');
        Pop3ServerPassword := ReadString('Pop3', 'ServerPassword', '');

        SmtplibServerName := ReadString('Smtplib', 'ServerName', '');
        SmtplibServerPort := StrToInt(ReadString('Smtplib', 'ServerPort', '25'));
        SmtplibServerUser := ReadString('Smtplib', 'ServerUser', '');
        SmtplibServerPassword := ReadString('Smtplib', 'ServerPassword', '');
        SmtplibAuthType := ReadInteger('Smtplib', 'SMTPAuthenticationType', 0);

        UserEmail := ReadString('Email', 'PersonalEmail', '');
      end;
    finally
      MailIni.Free;
    result := true;
  end;
end;

procedure TEmailSetup.FormCreate(Sender: TObject);
begin

  if Leggidaticonnessione = true then
    if Pop3ServerName <> '' then
      begin

```

```

POP3.text := Pop3ServerName;
end;
if Pop3ServerPort <> 0 then
begin
Port.Text := IntToStr(Pop3ServerPort);
end;
if Pop3ServerUser <> " then
begin
Account.Text := Pop3ServerUser;
end;
if Pop3ServerPassword <> " then
begin
Password.Text := Pop3ServerPassword;
end;
if SmtServerName <> " then
begin
SMTP.Text := SmtServerName;
end;
if SmtServerPort <> 0 then
begin
Portsmtp.Text := IntToStr(SmtServerPort);
end;
if SmtServerUser <> " then
begin
Accountsmtp.Text := SmtServerUser;
end;
if SmtServerPassword <> " then
begin
Passwordsmtp.Text := SmtServerPassword;
end;
if SmtAuthType <> -1 then
begin
Typesmt.ItemIndex := SmtAuthType;
end;
if UserEmail <> " then
Email.Text := UserEmail;
end;
end.

```

A tale proposito (parte rossa) si dovrà tenere presente che nel form principale (FIGURA 1) dovranno esserci delle variabili che raggruppino i dati del form di salvataggio e la USES che serva a salvare il file ini :

uses dom1, IniFiles;

Ecco le variabili del form principale (dom1) che andranno sotto la parte etichettata var :

```

Pop3ServerName: string;
Pop3ServerPort: Integer;
Pop3ServerUser: string;
Pop3ServerPassword: string;

```

```

SmtpServerName: string;
SmtpServerPort: Integer;
SmtpServerUser: string;
SmtpServerPassword: string;
SmtpAuthType: Integer;
UserEmail: string;

```

Infine queste variabili sempre in dom1 dovranno essere istanziate a dei valori neutri che possano essere controllati dentro il form di salvataggio :

```

procedure TForm1.FormCreate(Sender: TObject);
begin
  -----
  -----
  -----
  Pop3ServerName := "";
  Pop3ServerPort := 0;
  Pop3ServerUser := "";
  Pop3ServerPassword := "";
  SmtpServerName := "";
  SmtpServerPort := 0;
  SmtpServerUser := "";
  SmtpServerPassword := "";
  SmtpAuthType := -1 ;
  UserEmail := "";
end;

```

La function leggidaticonnessione garantisce che se ci sono dati già salvati non vengano inseriti nel form quelli di default. Ora si può passare al form principale e al controllo automatico dell'invio della posta, che verrà effettuato tramite SQL parametriche. Il form si presenterà in questo modo (FIGURA 4 non completa) con delle scadenze dei relativi siti dei clienti, fra 30 giorni, fra 15 giorni e già scaduti in un lasso di tempo di 30 giorni (oggi fino a 30 giorni fa.

The screenshot shows a Windows-style application window titled "INVIO E-MAIL". It contains three vertically stacked sections, each with a title bar and a table below it. The first section is titled "Sono in scadenza i seguenti contratti dei siti entro 15 giorni". The second section is titled "Sono in scadenza i seguenti contratti dei siti entro 30 giorni". The third section is titled "Sono scaduti i seguenti contratti entro un lasso di tempo di 30 giorni". Each table has three columns: a small icon column, a text column, and a date column. At the bottom of the window, there are four buttons: "Dati", "Setup Email", "Reinvia Email", and "Esci".

FIGURA 4

Per prima cosa bisogna inserire nel datamodule 3 datasource e 3 query (FIGURA 5).

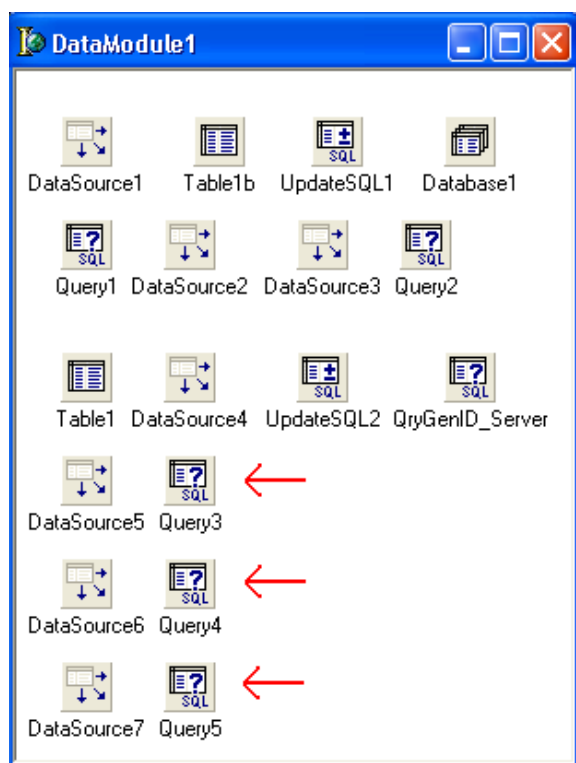


FIGURA 5

In ogni datasource bisogna inserire il nome del datate ovvero della query corrispondente
es. DataSource5 → **Query3**. In ogni query poi bisogna aggiungere il DatabaseName e la vera e propria query con delle variabili che la fanno divenire di tipo parametrico (voce SQL) :

```
select * from tabelladom where scadenza = :data1 ( per la prima )
select * from tabelladom where scadenza = :data2 ( per la seconda )
select * from tabelladom where scadenza between :data1 and :data2 ( per la terza )
```

Poi bisogna impostare il tipo di parametric che devono essere inseriti nella qsl parametrica (ovvero il tipo di parametro che corrisponde a data 1 e a data2). Selezionando PARAMS appare nell' Object Inspector la finestra del tipo di parametro. Qui si imposta che il DataType deve essere una data formattata in automatico così come vuole in database e cioè FTDATE (FIGURA 6).

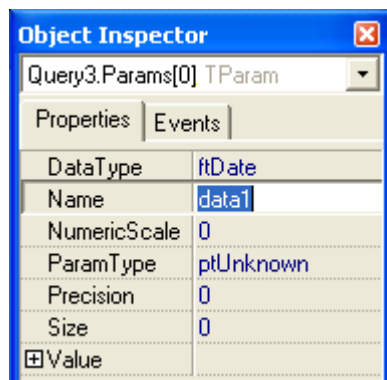


FIGURA 6

Per poter postare il datatype bisogna cliccare sulla variabile utilizzata che appare in una piccola finestra che riassume il nome delle variabili utilizzate nella SQL parametrica ecco i 3 casi (FIGURA 7 – 8 – 9).

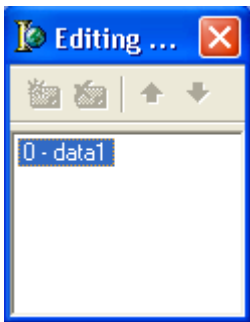


FIGURA 7 (QUERY 3)

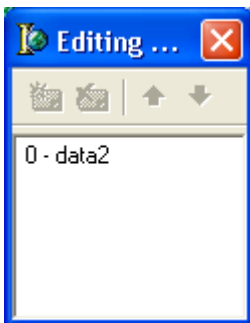


FIGURA 8 (QUERY 4)

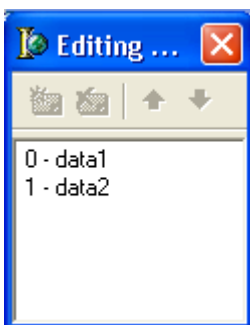


FIGURA 9 (QUERY 5)

Come si può notare vi sono le rispettive variabili, una per il primo e il secondo caso e 2 per il terzo (vedere le dichiarazioni precedenti).

Il codice da inserire nella routine che dovrebbe riempire dei DBGRID è il seguente (ricordarsi nei dbgrid di fare riferimento al DataSource corrispondente ad ogni query).

```
datamodule1.Query5.Close;  
datamodule1.Query5.ParamByName('Data1').AsDate:=Now-30;  
datamodule1.Query5.ParamByName('Data2').AsDate:=Now;  
DATAMODULE1.Query5.OPEN;  
  
datamodule1.Query3.Close;  
datamodule1.Query3.ParamByName('Data1').AsDate:=Now + 15;  
DATAMODULE1.Query3.OPEN;  
  
datamodule1.Query4.Close;  
datamodule1.Query4.ParamByName('Data2').AsDate:=Now + 30;  
DATAMODULE1.Query4.OPEN;
```

Da notare che in DataModule.Query5 il confronto fra le date deve essere effettuato obbligatoriamente con la prima data più vecchia e con la seconda attuale, in quanto la ricerca deve essere fatta in avanti (se si scambiano i dati non appare nulla del DBgrid corrispondente). Ecco cosa avviene dopo l'inserimento del codice precedente (FIGURA 10 non completa).

INVIO E-MAIL

Sono in scadenza i seguenti contratti dei siti entro 15 giorni

ID	NOME	COGNOME	RAGIONESOCIALE
4	poppoloni	poppoloni	pippo

Sono in scadenza i seguenti contratti dei siti entro 30 giorni

ID	NOME	COGNOME	RAGIONESOCIALE
1	franco	pippis2	pippo

Sono scaduti i seguenti contratti entro un lasso di tempo di 30 giorni

ID	NOME	COGNOME	RAGIONESOCIALE
2	pippo	depippis	pluto
3	fabio	Fava	fecco

Buttons: **Dati**, **Setup Email**, **Reinvia Email**, **Esci**

FIGURA 10

Ora si deve controllare che ogni DBgrid (per il primo e il secondo) abbia uno o più record corrispondenti ad altrettanti clienti che devono essere avvertiti in tempo e in modo automatico della scadenza dei loro contratti. Per prima cosa si devono inserire i controlli e l'esecuzione della ricerca nel database dei clienti con siti scaduti, all'interno della routine principale del form :

```
procedure TFrmEmail.FormCreate(Sender: TObject);
var
A:boolean;
begin
```

```

tempo := 0;
A := frmemail.leggidaticonnessione();
if A = true then
if controlladaticonnessione = true then
posta()
else
begin
// bisogna inserire i dati di connessione
showmessage('Attenzione bisogna inserire i dati di connessione[8]');
try
emailsetup:= Temailsetup.Create(Self);
emailsetup.Showmodal;
finally
FreeAndNil(emailsetup);
if controlladaticonnessione = true then
posta()
else
showmessage('Dati inseriti errati [10].');
close;
end;
end
else
showmessage('Posta NON inviata [7]');
end;

```

Per prima cosa si leggono i dati di connessione dal file .INI, si controlla che questi dati non siano quelli di default e si procede con la routine di invio posta (POSTA) , se invece i contenuti del file fossero riconosciuti non validi si apre la finestra di inserimento e si richiede (solo una volta) di inserire i dati di connessione (parte in rosso) (FIGURA 3).

Ecco la routine LEGGIDATI CONNESSIONE :

```

function TFrmEmail.leggidaticonnessione: boolean;
var
Maillni: TIniFile;
begin
try
Maillni := TIniFile.Create(ExtractFilePath(ParamStr(0)) + 'Mail.ini');
with Maillni do
begin
Pop3ServerName := ReadString('Pop3', 'ServerName', '');
Pop3ServerPort := StrToInt(ReadString('Pop3', 'ServerPort', '110'));
Pop3ServerUser := ReadString('Pop3', 'ServerUser', '');
Pop3ServerPassword := ReadString('Pop3', 'ServerPassword', '');

SmtpServerName := ReadString('Smtp', 'ServerName', '');
SmtpServerPort := StrToInt(ReadString('Smtp', 'ServerPort', '25'));
SmtpServerUser := ReadString('Smtp', 'ServerUser', '');
SmtpServerPassword := ReadString('Smtp', 'ServerPassword', '');
SmtpAuthType := ReadInteger('Smtp', 'SMTPAuthenticationType', 0);

```

```

UserEmail := ReadString('Email', 'PersonalEmail', '');
end;
finally
Maillni.Free; // talvolta questa riga da errore ( si può eliminare )
result := true;
end;
end;

```

Ecco la routine CONTROLLADATICONNESSIONE:

```

function tfrmemail.ControllaDatiConneSSIONe():boolean;
begin
if (Pop3ServerName = 'pop3.server.com') or (pop3ServerUser = 'your_login') or
(Pop3ServerPassword = 'your_password') or (SmtpServerName = 'smtp.server.com') or
(SmtpServerUser = 'your_login') or (SmtpServerPassword = 'your_password') then
result := false
else
if (Pop3ServerName <> 'pop3.server.com') or (pop3ServerUser <> 'your_login') or
(Pop3ServerPassword <> 'your_password') or (SmtpServerName <> 'smtp.server.com') or
(SmtpServerUser <> 'your_login') or (SmtpServerPassword <> 'your_password') then
result := true
else
result := false;
end;

```

INFINE ECCO LA ROUTINE PRINCIPALE : POSTA

```

procedure TFrmEmail.posta;
var
NumRec15,NumRec30,I,K: integer;
messaggio,email:string;
errore:Tstrings;
riepilogo:Tstrings;

begin
// creazione query ricerca dati e riempimento DBgrid
errore := TStringList.create;
riepilogo := TStringList.create;
try
errore.add('ERRORI');
riepilogo.add('RIEPILOGO');

datamodule1.Query5.Close;
datamodule1.Query5.ParamByName('Data1').AsDate:=Now-30;
datamodule1.Query5.ParamByName('Data2').AsDate:=Now;
DATAMODULE1.Query5.OPEN;

datamodule1.Query3.Close;
datamodule1.Query3.ParamByName('Data1').AsDate:=Now + 15;
DATAMODULE1.Query3.OPEN;

```

```

datamodule1.Query4.Close;
datamodule1.Query4.ParamByName('Data2').AsDate:=Now + 30;
DATAMODULE1.Query4.OPEN;

```

```

numrec15 := DATAMODULE1.Query3.recordcount;
numrec30 := DATAMODULE1.Query4.recordcount;

```

```

if ((numrec15 = 0) and (numrec30 = 0)) then
begin
exit;
end;

```

```

if (( numrec15 - 1) = -1) then //0
begin //1
//nulla
end // 1

```

```

else //0

```

```

begin //0+

```

```

for I := 0 to (numrec15 - 1) do //2
begin //2

```

```

if I = 0 then //3

```

```

begin //3

```

```

datamodule1.Query3.FindFirst;

```

```

if (controllaemail(datamodule1.query3.fieldbyname('email').AsString)) = true then
//4

```

```

begin //5

```

```

if datamodule1.query3.fieldbyname('email').AsString = " then

```

```

//6

```

```

begin //7

```

```

// errore

```

```

errore.Add('[A] Errore per l"azienda ' +

```

```

datamodule1.query3.fieldbyname('ragionesociale').AsString + ' campo e-mail vuoto.');
```

```

end //7

```

```

else //6

```

```

begin //8

```

```

// invio l'email

```

```

messaggio := 'Attenzione il vostro sito > ' +

```

```

datamodule1.query3.fieldbyname('sito').AsString + ' < scadrà in data > ' +

```

```

datetostr(datamodule1.query3.fieldbyname('scadenza').asdatetime) + '!';

```

```

email := datamodule1.query3.fieldbyname('email').AsString;

```

```

if inviaemail(messaggio,email) = true then //9

```

```

begin //10

```

```

// riepilogo

```

```

riepilogo.add('Invio e-mail riuscito per l"azienda ' +

```

```

datamodule1.query3.fieldbyname('ragionesociale').AsString + ' e-mail > ' +

```

```

datamodule1.query3.fieldbyname('email').AsString + ' <.';

```

```

end // 10

```

```

else //9

```

```

begin //1
//errore
errore.Add('[B] Errore per l"azienda ' +
datamodule1.query3.fieldbyname('ragionesociale').AsString + ' e-mail non inviata > ' +
datamodule1.query3.fieldbyname('email').AsString + ' <.');
end; //1
end; //8
end; //5
end; //3

if I > 0 then
begin //1
datamodule1.Query3.Findnext;

if controllaemail(datamodule1.query3.fieldbyname('email').AsString) = true then
begin //2
if datamodule1.query3.fieldbyname('email').AsString = " then
begin //3
// errore
errore.Add('[C] Errore per l"azienda ' +
datamodule1.query3.fieldbyname('ragionesociale').AsString + ' campo e-mail vuoto.');
end //3
else
begin //4
// invio l'email
messaggio := 'Attenzione il vostro sito > ' +
datamodule1.query3.fieldbyname('sito').AsString + ' < scadrà in data > ' +
datetostr(datamodule1.query3.fieldbyname('scadenza').asdatetime) + '.';
email := datamodule1.query3.fieldbyname('email').AsString;
if inviaemail(messaggio,email) = true then
begin //5
// riepilogo
riepilogo.add('Invio e-mail riuscito per l"azienda ' +
datamodule1.query3.fieldbyname('ragionesociale').AsString + ' e-mail > ' +
datamodule1.query3.fieldbyname('email').AsString + ' <.);
end; //5
end; //4
end //2
else
begin //6
//errore
errore.Add('[D] Errore per l"azienda ' +
datamodule1.query3.fieldbyname('ragionesociale').AsString + ' e-mail non inviata > ' +
datamodule1.query3.fieldbyname('email').AsString + ' <.);
end; //6
end; //1
end; //2
end; //0+

if ( numrec30 - 1) = -1 then

```

```

begin //0
//si esce
exit;
end //0
else
begin //1
for K := 0 to (numrec30 - 1) do
begin //2

if K = 0 then
begin //3
datamodule1.Query4.FindFirst;

if controllaemail(datamodule1.query4.fieldbyname('email').AsString) = true then
begin //4
if datamodule1.query4.fieldbyname('email').AsString = '' then
begin //5
// errore
errore.Add('[E] Errore per l''azienda ' +
datamodule1.query4.fieldbyname('ragionesociale').AsString + ' campo e-mail vuoto.');
```

```

end //5
else
begin //6
// invio l'email
messaggio := 'Attenzione il vostro sito > ' +
datamodule1.query4.fieldbyname('sito').AsString + ' < scadrà in data > ' +
datetostr(datamodule1.query4.fieldbyname('scadenza').asdatetime) + ':';
email := datamodule1.query4.fieldbyname('email').AsString;
if inviaemail(messaggio,email) = true then
begin //7
// riepilogo
riepilogo.add('Invio e-mail riuscito per l''azienda ' +
datamodule1.query4.fieldbyname('ragionesociale').AsString + ' e-mail > ' +
datamodule1.query4.fieldbyname('email').AsString + ' <.');
end //7
else
begin //8
//errore
errore.Add('[F] Errore per l''azienda ' +
datamodule1.query4.fieldbyname('ragionesociale').AsString + ' e-mail non inviata > ' +
datamodule1.query4.fieldbyname('email').AsString + ' <.');
end; //8
end; //6
end; //4
end; //3

if K > 0 then
begin //1b
datamodule1.Query4.Findnext;
```

```

if controllaemail(datamodule1.query4.fieldbyname('email').AsString) = true then
begin //2b
if datamodule1.query4.fieldbyname('email').AsString = " then
begin //3b
// errore
errore.Add('[G] Errore per l"azienda ' +
datamodule1.query4.fieldbyname('ragionesociale').AsString + ' campo e-mail vuoto.');
```

```

end //3b
else
begin //4b
// invio l'email
messaggio := 'Attenzione il vostro sito > ' +
datamodule1.query4.fieldbyname('sito').AsString + ' < scadrà in data > ' +
datetostr(datamodule1.query4.fieldbyname('scadenza').asdatetime) + ':';
email := datamodule1.query4.fieldbyname('email').AsString;
if inviaemail(messaggio,email) = true then
begin //5b
// riepilogo
riepilogo.add('Invio e-mail riuscito per l"azienda ' +
datamodule1.query4.fieldbyname('ragionesociale').AsString + ' e-mail > ' +
datamodule1.query4.fieldbyname('email').AsString + ' <.);
end //5b
else
begin //6b
//errore
errore.Add('[H] Errore per l"azienda ' +
datamodule1.query4.fieldbyname('ragionesociale').AsString + ' e-mail non inviata > ' +
datamodule1.query4.fieldbyname('email').AsString + ' <.);
end; //6b
end; //4b
end; //2b
end; //1b
end; //2

end; //1
//end;

memo1.lines := riepilogo;
memo2.lines := errore;
finally
riepilogo.free;
errore.free;
end;
end;
```

Per prima cosa si creano 2 array di tipo Tstrings (parte rossa) che contengano rispettivamente gli errori rilevati e il sunto delle operazioni fatte e andate a buon fine. Per questo nel form si devono mettere 2 controlli tipo MEMO per far apparire la lista degli errori e quella delle operazioni. Ecco come si presenta il form completo (FIGURA 11).

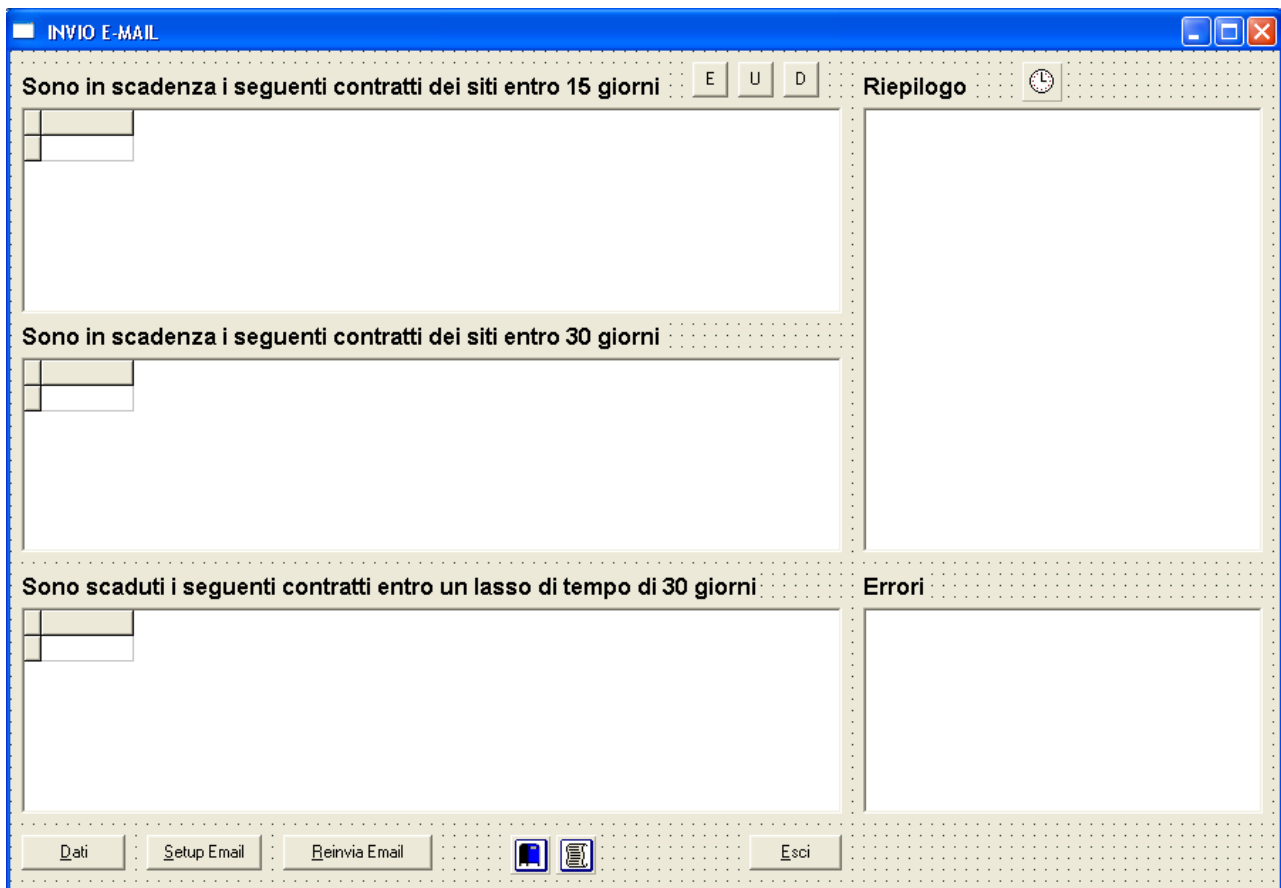


FIGURA 11

Si usano delle query parametriche per inserire le date relative alla ricerca di quei siti scaduti da 15 giorni, da 30 giorni ed infine da un lasso di tempo che va da oggi a 30 giorni fa (parte rossa) (FIGURA 5/9). A questo punto si calcolano i records presenti nella prima e nella seconda DBgrid. Questo servirà per fare una iterazione su tutti i records in modo da prelevare i 2 dati che a noi interessano, la scadenza (scadenza), l'indirizzo e-mail (email) del cliente a cui inviare l'avviso e il nome del corrispettivo sito (in quanto uno stesso cliente potrebbe avere più di un sito), e la ragione sociale della ditta in questione (ragione sociale) (alcuni di questi evidenziati in rosso). Le operazioni che vengono eseguite sono :

- 1) controllo della e-mail
- 2) emissione di un messaggio di errore o no
- 3) postare i dati per creare il messaggio da inviare
- 4) richiedere alla routine di invio l'esecuzione dell' operazioni di posta
- 5) emettere un messaggio di errore o di conferma
- 6) passare al successivo record

Come si sarà notato si usa FINDFIRST e FINDNEXT per trovare i record che corrispondono a quelli presenti nei 2 DBgrid.

ECCO INFINE LA ROUTINE DI INVIO DEL MESSAGGIO :

```
function TFrmEmail.InviaEmail(messaggio:string;email:string): boolean;
Var
  Mes:TStrings;
```

```

begin
  with frmemail.SEND do
  begin
    Mes:=TStringList.Create;
    Mes.Add('Comunicazione automatica dalla PIPPO Informatica ');
    mes.add(messaggio);

    Body.Assign(Mes);
    From.Text := UserEmail;
    Recipients.EmailAddresses := email;

    Subject := ' Scadenza Sito '; { Subject: header }
    Priority := TIdMessagePriority(frmemail.SEND.priority); { Message Priority }
    ReceiptRecipient.Text := 'pippo_pippo@libero.it';

end;

{authentication settings}

frmemail.SMTP.AuthenticationType := atLogin;

frmemail.SMTP.UserID := SmtplibServerUser;
frmemail.SMTP.Password := SmtplibServerPassword;

{General setup}
frmemail.SMTP.Host := SmtplibServerName;
frmemail.SMTP.Port := SmtplibServerPort;

{now we send the message}
frmemail.SMTP.Connect;

try // except
try
  begin
    frmemail.SMTP.Send(frmemail.SEND);
    // bisogna farlo aspettare un pò perchè le e-mail si accavallano
    result := true;
    timer1.Enabled := true;
  end;
finally
  //showmessage(email);
  frmemail.SMTP.Disconnect;
end;
Mes.free;
except
showmessage('Posta NON inviata [6]');
result := false;
end;
end;

```

In questo caso quello che si fa è:

- 1) Completare il messaggio
- 2) Inserire i dati di connessione che sono presenti nel file ini e residenti in variabili globali all'interno del programma
- 3) Connettersi ed inviare l'email
- 4) OPZIONALE – usare un timer per mettere tempo fra un invio e l'altro perché ci sono alcuni server che trovano difficoltà nel ricevere messaggi uno dietro l'altro. In questo caso il timer è settato a ½ secondo e completa il suo ciclo in 1 secondo per messaggio.

```
procedure TFrmEmail.Timer1Timer(Sender: TObject);  
// per mettere spazio fra l'invio di una e-mail e l'altra  
begin  
tempo := tempo + 1;  
if tempo = 2 then  
begin  
tempo := 0;  
timer1.Enabled := false;  
end;  
end;
```

Ecco i 2 modi per inviare la posta. Spero di essere stato abbastanza chiaro. Buona fortuna e buon lavoro.