

---

# **ttkthemes Documentation**

***Release v3.2.2***

**The ttkthemes authors**

**Feb 15, 2021**



---

## Contents:

---

<b>1</b>	<b>Authors</b>	<b>3</b>
1.1	GNU GPLv3 Themes . . . . .	3
1.2	Tcl-License Themes . . . . .	3
<b>2</b>	<b>Reference</b>	<b>5</b>
<b>3</b>	<b>Example</b>	<b>9</b>
<b>4</b>	<b>Themes</b>	<b>11</b>
4.1	Adapta . . . . .	12
4.2	Aquativo . . . . .	13
4.3	Arc . . . . .	14
4.4	Black . . . . .	15
4.5	Blue . . . . .	16
4.6	Breeze . . . . .	17
4.7	Clearlooks . . . . .	18
4.8	Elegance . . . . .	19
4.9	Equilux . . . . .	20
4.10	ITFT1 . . . . .	21
4.11	Keramik . . . . .	22
4.12	Kroc . . . . .	23
4.13	Plastik . . . . .	24
4.14	Radiance (Ubuntu) . . . . .	26
4.15	Scid themes . . . . .	29
4.16	Smog . . . . .	32
4.17	winxpblue . . . . .	33
4.18	yaru . . . . .	34
<b>5</b>	<b>Theming</b>	<b>35</b>
5.1	Choosing a theme . . . . .	35
5.2	Modifying a theme . . . . .	36
5.3	Examples . . . . .	36
5.4	Notes . . . . .	37
<b>6</b>	<b>Licenses</b>	<b>39</b>
<b>7</b>	<b>Installation</b>	<b>41</b>

7.1	PyPI . . . . .	41
7.2	Source . . . . .	41
<b>8</b>	<b>Theme Loading</b>	<b>43</b>
8.1	Loading . . . . .	43
8.2	Limitations . . . . .	44
8.3	Tcl-loading . . . . .	44
<b>9</b>	<b>License</b>	<b>45</b>
<b>10</b>	<b>Recommendations</b>	<b>47</b>
	<b>Index</b>	<b>49</b>

A group of themes for the ttk extensions for Tkinter gathered together by RedFantom and created by various authors.



The `ttkthemes` package contains many different themes created by many different authors. Some themes are even derived from other software, like `GTK-themes`.

### 1.1 GNU GPLv3 Themes

- `arc`: Copyright (c) 2015 Sergei Golovan
- `equilux`: Copyright (C) 2018 RedFantom
- `itft1`: Copyright (C) 2005 Jelco Huijser
- `ubuntu`: Copyright (C) Regents of the University of California, Sun Microsystems, Inc., Scriptics Corporation, and other parties. Copyright (C) 2018 RedFantom
- `yaru`: Copyright (c) 2020 RedFantom, 2015 Sergei Golovan, the [yaru\\_authors](#).

### 1.2 Tcl-License Themes

- `aquativo`: Copyright (C) 2004 Pat Thoyts
- `black`: Copyright (c) 2007-2008 Mats Bengtsson
- `blue`: Copyright (C) 2004 Pat Thoyts
- `clearlooks`: Copyright Regents of the University of California, Sun Microsystems, Inc., Scriptics Corporation, and other parties
- `elegance`: Copyright Regents of the University of California, Sun Microsystems, Inc., Scriptics Corporation, and other parties
- `keramik` and `keramik_alt`: Copyright (C) 2004 Pat Thoyts
- `kroc`: Copyright (C) 2004 David Zolli

- plastik: Copyright (C) 2004 Pat Thoyts
- radiance: Copyright Regents of the University of California, Sun Microsystems, Inc., Scriptics Corporation, and other parties
- scid and all its derivatives: Copyright (C) 2018 Uwe Klimmek
- smog: Copyright (C) 2005 Jelco Huijser
- winxpblue: Copyright (C) 2004 Pat Thoyts



**class** ttkthemes.themed\_style.**ThemedStyle** (\*args, \*\*kwargs)

Bases: ttk.Style, *ttkthemes.\_widget.ThemedWidget*

Style that supports setting the theme for a Tk instance. Can be used as a drop-in replacement for normal ttk.Style instances. Supports the themes provided by this package.

**\_\_init\_\_** (\*args, \*\*kwargs)

**Parameters** **theme** – Theme to set up initialization completion. If the theme is not available, fails silently.

**theme\_names** ()

Alias of get\_themes() to allow for a drop-in replacement of the normal ttk.Style instance.

**Returns** Result of get\_themes()

**theme\_use** (theme\_name=None)

Set a new theme to use or return current theme name

**Parameters** **theme\_name** – name of theme to use

**Returns** active theme name

**class** ttkthemes.themed\_tk.**ThemedTk** (\*args, \*\*kwargs)

Bases: Tkinter.Tk, *ttkthemes.\_widget.ThemedWidget*

Tk child class that supports the themes supplied in this package

A theme can be set upon initialization or during runtime. Can be used as a drop-in replacement for the normal Tk class. Additional options:

- Initial theme **theme**: Sets the initial theme to the theme specified. If the theme is not available, fails silently (there is no indication that the theme is not set other than it not appearing to the user).
- Toplevel background color **toplevel**: Hooks into the Toplevel.\_\_init\_\_ function to set a default window background color in the options passed. The hook is not removed after the window is destroyed, which is by design because creating multiple Tk instances should not be done in the first place.

- Tk background color `themebg`: Set the default background color of a Tk window to the default theme background color. For example: The background of windows may take on a dark color for dark themes. Backwards-compatible with the `background` keyword argument of v2.3.0 and earlier.
- GIF theme override `gif_override`: Forces ttkthemes to load the GIF version of themes that also provide a PNG version even if the PNG version can be loaded. Can only be set at object initialization. GIF themes may provide a higher UI performance than other themes.

`__getitem__` (*k*)

Return the resource value for a KEY given as string.

`__init__` (*\*args, \*\*kwargs*)

#### Parameters

- **theme** – Theme to set upon initialization. If theme is not available, fails silently.
- **toplevel** – Control Toplevel background color option, see class documentation for details.
- **themebg** – Control Tk background color option, see class documentation for details.
- **fonts** – Whether to enable the automatic change of default font selected for a theme

`cget` (*k*)

cget redirect to support additional options

`config` (*kw=None, \*\*kwargs*)

configure redirect to support additional options

`configure` (*kw=None, \*\*kwargs*)

Configure resources of a widget.

The values for resources are specified as keyword arguments. To get an overview about the allowed keyword arguments call the method keys.

`set_theme` (*theme\_name, topLevel=None, themebg=None*)

Redirect the set\_theme call to also set Tk background color

**class** `ttkthemes._widget.ThemedWidget` (*tk\_interpreter, gif\_override=False*)

Bases: `object`

Provides functions to manipulate themes in order to reduce code duplication in the ThemedTk and ThemedStyle classes.

`__init__` (*tk\_interpreter, gif\_override=False*)

Initialize attributes and call `_load_themes`

#### Parameters

- **tk\_interpreter** – tk interpreter for tk.Widget that is being initialized as ThemedWidget. Even if this Widget is just a single widget, the changes affect all widgets with the same parent Tk instance.
- **gif\_override** – Force loading of GIF-themes even if PNG-themes can be loaded

**current\_theme**

Property to get the currently enabled theme

`get_themes` ()

Return a list of names of available themes

`set_theme` (*theme\_name*)

Set new theme to use. Uses a direct tk call to allow usage of the themes supplied with this package.

**Parameters** `theme_name` – name of theme to activate

**set\_theme\_advanced** (*theme\_name*, *brightness=1.0*, *saturation=1.0*, *hue=1.0*, *pre-serve\_transparency=True*, *output\_dir=None*, *advanced\_name='advanced'*)

Load an advanced theme that is dynamically created

Applies the given modifiers to the images of the theme given and then creates a theme from these new images with the name 'advanced' and then applies this theme. Is not available without support for PNG-based themes, then raises `RuntimeError`.

**themes**

Property alias of `get_themes()`



## CHAPTER 3

---

### Example

---

Using `ttkthemes` is easiest through the usage of `ThemedTk`. The following example is written for Python 3.

```
from tkinter import ttk # Normal Tkinter.* widgets are not themed!
from ttkthemes import ThemedTk

window = ThemedTk(theme="arc")
ttk.Button(window, text="Quit", command=window.destroy).pack()
window.mainloop()
```



## CHAPTER 4

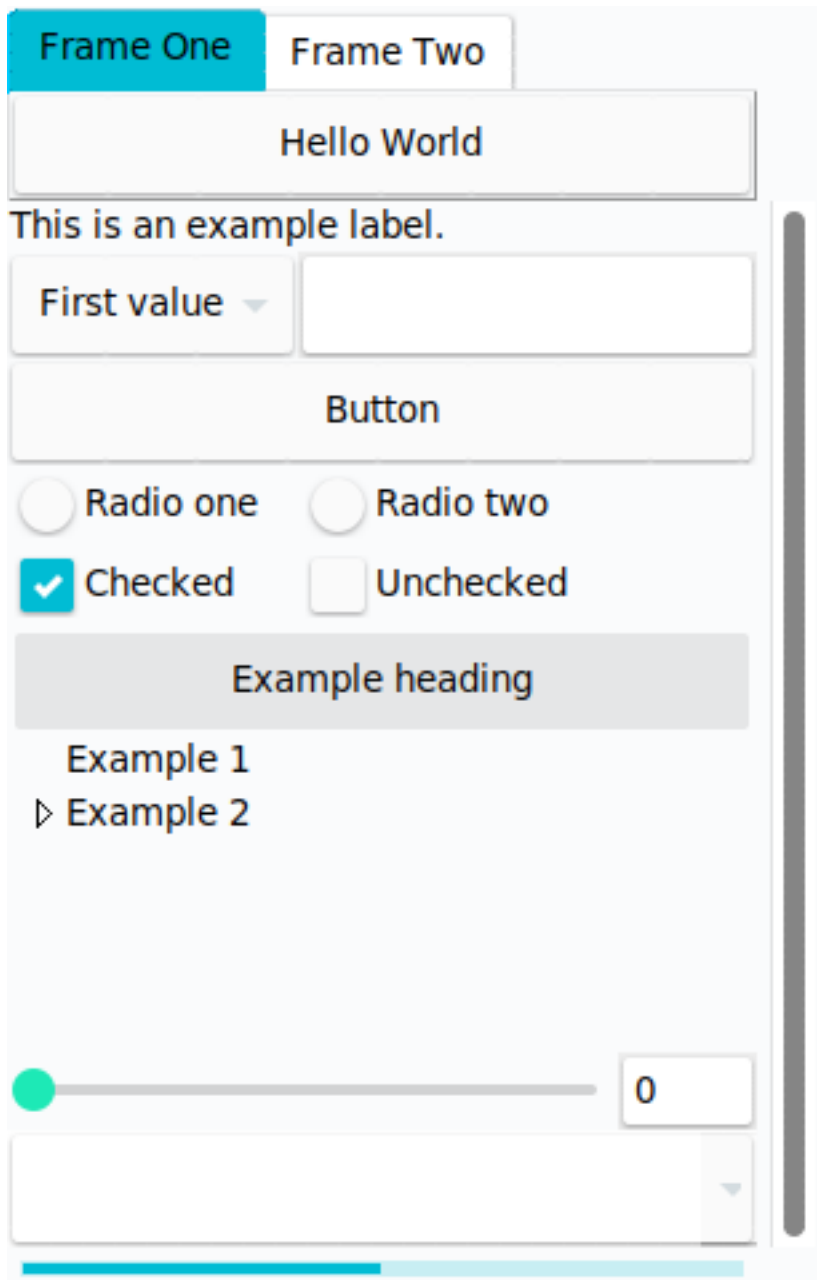
---

### Themes

---

`ttkthemes` includes a wide variety of different themes, and there is always room for more themes, no matter how ugly or obscure! Even though some themes may not be used in practice, the original goal of the project has not been forgotten: To gather and preserve *all* themes.

## 4.1 Adapta



The image displays a collection of Tkinter widgets styled with the Adapta theme. At the top, there are two tab-like frames: 'Frame One' (highlighted in blue) and 'Frame Two'. Below these is a rectangular box containing the text 'Hello World'. This is followed by a text label 'This is an example label.' and a dropdown menu with the text 'First value' and a downward arrow. Next is a button labeled 'Button'. Below the button are two radio buttons: 'Radio one' and 'Radio two'. Underneath the radio buttons are two checkboxes: 'Checked' (which is checked and highlighted in blue) and 'Unchecked'. Below the checkboxes is a grey rectangular box containing the text 'Example heading'. Under this heading are two list items: 'Example 1' and '▶ Example 2'. At the bottom, there is a horizontal slider with a green circle on the left and a numerical input field on the right showing the value '0'. Below the slider is a text entry field with a small downward arrow on its right side. A vertical scrollbar is visible on the right side of the widget area, and a blue progress bar is at the very bottom.

Frame One    Frame Two

Hello World

This is an example label.

First value ▼

Button

☐ Radio one    ☐ Radio two

☒ Checked    ☐ Unchecked

Example heading

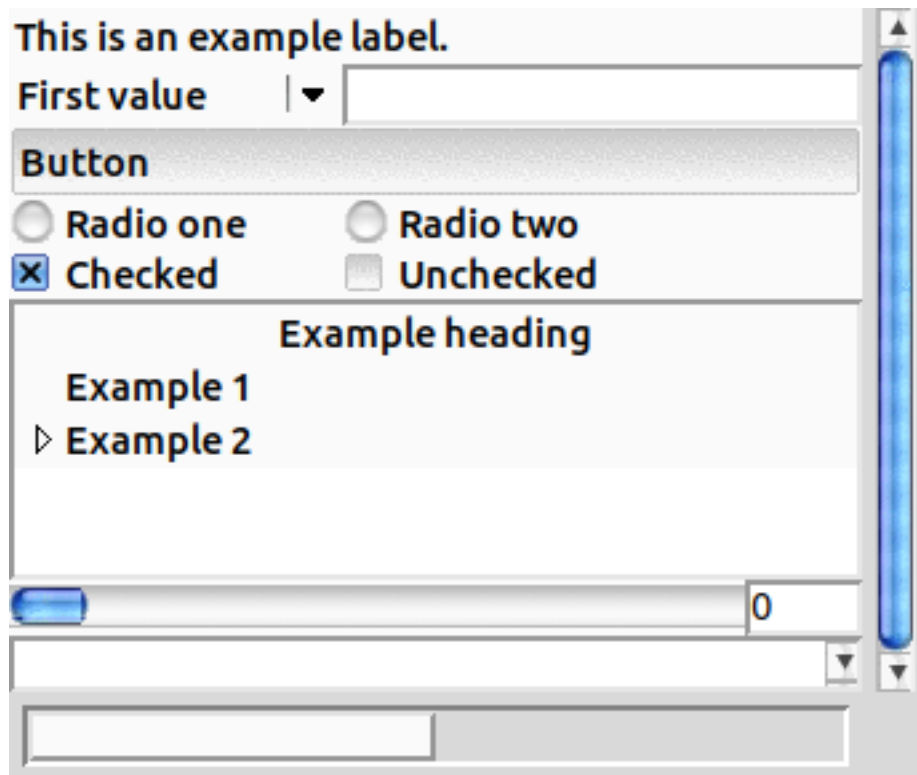
Example 1

▶ Example 2

0

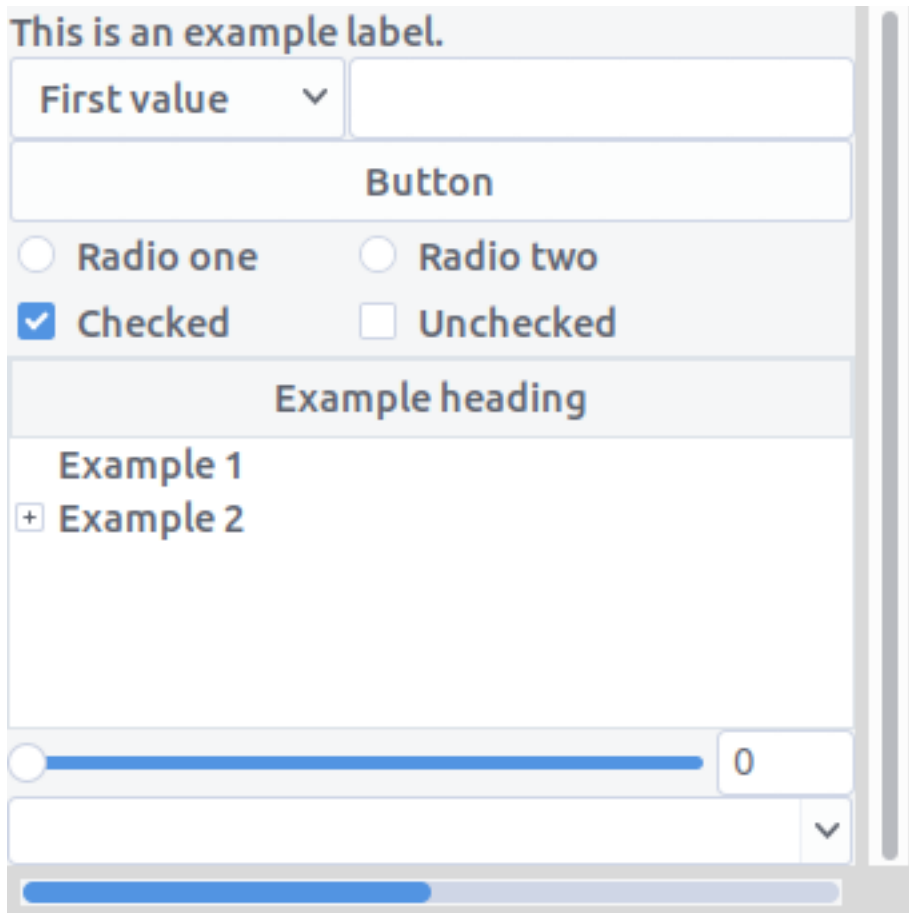


## 4.2 Aquativo



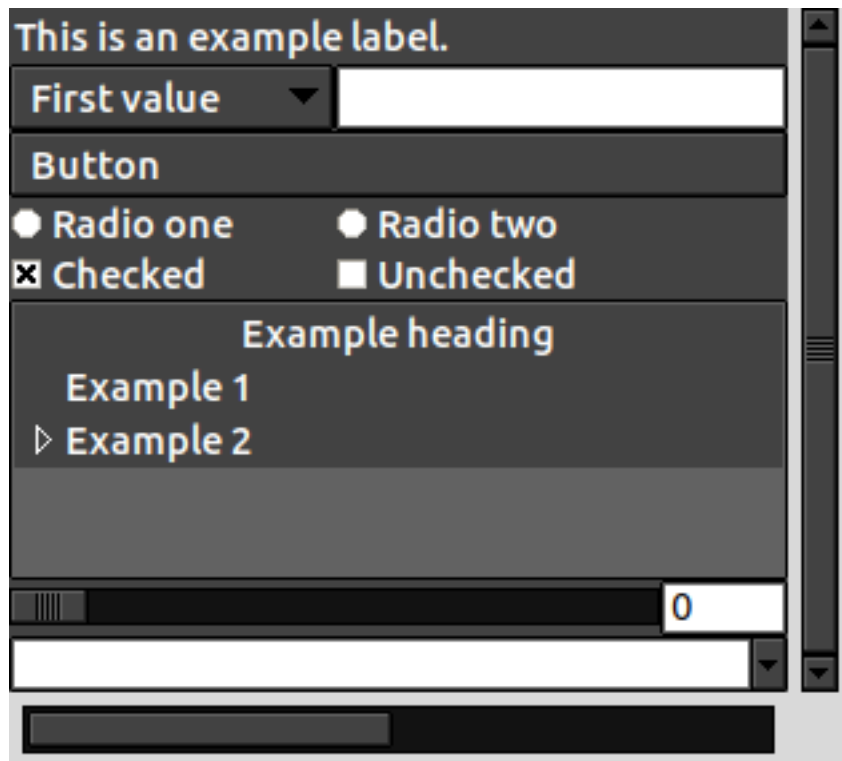
An Apple-style like theme by Pat Thoyts, created in 2004. The original site is listed to be <http://www.fewt.com>, but you will only be able to view this site using the Internet archive's Wayback machine. It appears the theme is related to the Ubuntu-based Linux distribution \_Fuduntu.

## 4.3 Arc



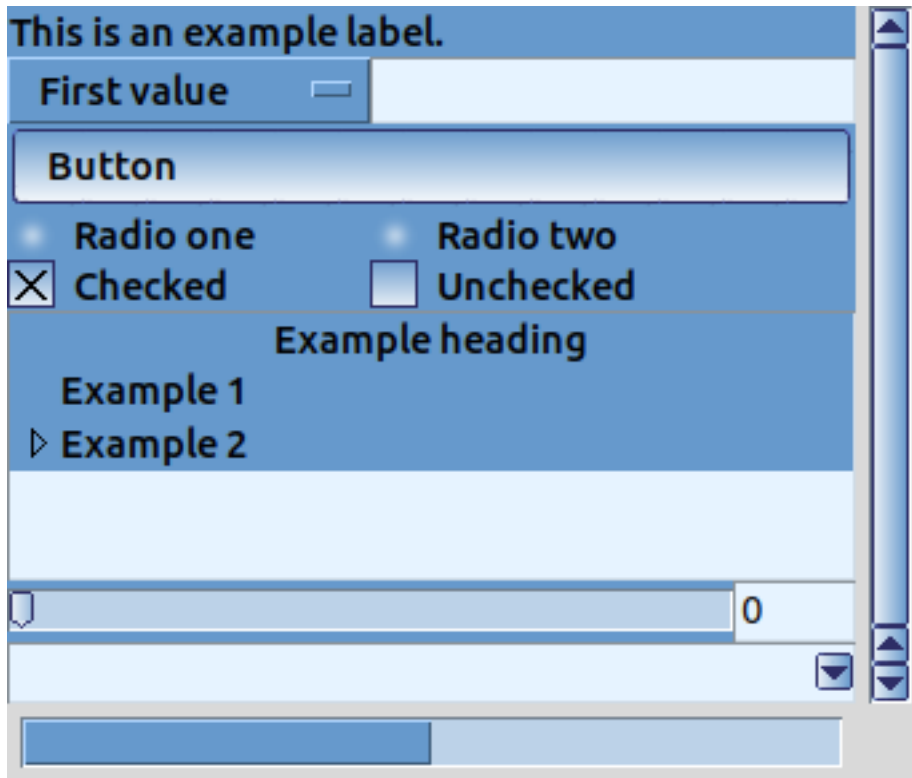
The newest theme of the bunch. Being created by Sergei Golovan in 2015 this theme requires Tk 8.6.0 in order to load. This is because the theme uses PNG images with transparency, making for a blue-tinted modern look and feel.

## 4.4 Black



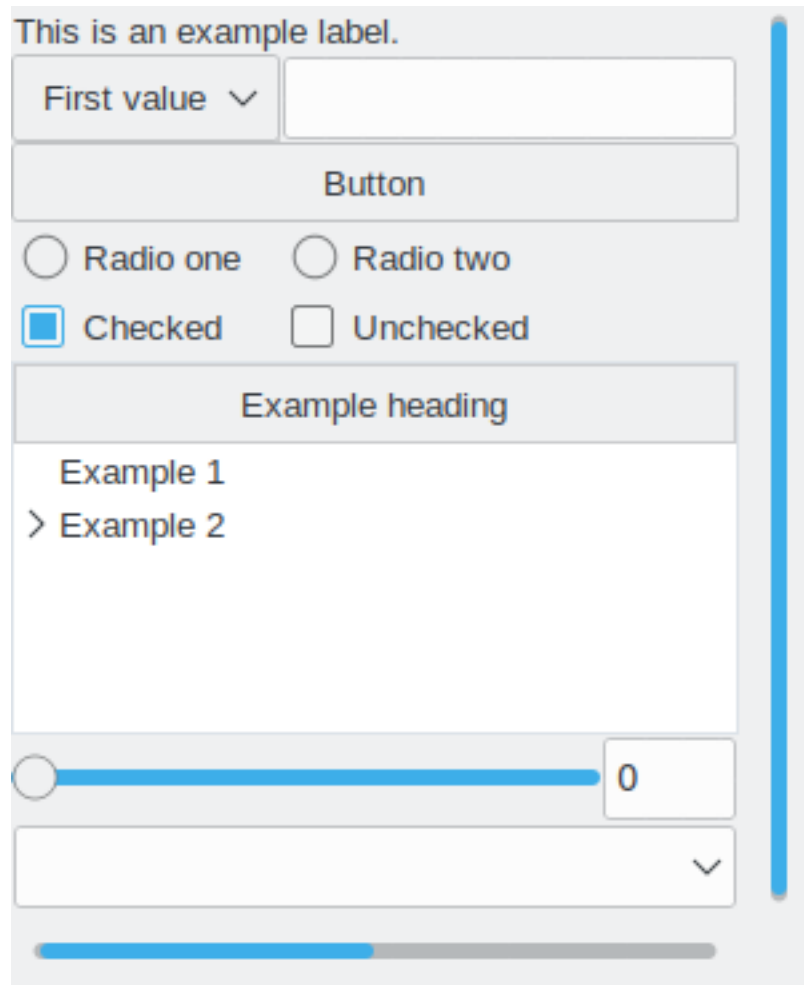
A simple yet quite popular theme, as it is very consistent in its use of dark colors in all widgets. Created by Mats Bengtsson in 2007.

## 4.5 Blue

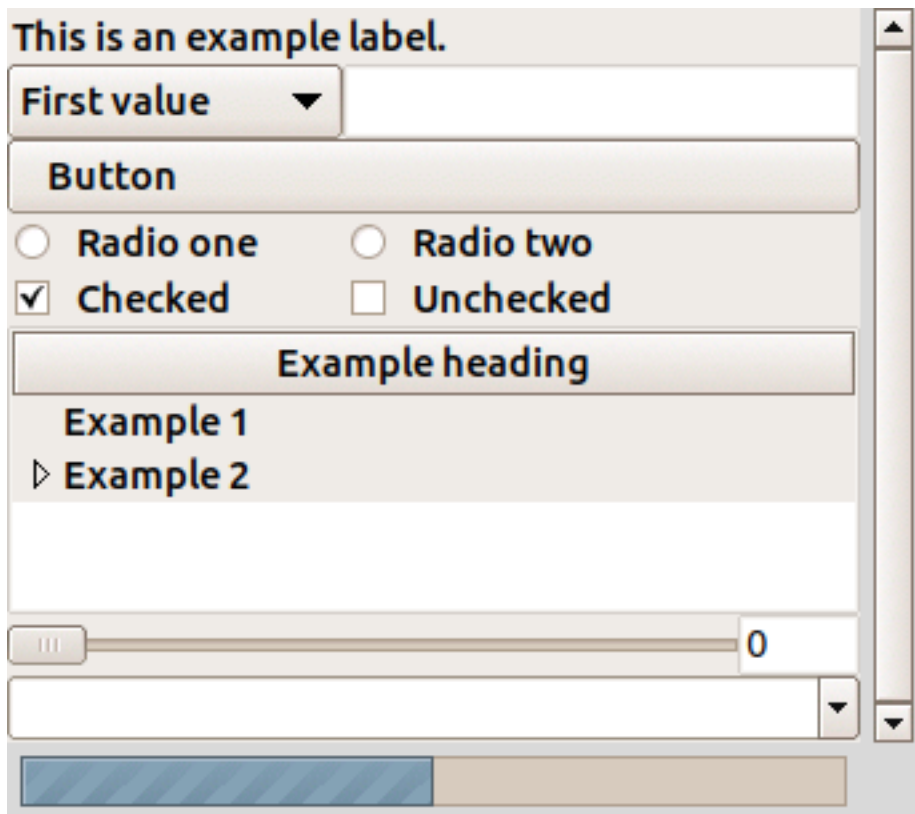


A theme that does live up to its name. This theme will burn your eyes out being so bright. Everything is blue, though in some widgets there is a nice color gradient. This theme was also created by Pat Thoyts in 2004.

## 4.6 Breeze

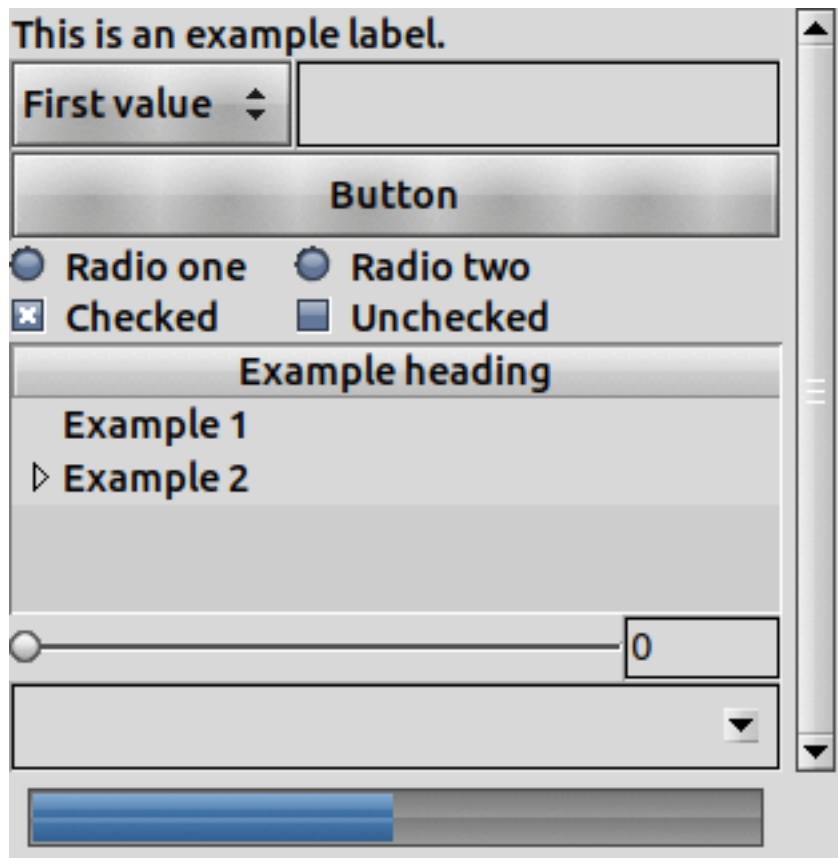


## 4.7 Clearlooks



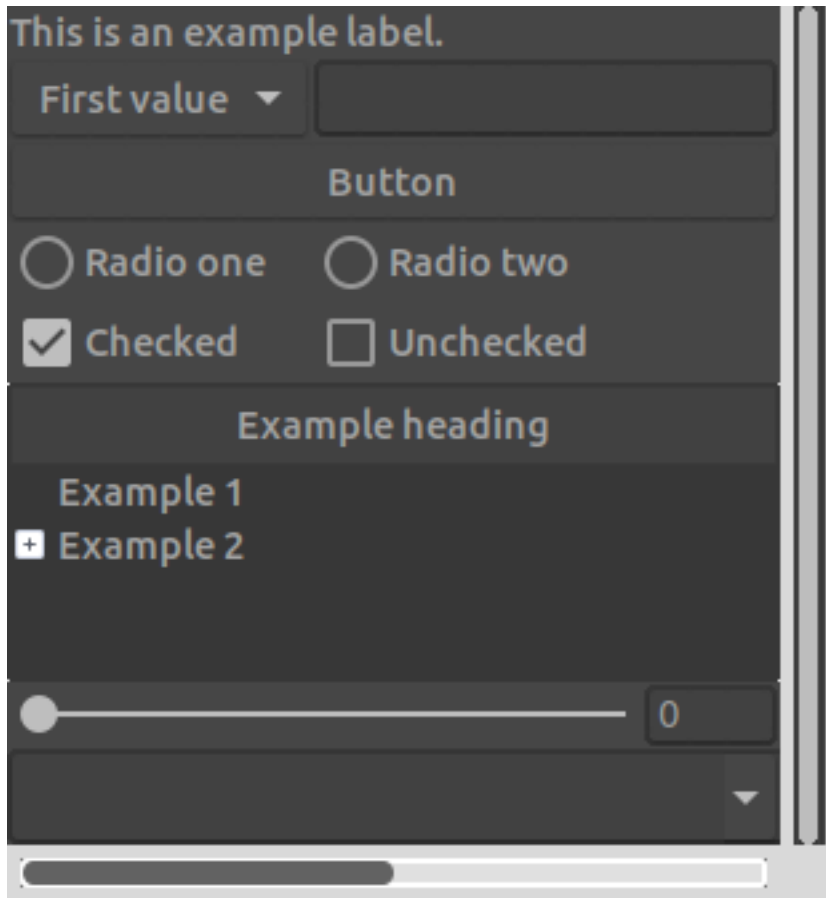
This theme was created by the Tcl/Tk developers team as a demo for what bitmap themes can do. The light tints move toward peach colors, giving this theme a feminine look. It still looks sleek and modern, and wouldn't be a bad choice. Just as with all themes though, the corners are rounded.

## 4.8 Elegance



A theme that appears to be created by the Tcl/Tk developers team. While attempts to tracing its exact origin have been unsuccessful so far, this theme was probably created around 2008. The theme can be found [here](#), but no author is listed.

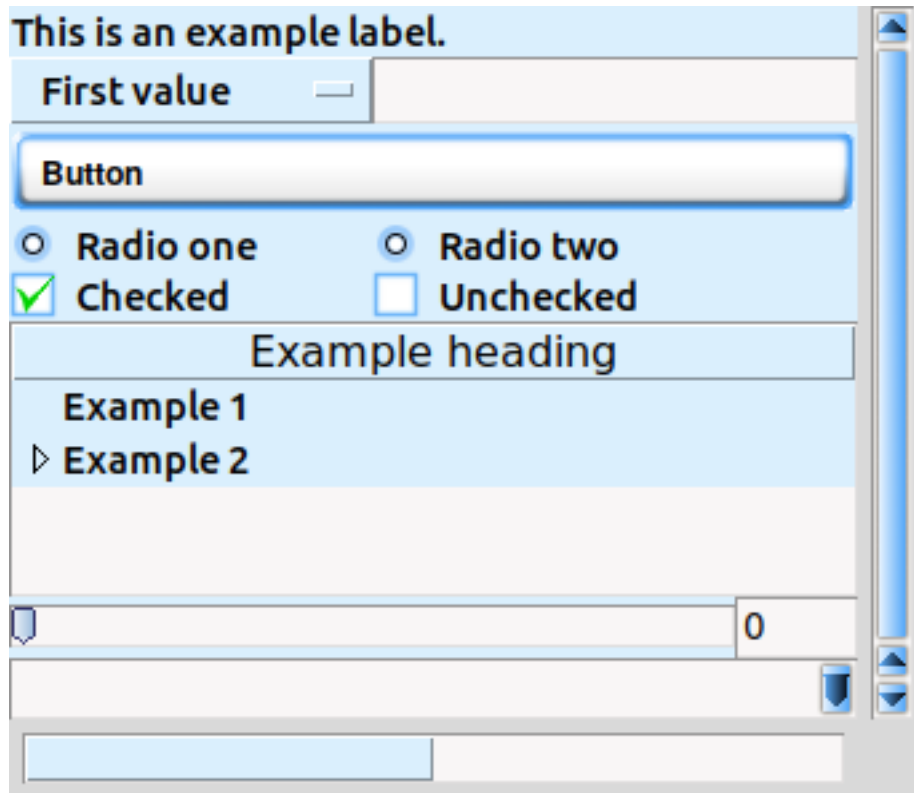
## 4.9 Equilux



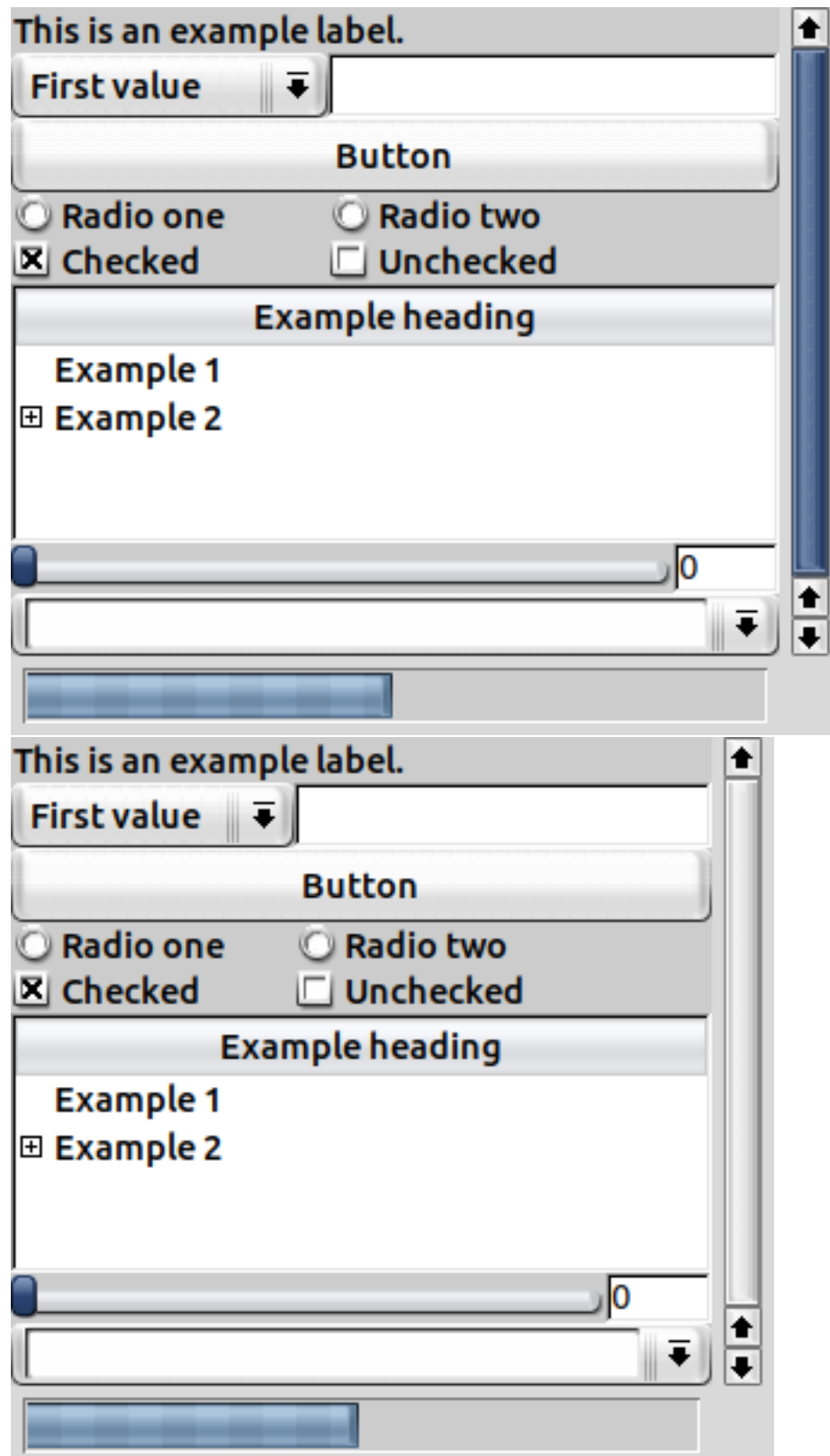
Originally [equilux](#) is a GTK3 theme.



## 4.10 ITFT1

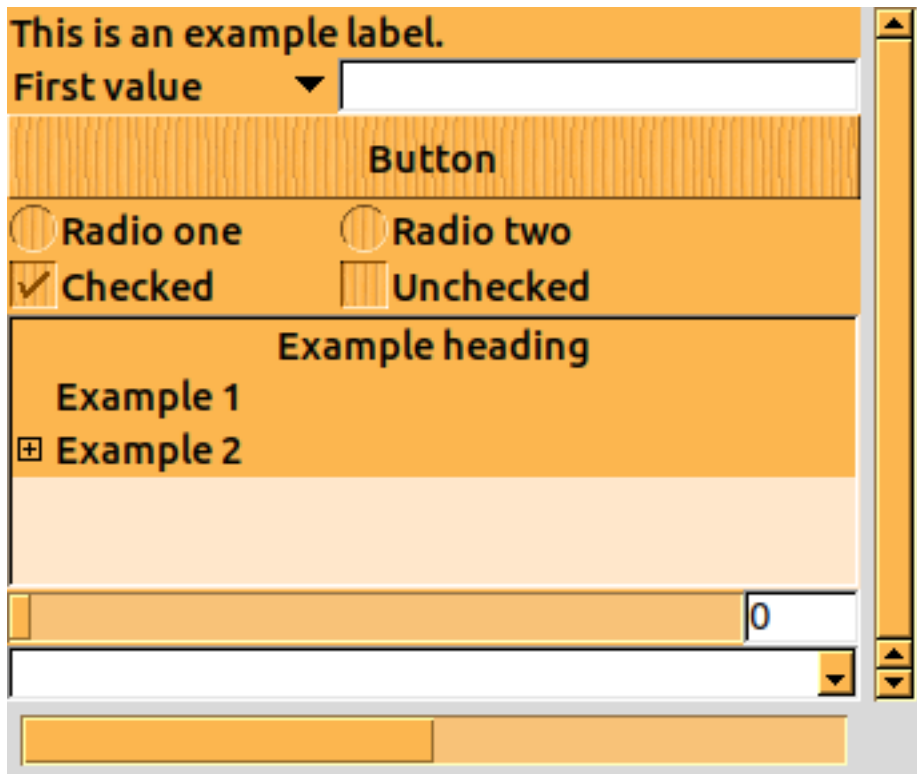


## 4.11 Keramik



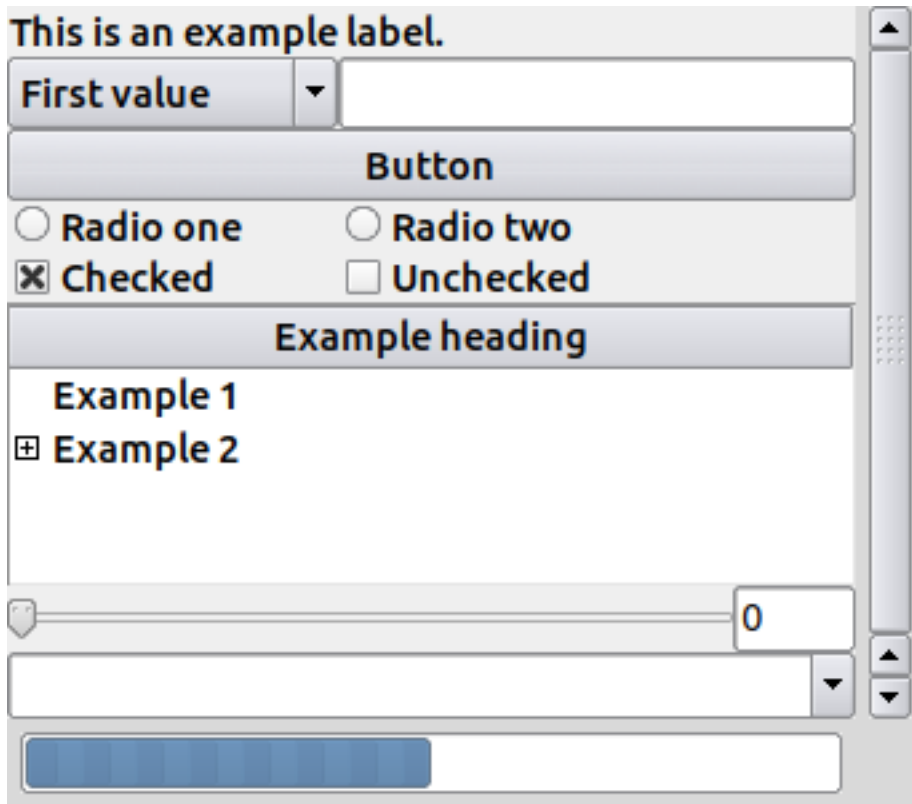
These two themes use a single file `keramik.tcl` and the differences between them are fairly limited. Originally developed by Pat Thoyts in 2004. These themes look the opposite of modern. They are futuristic, but in a bit of a wrong way. `Keramik_alt` uses a different color scrollbar element, namely silver instead of dark blue.

## 4.12 Kroc



This orange theme by David Zolli is busy on the eyes and has a wood-like grain in the Button widgets. Not a recommended choice for modern UI development.

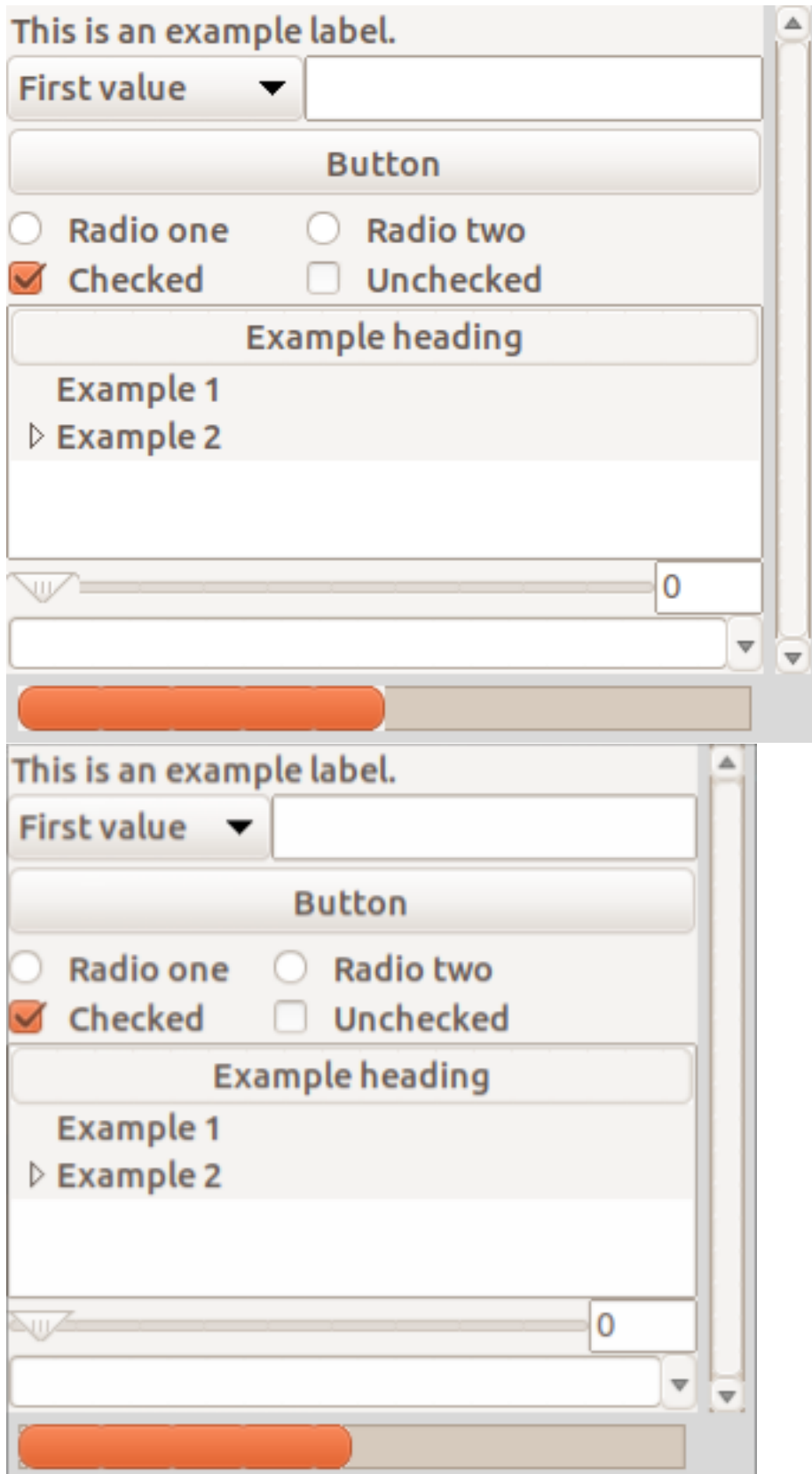
## 4.13 Plastik



A nice looking theme created by Pat Thoyts in 2005. While not bolstering the most distinctive features, it looks quite modern. Performance suffers heavily though, this is resolved by commenting out line 193 of the `plastik.tcl` file. The theme changes only slightly because of this change, but native performance is restored.



## 4.14 Radiance (Ubuntu)



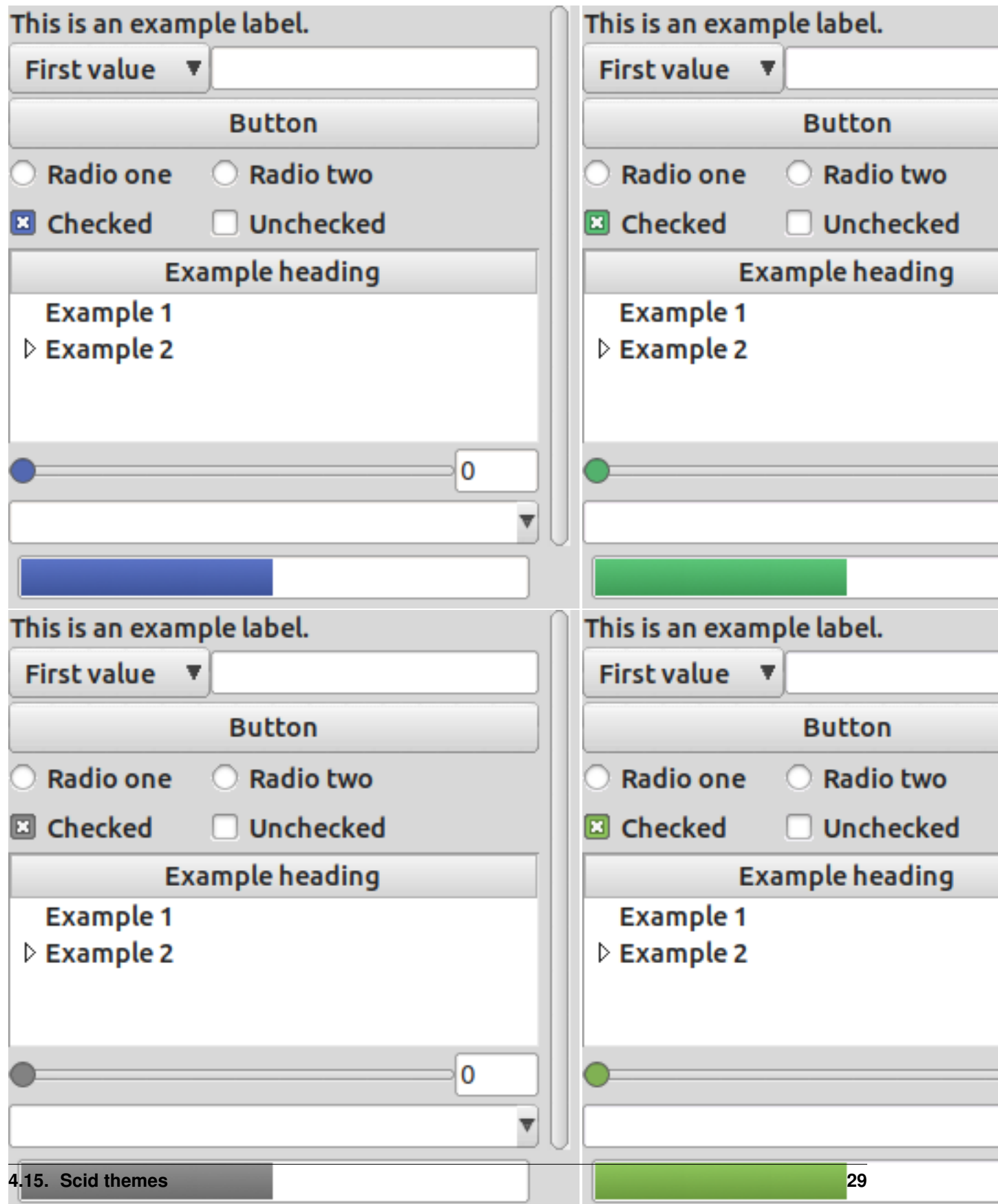
A theme that was developed by the Tcl/Tk developers team. This theme, as the name suggests, boasts the native look of Ubuntu's radiance theme, making for a good choice when targeting this platform. Large Progressbar widgets (namely in height) may look a bit mutated, so try sticking to the normal Progressbar height.

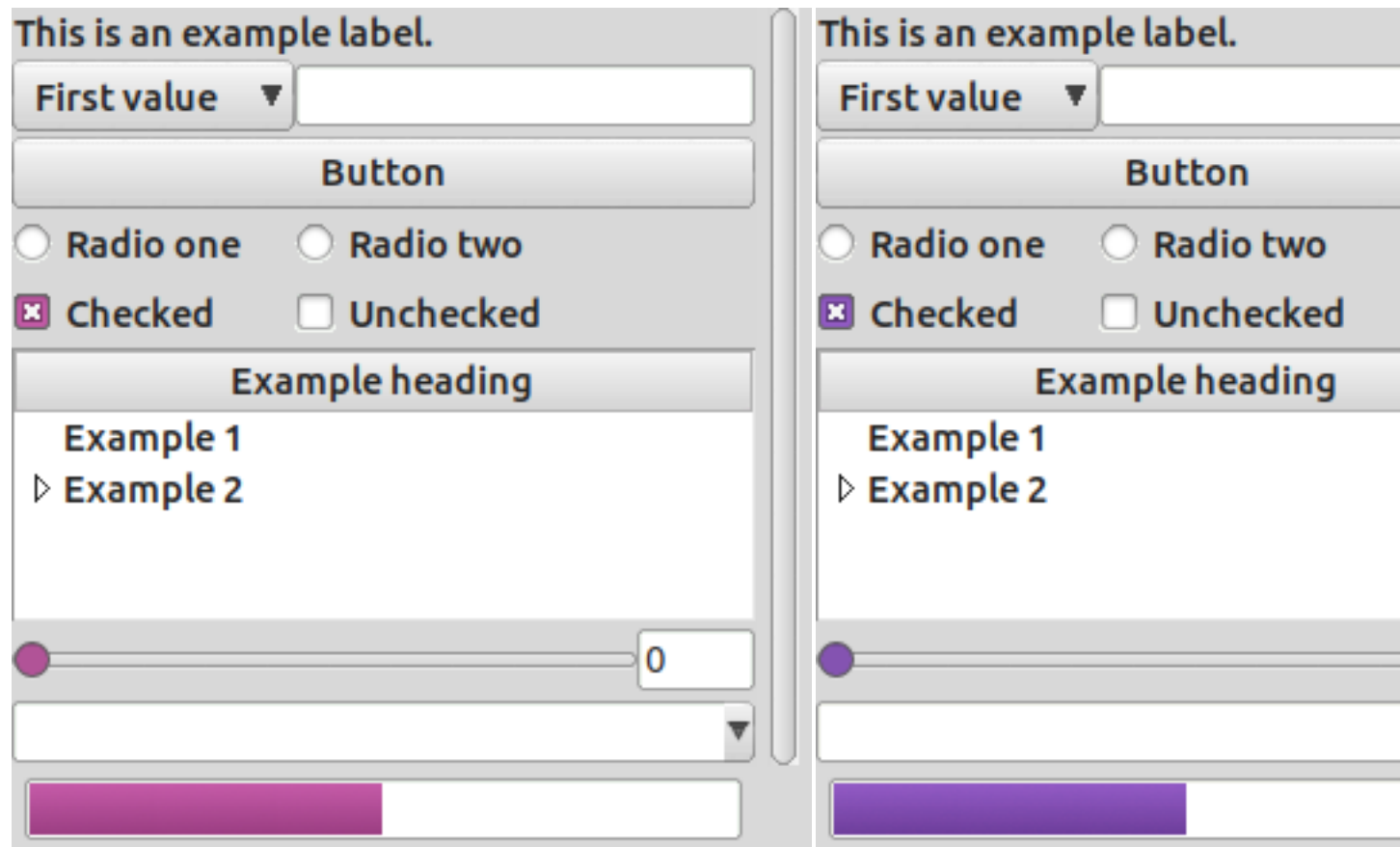
Theme `ubuntu` is equal to radiance, except it uses transparent elements in place of some white areas. Therefore, it looks a bit better when PNG-themes are available.

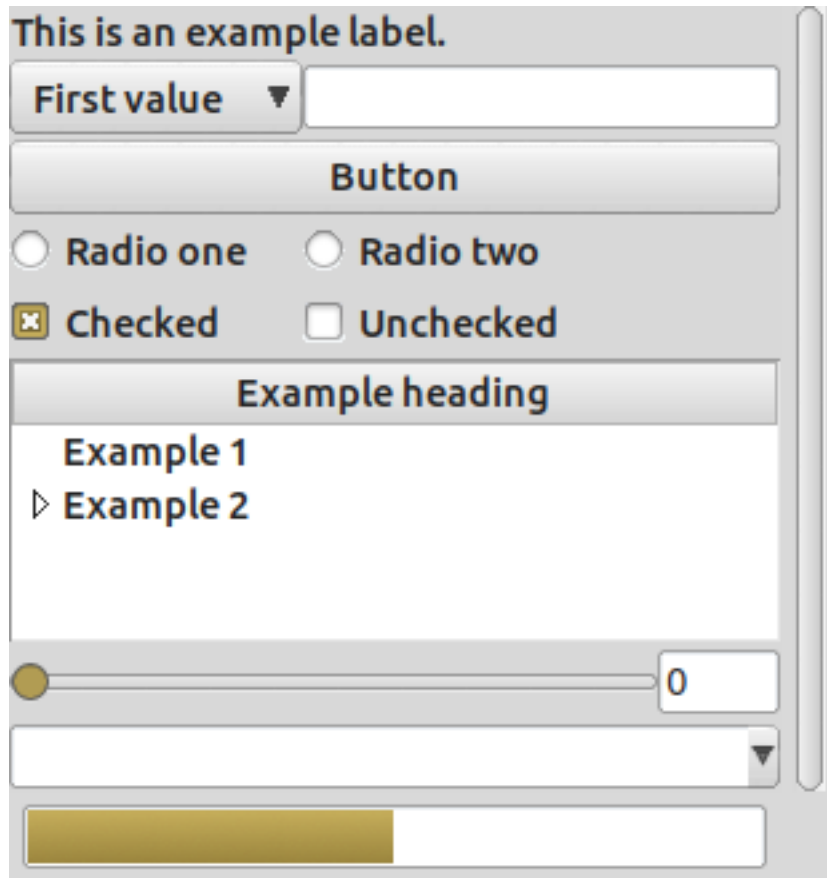




## 4.15 Scid themes

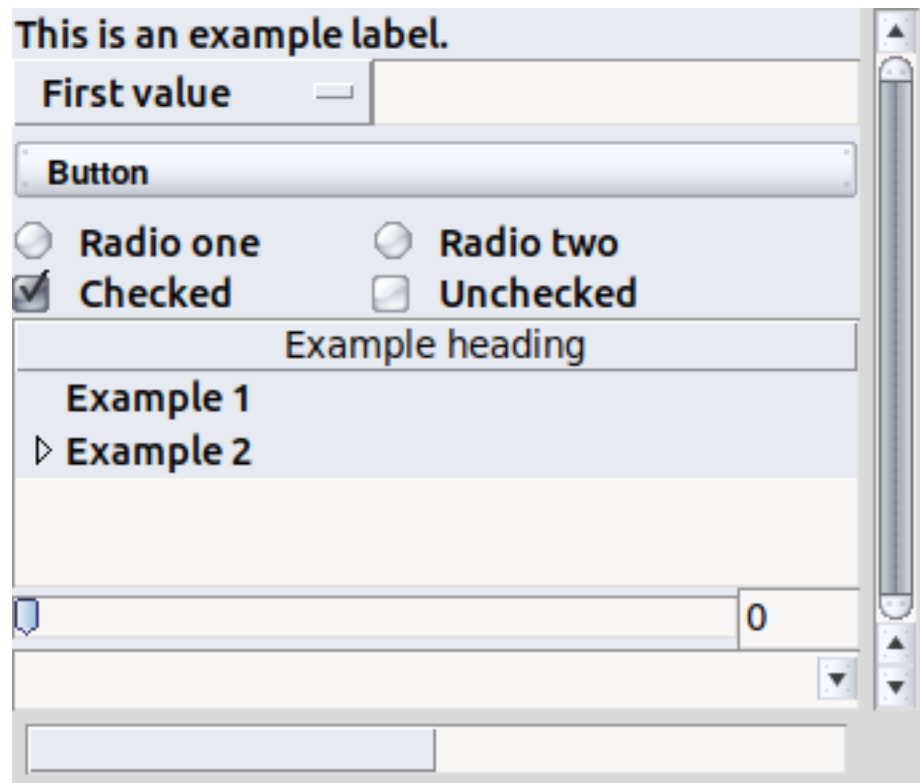




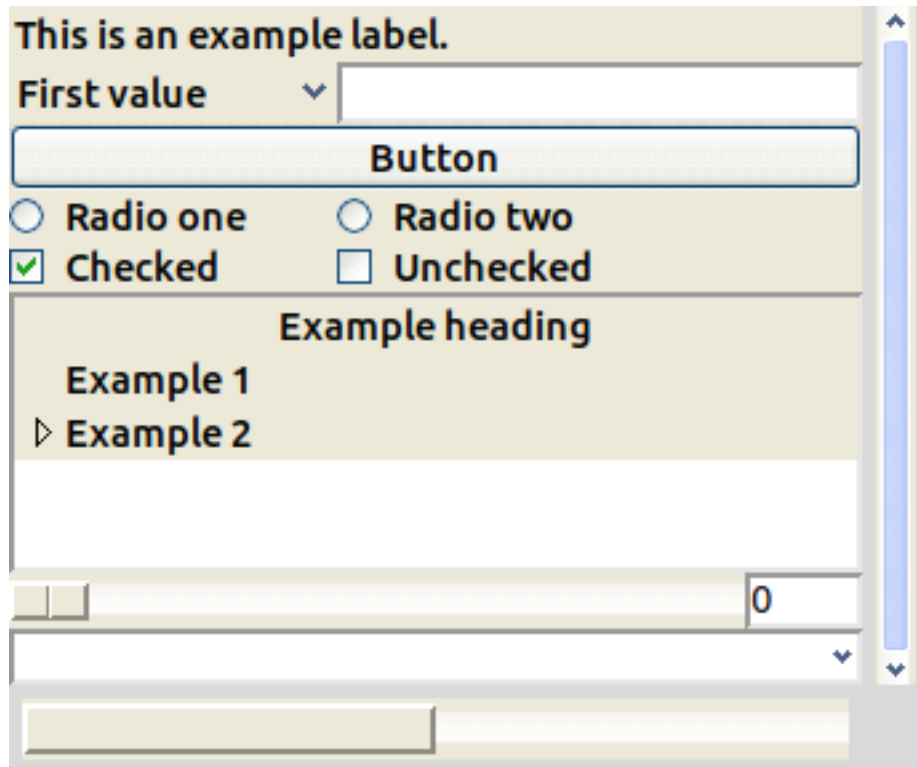


The scid themes form a group of themes that all share the same basic look but with differently colored accents.

## 4.16 Smog

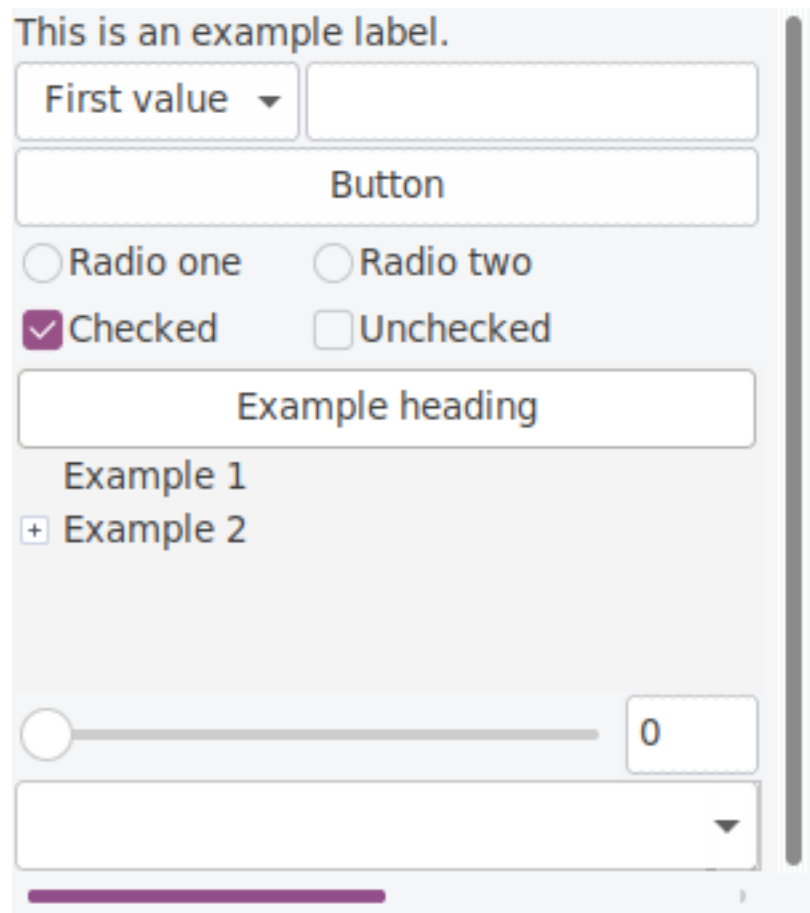


## 4.17 winxpblue



A theme that tries to imitate the Windows XP look and feel created by Pat Thoyts in 2004. Not recommended, even for Windows XP applications.

## 4.18 yaru



Theme derived from the new Ubuntu 20.04 default theme and theme ‘arc’.

`ttkthemes` supports the creation of custom themes based upon static themes during runtime. This is called dynamic theming. The functions to create dynamic themes are implemented in the `ThemedWidget` class. PNG-based theme support is required to apply a dynamic theme.

## 5.1 Choosing a theme

In order to use dynamic theming, you must first choose a theme to base your new theme on. The supported themes for dynamic themes are all pixmap themes that use files (and not a packed archive with files) directly to load pixmaps. An up-to-date list of supported pixmap themes may be found in the `pixmap_themes` class attribute of any `ThemedWidget` instance (such as `ThemedTk` or `ThemedStyle`). At the time of writing, the following list is available:

```
pixmap_themes = [  
    "arc",  
    "blue",  
    "clearlooks",  
    "elegance",  
    "kroc",  
    "plastik",  
    "radiance",  
    "winxpblue"  
]
```

It is recommended to choose a theme with noticeable colors for the best results. `radiance` and `blue` have proven to be quite suitable for this purpose. In order to use `blue`, it is recommended to also modify the `background` colors of all widgets you plan to use, as the theme colors are not changed during the operations.

Note that while being a pixmap theme, `equilux` is not included because using dynamic theming with that theme results in severe conversion artifacts.

## 5.2 Modifying a theme

In order to load an advanced theme, the following function is provided within any ThemedWidget:

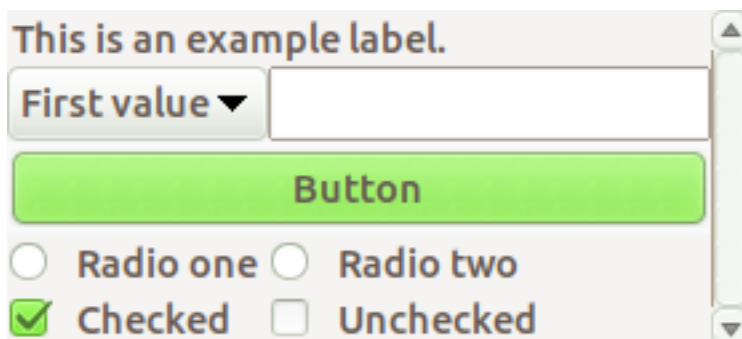
```
def set_theme_advanced(  
    self, theme_name, brightness=1.0, saturation=1.0, hue=1.0,  
    preserve_transparency=True, output_dir=None, advanced_name="advanced"  
)
```

As you might be able to deduce from the function definition, various parameters can be used to modify the pixmaps of the theme you choose:

- `theme_name`: The name of a valid pixmap theme to use for modification
- `brightness`: A modifier that is passed on to a `PIL.ImageEnhance.Brightness` enhancer. Values between 0.0 and 2.0 are expected.
- `saturation`: A modifier that is passed on to a `PIL.ImageEnhance.Color` enhancer. Values between 0.0 and 2.0 are expected.
- `hue`: A modifier that is used for the `ttkthemes._utils.shift_hue` function. Shifts the hue of an image by a certain amount. Note that the hue is the hue shift. Values between 0.0 and 2.0 are expected.
- `preserve_transparency`: When set to `True`, all resulting black pixels will be set to transparent. This is only required when modifying the hue of an image. During the conversion from RGBA to HSV image format, transparency is lost to black pixels, resulting in ugly black patches in images if not reversed.
- `output_dir`: Directory (to which write access is available) in which the new theme files should be placed. By default, a temporary directory is used provided by `tempfile`. Note that on most systems, this directory is cleared upon reboot.
- `advanced_name`: Name of the theme to generate. You can combine this with an `output_dir` parameter to actually create a custom theme that you can install on other machines as well. **NOTE THAT IT IS REQUIRED TO USE A DIFFERENT NAME EACH TIME IF YOU SET THE ADVANCED THEME FOR A SINGLE TK INTERPRETER INSTANCE MULTIPLE TIMES.**

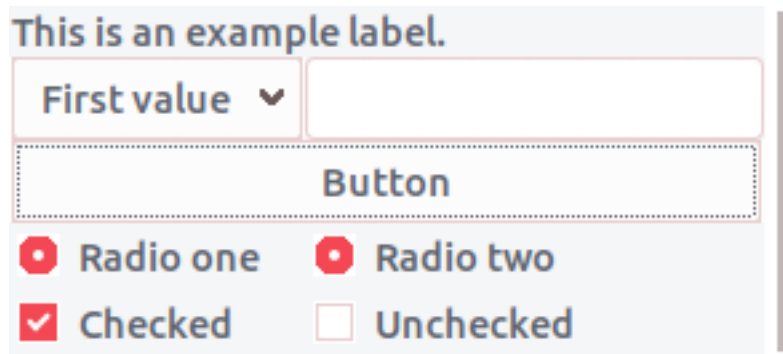
## 5.3 Examples

Some examples of what you can create using this function:



A modified radiance theme. The hue is changed so all originally orange features are a bright green.





A modified arc theme. The hue is changed as well as the brightness, though the latter only very slightly.

## 5.4 Notes

Note that the theme is generated during runtime, when the function `set_theme_advanced` is called. When the function is called, rather resource-expensive operations upon tens of images are performed, as well as disk I/O and loading all images into memory may cause a spike in memory usage, even though it is not all that much on most modern PCs.



## CHAPTER 6

---

### Licenses

---

The themes included in the `ttkthemes` package have been released under various different licenses, including the BSD-2-clause-like Tcl License, GNU GPLv2+ and GNU GPLv3. Note that the only license under which all themes are available together is GNU GPLv3. Most of the code is available under GNU GPLv3 only. If code is available under any other license, it is indicated in the specific files and folders.



The installation of `ttkthemes` is very simple. There are a few ways to install `ttkthemes`.

### 7.1 PyPI

Installation from PyPI is easiest. Simply use `pip` to fetch the package and install it:

There is a single installation option available for `ttkthemes`. In order to use the high-quality PNG-themes instead of the GIF-themes, an extension is required under Python 2 and some earlier distributions of Python 3. Simply install the extension with `pip` as well.

This option installs the separate package `tkimg`, for which more information is available from [here](#).

In some cases, it is required to update `setuptools` in order to install `ttkthemes`. It is not known which version of `setuptools` is exactly required at this time.

### 7.2 Source

Installation from git is also possible:

```
python3 -m pip install git+https://github.com/RedFantom/ttkthemes
```



---

## Theme Loading

---

Tkinter for Python runs a Tcl-interpreter with the Tk package. The actual UI is drawn and controlled by this Tcl-interpreter. In order to load themes into Tkinter, it must be done through this Tcl-interpreter. This page describes how this is done, and what the limitations per Python version are.

### 8.1 Loading

Each theme is a Tcl-package in itself. In order to load the theme, the package has to be loaded. Usually, stand-alone packages come with a `pkgIndex.tcl` file, which checks the dependencies of the package and then provides a `package ifneeded {name} {version}` line.

Then, in order to execute the code that creates the package, a `package require {name} {version}` is executed. Subsequently, the code provided after the `package ifneeded` is executed, and thus the files for the package are executed.

The code files that are executed to create the package then provide a `package provide {name} {version}` line, which creates the package. Then loading the package is complete, and the theme is available.

In the case of `ttkthemes`, there is a big difference. The themes are split into PNG-themes (which also have a GIF-version) and GIF-only themes. The appropriate `pkgIndex.tcl` files in the right folders are evaluated depending on whether the PNG-dependencies have been met. Instead of using `package ifneeded` with a `pkgIndex.tcl` file for each theme, `ttkthemes` loads all the themes instantly upon evaluation of the `pkgIndex.tcl` file of the whole folder of themes.

This is done so as to limit the amount of Tcl-code in the `ttkthemes` package. While the `ttkthemes` package could still be used with Tcl directly, it is intended for use within Python, and Tkinter's error reporting facilities are extremely limited (no full tracebacks are available), and thus the amount of Tcl-code is best minimized. All work that can be performed in Python is performed in Python.

For more information on packages and loading them, check the [tcl-lang](#) wiki website.

## 8.2 Limitations

All themes are available on all platforms that support Tk 8.4 or higher. Tk 8.4 is available with all Python distributions of 2.7 or higher. Python-distributions may be created with higher versions of Tk as well. In order to load the higher quality PNG version of themes for which it is available, either Tk 8.6 is required, or TkImg. Tk 8.6 is available with Python 3.6 in most Python binary distributions, including the Windows binaries and the Ubuntu PPA-version. However, it may be possible that there are binary distributions compiled with a different version of Tk out in the wild. In order to use the PNG version of themes on older Python versions, including Python 2.7, TkImg is required. While formerly provided with `ttkthemes`, the Tk-extension TkImg is now available for Python in a separate package: `tking`.

## 8.3 Tcl-loading

The `ttkthemes` themes can be loaded from a Tcl-interpreter directly. In fact, this is what the `_Widget`-class does for the `ttkthemes` Python-package. In this section, from this point onwards, `ttkthemes` will refer to the Tcl-package instead.

The themes provided are divided into two categories: themes with a PNG and GIF version and themes with only a GIF version. The GIF-only themes are provided in the folder `/ttkthemes/themes`. The `pkgIndex.tcl` file in this folder loads all themes in the folder when executed and provides the `ttkthemes` package.

For the other category of themes, *only* the GIF or PNG version can be loaded, as loading a second package with a name which is already used by another package is not possible.

Depending on whether the PNG or GIF version of themes should be loaded, the `pkgIndex.tcl` in the `ttkthemes/gif` or `ttkthemes/png` can be evaluated. The PNG themes can only be loaded if Tk is version 8.6 or TkImg is available.

```
package require Tk 8.6
source ttkthemes/themes/pkgIndex.tcl
source ttkthemes/png/pkgIndex.tcl
package require ttkthemes 1.0
ttk::setTheme plastik
```



ttkthemes: A group of themes for the ttk extensions of Tcl

- Copyright (C) 2017-2018 RedFantom
- Copyright (C) 2017-2018 Akuli
- Copyright (C) 2004 Pat Thoyts
- Copyright (C) 2004 David Zolli
- Copyright (C) 2007-2008 Mats Bengsston
- Copyright (C) 2015-2018 The materia and equilux authors
- Copyright (C) 2005 Jelco Huijser
- Copyright (C) Regents of the University of California, Sun Microsystems, Inc., Scriptics Corporation, and other parties.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Please note that while you can use some themes **only** under GPLv3 and that my code is also available **only** under GPLv3, some themes, of which the files have a header explicitly stating this, are also available under the BSD-like 2-clause Tcl license.



## CHAPTER 10

---

### Recommendations

---

The themes `plastik`, `clearlooks` and `elegance` are recommended to make your UI look nicer on all platforms when using `Tkinter` and the `ttk` extensions in Python. When you are targeting Ubuntu, consider using the great `radiance` theme.



## Symbols

`__getitem__()` (*ttkthemes.themedTk.ThemedTk method*), 6  
`__init__()` (*ttkthemes.\_widget.ThemedWidget method*), 6  
`__init__()` (*ttkthemes.themedStyle.ThemedStyle method*), 5  
`__init__()` (*ttkthemes.themedTk.ThemedTk method*), 6

## C

`cget()` (*ttkthemes.themedTk.ThemedTk method*), 6  
`config()` (*ttkthemes.themedTk.ThemedTk method*), 6  
`configure()` (*ttkthemes.themedTk.ThemedTk method*), 6  
`current_theme` (*ttkthemes.\_widget.ThemedWidget attribute*), 6

## G

`get_themes()` (*ttkthemes.\_widget.ThemedWidget method*), 6

## S

`set_theme()` (*ttkthemes.\_widget.ThemedWidget method*), 6  
`set_theme()` (*ttkthemes.themedTk.ThemedTk method*), 6  
`set_theme_advanced()` (*ttkthemes.\_widget.ThemedWidget method*), 7

## T

`theme_names()` (*ttkthemes.themedStyle.ThemedStyle method*), 5  
`theme_use()` (*ttkthemes.themedStyle.ThemedStyle method*), 5  
`ThemedStyle` (*class in ttkthemes.themedStyle*), 5  
`ThemedTk` (*class in ttkthemes.themedTk*), 5  
`ThemedWidget` (*class in ttkthemes.\_widget*), 6  
`themes` (*ttkthemes.\_widget.ThemedWidget attribute*), 7