ONE

Introduction to Delphi programming

During this course you will be learning to program in the computer language DELPHI. This is a Windows-based general purpose language which can carry out calculations, display and draw graphics, and save data on disc; it is a good language to use for a wide variety of computing projects.

Load the Delphi editing screen by clicking on the Delphi icon:

The main sections of the screen display are shown on page 2.

Drop down menus
By clicking on the words at the top line of the screen, you have access to a wide range of programming facilities.

Short cut buttons
The group of buttons in the top left corner of the screen provide a quick way of carrying out some of the more common programming tasks in Delphi. If you move the mouse pointer to any of the buttons and leave it stationary for a couple of seconds, a yellow help label will appear. Don't use a short cut button unless you understand its purpose!

Screen form window
This grey dotted grid is a representation of the program screen. You can use the grid to position the text, pictures and other components which will be displayed when your program is running.

Component selection
This menu provides the large toolkit of components which will be used to construct your programs.

Program unit window
The program unit window can be displayed by clicking the mouse on the *Toggle Form/Unit* short cut button:

Toggle
Form/Unit
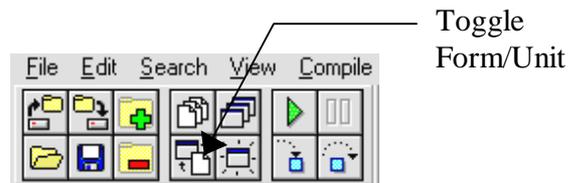
As a Delphi project is being developed , a program is written in the Unit window - partly by the computer and partly by the programmer.

component
selection

drop down
menus

screen form
window

short cut
buttons

Delphi - Project1

File Edit Search View Compile Run Options Tools Help

Standard / Additional / Data Access / Data Controls / Dialogs / System / VBX / Samples /

Object Inspec...

Form1: TForm1

ActiveControl	
AutoScroll	True
+BorderIcons	[biSystemMenu,
BorderStyle	bsSizeable
Caption	Form1
ClientHeight	273
ClientWidth	427
Color	clBtnFace
Ctl3D	True
Cursor	crDefault
Enabled	True
+Font	(TFont)
FormStyle	fsNormal
Height	300
HelpContext	0

Properties / Events /

Form1

UNIT1.PAS

```
unit Unit1;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes
  Forms, Dialogs;

type
  TForm1 = class(TForm)
  private
    { Private declarations }
  public
    { Public declarations }
```

1: 1   Modified   Insert

Unit1 /

Page 1   Sec 1   1/1   At 1"   Ln 1   Col 1   12:27 REC MRK EXT OVR WPH

Start | Dr Sol... | TextBri... | Micros... | Paint S... | Delphi... | Object... | UNIT... | Form1 | 12:27

object
inspector

program unit
window

Writing a Delphi program

Make sure that the *object inspector* window is visible on the screen. If not, you can make it appear by going to the VIEW drop down menu and clicking OBJECT INSPECTOR.

Scroll down the *properties* list to find WINDOW STATE. Click in the right column and use the small arrow to display a drop down menu. Select MAXIMIZED.
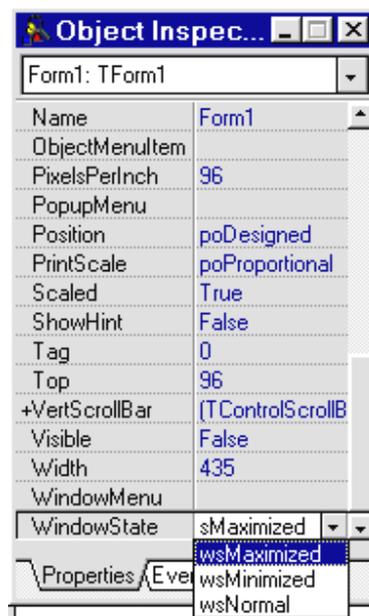
We are now ready to run our first Delphi program, although it will not do very much yet. Click on the *run* short cut button:
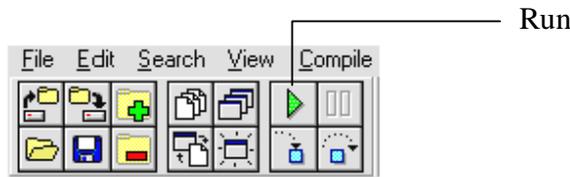
The program will produce a blank grey screen. To stop the program and return to the Delphi editing screen, click the cross in the top right corner:
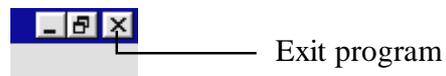
Saving the program on disc

Before going any further it would be a good idea to save the program on disc - always save your work at regular intervals in case anything goes wrong. A Delphi programming project consists of a number of separate files, so it is simplest to create a new directory for each project.

Use the START button in the bottom left of the screen to select WINDOWS EXPLORER. Create a NEW FOLDER in your work directory, and call this PROG1. Close the Windows Explorer screen.

Select the FILE drop down menu on the Delphi editing screen, and click on SAVE PROJECT. A file window will appear.
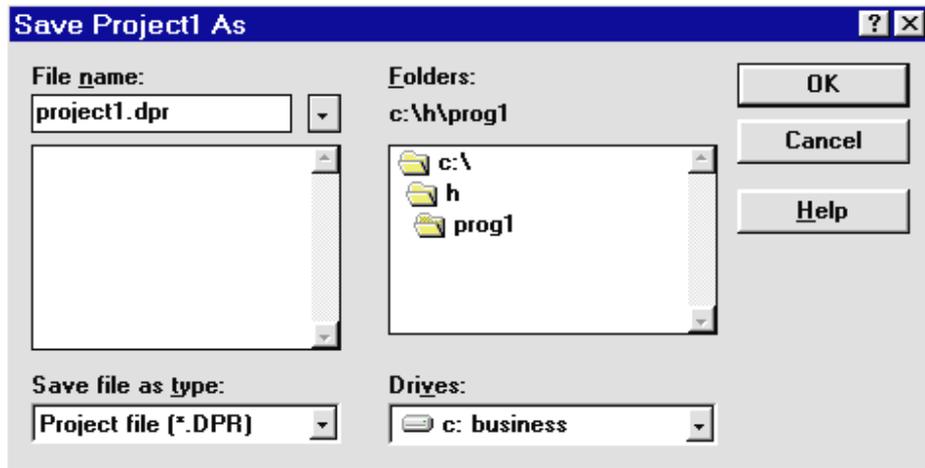
Change to the sub-directory PROG1 which you just created, and click '**OK**' to accept the file names *unit1.pas* and *project1.dpr* which are offered by the system.

Using components

Click on the grey dotted grid of the Form Window to bring this to the front of the editing screen. Using the mouse, drag this window larger so that the grey grid covers most of the screen.

Select the *button* component.

Move onto the grid and hold down the left mouse button to create the button component. Drag this near to the top right of the grid.
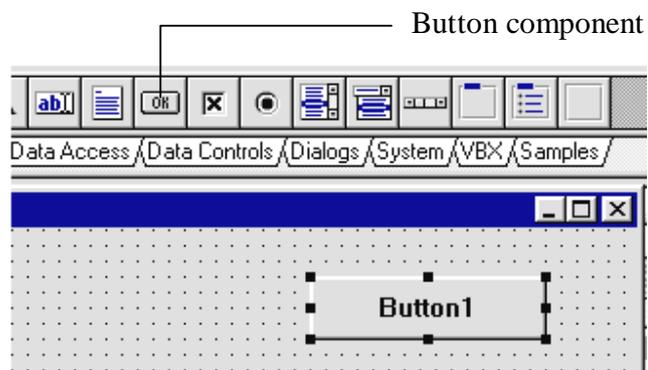
Press the ENTER key to display the object inspector for the button component, and change the *caption* entry so that the word 'Exit' appears on the button.
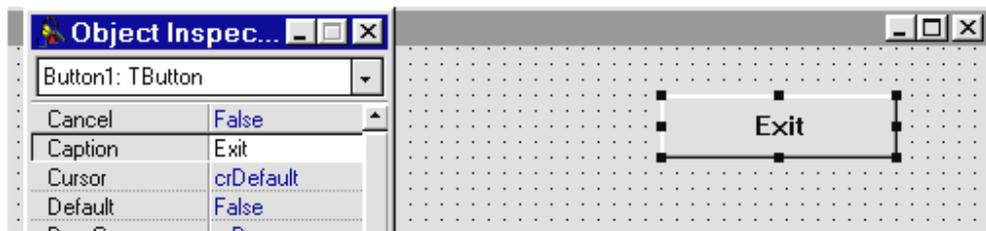
We are going to use this button as a more convenient way of stopping the program.

Quickly double-click the mouse on the Exit button, and the Program Unit screen will open:
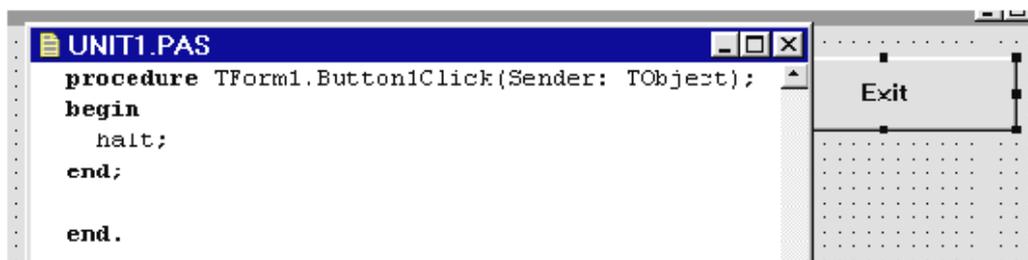
The computer has created a section of program which will be activated when the Exit button is pressed. This is known as an *event handling procedure*, and has been given the name *Button1.Click*.

To make the program end when the button is clicked, it is necessary to type the instruction

halt;

between the lines **begin** and **end**. Don't forget the semi-colon after the word **halt**.

If you enlarge and scroll the Unit Window, you will find that the computer has written quite a lot of program already. This need not bother us at this stage - you will learn the purpose of all these lines of program as the course progresses.

We are now ready to test the modified program. Before running it this time, go to the COMPILE drop down menu and click on COMPILE. The computer will carry out a check of the program and give an error message if there is any error. Once the program has compiled successfuly, click the RUN shortcut button.

The grey screen appears again, but this time clicking the Exit button will take you back to the Delphi editing screen.

Displaying a bitmap image

Photographs and clip art are often stored on disc as *bitmap images* - these have file names with the letters BMP at the end. Bitmap images can easily be displayed in Delphi programs.

Click the ADDITIONAL tab on the component menu, and select the *image* component :

image

Move onto the grid and drag the mouse to produce an image area. Double-click the mouse in the image area to open the *picture editor* window.

Press the LOAD button. A file CAMERA.BMP has been saved for you on disc. Locate the directory, select the file and press OK. A preview picture will appear:
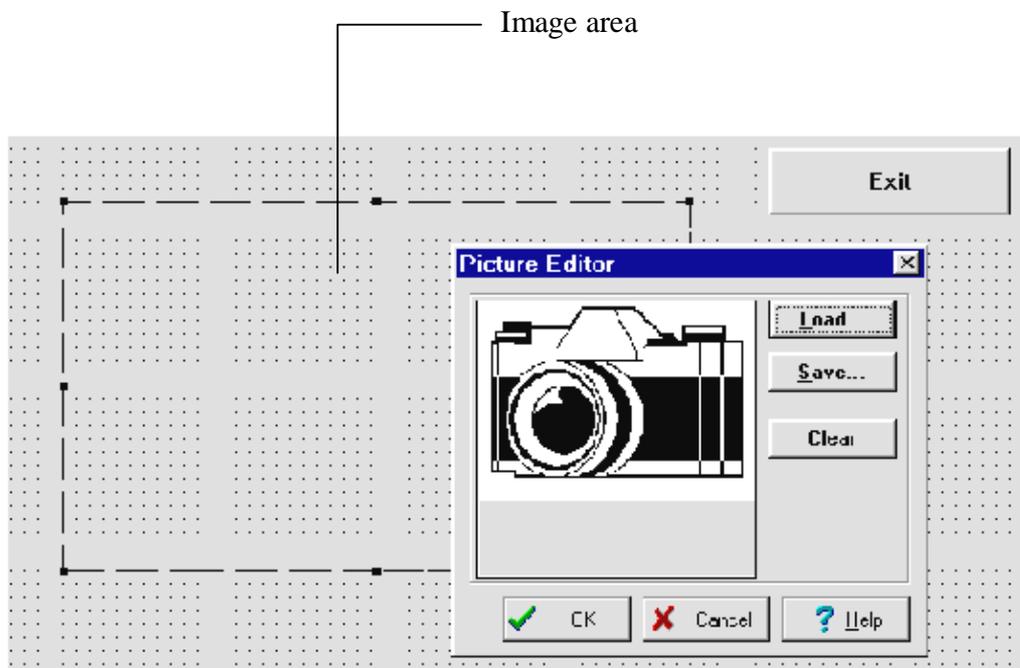
Image area

Click OK and the picture will be loaded into the image area.

Initially the picture is fixed in size, but this can be made adjustable. Press ENTER to bring up the *object inspector* window for the image box, then set the STRETCH property to TRUE.
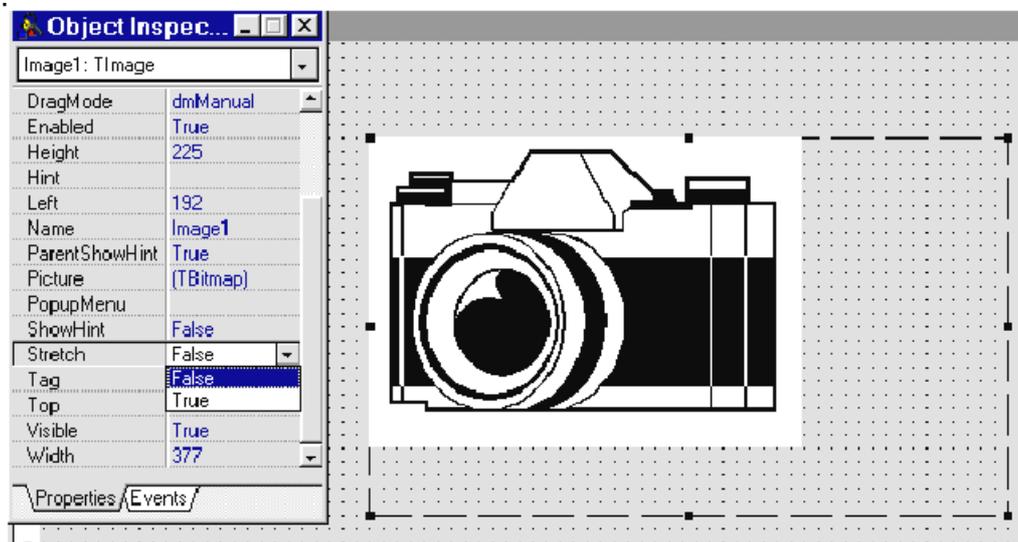
Once this is done, the picture will expand to fill the image area, and can be adjusted by dragging with the mouse.

Compile and run the program to check that the picture is displayed.

Finally, let's add buttons to the screen to allow the picture to be turned on and off while the program is running. Select the button component, and add two more buttons to the form. Label these 'Show picture' and 'Hide picture':
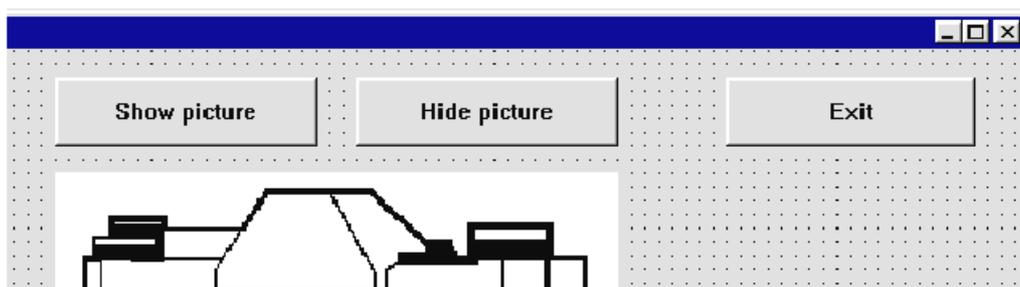
Double-click the mouse on the 'Show picture' button to create an event handler. Add the command:

```
image1.visible:=true;
```
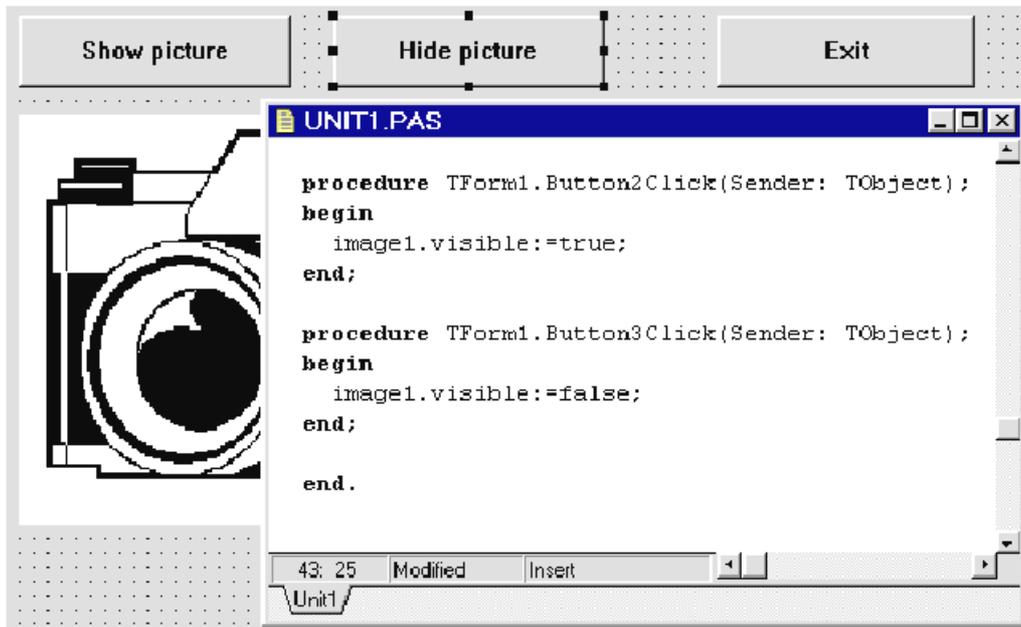
Double-click on the 'Hide picture' button, and add the command:
image1.visible:=false;

The program is refering to the picture as *image1*. It can be turned on and off by clicking the buttons, which set its *visible* property to be true or false. Compile and run the program to try this.

Save the completed project by clicking on the *save project* short cut button:

save
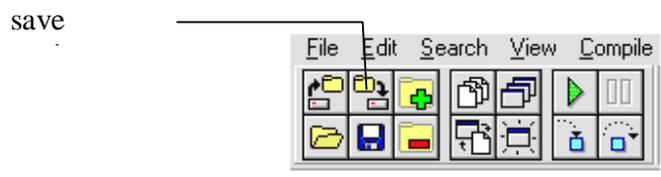
SUMMARY
In this section you have:
- loaded the Delphi editing screen and identified the sections of the screen display
- started a new project
- maximised the program window to fill the screen when it runs
- created a sub-directory in which the project can be saved
- placed a button on the form grid, and set up an event handling procedure to respond to the button being clicked
- placed an image box on the form grid, and loaded a bitmap image
- set the image *stretch* property to allow size adjustment
- added commands to event handler procedures to show and hide the image, and exit from the program

8