

Come si inserisce una Icon Try in un programma Delphi

Ecco come dovrebbe essere il risultato dell'inserimento di una Icon Try, partendo da un software che esibisca questa icona di programma (FIGURA 1 - 2 e 3).



FIGURA 1

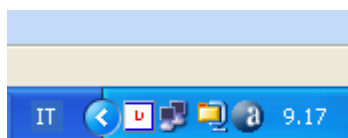


FIGURA 2

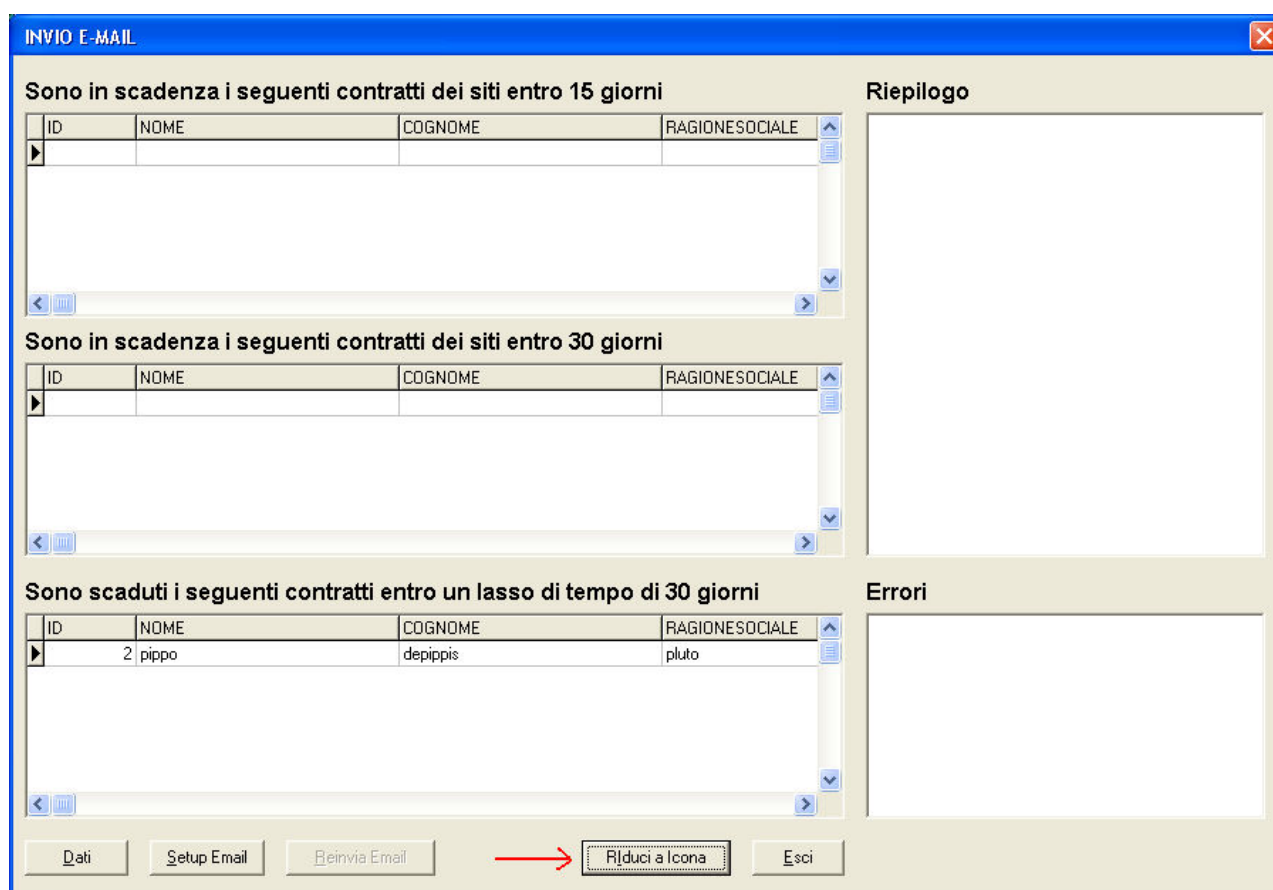


FIGURA 3

La parte di programma che andrò a presentare serve unicamente a ridurre ad icona, nell'Icon Try, un software e ad riportarlo alle sue condizioni normali se l'icona viene premuta. Per poi ridurlo di nuovo ad icona si dovrà premere il pulsante RIDUCI A ICONA (FIGURA 3). Ecco il codice che bisogna aggiungere al software:

Per prima cosa si deve aprire la finestra del progetto e inserire la riga rossa, prima dell'apertura del form principale :

```

begin
  Application.Initialize;
  Application.Title := 'Domini';
  Application.CreateForm(TDataModule1, DataModule1);
  // per la icon try
  Application.ShowMainForm := False;
  Application.CreateForm(TFrmEmail, FrmEmail);
  Application.Run;
end.

```

In questo modo all'avvio del form principale, in questo caso FRMEMAIL (riga successiva) questo apparirà direttamente nella icon try e non verrà visualizzato come un normale form. All'interno della finestra principale si dovrà poi aggiungere il seguente codice:

1) nelle USES : **ShellAPI**

2) Subito dopo nelle costanti :

```

const
  //my call back constant I could of just used wm_user
  WM_CALLBACK = WM_USER;

```

3) Nella dichiarazione TYPE della form :

```

procedure WM_CALLBACKPRO(var msg : TMessage); message wm_callback;

```

4) Poi nelle variabili globali VAR :

```

blah : HICON;
TrayIcon : TNotifyIconData;

```

5) Nella routine FormCreate il seguente codice:

```

FrmEmail.Visible := False;
//give the application icon handle to blah
blah := application.Icon.Handle;
//Get The Size Of Byte Of TNotifyIconData
TrayIcon.cbSize := SizeOf(TNotifyIconData);
//Say The Icon Is Coming From The Form
TrayIcon.Wnd := handle;
//Give A Tip Message With Mouse Over
TrayIcon.szTip := 'Hot Slut Ready!';
//Give It A Identifier of your icon
TrayIcon.uID := 1;
//give the hicon to the tray icon
TrayIcon.hIcon := blah;
//my callback function so when click it does something
TrayIcon.uCallbackMessage := WM_CALLBACK;
//TrayIcon flags to say what it can do
TrayIcon.uFlags := NIF_MESSAGE or NIF_ICON or NIF_TIP;
//Add The Icon To The Tray
Shell_NotifyIcon(NIM_ADD,@trayicon);

```

6) Poi si dovrà inserire la procedura precedentemente dichiarata :

```
procedure TFrmEmail.WM_CALLBACKpro(var msg : TMessage);
begin
//WM_LBUTTONDOWN is from the lparam so if left mouse button is pressed it does
something
case msg.LParam of WM_LBUTTONDOWN :
begin
//Delete The Tray Icon If Its Click And Show The Form Again
Shell_NotifyIcon(NIM_DELETE,@TrayIcon);
frmemail.Visible := True;
end;
end;
end;
```

7) Ed infine il codice del pulsante che riduce di nuovo la finestra ad icona, uguale a quello che abbiamo visto per la routine principale del form :

```
procedure TFrmEmail.ButtonIconaClick(Sender: TObject);
begin
FrmEmail.Visible := False;
//give the application icon handle to blah
blah := application.Icon.Handle;
//Get The Size Of Byte Of TNotifyIconData
TrayIcon.cbSize := SizeOf(TNotifyIconData);
//Say The Icon Is Coming From The Form
TrayIcon.Wnd := handle;
//Give A Tip Message With Mouse Over
TrayIcon.szTip := 'Hot Slut Ready!';
//Give It A Identifier of your icon
TrayIcon.uID := 1;
//give the hicon to the tray icon
TrayIcon.hIcon := blah;
//my callback function so when click it does something
TrayIcon.uCallbackMessage := WM_CALLBACK;
//TrayIcon flags to say what it can do
TrayIcon.uFlags := NIF_MESSAGE or NIF_ICON or NIF_TIP;
//Add The Icon To The Tray
Shell_NotifyIcon(NIM_ADD,@trayicon);
end;
```