

Circle crop images in FileMaker with MBS Plugin

Once again a client asked for some help with image editing in FileMaker.

An imported picture should be scaled to a given size and than be cropped to a round image. Of course there are no round images, just images with alpha channels to give the illusion the image may be round.



We created the following example script to first read in the image from a container field. We create a temporary image with transparent background of the source size and composite the existing image into that. Next we scale it to the requested size.

Next we create a new image where we draw the circle for the mask and we apply this mask to the scaled down image. Next we compose this image into our final image to apply the mask and write it to a container as PNG image file:

```
# Read in source image
Set Variable [ $SourceImage ; Value: MBS( "GMImage.NewFromContainer"; Clip Image::Input ) ]
Set Variable [ $sourceWidth ; Value: MBS( "GMImage.GetWidth"; $SourceImage ) ]
Set Variable [ $sourceHeight ; Value: MBS( "GMImage.GetHeight"; $SourceImage ) ]
#
# Scale to dest size
Set Variable [ $sourceSize ; Value: If($sourceWidth > $sourceHeight; $sourceHeight; $sourceWidth) ]
Set Variable [ $destWidth ; Value: GetAsNumber ( Clip Image::Width ) ]
Set Variable [ $destHeight ; Value: GetAsNumber ( Clip Image::Height ) ]
Set Variable [ $destSize ; Value: If($destWidth > $destHeight; $destHeight; $destWidth) ]
Set Variable [ $templImage ; Value: MBS( "GMImage.New"; $sourceSize & "x" & $sourceSize; "transparent" ) ]
Set Variable [ $r ; Value: MBS( "GMImage.CompositeXY"; $templImage; $sourceImage; -($sourceWidth-$sourceSize)/2; -($sourceHeight-$sourceSize)/2; 1) ]
Set Variable [ $r ; Value: MBS( "GMImage.Scale"; $templImage; $destSize & "x" & $destSize ) ]
#
# Build mask
Set Variable [ $cliImage ; Value: MBS( "GMImage.New"; $destSize & "x" & $destSize; "transparent" ) ]
Set Variable [ $r ; Value: MBS( "GMImage.SetFillColor"; $cliImage; "black" ) ]
Set Variable [ $radius ; Value: $destSize/2 ]
Set Variable [ $r ; Value: MBS( "GMImage.DrawCircle"; $cliImage; $radius; $radius; 0; $radius ) ]
#
```

```
# Apply mask
Set Variable [ $r ; Value: MBS( "GMImage.CopyChannel"; $templImage; "Opacity";
$cliplImage; "Opacity") ]
#
# Create destination image
Set Variable [ $destImage ; Value: MBS( "GMImage.New"; $destWidth & "x" &
$destHeight; "transparent") ]
Set Variable [ $r ; Value: MBS( "GMImage.CompositeXY"; $destImage; $templImage;
($destWidth-$destSize)/2; ($destHeight-$destSize)/2; 1) ]
Set Field [ Clip Image::Output ; MBS( "GMImage.WriteToPNGContainer"; $destImage) ]
#
# Memory cleanup
Set Variable [ $r ; Value: MBS( "GMImage.Destroy"; $SourceImage) ]
Set Variable [ $r ; Value: MBS( "GMImage.Destroy"; $templImage) ]
Set Variable [ $r ; Value: MBS( "GMImage.Destroy"; $destImage) ]
Set Variable [ $r ; Value: MBS( "GMImage.Destroy"; $cliplImage) ]
```

As you see this is a multi step approach and looks complicated, but sometimes you need to prepare an extra image for the mask (or alpha channel) you like and compose things together as needed. You can extend this and draw some text on top, some border around or watermark it with a logo.

PS: Looks like [MBS FileMaker Plugin](#) has more [GraphicsMagick](#) functions than FileMaker has built-in functions for calculations.