How to define application icon

With S60 1st and 2nd Edition

With S60 1st and 2nd Edition you could define an application icon using AIF file. The AIF file could be generated by using AIF builder or by using command line tools.

For command line tools you need to define AIF resource file, which for example could look like this (you could copy paste this to a file called MyApp.rss, for example):

```
#include <aiftool.rh>

RESOURCE AIF_DATA
{
        app_uid=0x12345678; // change to your own application UID
        caption_list=
                {
                CAPTION { code=ELangEnglish; caption="My app"; }
                };

        num_icons=2;

        embeddability= KAppNotEmbeddable;
        hidden=KAppNotHidden;
        newfile= KAppDoesNotSupportNewFile;
}
```

The file should have same name as your application file and rss ending, thus it is exactly same name as your application resource file, and for this reason it is advisable to make a separate folder for aif files.

Secondly you need icon bitmap files, you could use bmp images with 44x44 and 42x29 pixels in size, one with colors and one monochrome mask image. Then use bmconv to make a MBM (multibitmap file) from the icon bitmaps, for example by typing the following in command shell:

```
bmconv MyApp.mbm /c12s.bmp s_m.bmp /c12b.bmp b_m.bmp
```

where the s-images are the 42x29 in size and b-images are 44x44 in size.

After making the mbm file you can use aiftool to make the aif file by typing the following command:

```
aiftool MyApp MyApp.mbm
```

With S60 3rd Edition

In S60 3rd Edition icons are defined in the registration file. A registration file looks something like this:

```
UID2 KUidAppRegistrationResourceFile
UID3 0x12345678 // change to your own UID

RESOURCE APP_REGISTRATION_INFO
{
```

```
app_file="MyApp";
localisable_resource_file = "\\resource\\apps\\MyApp";
localisable_resource_id =R_LOCALISABLE_APP_INFO;
embeddability=KAppNotEmbeddable;
newfile=KAppDoesNotSupportNewFile;
group_name="My own folder";
}
```

The important parts inside the registration file are:

1. UID3, which needs to be your own application UID
2. app_file, which needs to be your application executables name without extension
3. localisable_resource_file, which needs to be your application resource files name with path (no drive and no extension)
4. localisable_resource_id, which needs to have the resource definition name used in your main resource for the application information.

The application information is then defined in your application's main resource file, for example like this:

```
RESOURCE LOCALISABLE_APP_INFO r_localisable_app_info
{
        short_caption = "My app name";
        caption_and_icon =
        {
                CAPTION_AND_ICON_INFO
                {
                        caption = " My application name ";
                        number_of_icons = 1;
                        icon_file = "\\Resource\\Apps\\MyApp_aif.mif";

                }
        };
}
```

The important parts inside the application information are:

1. Application viewable names (short_caption & caption), which can be localized for different languages
2. number_of_icons, with SVG this is 1, with bitmaps can be more, each bitmap icon also needs to have mask included.
3. icon_file, which is the name with path for the icon file (note that it also defined the extension, but doesn't need to define the drive)

Using bitmap icons

If you'd rather use a bitmap icon instead of SVG, follow these steps:

1. Install Inkscape drawing program and launch it.
2. Click *File* > *Open* and open your PNG/JPEG file. Nokia devices also support BMP format, however SVG-Tiny standard only recommends PNG and JPEG formats.
3. Click 'File > Save As *and save as "Plain SVG".*

Now you can use the saved SVG file by following the instructions for SVG file in the previous section.

If you don't want to use Inkscape, there is an alternative:

1. Prepare at 64x64 (for newer 240x320 phones) full color icon - 24bit color.
2. Prepare a 64x64 monochrome (1 bit) icon for b&w transparency mask, where black is where to draw, and white is where not to draw (transparent).
3. Name the icon files as follows: *img0.bmp* and *img0_mask.bmp*
4. Use the *mifconv* as follows:

   ```
   Symbian\9.3\S60_3rd_FP2_SDK\epoc32\tools\mifconv.exe MyApp.mif /c24,1 img0.bmp
   ```

The mifconv will automatically use the _mask file.

5. In the PKG file, add the following:

```
"..\gfx\MyApp.mif"                -"!:\resource\apps\MyApp_reg.mif"
"..\gfx\MyApp.mbm"                -"!:\resource\apps\MyApp_reg.mbm"
```

After installation restart the phone - the phone caches icons from previous installations. You have been warned.

Related Links

- How to create a Symbian application icon (SVG)
- Archived:Compatibility problem with binary-encoded SVG images (Known Issue)