

Images

(Information and examples taken from the DroidScript documentation)

Description

Create Image controls using the **CreateImage** method of the **app** object:

```
img = app.CreateImage( file, width, height, options );
```

Use the **SetOnTouch** method of the Image object to set the name of a function you want to be called when the Image is touched.

When Image controls are touched, they send an **event object** parameter to your callback function which contains details of the touch event, for example the **action** property of the event object contains **“Down”**, **“Up”** or **“Move”** as the user touches and moves their finger on the screen and the x and y properties contain arrays of touch **coordinates**.

If you don't set a size, the image object will match the original image size. If you set one dimension to a positive value and leave the other dimension as -1, then the image will maintain its original **aspect ratio**.

Specifying both width and height will **stretch** the image to fill the Image object, unless you can use the **“ScaleCenter”** option to keep the image at it's original size and centered within the Image object

Drawing On Images

You can use an image control like a **Canvas** by calling it's drawing methods, such DrawRectangle, DrawCircle, DrawImage etc.

It's possible to draw over a loaded image or you can start with a blank image by passing the value 'null' to the CreateImage method instead of a filename. You can set the background color of your blank image using the 'SetColor' method.

Methods

Some controls use the same methods.
For examples of the **same methods** look here.

Method	Description
Image.Clear()	

Method	Description
Image.Destroy()	Destroy image object. Release memory
Image.Draw(func, p1, p2, p3, p4, p5, p6, p7)	
Image.DrawArc(x1, y1, x2, y2, start, sweep)	
Image.DrawCircle(x,y,radius)	
Image.DrawImage(image, x, y, w, h, angle)	
Image.DrawImageMtx(image, matrix)	
Image.DrawLine(x1, y1, x2, y2)	
Image.DrawPoint(x, y)	Draw a point on the image canvas.
Image.DrawRectangle(x1, y1, x2, y2)	
Image.DrawText(txt, x, y)	Draw text on x,y position inside the image canvas.
Image.GetAbsHeight()	Get absolute pixel height size of the image.
Image.GetAbsWidth()	Get absolute pixel width size of the image.
Image.GetHeight()	
Image.GetName()	
Image.GetPixelData(format, left, top, width, height)	format can be "rawbase64", "rawbase64,grayscale" "pngbase64", "jpgbase64" or json (3x slower than others). Left,Top start pixel position. Width,Height size of pixel output.
Image.GetPosition()	returns an object with properties: left, top, width, height
Image.GetType()	
Image.GetVisibility()	
Image.GetWidth()	
Image.Move(x, y)	Move of a canvas
Image.Release()	Release object memory.
Image.Reset()	
Image.Rotate(angle, pivotX, pivotY)	Rotate of a canvas
Image.Save (fileName,quality)	quality parameter new since Vers 1.29 support for png files since Vers 1.29
Image.Scale(x, y)	Set zoom of a canvas
Image.SetAlpha(alpha)	Transparency of a canvas
Image.SetAutoUpdate(onoff)	Turn On/Off the update on action on image manipulation. By turning off update, call to Image.Update() will update the image all commands done on the image.
Image.SetBackColor(colorCode)	Set background color
Image.SetBackGradient(color1,color2,color3,p4,p5,p6,p7)	
Image.SetBackGradientRadial(x,y,r,color1,color2,color3,p7)	
Image.SetBackground(imagefile,options)	
Image.SetColor(color)	
Image.SetFontFile(file)	Set the font type to be draw on the image.
Image.SetImage(image,width,height,options)	

Method	Description
Image.SetLineWidth(width)	
Image.SetMargins(left,top,right,bottom)	
Image.SetMaxRate(ms)	Set the minimum amount of time (in ms) between onTouchMove events
Image.SetName(p1)	Set name canvas
Image.SetOnLoad(callback)	
Image.SetOnLongTouch(callback)	
Image.SetOnTouch(callback)	
Image.SetOnTouchDown(callback)	
Image.SetOnTouchMove(callback)	
Image.SetOnTouchUp(callback)	
Image.SetPadding(left, top, right, bottom)	
Image.SetPaintColor(color)	Set the color on draw action to the image.
Image.SetPaintStyle(style)	Paint style(line,fill)
Image.SetPosition(left, top, width, height)	
Image.SetScale(x,y)	Fract values (as usual): 1=original, -1=flip (mirror)
Image.SetSize(width, height)	
Image.SetTextSize(size)	
Image.SetTouchable(callback)	
Image.SetVisibility(HideShow)	
Image.Skew(p1,p2)	
Image.Transform(matrix)	
Image.Update()	
Image.Update2()	

Options

Button	Causes the image to depress like a button when touched
async	Loads the image from file in the background and allows your code to continue without waiting for it to load
FontAwesome	Use this option to write icons as Text on your image (see the sample at the bottom)
Resize	The Resize option internally scales down the original image to the display size, so it uses up less memory than the full size image (useful if you are displaying lots of thumbnail images)
ScaleCenter	you can use the option to keep the image at it's original size and centered within the Image object
Button	Causes image to depress like a button when touched

Example - Original Size

```
function OnStart()
{
```

```
lay = app.CreateLayout( "Linear", "VCenter,FillXY" );

img = app.CreateImage( "/Sys/Img/Droid1.png" );
img.SetOnTouch( img_OnTouch );
lay.AddChild( img );

app.AddLayout( lay );
}

function img_OnTouch( ev )
{
    if( ev.action=="Down" )
        app.ShowPopup( "Ouch!" );
}
```

Example - Stretched

```
function OnStart()
{
    lay = app.CreateLayout( "Linear", "VCenter,FillXY" );

    img = app.CreateImage( "/Sys/Img/Droid1.png", 0.5, 0.7 );
    img.SetOnTouch( img_OnTouch );
    lay.AddChild( img );

    app.AddLayout( lay );
}

function img_OnTouch( ev )
{
    if( ev.action=="Down" )
        app.ShowPopup( "Aaaargh!" );
}
```

Example - Maintain Aspect

```
function OnStart()
{
    lay = app.CreateLayout( "Linear", "VCenter,FillXY" );

    img = app.CreateImage( "/Sys/Img/Droid1.png", 0.5, -1 );
    img.SetOnTouch( img_OnTouch );
    lay.AddChild( img );

    app.AddLayout( lay );
}

function img_OnTouch( ev )
```

```
{
  if( ev.action=="Down" )
    app.ShowPopup( ev.x[0] + ", " + ev.y[0], "Short" );
}
```

Example - Draw Shapes

```
function OnStart()
{
  lay = app.CreateLayout( "Linear", "VCenter,FillXY" );

  img = app.CreateImage( null, 0.8, 0.8 );
  lay.AddChild( img );

  img.SetColor( "#8888ff" );
  img.SetPaintColor( "#ff0000" );
  img.DrawCircle( 0.5, 0.5, 0.1 );
  img.DrawRectangle( 0.7, 0.7, 0.05 );

  app.AddLayout( lay );
}
```

Example - FontAwesome

Some say there should always be **Hello World** code.

[helloimage.js](#)

```
function OnStart()
{
  lay = app.CreateLayout( "Linear", "VCenter,FillXY" );

  img = app.CreateImage( null, 0.8, 0.8, "FontAwesome" );
  lay.AddChild( img );

  img.SetColor( "#99ff99" );
  img.SetPaintColor( "#0000dd" );
  img.SetTextSize(42);
  img.DrawText( "[fa-globe] Hello World!", 0.3, 0.5 );

  app.AddLayout( lay );
}
```

From:

<https://wiki.droidsript.me.uk/> - **DroidScript wiki**

Permanent link:

https://wiki.droidsript.me.uk/doku.php?id=built_in:images

Last update: **2018/04/19 15:23**

