

Hub One – Mobility

# CheapToLight

User Guide

David DESVAUX



**Hub One**  
Une connexion d'avance

**Document change log**

Date	Author(s)	Change description
2014-12-24	David DESVAUX	Creation
2015-01-05	David DESVAUX	Added things...
2015-01-07	David DESVAUX	Added property "id" to "Imageltem" elements.
2015-01-15	David DESVAUX	Added installation informations

## Summary

<b>1</b>	<b>Introduction .....</b>	<b>4</b>
<b>2</b>	<b>Features .....</b>	<b>4</b>
2.1	Layouts .....	4
2.2	Elements.....	5
2.2.1	Displayer elements.....	6
2.2.2	Items.....	10
2.3	Miscellaneous options.....	12
2.3.1	Navigation bar .....	12
2.3.2	Battery level .....	12
2.3.3	Wi-Fi signal strength.....	12
2.3.4	Beep on refresh .....	13
2.4	Image list .....	14
2.5	Color mapping .....	15
2.6	Template parameters.....	16
2.7	Magic words .....	17

## 1 Introduction

The CheapToLight is meant to be used as an indicator showing where to pick or put objects. It can display arrows, texts, images or other elements which can be configured and placed into layouts. It can be controlled remotely using the Wireless Telnet.

## 2 Features

### 2.1 Layouts

A layout can be seen as a “page” containing several elements. A layout has a grid with a certain number of rows and columns. Elements can be placed freely, but this grid can be used to ease their placement.

The configuration can define multiple layouts, each of them having its own elements placed in different ways. The CheapToLight displays a single layout at a time, and the layout currently displayed can be changed at any time.

A layout has the same size as the screen minus optional bars like battery level.

#### Properties :

Name	Type	Description
<b>rows</b>	Integral	Number of rows of the grid.
<b>cols</b>	Integral	Number of columns of the grid.
<b>margin</b>	Integral	<i>(optional)</i> Margin between the cells, in pixels. Only affects elements positioned or sized using the grid.
<b>landscape</b>	Boolean	<i>(optional)</i> Indicates whether the CheapToLight is used in landscape mode, i.e whether the screen is rotated horizontally. This has an impact on things such as the orientation of the text in certain displayer elements.

#### Example :

```
<?xml version="1.0" encoding="UTF-8" ?>
<CheapToLight>
  <Layout rows="3" cols="3" margin="10" landscape="true">
    <Displayers></Displayers>
    <Items></Items>
  </Layout>
</CheapToLight>
```

The tags “Displayers” and “Items” can be omitted if the layout does not contain any displayer or item, respectively.

## 2.2 Elements

There are two categories of elements : displayers and items.

Displayers can be controlled using the Signature's PickToLight protocol, just like real displayers and are compatible with any software implementing that protocol.

An element is drawn within the rectangle defined by it's width and height. Anything that overflows that rectangle is not displayed.

All elements have several properties in common : the position and the size. Each of these properties can be defined by 3 different and **mutually exclusive** ways :

- Absolutely : a value in number of pixels ;
- Relatively : a percentage of the layout's size ;
- By coordinates : a number of cells in the layout's grid.

### Properties :

Position on horizontal axis, one of :

Name	Type	Description
<b>absx</b>	Integral	Number of pixels between the left side of the screen and the element.
<b>relx</b>	Real	Percentage of the screen width between the left side of the screen and the element. 1.0 is the screen width, 0.5 is half, etc...
<b>col</b>	Integral	Column in the layout's grid, starting from 0 for the first one.

Position on vertical axis, one of :

Name	Type	Description
<b>absy</b>	Integral	Number of pixels between the top side of the screen and the element.
<b>rely</b>	Real	Percentage of the screen height between the top side of the screen and the element. 1.0 is the screen height, 0.5 is half, etc...
<b>row</b>	Integral	Row in the layout's grid, starting from 0 for the first one.

Width, one of :

Name	Type	Description
<b>absw</b>	Integral	Width in pixels.
<b>relw</b>	Real	Width in percentage of the screen width. 1.0 is the screen width, 0.5 is half, etc...
<b>colspan</b>	Integral	Width in number of cells (the margins are added to that width).

Height, one of :

Name	Type	Description
<b>absh</b>	Integral	Height in pixels.
<b>relh</b>	Real	Width in percentage of the screen height. 1.0 is the screen height, 0.5 is half, etc...
<b>rowspan</b>	Integral	Height in number of cells (the margins are added to that width).

### 2.2.1 Displayer elements

Displayer elements are placed in the tag “Displayers” of the layout definition.

All displayers have several properties in common.

**Properties :**

Name	Type	Description
<b>addr</b>	Integral	Address (ID) to assign to the displayer.
<b>on</b>	Boolean	<i>(optional)</i> Power state, “true” for ON or “false” for OFF.
<b>color</b>	String	<i>(optional)</i> Color definition (see the color definition chapter).
<b>quantity</b>	String	<i>(optional)</i> Short text to display.

If the layout is in landscape mode, the quantities are rotated so that they can be read while the device screen is in landscape mode too.

### 2.2.1.1 Arrow displayer

Arrow displayers display an arrow and a quantity inside a disc. An angle can be specified for the arrow.

Name	Type	Description
<b>angle</b>	Integral	( <i>optional</i> ) Angle in degrees. Use a positive value for clockwise rotation and a negative value for counter clockwise.

Example :

```
<?xml version="1.0" encoding="UTF-8" ?>
<CheapToLight>
  <Layout rows="3" cols="3" margin="10" landscape="true">
    <Displayers>
      <ArrowDisplayer
        addr="1"
        on="true"
        row="0"
        col="0"
        rowspan="1"
        colspan="1"
        angle="-45"
        quantity="1A 2B" />
    </Displayers>
  </Layout>
</CheapToLight>
```

### 2.2.1.2 Uniform displayer

A displayers that shows a background of uniform color with a quantity inside a disc. No extra properties for this type.

Example :

```
<?xml version="1.0" encoding="UTF-8" ?>
<CheapToLight>
  <Layout rows="1" cols="1">
    <Displayers>
      <UniformDisplayer
        addr="1"
        on="true"
        row="0"
        col="0"
        rowspan="1"
        colspan="1"
        quantity="0" />
    </Displayers>
  </Layout>
</CheapToLight>
```

### 2.2.1.3 Image displayer

Image displayers can display images but no quantity. The image must exist in the image list (see the image list chapter). The ID of the image to display must be specified using the property “**quantity**” which is not an optional property in the case of an image displayer. An angle can also be specified. See also the “ImageItem” chapter.

Name	Type	Description
<b>angle</b>	Integral	( <i>optional</i> ) Angle in degrees. Use a positive value for clockwise rotation and a negative value for counter clockwise.

Example :

```
<?xml version="1.0" encoding="UTF-8" ?>
<CheapToLight>
  <Images>
    <Image id="123" src="$CTL/image.png" />
  </Images>
  <Layout rows="2" cols="3" margin="30">
    <Displayers>
      <ImageDisplayer
        addr="0"
        on="true"
        row="0"
        col="0"
        rowspan="1"
        colspan="1"
        angle="45"
        quantity="123" />
    </Displayers>
  </Layout>
</CheapToLight>
```

Using the quantity as an image identifier can be counter intuitive, but this is a mandatory constraint in order to keep the compatibility with the Signature’s PickToLight protocol.



### 2.2.1.4 Text displayer

Text displayers can display a text but no quantity over it. The text is defined using the property “quantity”. When using the Signature’s protocol, the text size is limited to 8 characters. For longer texts use a “TextItem” instead (see the “TextItem” chapter). The font size, name and colors and an angle can be specified.

Name	Type	Description
<b>font</b>	String	( <i>optional</i> ) Font name to use.
<b>size</b>	Integral	( <i>optional</i> ) Font size (height) in pixels.
<b>color</b>	String	( <i>optional</i> ) Color definition (see the color definition chapter).
<b>angle</b>	Integral	( <i>optional</i> ) Angle in degrees. Use a positive value for clockwise rotation and a negative value for counter clockwise.

Example :

```
<?xml version="1.0" encoding="UTF-8" ?>
<CheapToLight>
  <Layout rows="2" cols="1">
    <Displayers>
      <TextDisplayer
        addr="0"
        on="true"
        row="0"
        rely="0.2"
        absw="100"
        absh="150"
        font="Comic Sans MS"
        size="60"
        color="#FF7700"
        angle="45"
        quantity="bla bla" />
    </Displayers>
  </Layout>
</CheapToLight>
```

## 2.2.2 Items

Item elements are placed in the tag “Item” of the layout definition.

### 2.2.2.1 ImageItem

Displays an image. Unlike for the image displayer, the image is identified by either its file path using the property “src” or an identifier using the property “id” to indicate an image defined in the image list (see the image list chapter). The property “src” has the priority over the property “id”. Either one of these two properties is required. An angle can also be specified.

Name	Type	Description
<b>angle</b>	Integral	( <i>optional</i> ) Angle in degrees. Use a positive value for clockwise rotation and a negative value for counter clockwise.
<b>src</b>	String	Path to the image file.
<b>id</b>	Integral	ID of an image defined in the image list.

Example :

```
<?xml version="1.0" encoding="UTF-8" ?>
<CheapToLight>
  <Images>
    <Image id="5" src="$CTL/image.png" />
  </Images>
  <Layout rows="1" cols="1">
    <Items>
      <ImageItem
        absx="0"
        absy="0"
        relw="1.0"
        relh="0.5"
        angle="45"
        src="$CTL/image.png" />
      <ImageItem
        absx="0"
        absy="0.5"
        relw="1.0"
        relh="0.5"
        angle="45"
        id="5" />
    </Items>
  </Layout>
</CheapToLight>
```

### 2.2.2.2 TextItem

Displays a text of any length. The font size, name and colors and an angle can be specified.

Space characters (white space, tabulation, line returns) are ignored before and after the text, but those mixed within the text are preserved.

Name	Type	Description
<b>font</b>	String	Font name to use.
<b>size</b>	Integral	Font size (height) in pixels.
<b>color</b>	String	Color definition (see the color definition chapter).
<b>angle</b>	Integral	Angle in degrees. Use a positive value for clockwise rotation and a negative value for counter clockwise.

Example :

```
<?xml version="1.0" encoding="UTF-8" ?>
<CheapToLight>
  <Layout rows="1" cols="1">
    <Items>
      <TextItem
        relx="0.1"
        rely="0.3"
        relw="0.8"
        relh="0.6"
        font="Comic Sans MS"
        size="60"
        color="#FF7700"
        angle="45">
        Text to display here
      </TextItem>
    </Items>
  </Layout>
</CheapToLight>
```

## 2.3 Miscellaneous options

### 2.3.1 Navigation bar

It is possible to either show or hide the navigation bar using the property “**ShowNavigationBar**” of type boolean.

Exemple :

```
<?xml version="1.0" encoding="UTF-8" ?>
<CheapToLight>
  <Options>
    <ShowNavigationBar value="true" />
  </Options>
</CheapToLight>
```

### 2.3.2 Battery level

It is possible to either show or hide the battery level bar using the property “**ShowBatteryLevel**” of type boolean.

Exemple :

```
<?xml version="1.0" encoding="UTF-8" ?>
<CheapToLight>
  <Options>
    <ShowBatteryLevel value="false" />
  </Options>
</CheapToLight>
```

### 2.3.3 Wi-Fi signal strength

It is possible to either show or hide the signal strength bar using the property “**ShowSignalStrength**” of type boolean.

Exemple :

```
<?xml version="1.0" encoding="UTF-8" ?>
<CheapToLight>
  <Options>
    <ShowSignalStrength value="true" />
  </Options>
</CheapToLight>
```

### 2.3.4 Beep on refresh

It is possible to emit beeps when something is updated on the current layout using the property “**BeepOnRefresh**” of type boolean. If nothing has changed, no beep is emitted.

Example :

```
<?xml version="1.0" encoding="UTF-8" ?>
<CheapToLight>
  <Options>
    <BeepOnRefresh value="true" />
  </Options>
</CheapToLight>
```

## 2.4 Image list

A list of images can be defined to associate a file path and a **unique** image ID. This ID can then be used to reference an image in that list. This is used by the “ImageDisplayer” element. Each image has the following properties :

Name	Type	Description
<b>id</b>	Integral	Positive integral value identifying the image.
<b>src</b>	String	Absolute or relative path to the image file.

```
<?xml version="1.0" encoding="UTF-8" ?>
<CheapToLight>
  <Images>
    <Image id="111" src="$CTL/image-111.png" />
    <Image id="222" src="$CTL/thumbs/image-222.png" />
    <Image id="333" src="../image-333.png" />
  </Images>
</CheapToLight>
```

See also the chapter about the “ImageDisplayer” element.

## 2.5 Color mapping

Colors must be assigned to codes for the Signature's protocol. A color must be assigned to every code that can be received through that protocol.

Colors can be defined in 3 different ways :

- RGB as hexadecimal string : “#RRGGBB” ;
- ARGB as hexadecimal string “#AARRGGBB” (includes alpha, i.e transparency) ;
- Name : “red”, “blue”, “yellow”, etc...

Properties are :

Name	Type	Description
<b>code</b>	Integral	Positive integral value identifying the color.
<b>value</b>	String	Valid color definition.

```
<?xml version="1.0" encoding="UTF-8" ?>
<CheapToLight>
  <ColorMap>
    <Color code="0" value="red" /><!-- Red -->
    <Color code="1" value="#00FF00" /><!-- Green -->
    <Color code="2" value="#FF0000FF" /><!-- Blue -->
    <Color code="3" value="#7FFFFFF0" /><!-- Yellow -->
    <Color code="4" value="#FF00FF" /><!-- Magenta -->
    <Color code="5" value="cyan" /><!-- Cyan -->
    <Color code="6" value="#FF7F00" /><!-- Orange -->
    <Color code="7" value="#FFFFFFFF" /><!-- White -->
  </ColorMap>
</CheapToLight>
```

## 2.6 Template parameters

Parameters can be used inside the values of the properties of a layout and its elements. This allows to dynamically change certain properties.

Each layout has its own set of parameters. The parameters must respect the following syntax :

```
{{parameter-name}}
```

Each parameter can be given a default value which is used when the parameter is not found in the set of parameters of the layout. A default value is defined as follows :

```
{{parameter-name=default-value}}
```

The parameter name can include the characters in [a..z], [A..Z], [0..9], “-” (hyphen) and “\_” (underscore). The default value can include **any** character, except the pattern “}” which marks the end of the value. Anything after “}” is not considered to be part of the default value.

Parameters can only be used in the value of a property or in the text content of certain elements such as the “TextItem” element.

Parameters can be preceded or followed by other texts as long as the resulting value respects the property constraints. For example :

```
<?xml version="1.0" encoding="UTF-8" ?>
<CheapToLight>
  <Layout rows="{{NB_ROWS}}" cols="{{NB_COLS}}">
    <Items>
      <TextItem
        row="{{ROW=0}}"
        col="{{COL=0}}"
        relw="0.{{WIDTH_DECIMAL}}"
        relh="{{HEIGHT}}"
        font="Comic Sans MS"
        size="{{SIZE=60}}">
        Parameter of name "FOO" has a value of : {{FOO=bar}} and a
        default value of "bar".
      </TextItem>
    </Items>
  </Layout>
</CheapToLight>
```



## 2.7 Magic words

Magic words can be used inside file paths. Currently available magic words are :

- **\$CTL** : absolute path to the folder where the configuration is located, without a separator at the end. The path depends on the device in use. For example : `"/sdcard0/CheapToLight"`.

## 3 Installation

The CheapToLight is composed of an .APK file, and a dedicated folder containing its configuration as well as optionnal resources such as images.

To install the Cheap To Lightn follow these steps :

- 1) Place the .APK anywhere on the device and install it from the device ;
- 2) Locate the "internal storage" folder. This folder depends on the device in use. It can be for example : `"/storage/sdcard0/"` ;
- 3) Create a folder named "CheapToLight" in that folder;
- 4) Place the configuration file into that folder;
- 5) Run the CheapToLight.

If the CheapToLight folder or the configuration is not found, a pop-up will display a message containing the absolute path where the configuration file is supposed to be. This can be used if the location of the "internal storage" folder is not known.