

**: Drag and Drop :**

**: Sender 1**

Event

Sender

. TObject Sender

procedure TForm1.Button1Click(Sender: TObject);

Button1

. Button1

:

: Button

:

Button1

1

If sender = Button1 Then Showmessage ( ' You Click On Button1 ');

If sender = Button2 Then Showmessage ( ' You Click on Button2 ');

OnClick

Event

Button2

2

. Button1Click

:

2

: ListBox

Label

: 1

Listbox

Label

Label

Label

Listbox1

. dmAutomatic

DragMode

:

Listbox1

OnDragOver

Accept := Source is Tlabel;

:

Listbox1DragOver

. Listbox1

: Sender

.

Label

: Source

: X,Y

: State

)

True

: Accept

(

```

        . TLabel          Source          True          : Source Is TLabel
        :
        OnDragDrop
( sender as Tlistbox ).color := (Source as TLabel ).color ;
    Tlistbox          Sender          Sender As TlistBox
.
        . Source As TLabel
        ListBox1          Label
        : 2
        Picture          Image
Image1          dmAutomatic          DragMode
.
        :          Form1          OnDragOver
accept := Source is Timage;
.
        :          OnDragDrop
(Source as Timage).Top := y;
(Source as Timage).Left := X;
        Image1
        .ListBox

```

```

:
TImage      TForm      Canvas
.
:
x,y      Form1.Canvas.MoveTo(x,y)
. Form1      OnMouseDown
x,y      Form1.Canvas.LineTo(x,y)
. Form1      OnMouseUp
.
OnMouseMove
:
Form1.Canvas.LineTo(x,y);
:
:      Var
var
Form1: TForm1; Drawing : boolean = false;
:      OnMouseDown
Drawing := True;
Canvas.MoveTo(x,y);
:      OnMouseMove
if Drawing Then Form1.Canvas.LineTo(x,y);
:      OnMouseUp
Drawing := false;
.
OnMouseMove
.
Integer      X,y      TPoint
:      private
Origin : TPoint;

```

```

: OnMouseDown
Drawing := True; Canvas.MoveTo(x,y);
Origin := point(x,y);

```

```

: OnMouseMove
If Drawing then Begin
Canvas.MoveTo(origin.X,origin.y) ; Form1.Canvas.LineTo(x,y);
end;

```

```

: OnMouseUp
Canvas.MoveTo(origin.X,origin.Y);
canvas.LineTo(x,y);
Drawing := false;

```

```

OnMouseMove OnMouseUp
OnMouseMove
:
Tpoint Movept Private OnMoseMove
.
Origin,Movept :Tpoint;
: OnMouseDown
Movept := Point(x,y);
: OnMouseMove
If Drawing then begin
canvas.Pen.Mode := pmNotXor;
canvas.MoveTo(origin.X,origin.y) ; } (1)
Canvas.LineTo(Movept.x,movept.y); }
Canvas.moveto(origin.X,origin.y) ; } (2)
Canvas.LineTo(x,y);
end;
movept := point(x,y);
Canvas.Pen.Mode := pmCopy;
end;
pmNotXor pen Mode
PmCopy

```

http://www.alshater.net

لغة البرمجة دلفي

)

1

)

( Movept

(

pmNotXor

mode

2

OnMouseMove

(

)

:

```

If Drawing then begin
  canvas.Pen.Color := form1.Color ;
  canvas.MoveTo(origin.X,origin.y) ;
  Canvas.LineTo(Movept.x,movept.y);
  canvas.Pen.Color := clblack;
  canvas.moveto(origin.X,origin.y) ;
  Form1.Canvas.LineTo(x,y);
end;
movept := point(x,y);

```

...

:

:

New Button

Toolbar

. ( Painting

Gragh

)

:

Image Editor

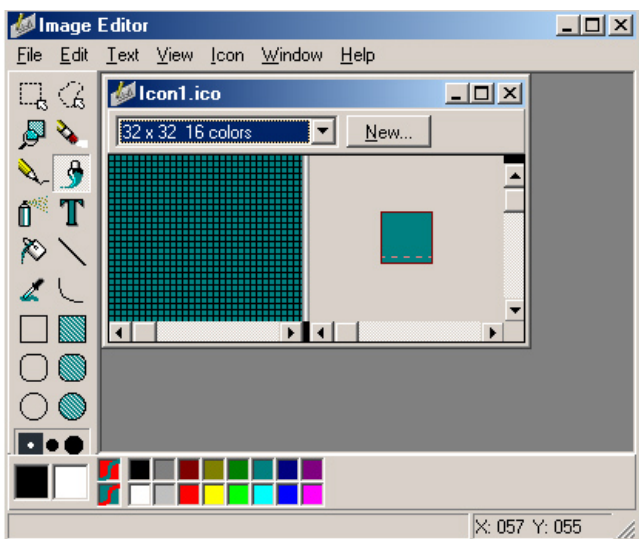
Tools

:

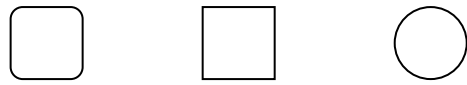
File → New → icon File

16 Color Small Icon

Icon1.ico



Line.ico



Add

Win32

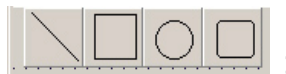
ImageList

ImageIndex

Images

Imagelist1

Toolbar1



. True

Flat

ToolBars

True

Grouped

tbsCheck

Style

Type

: Gragh

**type**

**TDrawingTool = (dtLine, dtRectangle, dtEllipse, dtRoundRect);**

TForm1 = class(TForm)

:

:

**const**

dtLine = 0;

dtRectangle = 1;

dtEllipse = 2;

dtRoundRect = 3;

: Private

DrawingTool: TDrawingTool;

:

DrawingTool := dtLine;

:

DrawingTool := dtRectangle;

:

<http://www.alshater.net>

```
DrawingTool := dtEllipse;
```

```
:
```

```
DrawingTool := dtRoundRect;
```

```
DrawingTool
```

```
OnMouseUp
```

```
DrawingTool
```

```
case DrawingTool of
```

```
dtLine:
```

```
begin
```

```
Canvas.MoveTo(Origin.X, Origin.Y);
```

```
Canvas.LineTo(X, Y)
```

```
end;
```

```
dtRectangle: Canvas.Rectangle(Origin.X, Origin.Y, X, Y);
```

```
dtEllipse: Canvas.Ellipse(Origin.X, Origin.Y, X, Y);
```

```
dtRoundRect: Canvas.RoundRect(Origin.X, Origin.Y, X, Y,  
(Origin.X - X) div 2, (Origin.Y - Y) div 2);
```

```
end;
```

```
Drawing := False;
```

```
End;
```

```
OnMuseMove
```

```
:
```

```
if Drawing then
```

```
begin
```

```
Canvas.Pen.Mode := pmNotXor;
```

```
case DrawingTool of
```

```
dtLine: begin
```

```
Canvas.MoveTo(Origin.X, Origin.Y);
```

```
Canvas.LineTo(MovePt.X, MovePt.Y);
```

```
Canvas.MoveTo(Origin.X, Origin.Y);
```

```
Canvas.LineTo(X, Y);
```

```
end;
```

```
dtRectangle: begin
```

```
Canvas.Rectangle(Origin.X, Origin.Y, MovePt.X, MovePt.Y);
```

```
Canvas.Rectangle(Origin.X, Origin.Y, X, Y);
```

```
end;
```

```
dtEllipse: begin
```

```
Canvas.Ellipse(Origin.X, Origin.Y, X, Y);
```

```
Canvas.Ellipse(Origin.X, Origin.Y, X, Y);
```

```
end;
```

```
dtRoundRect: begin
```

```
Canvas.RoundRect(Origin.X, Origin.Y, X, Y,
```

```
(Origin.X - X) div 2, (Origin.Y - Y) div 2);
```

```
Canvas.RoundRect(Origin.X, Origin.Y, X, Y,
```

```

http://www.alshater.net
(Origin.X - X) div 2, (Origin.Y - Y) div 2);
end;
end;
end;
MovePt := Point(X, Y);
Canvas.Pen.Mode := pmCopy;
end;

```

```

:
: ( Rectangle )

```

```

: ( Ellipse)

```

```

: ( RoundRect )

```

```

)

```

```

(

```

```

:

```

```

...

```

```

:

```

```

. False Visible Win32 Toolbar

```

```

. New Button

```

```

Toolbar2 ImageList1

```

```

)

```

C:\Program Files\Borland\Delphi6\Demos\Doc\Graphex

```

:

```

solid.bmp - dashed.bmp - dotted.bmp - dashdot.bmp - dashdot2.bmp - clear.bmp

```

:

```



```

tbsCheck Style True Grouped

```

```

New Separator toolbar1

```

```

: ImageList1

```



```

tbsCheck      Style
:
toolbar2.Visible := not toolbar2.Visible ;
.
: Toolbar2
canvas.Pen.Style := pssolid;
:
canvas.Pen.Style := psDash;
:
canvas.Pen.Style := psdot;
:
canvas.Pen.Style := psdashdot ;
:
canvas.Pen.Style := psdotdot ;
:
canvas.Pen.Style := psclear ;
:
: Tpen
. ColorDialog
( Separator )   Toolbar2
:
if ColorDialog1.Execute then
  Canvas.Pen.Color := ColorDialog1.Color;
. ( colors.bmp )
:
:
. ToolBar2      Edit
: Associate    Edit1      Win32      UpDown
: Edit1        OnChange
canvas.Pen.Width := strtoint(edit1.Text );
:
:
```

:

tbsCheck                      Style                      Toolbar1

:

Toolbar3.Visible := Not Toolbar3.Visible ;

Grouped

:

tbsCheck                      Style                      True



:

canvas.Brush.Style := bssolid;

:

canvas.Brush.Style := bsclear;

:

canvas.Brush.Style := bshorizontal;

:

Style

bsVertical – bsFdiagonal – bsbdiaognal – bsColor – bsDiagCross

:

: ToolBar3

If ColorDialog1.Excetue then canvas.Brush.Color := ColorDialog1.Color ;

:

Status bar

:

. Win32                      StatusBar

Panel Editor                      2

Add New

. 150                      Width                      0                      3

:                      OnMouseDown                      4

StatusBar1.Panels[0].Text := Format ('origin = %d , %d',[x,y]);

:                      OnMouseMove                      5

StatusBar1.Panels[1].Text := Format ('Point = %d , %d',[x,y]);