Theme-D-Gnome User Guide

Tommi Höynälänmaa

January 27, 2020

Contents

1 Copyright

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`. Symbol *rev* in the package names means the Debian revision of the packages. It is typically 1.

3 Software needed

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

- guile version 2.0 or 2.2
- guile-gnome-2 version 2.16.2, 2.16.4, or 2.16.5
- GNU make
- Theme-D

The proper Theme-D version depends on the version of Theme-D-Gnome you use. Theme-D-Gnome version 0.8.2-1 requires Theme-D version 1.4.1. Note that this Theme-D-Gnome version may not work for newer Theme-D releases. In that case a new version of Theme-D-Gnome shall be released.

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-2 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 2.2 with commands

```
apt-get install guile-2.2
apt-get install guile-2.2-dev
```

and guile-gnome-2 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-2 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-2 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 2.2 and library guile-gnome-2 version 2.16.5

Install the software with command

```
dpkg -i theme-d-gnome_0.8.2-rev_all.deb
```

in a root session.

4.1.2 Other Configurations

1. Create a new directory for building Theme-D-Gnomeand change into that directory. This directory shall be called the *build directory*.
2. Copy files `theme-d-gnome-0.8.2.tar.gz` and `theme-d-gnome_0.8.2-`*rev*`.debian.tar.xz` into the build directory.
3. Give command

   ```
   ln -s theme-d-gnome-0.8.2.tar.gz  theme-d-gnome_0.8.2.orig.tar.gz
   ```

4. Unpack the source code with command

   ```
   tar xvf theme-d-gnome-0.8.2.tar.gz
   ```

5. Give commands

   ```
   cd theme-d-gnome-0.8.2
   tar xvf ../theme-d-gnome_0.8.2-rev.debian.tar.xz
   ```

6. If you use guile version 2.0 change the value of variable `GUILE_VERSION` to 2.0 in file `debian/rules` (line 4).
7. If you use guile-gnome-2 version 2.16.2 or 2.16.4 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-0.8.2`.
9. Change to the parent directory and install the generated Debian package with command

   ```
   dpkg -i theme-d-gnome_0.8.2-rev_all.deb
   ```

   The command has to be run as root.

4.2 Other systems

1. Unpack the source package `theme-d-gnome-0.8.2.tar.gz` into some directory.
2. Change to the subdirectory `theme-d-gnome-0.8.2` and give command

   ```
   ./configure
   ```

   In case you have guile-gnome-2 version 2.16.2 use command

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

   instead. In case you have guile-gnome-2 version 2.16.4 use command

   ```
   ./configure --with-guile-gnome=2.16.4
   ```

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-0.8.2.tar.gz` into some directory and execute commands

```
./configure
make
```

in subdirectory `theme-d-gnome-0.8.2`. 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-0.8.2` 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

When you make your own programs using Theme-D-Gnome you can compile the programs in the same way as other Theme-D programs using command theme-d-compile but you have to link the programs with command theme-d-gnome-link. This ensures that the guile-gnome libraries are loaded by the target programs. 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-2. 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-2 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-2 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-2

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-2 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>