

# Theme-D-Gnome User Guide

Tommi Höynälänmaa

November 6, 2018

# Contents

<b>1</b>	<b>Copyright</b>	<b>1</b>
<b>2</b>	<b>General</b>	<b>2</b>
<b>3</b>	<b>Software needed</b>	<b>2</b>
<b>4</b>	<b>Installation</b>	<b>3</b>
4.1	Debian-based systems . . . . .	3
4.1.1	guile version 2.0 and library guile-gnome version 2.16.4 . .	3
4.1.2	Other Configurations . . . . .	3
4.2	Other systems . . . . .	4
4.3	Local mode . . . . .	5
<b>5</b>	<b>Removing the software</b>	<b>5</b>
5.1	Debian-based systems . . . . .	5
5.2	Other systems . . . . .	5
<b>6</b>	<b>Running the example programs</b>	<b>5</b>
<b>7</b>	<b>Making programs that use Theme-D-Gnome</b>	<b>6</b>
<b>8</b>	<b>Regenerating the Theme-D-Gnome wrapper modules</b>	<b>6</b>
<b>9</b>	<b>Documentation</b>	<b>6</b>
<b>10</b>	<b>Features different in Theme-D-Gnome and guile-gnome</b>	<b>7</b>
10.1	Making instances of classes . . . . .	7
10.2	Different procedures . . . . .	7
10.2.1	gtk-text-buffer-create-tag . . . . .	7
10.2.2	gtk-entry-new . . . . .	8
10.2.3	gtk-entry-create . . . . .	8
10.2.4	gtk-tree-or-list-store-set . . . . .	8
10.2.5	gtk-tree-selection-get-selected . . . . .	8
10.2.6	gtk-dialog-create . . . . .	9
10.2.7	gtk-dialog-new . . . . .	9
10.2.8	gtk-frame-create-alt . . . . .	10
10.2.9	gtk-frame-new-alt . . . . .	11
<b>11</b>	<b>Widget classes that have been checked</b>	<b>11</b>

## 1 Copyright

Copyright (C) 2015-2018 Tommi Höynälänmaa

Contents of subdirectories `theme-d-gen` and `gnome-examples` are licensed under GNU General Public License. See file `doc/GPL-3`. The contents of subdirectories `gnome` and `theme-d-gnome/runtime` are licensed under GNU Lesser

General Public License. See file `doc/LGPL-3`. All documentation (subdirectory `doc`) is licensed under the GNU Free Documentation License. See file `doc/GFDL-1.3`.

## 2 General

This guide covers only UNIX systems. Many commands in this guide have to be executed in a root session. A root session is launched with command `su root` or `sudo` depending on your system. In Ubuntu the command is `sudo`.

## 3 Software needed

You need at least the following software to use Theme-D-Gnome:

- guile version 2.0
- guile-gnome version 2.16.2, 2.16.4, or 2.16.5
- GNU make
- Theme-D version  $\geq 1.1.10$

TH Scheme Utilities and Theme-D can be obtained from

<http://www.iki.fi/tohoyn/theme-d/>.

Guile can be obtained from

<http://www.gnu.org/software/guile/>.

guile-gnome can be obtained from

<http://www.gnu.org/software/guile-gnome/>.

If you use Debian or Ubuntu Linux it is recommended that you install guile 2.0 with commands

```
apt-get install guile-2.0
apt-get install guile-2.0-dev
```

and guile-gnome with commands

```
apt-get install guile-gnome2-dev
apt-get install guile-gnome2-glib
apt-get install guile-gnome2-gtk
```

All of these commands have to be executed in a root session. Note that Theme-D-Gnome may not work for guile-gnome versions other than 2.16.2,

2.16.4, or 2.16.5. If you use such versions follow the instructions in section 8 to regenerate the wrapper libraries.

## 4 Installation

You can check the version number of your guile-gnome library with command

```
pkg-config --modversion guile-gnome-gtk-2
```

### 4.1 Debian-based systems

In particular these instructions apply to Debian and Ubuntu Linux distributions.

#### 4.1.1 guile version 2.0 and library guile-gnome version 2.16.4

Install the software with command

```
dpkg -i theme-d-gnome_0.7.3-1_all.deb
```

in a root session.

#### 4.1.2 Other Configurations

1. Create a new directory for building Theme-D-Gnome and change into that directory. This directory shall be called the *build directory*.
2. Copy files `theme-d-gnome-0.7.3.tar.gz` and `theme-d-gnome_0.7.3-1.debian.tar.xz` into the build directory.
3. Give command

```
ln -s theme-d-gnome-0.7.3.tar.gz theme-d-gnome_0.7.3.orig.tar.gz
```

4. Unpack the source code with command

```
tar xvf theme-d-gnome-0.7.3.tar.gz
```

5. Give commands

```
cd theme-d-gnome-0.7.3
tar xvf ../theme-d-gnome_0.7.3-1.debian.tar.xz
```

6. If you use guile version 2.2 change the value of variable `GUILLE_VERSION` to 2.2 in file `debian/rules` (line 4).

7. If you use guile-gnome version 2.16.5 or 2.16.2 change the value of variable `GUILE_GNOME_VERSION` in file `debian/rules` (line 5).
8. Build the Debian package with command

```
dpkg-buildpackage -uc -us
```

in subdirectory `theme-d-gnome-0.7.3`.

9. Change to the parent directory and install the generated Debian package with command

```
dpkg -i theme-d-gnome_0.7.3-1_all.deb
```

The command has to be run as root.

## 4.2 Other systems

1. Unpack the source package `theme-d-gnome-0.7.3.tar.gz` into some directory.
2. Change to the subdirectory `theme-d-gnome-0.7.3` and give command

```
./configure
```

In case you have guile-gnome version 2.16.2 use command

```
./configure --with-guile-gnome=2.16.2
```

instead. In case you have guile-gnome version 2.16.5 use command

```
./configure --with-guile-gnome=2.16.5
```

3. Give command

```
make
```

4. Install Theme-D-Gnome with command

```
make install-complete
```

in a root session.

## 4.3 Local mode

It is possible to use Theme-D-Gnome without installing it. This is called *local mode*. In order to do this unpack the source code package `theme-d-gnome-0.7.3.tar.gz` into some directory and execute commands

```
./configure
make
```

in subdirectory `theme-d-gnome-0.7.3`. This directory shall be called the *installation directory*. When you use Theme-D-Gnome in local mode and compile and link your programs you have to set the `-m` option so that it includes the installation directory. When you run your Theme-D-Gnome programs in local mode you have to set environment variable `GUILE_LOAD_PATH` so that it contains the installation directory.

## 5 Removing the software

### 5.1 Debian-based systems

Give command

```
dpkg --purge theme-d-gnome
```

in a root session.

### 5.2 Other systems

Give command

```
make uninstall-complete
```

in subdirectory `theme-d-gnome-0.7.3` of the directory where you unpacked Theme-D-Gnome. The command has to be run in a root session.

## 6 Running the example programs

Copy the example programs and makefiles (`user.mk`) into the directory where you want to build them. The examples programs can be found in directory `/usr/share/doc/theme-d-gnome/examples` or in subdirectory `gnome-examples` in the Theme-D-Gnome source package. Note that the example programs must be contained in subdirectory `gnome-examples` in the target directory. Give command `make -f user.mk XXX.go` where `XXX` is the name of the program (`hello` or `calc`). In order to run an application give command `rtp-gnome XXX.go`. You can also compile both the examples by command `make -f user.mk` in subdirectory `gnome-examples`. In order to run the Theme-D-Gnome demo build the program with command `make -f user.mk` and run it with command

```
rtp-gnome run-demo.go
```

in subdirectory `gnome-examples/theme-d-gnome-demo`.

## 7 Making programs that use Theme-D-Gnome

In order to use Theme-D-Gnome in your Theme-D programs you have to import the following guile modules in the target program:

- `(gnome gobject)`
- `(gnome gtk)`
- `(theme-d-gnome runtime gnome-support)`

Note that these modules are not imported as Theme-D modules. When you run your own programs using Theme-D-Gnome you have to ensure that necessary GNOME modules are linked into guile. This can be done by executing guile with command `guile-gnome`. Easiest way to fulfill these requirements is to use command

```
rtp-gnome MYPROGRAM.go
```

to run your programs. The program `rtp-gnome` accepts the same options `--no-verbose-errors`, `--backtrace`, and `--pretty-backtrace` as program `run-theme-d-program`, see the Theme-D User Guide.

## 8 Regenerating the Theme-D-Gnome wrapper modules

1. Change to the directory where you have unpacked the Theme-D-Gnome package.
2. Change to the subdirectory `theme-d-gen` and run command `make`.
3. Change to the subdirectory `gnome/wrap` and give command `./generate.sh`.
4. Change to the subdirectory `gnome` and run command `make`.

## 9 Documentation

Install the source code of `guile-gnome` and build it. GTK Scheme documentation is found in file `gtk/doc/gtk/guile-gnome-gtk.info`. Programming with Theme-D-Gnome resembles closely programming with `guile-gnome` and Scheme. See also <http://www.gtk.org/> for the documentation of the GTK C library.

## 10 Features different in Theme-D-Gnome and guile-gnome

### 10.1 Making instances of classes

In Theme-D-Gnome instances of classes are created by creator procedures. Creator procedures are named `xxx-create` or `xxx-create-yyy` and they correspond to GTK constructor procedures `xxx-new` and `xxx-new-yyy`. The GTK style constructor procedures are supported by Theme-D-Gnome, too. The difference between creators and GTK style constructor procedures is that the result type of a creator is the class to be created. GTK constructor procedures often have result type `<gtk-widget>` for the subclasses of `<gtk-widget>`.

### 10.2 Different procedures

The following Theme-D-Gnome procedures take Theme-D procedures as callbacks:

- `gtype-instance-signal-connect`
- `connect`
- `gtype-instance-signal-emit`
- `emit`
- `gtk-action-group-add-actions`
- `add-actions`
- `gtk-action-group-add-toggle-actions`
- `add-toggle-actions`
- `gtk-action-group-add-radio-actions`
- `add-radio-actions`

As procedure purities are not specified by the guile-gnome library we set the purity of all Theme-D-Gnome wrapper functions to `nonpure`.

#### 10.2.1 `gtk-text-buffer-create-tag`

*Syntax:*

```
(gtk-text-buffer-create-tag textbuffer str args...)
```

*Arguments:*

Name: `textbuffer`  
Type: `<gtk-text-buffer>`  
Description: A text buffer

Name: `str`

Type: `<string>`  
Description: The name of the new tag

Name: `args`  
Type: a list  
Description: The properties of the new tag

*Result value:* New tag  
*Result type:* `<gtk-text-tag>`

*Purity of the procedure:* nonpure

This procedure is a method of generic procedure `create-tag`.

### 10.2.2 `gtk-entry-new`

*Syntax:*

`(gtk-entry-new)`

No arguments.

*Result value:* New `<gtk-entry>` widget  
*Result type:* `<gtk-widget>`

*Purity of the procedure:* nonpure

### 10.2.3 `gtk-entry-create`

*Syntax:*

`(gtk-entry-create)`

No arguments.

*Result value:* New `<gtk-entry>` widget  
*Result type:* `<gtk-entry>`

*Purity of the procedure:* nonpure

### 10.2.4 `gtk-tree-or-list-store-set`

Could not find documentation for this procedure.

### 10.2.5 `gtk-tree-selection-get-selected`

*Syntax:*

```
(gtk-tree-selection-get-selected sel)
```

*Arguments:*

Name: `sel`  
Type: `<gtk-tree-selection>`  
Description: A tree selection

*Result value:* A pair of the tree model and tree iterator

*Result type:* `(:pair <object> <gtk-tree-iter>)`

*Purity of the procedure:* nonpure

This procedure is a method of generic procedure `get-selected`.

### 10.2.6 `gtk-dialog-create`

*Syntax:*

```
(gtk-dialog-create str-title modal? destroy-with-parent?)
```

*Arguments:*

Name: `str-title`  
Type: `<string>`  
Description: The title of the dialog

Name: `modal?`  
Type: `<boolean>`  
Description: The modality of the dialog

Name: `destroy-with-parent?`  
Type: `<boolean>`  
Description: Destroy the dialog with its parent

*Result value:* A new dialog

*Result type:* `<gtk-dialog>`

*Purity of the procedure:* nonpure

### 10.2.7 `gtk-dialog-new`

*Syntax:*

```
(gtk-dialog-new str-title modal? destroy-with-parent?)
```

*Arguments:*

Name: `str-title`  
Type: `<string>`  
Description: The title of the dialog

Name: `modal?`  
Type: `<boolean>`  
Description: The modality of the dialog

Name: `destroy-with-parent?`  
Type: `<boolean>`  
Description: Destroy the dialog with its parent

*Result value:* A new dialog

*Result type:* `<gtk-widget>`

*Purity of the procedure:* nonpure

### 10.2.8 `gtk-frame-create-alt`

*Syntax:*

```
(gtk-frame-create-alt str-label s-shadow-type i-width-request i-height-request)
```

*Arguments:*

Name: `str-label`  
Type: `<string>`  
Description: The label of the frame

Name: `s-shadow-type`  
Type: `<symbol>`  
Description: The shadow type of the frame

Name: `i-width-request`  
Type: `<integer>`  
Description: The requested width of the frame

Name: `i-height-request`  
Type: `<integer>`  
Description: The requested height of the frame

*Result value:* A new frame

*Result type:* `<gtk-frame>`

*Purity of the procedure:* nonpure

The shadow type is one of the following symbols:

- none
- in
- out
- etched-in
- etched-out

### 10.2.9 gtk-frame-new-alt

*Syntax:*

```
(gtk-frame-new-alt str-label s-shadow-type i-width-request i-height-request)
```

*Arguments:*

Name: `str-label`  
 Type: `<string>`  
 Description: The label of the frame

Name: `s-shadow-type`  
 Type: `<symbol>`  
 Description: The shadow type of the frame

Name: `i-width-request`  
 Type: `<integer>`  
 Description: The requested width of the frame

Name: `i-height-request`  
 Type: `<integer>`  
 Description: The requested height of the frame

*Result value:* A new frame

*Result type:* `<gtk-widget>`

*Purity of the procedure:* nonpure

## 11 Widget classes that have been checked

- `<gtk-action-group>`
- `<gtk-box>`
- `<gtk-button>`
- `<gtk-button-box>`
- `<gtk-cell-renderer-toggle>`

- <gtk-cell-renderer-text>
- <gtk-check-button>
- <gtk-combo-box>
- <gtk-container>
- <gtk-dialog>
- <gtk-entry>
- <gtk-frame>
- <gtk-hbox>
- <gtk-hbutton-box>
- <gtk-hpaned>
- <gtk-image>
- <gtk-label>
- <gtk-list-store>
- <gtk-menubar>
- <gtk-message-dialog>
- <gtk-paned>
- <gtk-scrolled-window>
- <gtk-size-group>
- <gtk-statusbar>
- <gtk-table>
- <gtk-text-buffer>
- <gtk-text-iter>
- <gtk-text-view>
- <gtk-toolbar>
- <gtk-tree-selection>
- <gtk-tree-store>
- <gtk-tree-view>
- <gtk-tree-view-column>
- <gtk-ui-manager>
- <gtk-vbox>

- `<gtk-vbutton-box>`
- `<gtk-vpaned>`
- `<gtk-widget>`
- `<gtk-window>`