How to create a skin for the SCA UI v3.0

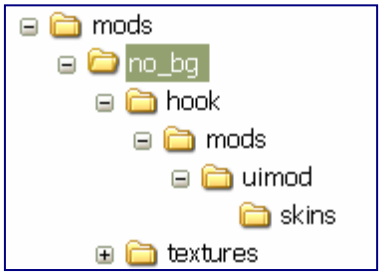
First things first: Don't be scared by the length of this document! Most of it is taken up with examples in the appendices. If anything isn't covered in enough detail then ask Goom or Cleopatre on the GPG forums, we'd be happy to help you out.

The skinning abilities of this mod are very powerful, but to use them, you need to stick to a simple but strict format. First you should understand that skinning the interface consists of two parts.

- ***The skin*** which concerns the backgrounds and the position of the components inside a control.
- ***The layout*** which sets the position of all the controls.

For ease of use both are mixed in the normal layout manager, but to create a skin you must dissociate them if you want to understand how they work. Just have a look at the advanced layout panel. You can change the skin of an interface without changing the position of the windows and vice-versa.

I. Overview



The mod folder must contain certain files. Be aware that two mods can not have the same name or unexpected behaviour may occur. Here the directory name is no_bg but should be replace by your mod name.

There are two important files, which must be in any skin mod:

- '/mods/myskin/mod_info.lua, it contains the info for the mod and for the mod manager.
- '/mods/myskin/hook/mods/uimod/skins/sca.skin', this contains all the information about textures and layouts for your skin.

You must also have your textures somewhere in your skin folder. The paths are set up in sca.skin so you can use whatever folder structure you find most convenient.

When you have created your files and directory (the easiest way is to start from an existing skin mod). You can start to create your skin/layout.

II. Create a new skin(changing the appearance):

a. The gory details.

Create a text file in the 'mods/ *myskinfolder*/hook/mods/uimod/skins' directory, and name it '**sca.skin**', (where *myskinfolder* can be anything you like).

The format for sca.skin is as follows:

```
do
    table.insert(skinlist,"myskinname")
    myskinname= {
          path = '/mods/myskinfolder/textures/',
          info = "info on this skin",
          first control to be skinned = {stuff},
          second control to be skinned = {more stuff},
          etc...
    }
end
```

The 'do' at the start and the 'end' at the end are part of the supreme commander mod system and shouldn't be forgotten. The table.insert(skinlist, 'myskin') is needed to tell the ui mod the name of your skin. Your skin name needs to be unique to avoid conflicts in game. 'myskin' **must** be written in exactly the same way as the table that contains your skin, and can contain no spaces (use _ for space). Next the **path argument** is root folder where you are saving your textures. Here we have created a '`/mods/myskinfolder/textures/`' directory (texture could also be the one used by the uimod itself then the `path should be "/mods/uimod/skins/sca/")`

Now we move on to actually creating your skin.

```
myskinname = {
    first control to be skinned = {component},
    second control to be skinned = {more component},
    etc...
}
```

The contents of the table for each control should be a list of all the components of the control you want to reskin (also as tables). Any controls named in this table will be reskinned. Any controls which are not present will use the default SCA skin.

Here is the list of the different controls you can skin:

- mfd
- minimap
- options
- mass
- energy
- orders
- selection
- construction
- queue
- unitview
- unitviewdetail
- borders
- score

For the full list of all the components each control possesses you should look at the *DEFAULT table* in appendix I (or look directly in the original sca.skin)

Component tables can contain any of the following variables (the only required variable for a main control is texture, for a sub-component texture, x and y are required)

```
component = { type = '',
              texture = '',
              x = #,
              y = #,
              x_right = #,
              y_bottom = #,
              type_helper = '',
              code = ''   },
```

Here"s a table that sum up the different possible values for *type* and *texture*:

type	**Comments**	***texture* argument and file format**	**Files link in the folder directory**
no type specify	For a normal bitmap	'folder/*name*'	1 file: *name*.dds
'9part'	for controls resizable in both axes	'folder/*name*'	9 files: *name*_tl.dds, *name*_t.dds, *name*_tr.dds, *name*_l.dds, *name*_m.dds, *name*_r.dds, *name*_bl.dds, *name*_b.dds, *name*_br.dds
'3part_vert'	for controls resizable in the y axis	'folder/*name*'	3 files:*name*_t.dds, *name*_m.dds, *name*_b.dds
'3part_horiz'	for controls resizable in the x axis	'folder/*name*'	3 files:*name*_l.dds, *name*_m.dds, *name*_r.dds
'button'	for buttons	'folder/*name*'	4 files: *name*_up.dds, *name*_over.dds, *name*_down.dds , *name*_dis.dds
'checkboxAs Button'	for checkboxes but using button textures	'folder/*name*'	4 files: *name*_up.dds, *name*_over.dds, *name*_down.dds , *name*_dis.dds
'checkbox'	for checkboxes	'folder/*name*'	6 files: *name*_up.dds, *name*_upsel.dds, *name*_over.dds, *name*_oversel.dds, *name*_dis.dds, *name*_dissel.dds

Note that buttons/checkboxes must be used as in the default sca.skin, so if you aren't sure if you are skinning a button or a checkbox then look at the table in appendix I. Multi-part textures are up to you to

decide which to use. *Type* defines the requested argument for *texture* and the format of the name of the file you have to respect (note: _tl is for topleft; _m for middle; _br for bottomright and so on.). You don't have to include the dds extension in the texture field.

For a main element of a control (usually the background, but check the default skin to see which controls don't specify x/y) that is all you need. The positions are set up as part of the layout.

For sub-components you must specify x and y positions, these are usually (but not always) relative to the background of the control.

x = relative x position in the control
y = relative y position in the control

The rest of the options are used for more precise control over position.
x_right and y_bottom are useful if a component has to stretch as the component resizes (e.g. the scroll buttons on the default construction panel)

x_right = distance of the right side of the component from the right edge of the control
y_bottom = distance of the bottom side of the component from the bottom edge of the control

type_helper will allow you to specify the relative position your component takes in the control. By default *type_helper* is unspecified and x/y are from the topleft of the control. If *type_helper* is specified, the component will be placed from a different position. The possibilities are:

- 'RightTop' -relative to the top right corner
- 'Center' - relative to the control center
- 'HorizontalCenter' – y from the top, x relative to control horizontal center
- 'VerticalCenter' – x from the left, y relative to control vertical center

b. Examples:

1. basic example

This first skin will just replace the background of construction with a simple bitmap. Be aware that as a simple texture you can not resize the windows (you should use multipart textures if you want a control to be resizable, see the other examples).

```
do
    table.insert(skinlist,"constructionskin")
    constructionskin= {
          path = '/mods/myawesomeskin/textures/',
          info = "Totally awesome new skin for construction!",
          construction = {
                bg = {texture = 'myfile.dds'},
          },
   }
end
```

2. resizable example

This skin would just replace the background of the queue with a 3 part texture (resizeable horizontally), you need to put 3 files in the directory '`mods/myskin/' and they should be` named:
`myqueuetexture_l.dds, myqueuetexture_r.dds, myqueuetexture_m.dds`

```
do
    table.insert(skinlist,"myskinname")
    myskinname= {
          path = '/mods/myskinfolder/textures/',
          info = "Even better queue skin.",
          queue = {
                bg = {type = "3part_horiz", texture = 'myqueuetexture'},
          },
   }
end
```

3. Third example: complete skinning

This example will do a complete skinning for queue for both normal and vertical options (the second paragraph). It's also setting the buttons, the checkboxes and the icon grid, and is resizable in both axes.

```
do
  table.insert(skinlist,"myskinname")
  myskinname= {
    path = '/mods/myskinfolder/textures/',
    info = "info on this mod",
    queue = {
      bg = {type = "9part", texture = 'construction/cstr_horiz_default'},
      infinitebtn = {type = "checkbox", texture = 'queue/infinite_btn', x = 3, y = 3},
      pausebtn = {type = "checkbox", texture = 'queue/pause_btn', x = 28, y = 3},
      scrollup = {type = "button", texture = 'construction/arrow_btn', x = 55, y = 10,
      scrolldown = {type = "button", texture = 'construction/arrow_btn', x = 55, y = 40,
      grid = {left = 66, top = 11, right = 13, bottom = 11},

      verticalbg = {type = "9part", texture = 'construction/cstr_vert_default'},
      infinitebtn_v = {type = "checkbox", texture = 'queue/infinite_btn_v', x = 2, y = 3},
      pausebtn_v = {type = "checkbox", texture = 'queue/pause_btn_v', x = 2, y = 28},
      scrollup_v = {type = "button", texture = 'construction/arrow-up_scr', x = 7, y = 52},
      scrolldown_v = {type = "button", texture = 'construction/arrow-down_scr',x= 7,y= 79},
      grid_v = {left = 45, top = 3, right = 3, bottom = 3},
    },
  }
end
```

c. Advanced use:

It's possible to do more complex operations with the code field. For example, all the layouts included in the mod are created with this skin system. If you want to discover more about the code option then look at the default SCA skin, either in appendix I or directly in the mod itself. We used it to cause the tech buttons on the construction panel to be sized relative to the background.

Another trick we use is an offsets field, this is only really useful for selection – because selection can't be placed separately from construction, you can create an offsets table to set it to a different position. E.G. for our GPG vertical skin the selection background is set up as follows:

```
verticalbg = {type="3part_vertical", texture="selection/panel-v_bmp", offsets = {Bottom = -78}},
```

This is because the bottom part of selection needs to cover the queue for that layout.

III. Create a layout(changing the position):

a. Simple moving windows

At the end of the skin part, you could add a number of different layouts to be used with your skin. The syntax is to give like a component before

```
do
    table.insert(skinlist,"myskinname")
    myskinname= {
          path = '/mods/myskinfolder/textures/',
          info = "info on this skin",
          first control to be skinned = {stuff},
          second control to be skinned = {more stuff},
          etc...
          layouts = {
             name_of_the_layout = {
             preview = 'name_of_file_preview.dds',
             control1 = function() return {Left = #, Bottom = screen.Bottom()} end,
             control2 = etc
             data = {
                … one or all the option of the mod …
                },
             code = { function()… layout specific code …   end},
          }
     }
end
```

A layout has to place every control, or you will get an error when starting the game in your layout. The full list of controls is:

- mfd
- minimap
- options
- mass
- energy
- orders
- construction
- queue
- unitview
- borders
- score

selection is placed with construction, and unitviewdetail is placed with unitview.

The positions are then specified as `function() return {table of positions} end`,

The function() part is to allow layouts to be set to any resolution. The table can contain {Left = #, Top = #, Right = #, Bottom = #}.
For a control with a single texture you should only specify 1 each of Left/Right and Top/Bottom.
For a 3 part horizontal texture you should specify Left, Top, Right.
For a 3 part vertical you should specify Left, Top, Bottom.
For a 9 part texture you need to specify all 4 of Left, Top, Bottom, Right.
Note that unitview and score work strangely, and you always need to specify all 4 sides for these two.
Failure to use the correct format in this section *will* lead to quite serious looking errors in the game log, and probably cause the layout not to load.

Using plain numbers for the positions will set the control relative to the top left corner of the screen. You can also use `screen.Right() - #` and `screen.Bottom - #` to set a position relative to the right or bottom of the screen, this is very useful for creating layouts that will work at any resolution.

b. Using the existing option of the mod

You can also set most of the mod options from a layout. Here's a complete list of the options and what they do, any you don't include will simply not be altered by your layout and will use whatever the last layout used.

Type and example value	*information*
transparencyCheckBox=1	The checkbox to allow transparency (1=check)
transparenceSlider=0.6	The value of the transparency between 0 and 1
inverse_barre=1,	Invert the energy bar
affiche_construc=1	display the windows construction/orders/queue/selection when nothing is selected. if 1 -->no
affiche_help=0,	display the help tips (building/units information)
hide_orders=0,	Hide the orders window
hide_mfd=0,	Hide the mfd window (the one with the minimap etc…)
hide_mass=0,	Hide the mass bar
hide_energy=0,	Hide the energy bar
hide_options=0,	Hide the windows option (the one with pause etc…)
hide_queue=0,	Hide the queue window
hide_construction=0,	Hide the construction window
hide_unitview=0,	Hide the unitview window
new_unit_view=1,	Select the unit view style (0 old one, 1 the mini ui style)
popzoom = 0.35,	The level popzoom pops to (0 to 1)
iconsize = 0.44,	The iconsize in queue and construction (0 to 1, from 32 pixels to 96)
verticalenergy = 0,	Display the energy bar vertically
verticalmass = 0,	Display the mass bar vertically
verticaloptions = 0,	Display the option window vertically
verticalMFD = 1,	Display the mfd window vertically
straticons = 1,	Display the stategic icons on the build icons
altconstruction = 0,	Use the vertical construction style (tech buttons on top)
vertconstruction = 0,	Use the vertical style of construction window
miniui_econ = 0,	Use the special miniui style for the economy bar, mfd and options (and disable the normal econ bars, option and mfd)
showborders = 1,	Show the borders around the screen
ordersgrid = 34,	The style of orders: 3x4 icons, 4x3 or 6by2. (use 34/43/62)
vertqueue = 0,	Use the vertical style of queue window
const_sizing = 0,	Allow the queue and construction windows to autoresize to have the minimal size with number of units as in the mini ui.

For example: the following code will change only these two values and let the others unchanged.

```
data = {
       affiche_help = 1,
       hide_orders = 0,
   },
```

c. Advanced code options

It is also possible to do more complex operations. As we do for many of the layouts you see in the mod. We are using the same system for our own layouts and even to recreate the original layout of gpg! For more information on this just have a look to the code part of the skin, there is some in the layout part and some in the skin part. This enables you to do conditionals and complex operations. Just have a look at the file '\mods\uimod\skins\sca.skin' in the mod. This does however require some knowledge of LUA, and how the UI works. An example of what we use code for is in the gpg bottom layout setting the construction/queue bars to different positions depending on if the minimap is shown (have also a look at appendix III).

d. Factional skins

Another possibility is to make a skin which changes for each faction. You can do it by giving the name of the skin used for each faction with

```
    factionskins = {"SCA_Aeon", "SCA_Cybran", "SCA_UEF"},
```

This way you only need to set your layout up once, and your additional factional skins can be very short (cf. an example at the appendix II).

Appendix I : Default layout and table of all the components available

Extract of '\mods\uimod\skins\sca.skin'

```
  Default = {
    path = '/mods/uimod/skins/sca/default/',
    info = 'Default SCA skin.',
    equivalentSkins = {      'SCA_Aeon', 'SCA_Cybran', 'SCA_UEF',},

    score = {
            tl = {texture = 'score/score_brd_ul.dds'},
            tm = {texture = 'score/score_brd_horz_um.dds'},
            tr = {texture = 'score/score_brd_ur.dds'},
            lm = {texture = 'score/score_brd_vert_l.dds'},
            m = {texture = 'score/score_brd_m.dds'},
            rm = {texture = 'score/score_brd_vert_r.dds'},
            bl = {texture = 'score/score_brd_ll.dds'},
            bm = {texture = 'score/score_brd_lm.dds'},
            br = {texture = 'score/score_brd_lr.dds'},
            units = {x = 15, y = 6},
            time = {x = 16, y = 18},
    },
    mfd = {
            bg = {texture = 'mfd/mfd_horiz_button_default_bmp.dds'},
            control = {type = "button", texture = 'mfd/control_btn', x = 170, y = 2},
            economy = {type = "checkboxAsButton", texture = 'mfd/economy_btn', x = 82, y = 2},
            intelligence = {type = "checkboxAsButton", texture = 'mfd/intelligence_btn', x = 38, y = 2},
            military = {type = "checkboxAsButton", texture = 'mfd/military_btn', x = -6, y = 2},
            ['team-color'] = {type = "checkboxAsButton", texture = 'mfd/team-color_btn', x = 126, y = 2},

            verticalbg = {texture = 'mfd/mfd_button_default_bmp.dds'},
            control_v = {type = "button", texture = 'mfd/control_btn', x = -6, y = 161},
            economy_v = {type = "checkboxAsButton", texture = 'mfd/economy_btn', x = -6, y = 83},
            intelligence_v = {type = "checkboxAsButton", texture = 'mfd/intelligence_btn', x = -6, y = 44},
            military_v = {type = "checkboxAsButton", texture = 'mfd/military_btn', x = -6, y = 5},
            ['team-color_v'] = {type = "checkboxAsButton", texture = 'mfd/team-color_btn', x = -6, y = 122},
    },
    minimap = {
            bg = {type = '9part', texture = 'minimap/minimapBG_default'},
            glow = {type = '9part', texture = 'minimap/minimap_glow'},
            borders = {left = 10, top = 10, right = 10, bottom = 10},
    },
    options = {
            horizontal = {texture = 'options/options_bg_default_horiz.dds'},
            exit_menu = {type = "button", texture = 'options/menu_btn', x = 0, y = 2},
            objectives = {type = "button", texture = 'options/objectives_btn', x = 46, y = 2},
            score = {type = "button", texture = 'options/score_btn', x = 92, y = 2},
            inbox = {type = "button", texture = 'options/inbox_btn', x = 138, y = 2},
            diplomacy = {type = "button", texture = 'options/diplomacy_btn', x = 184, y = 2},
            pause = {type = "checkbox", texture = 'options/pause_btn', x = 230, y = 2},

            vertical = {texture = 'options/options_bg_default_vert.dds'},
            exit_menu_v = {type = "button", texture = 'options/menu_btn', x = 9, y = -2},
            objectives_v = {type = "button", texture = 'options/objectives_btn', x = 9, y = 32},
            score_v = {type = "button", texture = 'options/score_btn', x = 9, y = 65},
            inbox_v = {type = "button", texture = 'options/inbox_btn', x = 9, y = 98},
            diplomacy_v = {type = "button", texture = 'options/diplomacy_btn', x = 9, y = 131},
            pause_v = {type = "checkbox", texture = 'options/pause_btn', x = 9, y = 164},
    },
    unitview = {
            new = {texture = 'unitview/unit-over-back_bmp.dds'},

            classic = {texture = 'unitview/unit-over_bmp.dds'},
            icon = {x = 4, y = 45, height = 46, width = 46},
            straticon = {x = 60, y = 53, height = 14, width = 14},
            techpips = {x = 78, y = 62},
            healthbar = {foreground = 'unitview/bar01_bmp.dds', background = 'unitview/health-bars-back-1_bmp.dds', x = 60, y = 68},
            progbar = {foreground = 'unitview/bar01_bmp.dds', background = 'unitview/health-bars-back-1_bmp.dds', x = 61, y = 73,
range = {0,1}},
            name = {x = 115, y = 48, width = 300},
            veticons = {x = 54, y = 40},
            mass = {x = 143, y = 68},
            energy = {x = 204, y = 68},
            extrainfo = {x = 265, y = 70},
            focusbg = {texture = 'unitview/unit-over-build_bmp.dds', x = 412, y = -4},
            focusicon = {x = 42, y = 2, height = 46, width = 46},
            focushealthtext = {x = 99, y = 28},
            focushealthbar = {foreground = 'unitview/bar01_bmp.dds', background = 'unitview/health-bars-back-1_bmp.dds', x = 136, y
= 33, width = 184},
            focusname = {x = 101, y = 9},
            abilitybox = {x = 18, y = 69, width = 20},
    },
    unitviewdetail = {
            new = {texture = 'unitview/build-over-back_bmp.dds'},

            classic = {texture = 'unitview/unit-over-back02.dds'},
            name = {x = 14, y = 18},
            description = {x = 5, y = 0},
            icon = {x = 8, y = 46, height = 46, width = 46},
            healthstat = {x = 64, y = 41},
            timestat = {x = 64, y = 58},
            shieldstat = {x = 64, y = 75},
            buildcostgroup = {x = 205, y = 40},
            upkeepgroup = {x = 305, y = 40},
            abilitybox = {x = 18, y = 102, width = 20},
            helpbox = {x = 429, y = 4},
    },
    borders = {
            bg = {type = '9part', texture = 'borders/back_brd'},
    },
    mass = {
            bg = {texture = 'resources/mass-back_default_bmp.dds'},
            dropdown = {texture = 'resources/econ-advanced_bmp.dds', x = 314, y = 4},
            dropover = {texture = 'resources/econ-advanced_over_bmp.dds', x = 0, y = 0, type_helpers = "Center"},
            rateContainer = {type = 'button', texture = 'resources/resource_rate_bmp', x = 246, y = 3},
            label = {texture = 'resources/mass_btn_up.dds', x = -9, y = -7},
```

```
            warning = {texture = 'resources/glow-02_bmp.dds', x = 0, y = 0, type_helpers = "Center"},
            storagebar = {foreground = 'resources/mass-bar_bmp.dds', background = 'resources/mass-bar-back_bmp.dds', x = 54, y = 12,
slidepercentage = 7/190},
            storage = {text = "", size = 12, color = false, font = "Arial", type_helpers = "Center", x = 0, y = 13},
            rate = {text = "", size = 20, color = false, font = "Arial", type_helpers = "Center", x = 0, y = 0},
            income = {text = "", size = 10, color = 'green', font = "Arial", type_helpers = "Center", x = 0, y = -6},
            expense = {text = "", size = 10, color = 'red', font = "Arial", type_helpers = "Center", x = 0, y = 6},
            warningText = {text =  LOC("<LOC Econ_0000>Mass Needed"), size = 16, color = false, font = "Zeroes Three", type_helpers
= "HorizontalCenter", x = 0, y = -4},

            verticalbg = {texture = 'resources/alt-econ-back_default.dds'},
            dropdown_v = {texture = 'resources/econ-advanced_bmp_v.dds', x = -1, y = 170},
            dropover_v = {texture = 'resources/econ-advanced_over_bmp_v.dds', x = 0, y = 0, type_helpers = "Center"},
            rateContainer_v = {type = 'button', texture = 'resources/resource_rate_bmp_v', x = 45, y = 170},
            label_v = {texture = 'blank.dds', x = 0, y = 0},
            warning_v = {texture = 'blank.dds', x = 0, y = 0, type_helpers = "Center"},
            storagebar_v = {foreground = 'resources/mass-bar-v_bmp.dds', background = 'resources/mass-bar-back-v_bmp.dds', x = 21, y
= 28, vertical = true, slidepercentage = 7/140},
            rateposbar = {foreground = 'resources/resource-rate-pos.dds', background = 'resources/resource-rate-pos-back.dds', x =
72, y = 30, vertical = true, slidepercentage = 0.1},
            ratenegbar = {foreground = 'resources/resource-rate-neg.dds', background = 'resources/resource-rate-neg-back.dds', x =
72, y = 99, vertical = true, inverse = true, slidepercentage = 0.1},
            storage_v = {text = "", size = 11, color = false, font = "Arial", type_helpers = "Center", x = 0, y = -7},
            storageMax_v = {text = "", size = 11, color = false, font = "Arial", type_helpers = "Center", x = 0, y = 7},
            rate_v = {text = "", size = 19, color = false, font = "Arial", type_helpers = "Center", x = -1, y = -1},
            income_v = {text = "", size = 10, color = 'green', font = "Arial", type_helpers = "Center", x = 0, y = -13},
            expense_v = {text = "", size = 10, color = 'red', font = "Arial", type_helpers = "Center", x = 0, y = 13},
            warningText_v = {text = "Mass", size = 22, color = false, font = "Zeroes Three", type_helpers = "HorizontalCenter", x =
0, y = 1},
    },
    energy = {
            bg = {texture = 'resources/energy-back_default_bmp.dds'},
            dropdown = {texture = 'resources/econ-advanced_bmp.dds', x = 11, y = 4},
            dropover = {texture = 'resources/econ-advanced_over_bmp.dds', x = 0, y = 0, type_helpers = "Center"},
            rateContainer = {type = 'button', texture = 'resources/resource_rate_bmp', x = 55, y = 3},
            label = {texture = 'resources/energy_btn_up.dds', x = 318, y = -5},
            warning = {texture = 'resources/glow-02_bmp.dds', x = 0, y = 0, type_helpers = "Center"},
            storage = {text = "", size = 12, color = false, font = "Arial", type_helpers = "Center", x = 0, y = 13},
            storagebar = {foreground = 'resources/energy-bar_bmp.dds', background = 'resources/energy-bar-back_bmp.dds', x = 128, y
= 12, slidepercentage = 7/190},
            rate = {text = "", size = 20, color = false, font = "Arial", type_helpers = "Center", x = 0, y = 0},
            income = {text = "", size = 10, color = 'green', font = "Arial", type_helpers = "Center", x = 0, y = -6},
            expense = {text = "", size = 10, color = 'red', font = "Arial", type_helpers = "Center", x = 0, y = 6},
            warningText = {text =  LOC("<LOC Econ_0001>Energy Needed"), size = 16, color = false, font = "Zeroes Three",
type_helpers = "HorizontalCenter", x = 0, y = -4},

            bg_i = {texture = 'resources/mass-back_default_bmp.dds'},
            dropdown_i = {texture = 'resources/econ-advanced_bmp.dds', x = 314, y = 4},
            dropover_i = {texture = 'resources/econ-advanced_over_bmp.dds', x = 0, y = 0, type_helpers = "Center"},
            rateContainer_i = {type = 'button', texture = 'resources/resource_rate_bmp', x = 246, y = 3},
            label_i = {texture = 'resources/energy_btn_up.dds', x = -1, y = -5},
            warning_i = {texture = 'resources/glow-02_bmp.dds', x = 0, y = 0, type_helpers = "Center"},
            storagebar_i = {foreground = 'resources/energy-bar_bmp.dds', background = 'resources/energy-bar-back_bmp.dds', x = 54, y
= 12, slidepercentage = 7/190},
            storage_i = {text = "", size = 12, color = false, font = "Arial", type_helpers = "Center", x = 0, y = 13},
            rate_i = {text = "", size = 20, color = false, font = "Arial", type_helpers = "Center", x = 0, y = 0},
            income_i = {text = "", size = 10, color = 'green', font = "Arial", type_helpers = "Center", x = 0, y = -6},
            expense_i = {text = "", size = 10, color = 'red', font = "Arial", type_helpers = "Center", x = 0, y = 6},
            warningText_i = {text =  LOC("<LOC Econ_0001>Energy Needed"), size = 16, color = false, font = "Zeroes Three",
type_helpers = "HorizontalCenter", x = 0, y = -4},

            verticalbg = {texture = 'resources/alt-econ-back_default.dds'},
            dropdown_v = {texture = 'resources/econ-advanced_bmp_v.dds', x = -1, y = 170},
            dropover_v = {texture = 'resources/econ-advanced_over_bmp_v.dds', x = 0, y = 0, type_helpers = "Center"},
            rateContainer_v = {type = 'button', texture = 'resources/resource_rate_bmp_v', x = 45, y = 170},
            label_v = {texture = 'blank.dds', x = 0, y = 0},
            warning_v = {texture = 'blank.dds', x = 0, y = 0, type_helpers = "Center"},
            storagebar_v = {foreground = 'resources/energy-bar-v_bmp.dds', background = 'resources/energy-bar-back-v_bmp.dds', x =
21, y = 28, vertical = true, slidepercentage = 7/140},
            rateposbar = {foreground = 'resources/resource-rate-pos.dds', background = 'resources/resource-rate-pos-back.dds', x =
72, y = 30, vertical = true, slidepercentage = 0.1},
            ratenegbar = {foreground = 'resources/resource-rate-neg.dds', background = 'resources/resource-rate-neg-back.dds', x =
72, y = 99, vertical = true, inverse = true, slidepercentage = 0.1},
            storage_v = {text = "", size = 11, color = false, font = "Arial", type_helpers = "Center", x = 0, y = -7},
            storageMax_v = {text = "", size = 11, color = false, font = "Arial", type_helpers = "Center", x = 0, y = 7},
            rate_v = {text = "", size = 19, color = false, font = "Arial", type_helpers = "Center", x = -1, y = -1},
            income_v = {text = "", size = 10, color = 'green', font = "Arial", type_helpers = "Center", x = 0, y = -13},
            expense_v = {text = "", size = 10, color = 'red', font = "Arial", type_helpers = "Center", x = 0, y = 13},
            warningText_v = {text = "Energy", size = 22, color = false, font = "Zeroes Three", type_helpers = "HorizontalCenter", x
= 2, y = 1},
    },
    orders = {
            ['3by4'] = {texture = 'orders/orders-34_default.dds'},
            ['3by4Grid'] = {x = -1, y = 0, width = 50, height = 50, cols = 3, rows = 4},
            ['4by3'] = {texture = 'orders/orders-43_default.dds'},
            ['4by3Grid'] = {x = -1, y = 0, width = 50, height = 50, cols = 4, rows = 3},
            ['6by2'] = {texture = 'orders/orders-62.dds'},
            ['6by2Grid'] = {x = -1, y = 0, width = 50, height = 50, cols = 6, rows = 2},
            buttonpath = '/textures/ui/common/game/orders/',
    },
    selection = {
            bg = {type = "9part", texture = 'construction/cstr_horiz_default'},
            scrollup = {type = "button", texture = 'construction/arrow_btn', x = 55, y = 10, y_bottom = 10, code = function(self)
import('/mods/uimod/sca_utils/uimod.lua').TempScrollArrowFctn(self, "left") end},
            scrolldown = {type = "button", texture = 'construction/arrow_btn', x = 3, y = 10, y_bottom = 10, type_helpers =
"RightTop", code = function(self) import('/mods/uimod/sca_utils/uimod.lua').TempScrollArrowFctn(self, "right") end},
            grid = {left = 66, top = 11, right = 13, bottom = 11},

            verticalbg = {type = "9part", texture = 'construction/cstr_vert_default'},
            scrollup_v = {type = "button", texture = 'construction/arrow-up_scr', x = 7, y = 3},
            scrolldown_v = {type = "button", texture = 'construction/arrow-down_scr', x = 7, y = 32},
            grid_v = {left = 45, top = 3, right = 3, bottom = 3},
    },
    construction = {
            bg = {type = "9part", texture = 'construction/cstr_horiz_default'},
            tech1tab = {type = "checkbox", texture = 'tech buttons/tech-tab_btn', x = 2, y = -2, code=function(self)
self.Top:Set(function() return obj.construction.bg.Top() - math.min(math.floor(obj.construction.bg.Height()/110), 1) end)
self.Bottom:Set(function() return self.Top() + math.min(obj.construction.bg.Height()/4.2, 30) end) end},
```

```
            tech2tab = {type = "checkbox", texture = 'tech buttons/tech-tab_btn', x = 2, y = 20, code=function(self) self.Top:Set(function() return obj.construction.tech1tab.Bottom() - math.min(math.floor(obj.construction.bg.Height()/18), 6) end) self.Bottom:Set(function() return self.Top() + math.min(obj.construction.bg.Height()/4.2, 30) end) end},
            tech3tab = {type = "checkbox", texture = 'tech buttons/tech-tab_btn', x = 2, y = 42, code=function(self) self.Top:Set(function() return obj.construction.tech2tab.Bottom() - math.min(math.floor(obj.construction.bg.Height()/18), 6) end) self.Bottom:Set(function() return self.Top() + math.min(obj.construction.bg.Height()/4.2, 30) end) end},
            tech4tab = {type = "checkbox", texture = 'tech buttons/tech-tab_btn', x = 2, y = 64, code=function(self) self.Top:Set(function() return obj.construction.tech3tab.Bottom() - math.min(math.floor(obj.construction.bg.Height()/18), 6) end) self.Bottom:Set(function() return self.Top() + math.min(obj.construction.bg.Height()/4.2, 30) end) end},
            tech1icon = {texture = 'tech buttons/tech-1_icon.dds', type_helpers = 'HorizontalCenter', x = 0, y = 0, code=function(self) self.Top:Set(function() return obj.construction.tech1tab.Top() - 4 end) self.Bottom:Set(function() return obj.construction.tech1tab.Bottom() + 3 end) end},
            tech2icon = {texture = 'tech buttons/tech-2_icon.dds', type_helpers = 'HorizontalCenter', x = 0, y = 0, code=function(self) self.Top:Set(function() return obj.construction.tech2tab.Top() - 4 end) self.Bottom:Set(function() return obj.construction.tech2tab.Bottom() + 3 end) end},
            tech3icon = {texture = 'tech buttons/tech-3_icon.dds', type_helpers = 'HorizontalCenter', x = 0, y = 0, code=function(self) self.Top:Set(function() return obj.construction.tech3tab.Top() - 4 end) self.Bottom:Set(function() return obj.construction.tech3tab.Bottom() + 3 end) end},
            tech4icon = {texture = 'tech buttons/tech-4_icon.dds', type_helpers = 'HorizontalCenter', x = 0, y = 0, code=function(self) self.Top:Set(function() return obj.construction.tech4tab.Top() - 4 end) self.Bottom:Set(function() return obj.construction.tech4tab.Bottom() + 3 end) end},
            enhancebtn = {type = "button", texture = 'construction/enhance_btn', x = 4, y = 92, code=function(self) self.Top:Set(function() return obj.construction.tech4tab.Bottom() - math.min(math.floor(obj.construction.bg.Height()/30), 3) end) self.Bottom:Set(function() return self.Top() + math.min(obj.construction.bg.Height()/4.84, 26) end ) end},
            scrollup= {type = "button", texture = 'construction/arrow_btn', x = 55, y = 10, y_bottom = 10, code = function(self) import('/mods/uimod/sca_utils/uimod.lua').TempScrollArrowFctn(self, "left") end},
            scrolldown = {type = "button", texture = 'construction/arrow_btn', x = 3, y = 10, y_bottom = 10, type_helpers = "RightTop", code = function(self) import('/mods/uimod/sca_utils/uimod.lua').TempScrollArrowFctn(self, "right") end},
            grid = {left = 66, top = 11, right = 13, bottom = 11},

            verticalbg = {type = "9part", texture = 'construction/cstr_vert_default'},
            tech1tab_v = {type = "checkbox", texture = 'tech buttons/tech-tab_btn_v', x = 1, y = -2},
            tech2tab_v = {type = "checkbox", texture = 'tech buttons/tech-tab_btn_v', x = 1, y = 20},
            tech3tab_v = {type = "checkbox", texture = 'tech buttons/tech-tab_btn_v', x = 1, y = 42},
            tech4tab_v = {type = "checkbox", texture = 'tech buttons/tech-tab_btn_v', x = 1, y = 64},
            tech1icon_v = {texture = 'tech buttons/tech-1_icon.dds', type_helpers = 'Center'},
            tech2icon_v = {texture = 'tech buttons/tech-2_icon.dds', type_helpers = 'Center'},
            tech3icon_v = {texture = 'tech buttons/tech-3_icon.dds', type_helpers = 'Center'},
            tech4icon_v = {texture = 'tech buttons/tech-4_icon.dds', type_helpers = 'Center'},
            enhancebtn_v = {type = "button", texture = 'construction/enhance_btn_alt', x = 5, y = 93},
            scrollup_v = {type = "button", texture = 'construction/arrow-up_scr', x = 7, y = 124},
            scrolldown_v = {type = "button", texture = 'construction/arrow-down_scr', x = 7, y = 153},
            grid_v = {left = 45, top = 3, right = 3, bottom = 3},

            tech1tab_mini = {type = "checkbox", texture = 'tech buttons/tech1-tab02_btn', x = -8, y = -31},
            tech2tab_mini = {type = "checkbox", texture = 'tech buttons/tech2-tab02_btn', x = 27, y = -31},
            tech3tab_mini = {type = "checkbox", texture = 'tech buttons/tech2-tab02_btn', x = 66, y = -31},
            tech4tab_mini = {type = "checkbox", texture = 'tech buttons/tech4-tab02_btn', x = 105, y = -31},
            tech1icon_mini = {texture = 'tech buttons/tech-1_icon.dds', type_helpers = 'Center'},
            tech2icon_mini = {texture = 'tech buttons/tech-2_icon.dds', type_helpers = 'Center'},
            tech3icon_mini = {texture = 'tech buttons/tech-3_icon.dds', type_helpers = 'Center'},
            tech4icon_mini = {texture = 'tech buttons/tech-4_icon.dds', type_helpers = 'Center'},
            enhancebtn_mini = {type = "button", texture = 'construction/enhance_btn_alt', x = 3, y = 3},
            scrollup_vmini = {type = "button", texture = 'construction/arrow-up_scr', x = 7, y = 52},
            scrolldown_vmini = {type = "button", texture = 'construction/arrow-down_scr', x = 7, y = 79},
    },
    queue = {
            bg = {type = "9part", texture = 'construction/cstr_horiz_default'},
            infinitebtn = {type = "checkbox", texture = 'queue/infinite_btn', x = 3, y = 3},
            pausebtn = {type = "checkbox", texture = 'queue/pause_btn', x = 28, y = 3},
            scrollup = {type = "button", texture = 'construction/arrow_btn', x = 55, y = 10, y_bottom = 10, code = function(self) import('/mods/uimod/sca_utils/uimod.lua').TempScrollArrowFctn(self, "left") end},
            scrolldown = {type = "button", texture = 'construction/arrow_btn', x = 3, y = 10, y_bottom = 10, type_helpers = "RightTop", code = function(self) import('/mods/uimod/sca_utils/uimod.lua').TempScrollArrowFctn(self, "right") end},
            grid = {left = 66, top = 11, right = 13, bottom = 11},

            verticalbg = {type = "9part", texture = 'construction/cstr_vert_default'},
            infinitebtn_v = {type = "checkbox", texture = 'queue/infinite_btn_v', x = 2, y = 3},
            pausebtn_v = {type = "checkbox", texture = 'queue/pause_btn_v', x = 2, y = 28},
            scrollup_v = {type = "button", texture = 'construction/arrow-up_scr', x = 7, y = 52},
            scrolldown_v = {type = "button", texture = 'construction/arrow-down_scr', x = 7, y = 79},
            grid_v = {left = 45, top = 3, right = 3, bottom = 3},
```

Appendix II : An example of a faction skin

```
    SCA_Aeon={
        path = '/mods/uimod/skins/sca/aeon goom/',
        info = 'Minimalist skin based on the Aeon faction.',
        equivalentSkins = {   'Default', 'SCA_Cybran', 'SCA_UEF',},
        icon = 'aeon_icon.dds',
        mfd = {
                bg = {texture = 'mfd/mfd_horiz_button_aeon_bmp.dds'},
                verticalbg = {texture = 'mfd/mfd_button_aeon_bmp.dds'},
        },
        minimap = {
                glow = {type = '9part', texture = 'minimap/minimapFG_aeon'},
        },
        options = {
                horizontal = {texture = 'options/options_bg_aeon_horiz.dds'},
                vertical = {texture = 'options/options_bg_aeon_vert.dds'},
        },
        mass = {
                bg = {texture = 'resources/mass-back_aeon_bmp.dds'},
                verticalbg = {texture = 'resources/alt-econ-m-back_aeon.dds'},
        },
        energy = {
                bg = {texture = 'resources/energy-back_aeon_bmp.dds'},
                bg_i = {texture = 'resources/mass-back_aeon_bmp.dds'},
                verticalbg = {texture = 'resources/alt-econ-e-back_aeon.dds'},
```

```
        },
        orders = {
                ['3by4'] = {texture = 'orders/orders-34_aeon.dds'},
                ['4by3'] = {texture = 'orders/orders-43_aeon.dds'},
        },
        selection = {
                bg = {type="9part", texture="construction/cstr_horiz_aeon"},
                verticalbg = {type="9part", texture="construction/cstr_vert_aeon"},
        },
        construction = {
                bg = {type="9part", texture="construction/cstr_horiz_aeon"},
                verticalbg = {type="9part", texture="construction/cstr_vert_aeon"},
        },
        queue = {
                bg = {type="9part", texture="construction/cstr_horiz_aeon"},
                verticalbg = {type="9part", texture="construction/cstr_vert_aeon"},
        },
    }
```

As you can see, once all the hard work has been done of creating a base skin in the first place, it's very little code to add another skin based on it to be used as a factional skin. The field equivalentSkins ={} is to say which skins this skin should share layouts with for the layout manager.

Appendix III : code examples

There is two possibilities of code: one for the skin part and one for the layout part; here some examples.

a. Skin code:

You should add at the top of your file, like in sca.skin.

```
-- don't remove these lines, could obj be use to make calculus on object
local skinning= import('/mods/uimod/sca_utils/skinning.lua')
local obj = skinning.obj
local LayoutHelpers = import('/lua/maui/layouthelpers.lua')
----------------------------
```

With theses lines you can make things like :

```
construction = {
       bg = {type = "9part", texture = 'construction/cstr_horiz_default'},
          tech1tab = {
                type = "checkbox", texture = 'tech buttons/tech-tab_btn', x = 2, y = -2,
code=function(self)
       self.Top:Set(function() return obj.construction.bg.Top() -
             math.min(math.floor(obj.construction.bg.Height()/110), 1) end)
       self.Bottom:Set(function() return self.Top() +
             math.min(obj.construction.bg.Height()/4.2, 30) end)
end},

          tech2tab = {
       type = "checkbox", texture = 'tech buttons/tech-tab_btn', x = 2, y = 20,
code=function(self)
       self.Top:Set(function() return obj.construction.tech1tab.Bottom() -
             math.min(math.floor(obj.construction.bg.Height()/18), 6) end)
       self.Bottom:Set(function() return self.Top() +
             math.min(obj.construction.bg.Height()/4.2, 30) end)
 end},
  },
```

Here the checkbox tech1 and the tech2 are calculated with the height of the construction windows. In a more simple way, if you use the new code, you will align the top of the bitmap to the top of construction and put his bottom 10 pixels under (so setting is height to 10).

```
code=function(self)
       self.Top:Set(function() return obj.construction.bg.Top() - 10 end)
       self.Bottom:Set(function() return self.Top() + 10) end)
end},
```

b. Layout Code:

```
code = function()
   if import('/lua/ui/game/multifunction.lua').controls.miniMap.BG:IsHidden() then

import('/lua/ui/game/construction.lua').controls.constructionGroup.Resizer.Left:Set(import('/lua/ui/game/m
ultifunction.lua').controls.bg.Resizer.Right)

import('/lua/ui/game/queue.lua').controls.queueGroup.Resizer.Left:Set(import('/lua/ui/game/multifunction.l
ua').controls.bg.Resizer.Right)
   end
   import('/lua/ui/game/multifunction.lua').controls.miniMap.BG.OnHide = function(self, hidden)
      if hidden then

import('/lua/ui/game/construction.lua').controls.constructionGroup.Resizer.Left:Set(import('/lua/ui/game/m
ultifunction.lua').controls.bg.Resizer.Right)

import('/lua/ui/game/queue.lua').controls.queueGroup.Resizer.Left:Set(import('/lua/ui/game/multifunction.l
ua').controls.bg.Resizer.Right)
      else

import('/lua/ui/game/construction.lua').controls.constructionGroup.Resizer.Left:Set(import('/lua/ui/game/m
ultifunction.lua').controls.miniMap.BG.Resizer.Right)

import('/lua/ui/game/queue.lua').controls.queueGroup.Resizer.Left:Set(import('/lua/ui/game/multifunction.l
ua').controls.miniMap.BG.Resizer.Right)
      end
   end
end,
```

This is the code we use for our GPG bottom layout to reposition the construction and queue according to if the minimap is hidden. Basically it does this:

```
if 'minimap is hidden' then
     Set construction and queue left to mfd right
Set minimap hide function:
     If hidden then
            Set construction and queue left to mfd right
     Otherwise
            Set construction and queue left to minimap right
```

If you didn't understand a word of that, but you want to include this kind of behaviour in your skin, then don't hesitate to contact Goom or Cleopatre on the GPG forums and we'll help you out with it.

SCA TEAM:

- Saya, Cleopatre, Goom and Associate

For any remarks, questions or suggestions, you can contact us on the gpgnet forums at http://forums.gaspowered.com/viewtopic.php?t=59&start=0