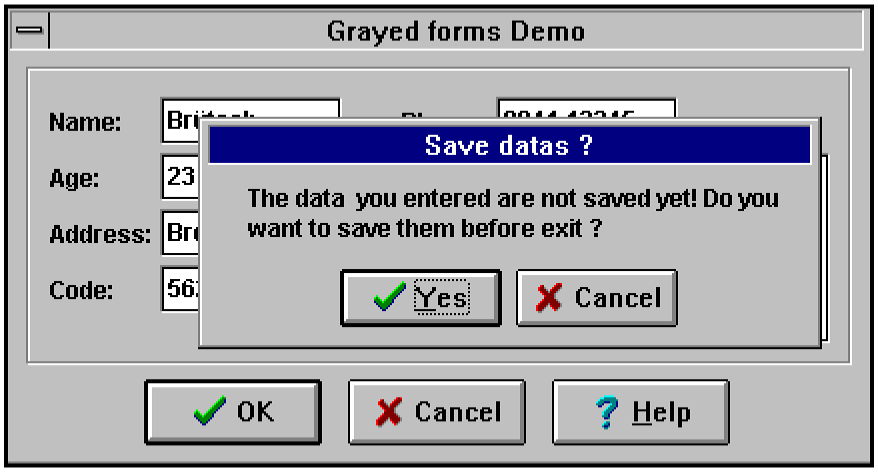
delphi *"Grayed Forms"* utility

SHAREWARE - Version 1.1, (c)copyright 1995 by Martin Brütsch, CIS ID 100430,3660
>>Registers vs COMPUSERVE, SWREG ID #8027.

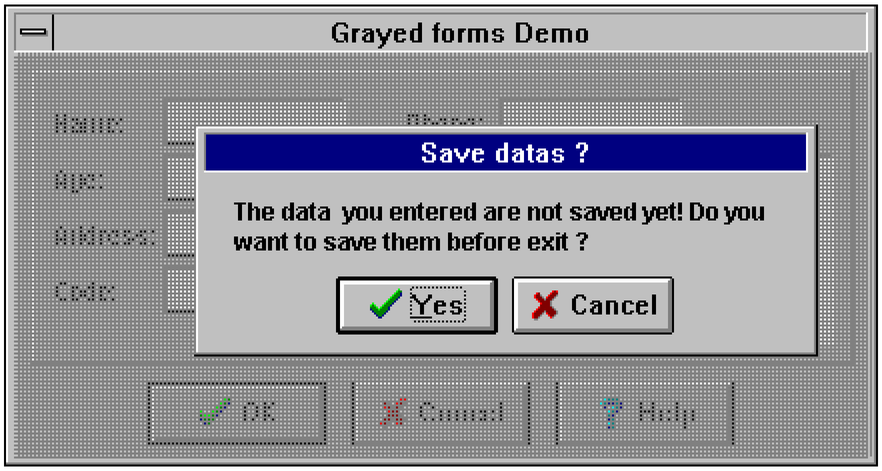
You know the following problem **certainly:**



The data you entered are not saved yet! Do you want to save them before exit ?

Thousands of very nice buttons (especially 2 x OK, 2 x CANCEL a.s.o.), but which one in the hell the user has to fire to go on with your smart delphi application ?! Be sure, a lot of people will first press one of the the wrong (disabled) buttons and curse of, because absolutly nothing happens (except a nice windows "beep").

The solution: "**Delphi Grayed Forms**" !



The data you entered are not saved yet! Do you want to save them before exit ?

Delphi "Grayed Forms" is a very small delphi utility unit, that temporary paints a grayed net across forms, that are not accessable by the user, e.g. because another modal form or a "messagedlg" is currently active. The unit manages up to 99 grayed forms at one time.

Specifications: - Sizeable-, Dialog-, NoneBorder- and SingleBorder-windows
- For modal *and* none-modal windows
- Manages upt to 99 grayed forms at one time.
- Full VCL-compatible
- small unit with less than 8KB

How to install the "*Grayed Forms*" utility unit ?

(1) Copy the files GRAYFORM.DCU and GRAYFORM.DFM into your project directory.

(2) Add the unit-name GRAYFORM to the uses-statement of EACH form, you want to be grayed, while it's unaccessable.

Example:

```
uses
   SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics,
   Controls, Forms, Dialogs, ExtCtrls, StdCtrls, Buttons,
   Show2, GrayForm;
```

(3) Write "GRAYEDFORM(SELF);" before you call a modal window or messagedlg and

use "UNGRAYEDFORM(SELF);" after calling them.

Example 1:

```
procedure TForm1.BitBtn1Click(Sender: TObject);
begin
      GrayedForm(SELF);    { <- gray form }
      Form2.ShowModal;     { show another modal form }
     UnGrayedForm(SELF);  { <- ungray form }
end;
```

Example 2:

```
procedure TForm1.BitBtn1Click(Sender: TObject);
begin
      GrayedForm(SELF);    { <- gray form }
      MessageDlg("Hello world...",mterror,[mbok],0);
     UnGrayedForm(SELF);  { <- ungray form }
end;
```

(4) That's all, so just compile it and enjoy the result !!!

You can gray the actual form too, while you're displaying a non-modal form for showing e.g. the process of a complex operation, that takes a lot of time.

There's a sample delphi application included in this package that demonstrates different ways to use "*Grayed Forms*" for modal and non-modal windows.

Files within this archive: - GRAYFORM.DCU
- GRAYFORM.DFM
- README.WRI

IMPORTANT:

The unregistered version of "*Grayed Forms*" has no functionality limitation, but it works only while the DELPHI development system (IDE) is running.

If you register for only $12, you'll get a copy that works without running DELPHI IDE. So test the utility as long as you need to find out, if it works properly together with your application. Do this tests before you register !!!

Registration can be done through Compuserve: **GO SWREG, ID # 8027**

Also watch out for my other DELPHI products: TABLE-PRINTER + BLINKLABEL, that can be found within the CS "BORGMBH" forum, section 4/DELPHI.

Have fun: *Martin Brütsch*

(CIS ID 100430,3660)