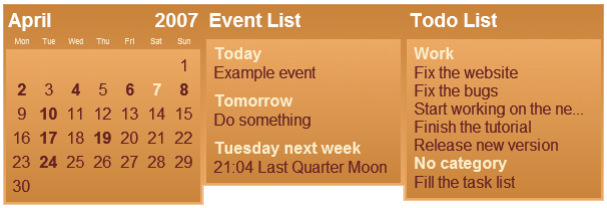
# Skinning Tutorial




In this tutorial we'll create a very simple calendar skin from a scratch. The skin will not use any images so it should be quite easy to modify. You should have some basic understanding of xml files to be able to follow the tutorial.

Before you start to create skins you should make sure that you have a text editor which supports utf-8 encoding. Do not edit the files with Notepad! Get a proper editor like Notepad++ instead. In Linux gedit or kate should be fine.

Note that if you are trying to edit an existing skin you can change the .r2skin file extension to .zip and uncompress the file to access the xml and image files.

To start we should create a folder for our new skin. So, go to Rainlendar's skins subfolder in the application folder and create a new folder. We'll call it *Nahka* (that's "skin" in Finnish :-). This will also be the name of our skin.

Inside the *Nahka* folder create a new text file. The name of the file must be *skin.xml*. Rainlendar is able to use the xml-skins also in uncompressed form which makes things easier during the skin development phase as we can just refresh the calendar to see the changes.

Now, open the *skin.xml* with a text editor.

The xml file needs a specific header so just copy the text below to the editor.

```
<?xml version="1.0" encoding="UTF-8"?>
```

The main tag in Rainlendar2 skins is <skin> which must contain all the other tags. So, lets add it.

```
<skin version="1.0">
</skin>
```

Note the version="1.0" in the opening tag. It defines the version of Rainlendar r2skin format. Currently it should always be set to "1.0" but if in the future the skin format changes this value is used to determine the format details. Note that the version="1.0" in the <?xml> tag is completely different so do not change that.

All skins need to have an info section which defines the name of the skin and some other details. Here's an example

```
<info>
  <version>1.0</version>
  <author>Rainy</author>
  <email>rainy@iki.fi</email>
  <homepage>https://www.rainlendar.net</homepage>
  <comment>The result of the skinning tutorial.</comment>
</info>
```

The version element contains the version number of your skin. You can increase it when you make changes to the skin so that the people who use your skin know that it has been updated. The email and homepage elements can be left out if you don't want to publish that kind of information. The comment element can contain e.g. instructions how to use the skin or whatever you want to put in there.

Prev - Next

---

The skin.xml should be like this now (it doesn't work yet so no need to test it):

```
<?xml version="1.0" encoding="UTF-8"?>
<skin version="1.0">
  <info>
    <version>1.0</version>
    <author>Rainy</author>
    <email>rainy@iki.fi</email>
    <homepage>https://www.rainlendar.net</homepage>
    <comment>The result of the skinning tutorial.</comment>
  </info>
</skin>
```

## The Calendar Window

Before we start to create the main calendar window we have to define the graphical elements for it. These include the fonts and background graphics. The graphical elements are defined with <element> tag. First we'll create just a font which is used to draw the days on the calendar.

```
<elements>
  <font id="font.normal" facename="Arial" size="12" />
</elements>
```

It *id* attribute is the identification string which is used when the element is referred. It can be any string as long as you don't use the same string twice.

Then we'll create the window for the calendar. A window is just a container for items. It doesn't know or care about the things it contains so there are no calendar windows, event list windows and such; there are just windows which contain items. The window is not limited to contain just a single item. In this example it actually will have multiple items (calendar, year, month and couple of images) and it could also contain the event list and todo list as well.

A window is defined with <window> tags.

```
<window id="Calendar" w="200" h="200" threshold="48" default="1">
</window>
```

The *id* is again the identifier of the window. It also defines the name of the window which is shown in the skin options. The *w* and *h* are the width and height of the window. It's not always necessary to define those as Rainlendar can also automatically calculate the dimensions. However this does not work unless the contents has a specific size and position, so unless you're making a window which can have different size (e.g. by using a event or todo list) it's useful to define the exact dimensions for the window.

The *threshold* is used with the region transparency and it defines the level (0-255) for the transparent pixels. To define a good value for this switch to the region transparency from the advanced options and see which value gets the best results.

If the *default* attribute is set to 1 the window is shown by default when the skin is selected in the simple mode. You should set it for all the common windows (calendar, event list, todo list).

Inside the window we'll create the calendar.

```
<calendar id="Calendar.calendar" x="0" y="0" w="200" h="200" layout="GRID">
  <days>
    <appearance layer="0" priority="0" element="font.normal" color="102, 35, 36" align="CENTER" />
  </days>
  <today>
    <appearance layer="0" priority="100" element="font.normal" color="249, 237, 204" align="CENTER" />
  </today>
  <events>
    <appearance layer="0" priority="1" element="font.normal" color="102, 35, 36" align="CENTER" />
  </events>
</calendar>
```

The calendar is defined to fill the whole window for now. The *layout* defines how the days are laid out on the calendar. The GRID is a standard 7x6 table.

The <days>, <today> and <events> define how the normal days, today and events should look like. We'll use the font, which was defined in the <elements>, for all of them. All the appearances use the same layer which means that only the one which has the highest priority gets drawn on one particular day. In our case the today has very high priority so it's drawn no matter what. If a day has an event its appearance is drawn on the calendar and if not the day will show just the normal day number. Note that the event priority gets adjusted by the event repeat type and the duration of the event so even if we define it as 1 it can have values from 1 to 51 + length in days (see the skin reference for details). This is also the reason why the today's priority is 100 instead of e.g. 2.

The *align* attribute sets the numbers to the center of the calendar grid's cells.

You might have noticed that I used the same color for the events and normal days which isn't very clever since it's not possible to see which days have events. We'll fix that by defining a new font and using it in the events and today.

```
<elements>
  <font id="font.normal" facename="Arial" size="12" />
  <font id="font.bold" facename="Arial" weight="BOLD" size="12" />
</elements>

<calendar id="Calendar.calendar" x="0" y="0" w="200" h="200" layout="GRID">
  <days>
    <appearance layer="0" priority="0" element="font.normal" color="102, 35, 36" align="CENTER" />
  </days>
  <today>
    <appearance layer="0" priority="100" element="font.bold" color="249, 237, 204" align="CENTER" />
  </today>
  <events>
    <appearance layer="0" priority="1" element="font.bold" color="102, 35, 36" align="CENTER" />
  </events>
</calendar>
```

We can now start Rainlendar and take a look at our (unfinished) skin. Just go to Options->Skins and you should see the MyFirstSkin in the list. If you are in the Advanced mode you should switch to the Simple mode. Select the skin from the list and you should see the details that you added to the <info> element for the skin. Click OK and you should see a window that looks something like this:


Quite plain looking and the desktop wallpaper shows through so it's pretty hard to move the window or open the context menu.

Next we'll do something to the background so that the text is more visible. To do this we need to create the background element first and then use it in the window. For background we'll be using two gradients which go to opposite directions.

The elements:

```
<elements>
  <font id="font.normal" facename="Arial" size="12" />
  <font id="font.bold" facename="Arial" weight="BOLD" size="12" />
  <gradient id="gradient.background1" color1="201, 131, 59" color2="237, 170, 100" direction="VERTICAL" />
  <gradient id="gradient.background2" color1="237, 170, 100" color2="201, 131, 59" direction="VERTICAL" />
</elements>
```

...and the window:

```
<window id="Calendar" w="200" h="200" threshold="48" default="1">
  <image id="Calendar.background1" element="gradient.background1" x="0" y="0" w="200" h="200" />
  <image id="Calendar.background2" element="gradient.background2" x="3" y="3" w="194" h="194" />
  <month id="Calendar.month" element="font.large" x="0" y="0" ALIGN="TOP-LEFT" />
  <year id="Calendar.year" element="font.large" x="200" y="0" ALIGN="TOP-RIGHT" />
  <calendar id="Calendar.calendar" x="0" y="0" w="200" h="200" layout="GRID">
    <days>
      <appearance layer="0" priority="0" element="font.normal" color="102, 35, 36" align="CENTER" />
    </days>
    <today>
      <appearance layer="0" priority="100" element="font.bold" color="249, 237, 204" align="CENTER" />
    </today>
    <events>
      <appearance layer="0" priority="1" element="font.bold" color="102, 35, 36" align="CENTER" />
    </events>
  </calendar>
</window>
```

You should note that the background images must be the first entry for the window. The items are drawn in the same order as they are defined so if you overlap the items the one you want to have behind the other must be defined first.

Our calendar window looks now like this:


Now we'll have to make some room for the month and the year. We'll also use a slightly larger font so let's add the definition to the elements section.

```
<elements>
  <font id="font.normal" facename="Arial" size="12" />
  <font id="font.bold" facename="Arial" weight="BOLD" size="12" />
  <gradient id="gradient.background1" color1="201, 131, 59" color2="237, 170, 100" direction="VERTICAL" />
  <gradient id="gradient.background2" color1="237, 170, 100" color2="201, 131, 59" direction="VERTICAL" />
  <font id="font.large" facename="Arial" weight="BOLD" size="14" />
</elements>
```

The month and year go to the top part of the window so we need to move the calendar to start slightly lower. We'll also move the edges of the calendar slightly inwards so it fits better for the background.

```
<window id="Calendar" w="200" h="200" threshold="48" default="1">
  <image id="Calendar.background1" element="gradient.background1" x="0" y="0" w="200" h="200" />
  <image id="Calendar.background2" element="gradient.background2" x="3" y="33" w="194" h="164" />
  <month id="Calendar.month" element="font.large" x="5" y="5" align="TOP-LEFT" />
  <year id="Calendar.year" element="font.large" x="195" y="5" align="TOP-RIGHT" />
  <calendar id="Calendar.calendar" x="5" y="35" w="190" h="160" layout="GRID">
    <days>
      <appearance layer="0" priority="0" element="font.normal" color="102, 35, 36" align="CENTER" />
    </days>
    <today>
      <appearance layer="0" priority="100" element="font.bold" color="249, 237, 204" align="CENTER" />
    </today>
    <events>
      <appearance layer="0" priority="1" element="font.bold" color="102, 35, 36" align="CENTER" />
    </events>
  </calendar>
</window>
```

So, now we have month and year in our calendar. The month is in the left side of the calendar in the 5,5 coordinates and the year is on the right side in 195,5. The alignment of the year must be set to TOP-RIGHT so that the text flows to the left. Otherwise it would go outside the window.


The last thing we need to do to the calendar window is to add the weekdays over the day numbers. Rainlendar will position them automatically so we don't have to worry about that. What we need to do is to create a new font for the weekdays and adjust the position of the background on the calendar slightly (and define the weekdays of course).

```
<elements>
  <font id="font.normal" facename="Arial" size="12" />
  <font id="font.bold" facename="Arial" weight="BOLD" size="12" />
  <font id="font.large" facename="Arial" weight="BOLD" size="14" />
  <font id="font.small" facename="Arial" size="6" />
  <gradient id="gradient.background1" color1="201, 131, 59" color2="237, 170, 100" direction="VERTICAL" />
  <gradient id="gradient.background2" color1="237, 170, 100" color2="201, 131, 59" direction="VERTICAL" />
</elements>

<window id="Calendar" w="200" h="200" threshold="48" default="1">
  <image id="Calendar.background1" element="gradient.background1" x="0" y="0" w="200" h="200" />
  <image id="Calendar.background2" element="gradient.background2" x="3" y="43" w="194" h="154" />
  <month id="Calendar.month" element="font.large" x="5" y="5" align="TOP-LEFT" />
  <year id="Calendar.year" element="font.large" x="195" y="5" align="TOP-RIGHT" />
  <calendar id="Calendar.calendar" x="5" y="25" w="190" h="170" layout="GRID">
    <weekdays abbreviate="1">
      <appearance element="font.small" color="249, 237, 204" align="CENTER" />
    </weekdays>
    <days>
      <appearance layer="0" priority="0" element="font.normal" color="102, 35, 36" align="CENTER" />
    </days>
    <today>
      <appearance layer="0" priority="100" element="font.bold" color="249, 237, 204" align="CENTER" />
    </today>
    <events>
      <appearance layer="0" priority="1" element="font.bold" color="102, 35, 36" align="CENTER" />
    </events>
  </calendar>
</window>
```


The *abbreviate* attribute is used to shorten the weekday names since the full names wouldn't fit in the reserved space.

Well, that's it for the calendar window. Next we'll create the event and todo lists and the other windows.

Prev - Next

---

The skin.xml should be like this now:

```
<?xml version="1.0" encoding="UTF-8"?>
<skin version="1.0">

  <info>
    <version>1.0</version>
    <author>Rainy</author>
    <email>rainy@iki.fi</email>
    <homepage>https://www.rainlendar.net</homepage>
    <comment>The result of the skinning tutorial.</comment>
  </info>

  <elements>
    <font id="font.normal" facename="Arial" size="12" />
    <font id="font.bold" facename="Arial" weight="BOLD" size="12" />
    <font id="font.large" facename="Arial" weight="BOLD" size="14" />
    <font id="font.small" facename="Arial" size="6" />
    <gradient id="gradient.background1" color1="201, 131, 59" color2="237, 170, 100" direction="VERTICAL" />
    <gradient id="gradient.background2" color1="237, 170, 100" color2="201, 131, 59" direction="VERTICAL" />
  </elements>

  <window id="Calendar" w="200" h="200" threshold="48" default="1">
    <image id="Calendar.background1" element="gradient.background1" x="0" y="0" w="200" h="200" />
    <image id="Calendar.background2" element="gradient.background2" x="3" y="43" w="194" h="154" />
    <month id="Calendar.month" element="font.large" x="5" y="5" align="TOP-LEFT" />
    <year id="Calendar.year" element="font.large" x="195" y="5" align="TOP-RIGHT" />
    <calendar id="Calendar.calendar" x="5" y="25" w="190" h="170" layout="GRID">
      <weekdays abbreviate="1">
        <appearance element="font.small" color="249, 237, 204" align="CENTER" />
      </weekdays>
      <days>
        <appearance layer="0" priority="0" element="font.normal" color="102, 35, 36" align="CENTER" />
      </days>
      <today>
        <appearance layer="0" priority="100" element="font.bold" color="249, 237, 204" align="CENTER" />
      </today>
      <events>
        <appearance layer="0" priority="1" element="font.bold" color="102, 35, 36" align="CENTER" />
      </events>
    </calendar>
  </window>

</skin>
```

## The Other Windows

There are four other windows the skin still needs: Event list, Todo list, Alarm and the Tooltip. We could also define the tray icon but since it requires images we'll settle for the build-in icon.

For event list we'll create a new window, set the background for it and define the actual list item.

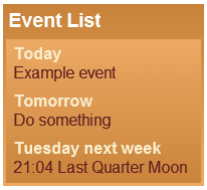
```
<window id="Event List" w="200" threshold="48" default="1">
  <image id="Event List.background1" element="gradient.background1" origin1="TOP-LEFT" x1="0" y1="0" origin2="BOTTOM-RIGHT" x2="0" y2="0" color="249, 237, 204" minsizefromimage="0" />
  <image id="Event List.background2" element="gradient.background2" origin1="TOP-LEFT" x1="3" y1="35" origin2="BOTTOM-RIGHT" x2="-3" y2="-3" color="249, 237, 204" minsizefromimage="0"
  <text id="Event List.header" element="font.large" x="5" y="5" align="TOP-LEFT" text="Event List" />
  <eventlist id="Event List.eventlist" origin1="TOP-LEFT" x1="10" y1="40" origin2="BOTTOM-RIGHT" x2="-10" y2="-10">
    <header>
      <appearance element="font.bold" color="249, 237,204" align="TOP-LEFT" />
    </header>
    <item>
      <appearance element="font.normal" color="102, 35, 36" align="TOP-LEFT" />
    </item>
    <separator id="Event List.separator" padding="0, 10, 0, 0" />
  </eventlist>
</window>
```

Couple of things you should note about the event list window: the window doesn't define any height and the background images and the event list do not define absolute size. This is because the window will automatically

adjust its height from the contents of the list. The background images are defined so that it always fills the whole window from top left to bottom right. The *minsizefromimage* must be set to "0" for the images. This is because the gradient is really a 1x256 size image so unless we tell Rainlendar that it's ok to shrink the image it would set the minimum size of the window to 256 which wouldn't look too good.

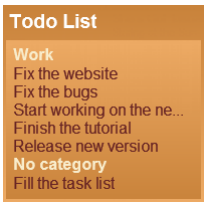
The text element just draw the header to the window. The text gets automatically translated too if it can be found from the translated strings.

The actual event list's position is defined the same way as the background but with some additional padding in the edges. Event list contains header and item which define the appearance for the texts. The separator is used to add some space between the items (it could also draw e.g. an image between the items).



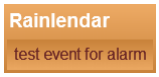

The todo list is very similar as the event list. In fact the only difference is the *id* attributes and that we'll create a <todolist> instead of <eventlist>.

```
<window id="Todo List" w="200" threshold="48" default="1">
  <image id="Todo List.background1" element="gradient.background1" origin1="TOP-LEFT" x1="0" y1="0" origin2="BOTTOM-RIGHT" x2="0" y2="0" color="249, 237, 204" minsizefromimage="0" />
  <image id="Todo List.background2" element="gradient.background2" origin1="TOP-LEFT" x1="3" y1="35" origin2="BOTTOM-RIGHT" x2="-3" y2="-3" color="249, 237, 204" minsizefromimage="0"
  <text id="Todo List.header" element="font.large" x="5" y="5" align="TOP-LEFT" text="Todo List" />
  <todolist id="Todo List.todolist" origin1="TOP-LEFT" x1="10" y1="40" origin2="BOTTOM-RIGHT" x2="-10" y2="-10">
    <header>
      <appearance element="font.bold" color="249, 237, 204" align="TOP-LEFT" />
    </header>
    <item>
      <appearance element="font.normal" color="102, 35, 36" align="TOP-LEFT" />
    </item>
    <separator id="Todo List.separator" padding="0, 10, 0, 0" />
  </todolist>
</window>
```




The alarm window isn't much different than the rest.

```
<alarm id="Alarm" threshold="48" textmargins="10, 40, 10, 10">
  <image id="Alarm.background1" element="gradient.background1" origin1="TOP-LEFT" x1="0" y1="0" origin2="BOTTOM-RIGHT" x2="0" y2="0" color="249, 237, 204" minsizefromimage="0" />
  <image id="Alarm.background2" element="gradient.background2" origin1="TOP-LEFT" x1="3" y1="35" origin2="BOTTOM-RIGHT" x2="-3" y2="-3" color="249, 237, 204" minsizefromimage="0" />
  <text id="Alarm.header" element="font.large" x="5" y="5" align="TOP-LEFT" text="Rainlendar" />
  <item id="Alarm.item">
    <appearance element="font.normal" color="102, 35, 36" align="TOP-LEFT" />
  </item>
  <button id="Alarm.button.dismissall" origin1="TOP-LEFT" x1="0" y1="0" origin2="BOTTOM-RIGHT" x2="0" y2="0" action="Global_DismissAll()" />
</alarm>
```




Note that the alarm is not a <window> but uses a <alarm> tag instead. You never define the size of the alarm as it is calculated automatically. The *textmargins* define the area around the alarm window which is added as margins for the text. The alarm window can contain all the same items as a normal window but in addition to that it also defines the list's <item> element which contains the appearance for the alarm items.

In the alarm we also define a transparent button (it's transparent because we don't define any graphical element for it) which covers the whole window. Pressing the button will dismiss all alarms that the window contains. The *Global_DismissAll()* is the name of the script that is executed when the button is pressed. You can check the Rainlendar's scripts folder for additional scripts that can be executed. In the alarm window you might want to have also a button which snoozes the alarms.

The last window we'll have to define is the tooltip.

```
<tooltip id="Tooltip" threshold="48" textmargins="10, 10, 10, 10" offset="0, 0">
  <image id="Tooltip.background1" element="gradient.background1" origin1="TOP-LEFT" x1="0" y1="0" origin2="BOTTOM-RIGHT" x2="0" y2="0" color="249, 237, 204" minsizefromimage="0" />
  <image id="Tooltip.background2" element="gradient.background2" origin1="TOP-LEFT" x1="3" y1="3" origin2="BOTTOM-RIGHT" x2="-3" y2="-3" color="249, 237, 204" minsizefromimage="0" />
  <header>
    <appearance element="font.bold" color="102, 35, 36" align="TOP-LEFT" />
  </header>
  <item>
    <appearance element="font.normal" color="102, 35, 36" align="TOP-LEFT" />
  </item>
  <separator id="Tooltip.separator" padding="0, 10, 0, 0" />
</tooltip>
```


That's pretty much the same as the alarm window. The tooltip has a special element just like the alarm and it also defines the text margins. The *offset* defines how far the window is opened from the mouse pointer. This is useful if the tooltip has e.g. an arrow which should be under the mouse. In our example the tooltip is just a rectangle so we don't define any offset.

The rest is pretty much a copy from the alarm window with the exception that the tooltip doesn't have any title (it could though...).

Prev - Next

---

The finished skin.xml should be like this now:

```
<?xml version="1.0" encoding="UTF-8"?>
<skin version="1.0">
  <info>
    <version>1.0</version>
    <author>Rainy</author>
    <email>rainy@iki.fi</email>
    <homepage>https://www.rainlendar.net</homepage>
    <comment>The result of the skinning tutorial.</comment>
  </info>

  <elements>
    <font id="font.normal" facename="Arial" size="12" />
    <font id="font.bold" facename="Arial" weight="BOLD" size="12" />
    <font id="font.large" facename="Arial" weight="BOLD" size="14" />
    <font id="font.small" facename="Arial" size="6" />
    <gradient id="gradient.background1" color1="201, 131, 59" color2="237, 170, 100" direction="VERTICAL" />
    <gradient id="gradient.background2" color1="237, 170, 100" color2="201, 131, 59" direction="VERTICAL" />
  </elements>
```

```
  <window id="Calendar" w="200" h="200" threshold="48" default="1">
    <image id="Calendar.background1" element="gradient.background1" x="0" y="0" w="200" h="200" />
    <image id="Calendar.background2" element="gradient.background2" x="3" y="43" w="194" h="154" />
    <month id="Calendar.month" element="font.large" x="5" y="5" align="TOP-LEFT" />
    <year id="Calendar.year" element="font.large" x="195" y="5" align="TOP-RIGHT" />
    <calendar id="Calendar.calendar" x="5" y="25" w="190" h="170" layout="GRID">
      <weekdays abbreviate="1">
        <appearance element="font.small" color="249, 237, 204" align="CENTER" />
      </weekdays>
      <days>
        <appearance layer="0" priority="0" element="font.normal" color="102, 35, 36" align="CENTER" />
      </days>
      <today>
        <appearance layer="0" priority="100" element="font.bold" color="249, 237, 204" align="CENTER" />
      </today>
      <events>
        <appearance layer="0" priority="1" element="font.bold" color="102, 35, 36" align="CENTER" />
      </events>
    </calendar>
  </window>

  <window id="Event List" w="200" threshold="48" default="1">
    <image id="Event List.background1" element="gradient.background1" origin1="TOP-LEFT" x1="0" y1="0" origin2="BOTTOM-RIGHT" x2="0" y2="0" color="249, 237, 204" minsizefromimage="0"
    <image id="Event List.background2" element="gradient.background2" origin1="TOP-LEFT" x1="3" y1="35" origin2="BOTTOM-RIGHT" x2="-3" y2="-3" color="249, 237, 204" minsizefromimage="
    <text id="Event List.header" element="font.large" x="5" y="5" align="TOP-LEFT" text="Event List" />
    <eventlist id="Event List.eventlist" origin1="TOP-LEFT" x1="10" y1="40" origin2="BOTTOM-RIGHT" x2="-10" y2="-10">
      <header>
        <appearance element="font.bold" color="249, 237,204" align="TOP-LEFT" />
      </header>
      <item>
        <appearance element="font.normal" color="102, 35, 36" align="TOP-LEFT" />
      </item>
      <separator id="Event List.separator" padding="0, 10, 0, 0" />
    </eventlist>
  </window>

  <window id="Todo List" w="200" threshold="48" default="1">
    <image id="Todo List.background1" element="gradient.background1" origin1="TOP-LEFT" x1="0" y1="0" origin2="BOTTOM-RIGHT" x2="0" y2="0" color="249, 237, 204" minsizefromimage="0" /
    <image id="Todo List.background2" element="gradient.background2" origin1="TOP-LEFT" x1="3" y1="35" origin2="BOTTOM-RIGHT" x2="-3" y2="-3" color="249, 237, 204" minsizefromimage="0
    <text id="Todo List.header" element="font.large" x="5" y="5" align="TOP-LEFT" text="Todo List" />
    <todolist id="Todo List.todolist" origin1="TOP-LEFT" x1="10" y1="40" origin2="BOTTOM-RIGHT" x2="-10" y2="-10">
      <header>
        <appearance element="font.bold" color="249, 237, 204" align="TOP-LEFT" />
      </header>
      <item>
        <appearance element="font.normal" color="102, 35, 36" align="TOP-LEFT" />
      </item>
      <separator id="Todo List.separator" padding="0, 10, 0, 0" />
    </todolist>
  </window>

  <tooltip id="Tooltip" threshold="48" textmargins="10, 10, 10, 10" offset="0, 0">
    <image id="Tooltip.background1" element="gradient.background1" origin1="TOP-LEFT" x1="0" y1="0" origin2="BOTTOM-RIGHT" x2="0" y2="0" color="249, 237, 204" minsizefromimage="0" />
    <image id="Tooltip.background2" element="gradient.background2" origin1="TOP-LEFT" x1="3" y1="3" origin2="BOTTOM-RIGHT" x2="-3" y2="-3" color="249, 237, 204" minsizefromimage="0" /
    <header>
      <appearance element="font.bold" color="102, 35, 36" align="TOP-LEFT" />
    </header>
    <item>
      <appearance element="font.normal" color="102, 35, 36" align="TOP-LEFT" />
    </item>
    <separator id="Tooltip.separator" padding="0, 10, 0, 0" />
  </tooltip>

  <alarm id="Alarm" threshold="48" textmargins="10, 40, 10, 10">
    <image id="Alarm.background1" element="gradient.background1" origin1="TOP-LEFT" x1="0" y1="0" origin2="BOTTOM-RIGHT" x2="0" y2="0" color="249, 237, 204" minsizefromimage="0" />
    <image id="Alarm.background2" element="gradient.background2" origin1="TOP-LEFT" x1="3" y1="3" origin2="BOTTOM-RIGHT" x2="-3" y2="-3" color="249, 237, 204" minsizefromimage="0" />
    <text id="Alarm.header" element="font.large" x="5" y="5" align="TOP-LEFT" text="Rainlendar" />
    <item id="Alarm.item">
      <appearance element="font.normal" color="102, 35, 36" align="TOP-LEFT" />
    </item>
    <button id="Alarm.button.dismissall" origin1="TOP-LEFT" x1="0" y1="0" origin2="BOTTOM-RIGHT" x2="0" y2="0" action="Global_DismissAll()" />
  </alarm>

</skin>
```

## Finishing Things Up

The last thing we should do is to package the skin into a r2skin file. The file is a standard zip-archive so we can use e.g. the build-in archiving functionalty in Windows. When creating the package we must make sure that the skin.xml is in the root folder of the archive. If there is an extra subfolder in the archive Rainlendar is not able to find the skin.xml from it and the skin won't be visible in the skin list.

In Windows you can right click on the skin.xml and select "Send to"->"Compressed (zipped) Folder" (if your skin contains multiple files, select them all). This will create a new file called skin.zip into the *Nahka* folder. Rename it to Nahka.r2skin and move it to the parent folder. You should also delete or rename the old *Nahka* folder. The Nahka.r2skin is now ready to be distributed to other users.

The Nahka-skin is all in single file but if you are implementing a complex skin it might be useful to define the different windows in separate xml-files so that the file doesn't grow too big (Rainlendar doesn't mind about that but it's probably easier to find things if the files are smaller). You can check the Shadow4-skin as an example how to do that.

Prev - Next