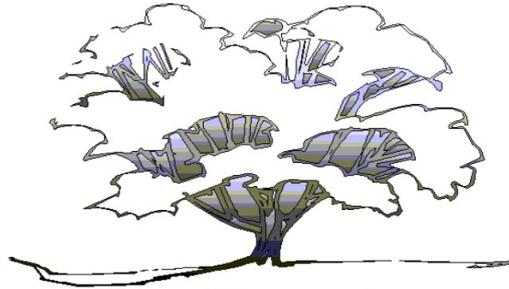


April 18 1995



# Jacaranda Designs

P.O. Box 4232  
KALGOORLIE. 6430.  
Western Australia.

BBS: The MAGICal C Emporium  
61-90-217533 28.8k 8N1  
Compuserve : Mark Bailey - 100026,1750

Hi Delphi developers.

Here is a small component (our first) , born out of a need for a real application that after writing we thought others may see some value in. It is derived from the PaintBox component and provides the developer with a mesh that overlays the PaintBox. Hence, it is called MeshedPaintBox. And for only \$25 AUS it may save you many hours work.

1. What does it do?
2. Ordering Information
3. Installation
4. How do I use it.

## **1. What does it do?**

The MeshedPaintBox is a component will allow you to add a Mesh to a PaintBox with full control over the Mesh. You may have a Mesh that consists of Lines, Dots or Crosses, in any colour you like. In addition to this you are able to control the spacing of the horizontal and vertical spacing of the Mesh as well as the inclination of the Mesh both horizontally and vertically. Also included is a new type called TFPoint which is identical to TPoint except that the X and Y values are of a Single type instead of Integer. This is used when retrieving mesh coordinates. Any application that maintains objects at set spacings needs to store the coordinates as floating point values to preserve resolution. If rounding occurs when resizing the layout of the objects the spacing with get out of whack and you will not be able to resize back to the original values, if you get my meaning.

## **2. Ordering Information**

The MeshedPaintBox component can be ordered for the measly sum of \$25AUS (this is only about \$19 US) by sending a Bank Cheque or Money Order in AUS\$ to,

Jacaranda Designs  
P.O. 4232  
KALGOORLIE. 6430.  
Western Australia.

For your effort you will get the unlimited MeshedPaintBox component including the source code and detailed explanation of how it works.

By unlimited we mean that there will be no limitations on the registered version. **The demo version included here will only allow you to have a MeshedPaintBox 200x200 pixels.**

You will also be able to download the latest version from our BBS, The MAGICal C Emporium

### 3. Installation

MESHBOX.EXE is a self extracting archive. Copy this file into a new directory on your hard disk then type MESHBOX and press Enter.

The file will decompress to the following files.

MESHBOX.EXE	- The original self extracting file.
MESH.DCU	- The demo component MeshedPaintBox
MESH.DCR	- The resource file for the MeshedPaintBox. Contains the BMP for the palette button.
MESHDEMO.EXE	- Demo Program showing MeshedPaintBox
README.WRI	- This file.

Now copy Mesh.DCU and Mesh.DCR to your Delphi\Lib directory or even your root Delphi directory will do. See warning below.

Firstly it is best to try out MESHDEMO.EXE. Start in Windows in the usual way. ie File/New/Item etc and point to this file. The icon is the default Delphi icon.

Before you can try out the MeshedPaintBox yourself, you must install the component into the Delphi palette and rebuild the component library. From within Delphi, select "Options|Install components" from the main menu, select "Add" from the available buttons at the right of the dialog, enter MESH, and select the OK button. Before closing the "Install components" dialog, enter the path name where you extracted the MESHBOX files into the "Search path" edit control (it is usually easier to use the Browse to select the file to install). Select OK to have Delphi compile and add the MeshedPaintBox component to the component library. Note: It will be installed to a palette page called MyStuff.

**Warning!** I have found it best to put new components to be installed in the Delphi\Lib directory. It seems the path has limited length and if you already have a fair few components installed then the new path may be truncated. If you get the message " could not find file Mesh.DCU" then this is the problem. This may be the hard way so any tips here might be helpful to all.

### 4. How do I use it?



After installing the component on the Delphi component palette you will see this icon in the MyStuff page of the palette.

Click the icon and then click on your form. You will then see a normal PaintBox type area added to your form. In the object inspector you will see some extra parameters which you can play around with as follows.

- MeshEnabled - When True the Mesh is displayed. When False it isn't.
- MeshPen - This lets you change the Pen for the lined mesh. Change the usual Pen parameters, namely, Color, Mode, Style and Width.
- MeshType - The MeshType can be mtLine, mtDot, or mtCross.
- XSpacing, YSpacing - Pretty self explanatory. Changes the spacing in the X or Y direction.
- XOffset, YOffset - Controls the angle of the Mesh in the X or Y direction.

As you change any of these parameters you will see the effect immediately on the MeshedPaintBox. Apart from these changes the PaintBox behaves in the same way as a normal PaintBox. The demo will show all of the above in action.

You can also find the closest mesh intersection by supplying the MeshedPaintBox a TPoint as follows,

```
var
```

```
Intersection : TFPPoint;
```

```
MeshedPaintBox1.GivenCoords := Point(X,Y);
```

```
Intersection := MeshedPaintBox1.MeshCoords;
```

Intersection will now contain the coordinates of the closest intersection on the mesh to the original X,Y supplied to GivenCoords. Try this out in the demo by clicking on the mesh. A filled dot will appear on the mesh at the nearest intersection.