# Harmattan Theming for Dumm1es

Authors: Tuomas Kuosmanen <tuomas.kuosmanen@nokia.com

## *In Brief*

Maemo Harmattan is built on top of the “Direct Ui” framework, and it uses a somewhat different approach to theming compared maemo5 and other earlier, gtk-toolkit based maemo releases.

The system provides a set of common building blocks for applications, called common components (programmers often call them “widgets” as well). Those common components are themed uniformly throughout the system, so all applications can share the same look and feel.

Applications can also have custom styling for elements specific to just that application. This is explained further later, when the theme folder structure is described.

## *All your base are belong to us*

Maemo6 “Harmattan” containst one “base” theme that all other themes inherit from. This is useful so you can benefit from the basic grounwork and definitions in base, and then build your own theme on top of that, instead of building everything from scratch.

## *Basic theming technology*

The theme graphics are composed of a number of vector graphics in SVG file format (currently svg-tiny subset is supported), PNG/JPG bitmap graphics and a number of CSS stylesheet files that map the graphics to the user interface components.

The CSS files tie our graphics, sounds and other theme bits to component states. For example, that a piece of SVG called “duibutton-background-pressed” is to be used as a “skin” to a component called “duibutton” when the user presses it. The SVG is a subset of the w3c stylesheet standard, and has custom parameters for the framework use, so always refer to “base” theme's CSS files, as those should reflect the supported parameters. The syntax is also limited, so not all possible selector methods work. *FIXME: this needs to be explained better.*

Each SVG element used in theming is referenced by its svg group-id, and each bitmap by its filename, with the extension stripped off. If you have several files using the same basename, with different extension (“maemo.jpg”, “maemo.jpeg” and “maemo.png”) bad things may happen, so better keep them unique by their basename. Also, as it is possible to have the same SVG group id in two separate files, it is important to maintain to a naming scheme that avoids this.

To make it easier to maintain a consistent look and feel, themes define various “constants” like commonly used fonts and a colors in a file called “constants.ini”. These can be used in stylesheets to keep things consistent across the theme. This

way you can later tweak those settings in one file, rather than having to edit all your stylesheets if you want to change something.

Themes can also contain vibra and sound feedbacks assigned to widget events, these can consist of a sound file (wav) or a vibra motor effect file, or both. *FIXME: these need to be explained in more detail.*

### *Theme folder structure*

A theme consists of a folder tree located in the system theme path under a *theme root folder*, for example */usr/share/themes/**foobar*** and the theme contents are organized in a number of subfolders. Since this system theme path is shared by different software projects, all directui themes live inside a folder called “dui” inside the theme root folder.

The common user interface components theme is in the subfolder “libdui”, and application specific theming is in a subfolder named after the application name. Otherwise the structure is the same on either case, and the subfolders are always as follows:

- **feedbacks** folder is used to define different vibra/sound event feedbacks. These can be bound into widget states and events through CSS. Sound files are wav format. *FIXME: Needs more information about how this works.*
- **icons** has all the icon files, one svg file per icon. Even though icons are SVG, the filename is used as an identifier for icons, and svg group id does not matter. *Note: his does not currently work yet! Currently icons need to be in the svg folder, referenced by svg id – we try to fix this in a future release, so be warned!*
- **images** contains bitmap images, for example wallpaper files and application background graphics. Referenced by basename, (filename without the extension). Take care to keep the basenames unique, as Bad Things mayl happen if you have “maemo.jpg” and “maemo.png” there and try to use “maemo” as your graphic id...
- **style** contains the stylesheets for theme, and there is one “master stylesheet” that is loaded, and other stylesheet can be included with the @import command. The master stylesheet name is the same as the parent folder name: libdui.css or myapplicationname.css, depending on whether it is part of common components or application specific theming. By using the import command you can split the CSS into logical parts, for example grouped by component.
- **svg** contains all of our SVG graphics. You can split the graphics across separate files if you like, or you can just have one huge SVG that contains all your graphics, it's up to you. The only important thing is that you need to make sure each referenced svg group ID is unique – so be careful with naming if you have separate files.
- *FIXME: needs information about the other files like .conf, index.theme and constants.ini etc...*

## *Creating your own theme*

Finally the part you are looking for! :-) Probably the simplest way is to learn by example, so create a duplicate folder structure of the “base” theme, and study its contents and modify and copy whatever you changed into your own theme folder structure. Whatever you leave out of your own theme is used from the “base” theme, so you can change the theme step by step, and still have a working theme you can test.

### *Technical bits*

The important part is that since SVG elements are referenced by their group-ID, you need to keep those id's in sync with your CSS files – so if you draw new elements next to the old ones, be sure to rename the new one after deleting the old, or just work inside the “group” to begin with. If you are using Adobe Illustrator, the group ID is the same as the layer dialog's group name. In Inkscape you should set the ID from the Object Properties dialog (Object → Properties in the menu). On both applications doubleclicking the group “dives” into the group, so you can edit and replace elements, while keeping the group name intact. This is just a thing to remember, especially when you wonder why your new shiny graphic is not showing up... :-)

### *Application specific theming*

Within the `theme_root_folder/dui` there can be two things: common component theming and application specific parts. Common component theming is always in a subfolder called “libdui”. Applications can install theming in a subfolder named after the application name (“mediaplayer” or “duihome” or “widgetsgallery” for example).

## *Installing a theme*

TODO