Theme-D-Gnome User Guide

Tommi Höynälänmaa

June 23, 2018

Contents

1 Copyright

Contents of subdirectories `theme-d-gen` and `gnome-examples` are licensed under GNU General Public License. See file `doc/gpl.txt`. The contents of subdirectories `gnome` and `runtime` and files `theme-d-config.scm.in` and `theme-d-config-maint.scm.in` are licensed under GNU Lesser General Public License. See file `doc/lgpl.txt`. All documentation (files `*.tex` and `*.pdf`

in subdirectory `doc`) is licensed under the GNU Free Documentation License. See file `code/fdl.txt`.

2 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
- TH Scheme Utilities version >= 1.3
- Theme-D version >= 1.1.1

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

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

and guile-gnome with commands

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

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 7 to regenerate the wrapper libraries.

3 Installation

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

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

3.1 Debian-based systems

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

3.1.1 Amd64/Intel 64-bit x86 Processor Architecture and guile-gnome version 2.16.4

Install the software with command

```
sudo dpkg -i theme-d-gnome_0.6.15_amd64.deb
```

3.1.2 Other Configurations

1. Unpack the Theme-D-Gnome source code.
2. If your processor architecture is not amd64 change the Architecture entry in file `debian/control` (9th line).
3. If you use guile-gnome version 2.16.2 uncomment the line

   ```
   #DEB_CONFIGURE_EXTRA_FLAGS = --with-guile-gnome=2.16.2
   ```

   in file `debian/rules` (6th line). If you use guile-gnome version 2.16.5 uncomment the line

   ```
   #DEB_CONFIGURE_EXTRA_FLAGS = --with-guile-gnome=2.16.5
   ```

4. Build the Debian package with command

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

   in subdirectory `theme-d-gnome-0.6.15`.
5. Change to the parent directory and install the generated Debian package with command

   ```
   sudo dpkg -i theme-d-gnome_0.6.15_arch.deb
   ```

   where *arch* is your processor architecture.

3.2 Other systems

1. Unpack the source package `theme-d-gnome_0.6.15.tar.gz` into some directory.
2. Change to the subdirectory `theme-d-gnome-0.6.15` 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

   ```
   make
   ```

4. Install Theme-D-Gnome with command

   ```
   sudo make install-complete
   ```

   If you don't have `sudo` you may try command

   ```
   su root make install-complete
   ```

5. Give command

   ```
   make compile-scheme-code
   ```

4 Removing the software

4.1 Debian-based systems

Give command

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

4.2 Other systems

Give command

```
sudo make uninstall-complete
```

in subdirectory `theme-d-gnome-0.6.15` of the directory where you unpacked Theme-D-Gnome.

5 Running the example programs

Unpack the Theme-D-Gnome source package and change to the subdirectory `gnome-examples`. Give command `make XXX.go` where `XXX` is the name of the program (`hello` or `calc`). If you compile the programs outside your home directory use command `sudo make XXX.go` instead. In order to run an application give command `rtp-gnome XXX.go`. You can also compile all the examples by command `make` (or `sudo make`) in subdirectory `gnome-examples`. In order to run the Theme-D-Gnome demo build the program with command `make` and run it with command

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

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

6 Making programs that use Theme-D-Gnome

In order the use Theme-D-Gnome in your Theme-D programs import modules `(gnome gobject)` and `(gnome gtk)`. When you run your own programs using Theme-D-Gnome you have to ensure that necessary GNOME modules are linked into guile. Easiest way to do this is to execute guile with command guile-gnome. Your program has to include file `gnome/gnome-support.go`, too. 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` and `--pretty-backtrace` as program `run-theme-d-program`, see the Theme-D User Guide.

7 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`.

8 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.

9 Features different in Theme-D-Gnome and guile-gnome

9.1 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>`.

9.2 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`.

9.2.1 `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`.

9.2.2 `gtk-entry-new`

Syntax:

```
(gtk-entry-new)
```

No arguments.

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

Purity of the procedure: nonpure

9.2.3 `gtk-entry-create`

Syntax:

```
(gtk-entry-create)
```

No arguments.

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

Purity of the procedure: nonpure

9.2.4 `gtk-tree-or-list-store-set`

Could not find documentation for this procedure.

9.2.5 `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`.

9.2.6 `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

9.2.7 `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

9.2.8 `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`

9.2.9 `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

10 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>