
Subject: StaticImage enhancement
Posted by [Didier](#) on Tue, 16 Apr 2013 21:15:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi all,

I noticed that Static Image displayed a custom image so I looked at how it was done and managed to display my own image and setting the image through the layout designer (and not directly in the '.usc' file)

I was wondering if there was a way to use the img pathname set in the layout designer to directly set the image at runtime:

The image path names look like this: Controls4U:Controls4U.iml:ImageSample

Controls4U:Controls4U.iml:ImageSample
Controls4U::Controls4U.iml::ImageSample (this also works)

This would be great !!
All would get displayed and set by the layout designer !!

Subject: Re: StaticImage enhancement
Posted by [koldo](#) on Wed, 17 Apr 2013 06:31:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Didier

usc language has some limitations and AFAIK one of them is that it is not possible to open an image with its file name.

This is a subject I have not insisted enough to Mirek, but I would like to add more functions to usc. For example, the circles, ellipses and related trigonometry in Controls4U are made by hand inside usc code.

Subject: Re: StaticImage enhancement
Posted by [Didier](#) on Wed, 17 Apr 2013 21:49:37 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Koldo,

USC language may not be able to open images by they're file name but it can open images from iml files (which I think is sufficient)

The only problem is the code generated in the '.lay' file ==> it doesn't compile

I think converting it to a character string would be enough (this type of processing is already done for Text and Label) so no big deal probably.

The main difficulty is how to select the iml image using this character string ???

Here is the modified StaticImage Usc:

```
ctrl ModifiedStaticImage {
  group "TEST";

  GetMinSize() { return Size(0, 0); }
  GetStdSize() { return Size(64, 24); }

  Frame    SetFrame @1;
  ImageFit SetFit;
  ImageAngle SetAngle;
  Color    SetBackground;
  bool    UseAsBackground = false;
  Image    SetImage;          // ----- NEW PROPERTY the type is completely arbitrary -----

  Paint(w) {
    r = GetRect();

    DrawCtrlFrame(w, r, .SetFrame);

    sz = Size(r.right - r.left, r.bottom - r.top);

    DeflateRect(r);
    sz = Size(r.right - r.left, r.bottom - r.top);
    w.DrawRect(r.left, r.top, sz.cx, sz.cy, .SetBackground);

    img = .SetImage;  ----- the iml pathName is retrieved from the property -----

    if (.SetFit == "0") {
      imagesize = GetImageSize(img);
      rectaspect = sz.cx/sz.cy;
      imageaspect = imagesize.cx/imagesize.cy;
      if (rectaspect > imageaspect)
        w.DrawImage(r.left+(sz.cx-imageaspect*sz.cy)/2, r.top, imageaspect*sz.cy, sz.cy, img);
      else
        w.DrawImage(r.left, r.top+(sz.cy-sz.cx/imageaspect)/2, sz.cx, sz.cx/imageaspect, img);
    } else if (.SetFit == "1")
      w.DrawImage(r.left, r.top, sz.cx, sz.cy, img);
    else if (.SetFit == "2")
      w.DrawImage(r.left, r.top, img);
    else if (.SetFit == "3") {
      imagesize = GetImageSize(img);
```

```

top = r.top;
for (left = r.left; left < r.right; left += imagesize.cx)
  for (top = r.top; top < r.bottom; top += imagesize.cy)
    w.DrawImage(left, top, img);
}
if (.UseAsBackground) {
  PaintCenterText(w, (r.right+r.left)/2, (r.top+r.bottom)/2, "Background", Arial(11), :SBlack);
  PaintCenterText(w, 1+(r.right+r.left)/2, 1+(r.top+r.bottom)/2, "Background", Arial(11), :SWhite);
}
}
}
}

```

BUTthis gives the following layout file

```

LAYOUT(SplashScreenLayout, 240, 320)
  ITEM(ModifiedStaticImage, splashImg,
SetImage(Controls4U:Controls4U.iml:ImageSample).HSizePosZ(0, 0).VSizePosZ(0, 0))
END_LAYOUT
This of coarse cannot compile

```

But this will

```

LAYOUT(SplashScreenLayout, 240, 320)
  ITEM(ModifiedStaticImage, splashImg,
SetImage("Controls4U:Controls4U.iml:ImageSample").HSizePosZ(0, 0).VSizePosZ(0, 0))
END_LAYOUT

```

Subject: Re: StaticImage enhancement
 Posted by [koldo](#) on Fri, 19 Apr 2013 10:57:12 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Didier

It is possible to call a .iml Image by program "manually" but AFAIK this is only possible for the IMAGECLASS and IMAGEFILE defined in source.

This way StaticImage code cannot open an IMAGEFILE defined in your source.

However, I think it is possible to overcome this but I do not know how. The function to define could work like this:

```
Image img = LoadFromIml("Controls4U:Controls4U.iml:ImageSample_90");
```

.. But this would require to have the iml files accessible to exe instead of being embedded inside the exe.

Subject: Re: StaticImage enhancement
Posted by [Didier](#) on Fri, 19 Apr 2013 13:44:35 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Koldo,

Quote:It is possible to call a .iml Image by program "manually" but AFAIK this is only possible for the IMAGECLASS and IMAGEFILE defined in source.

This way StaticImage code cannot open an IMAGEFILE defined in your source.
The ModifiedStaticImage can open images from iml files that come from any package (at least the ones added in you're project : I didn't verify for other cases).

Quote:Image img = LoadFromIml("Controls4U:Controls4U.iml:ImageSample_90");
I think iml images can be retrieved by name so "all their is to do" is being able to get an iml image from this longer 'namePath' pattern.

==> Their are probably only 2 things to do:

modify '.lay' files and <CtrlCore/lay.h> so that the file namePath gets automatically added to the generated iml class
Add a global object that can retrieve iml class instances from the file pathName and then we could get the image using 'ImlManager.Get("imlPath").Get("imageName")'

NB: point 1 requires a modification in the layout designer of thelde

Subject: Re: StaticImage enhancement
Posted by [koldo](#) on Fri, 19 Apr 2013 22:05:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Didier

I partially disagree. Quick and dirty summary: .usc files can open whatever .iml image, however .cpp files cannot.

However I do not think it is so important. I would prefer to implement first an atan() or something like a DrawCircle() function in .usc.

Subject: Re: StaticImage enhancement
Posted by [Didier](#) on Mon, 22 Apr 2013 18:14:18 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hy Koldo,

Quote: implement first an atan() or something like a DrawCircle() function in .usc.
I agree that these functions are missing : although I use you're PaintCircle USC function

Subject: Re: StaticImage enhancement

Posted by [koldo](#) on Tue, 23 Apr 2013 07:09:59 GMT

[View Forum Message](#) <> [Reply to Message](#)

Didier wrote on Mon, 22 April 2013 20:14Hy Koldo,

Quote: implement first an atan() or something like a DrawCircle() function in .usc.

I agree that these functions are missing : although I use you're PaintCircle USC function Hello Didier

Does it exist an usc PaintCircle() function?. I could not found it.

Subject: Re: StaticImage enhancement

Posted by [Didier](#) on Tue, 23 Apr 2013 22:02:43 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi Koldo,

Quote:Does it exist an usc PaintCircle() function?. I could not found it.

I think you are working to much

these functions come from you're package : Controls4U (although I misspelled : it's not PaintCircle() but PaintEllipse())

```
fn PaintEllipse(w, left, top, right, bottom, width, color)
```

```
{
  if (width < 1)
    width = 1;
  a = (right-left)/2.;
  b = (bottom-top)/2.;
  width_2 = width/2.;
  delta = Pi()/20.;
  maxi = 2.*Pi();
  for (i = 0; i < maxi; i += delta) {
    if (i == 0) {
      x0 = left + a + (a - width_2);
      y0 = top + b;
    } else {
      x0 = x1;
      y0 = y1;
    }
    x1 = left + a + (a - width_2) * cos(i + delta);
```

```

y1 = top + b + (b - width_2) * sin(i + delta);
w.DrawLine(x0, y0, x1, y1, width, color);
}
}

fn DrawCircle(w, cx, cy, R, width, color) {
  PaintEllipse(w, cx-R-width/2., cy-R-width/2., cx+R+width/2., cy+R+width/2., width, color);
}

fn PaintArc(w, cx, cy, R, ang0, ang1, direction, width, color)
{
  if (direction == -1) {
    c = ang0;
    ang0 = ang1;
    ang1 = c;
  }
  ang0 = ang0*Pi()/180;
  ang1 = ang1*Pi()/180;
  delta = 3*Pi()/180;
  if (ang0 > ang1)
    ang1 += 2*Pi();
  for (i = ang0; i < ang1; i += delta) {
    if (i == ang0) {
      x0 = cx + R*cos(i);
      y0 = cy - R*sin(i);
    } else {
      x0 = x1;
      y0 = y1;
    }
    x1 = cx + R*cos(i + delta);
    y1 = cy - R*sin(i + delta);
    w.DrawLine(x0, y0, x1, y1, width, color);
  }
}

fn FillEllipse(w, left, top, right, bottom, background)
{
  a = (right-left)/2.;
  b = (bottom-top)/2.;
  if (a <= 0.5 || b <= 0.5) {
    w.DrawLine(left, top, right, bottom, 1, background);
    return;
  }
  delta = Pi()/10.;
  x0 = left + a;
  y0 = top + b;

  for (i = delta; i < Pi()/2.; i += delta) {

```

```
x1 = a * cos(i);
y1 = b * sin(i);
w.DrawRect(x0-x1 , y0-y1, 2*x1 , 2*y1, background);
}
width = min(a, b)/4.;
if (width > 1)
    PaintEllipse(w, left, top, right, bottom, width, background);
}

fn FillCircle(w, cx, cy, R, color) {
    FillEllipse(w, cx-R, cy-R, cx+R, cy+R, color);
}
```

Subject: Re: StaticImage enhancement
Posted by [koldo](#) on Wed, 24 Apr 2013 06:29:16 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Didier

I meant I would prefer to have those kind of functions in U++ instead of reinventing the wheel programming them in .usc .

Subject: Re: StaticImage enhancement
Posted by [Didier](#) on Thu, 25 Apr 2013 18:45:47 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Koldo,

I though it might be the case, but I wanted to joke a little bit