Theme-D-Intr User Manual

Tommi Höynälänmaa

Copyright © 2020-2026 Tommi Höynälänmaa

This file is part of Theme-D-Intr version 0.20.5. This document has been last updated May 5, 2026.

You can redistribute and/or modify this file under the terms of the GNU Free Documentation License as published by the Free Software Foundation, either version 1.3 of the License, or (at your option) any later version.

See file `doc/GFDL-1.3` in the source code package for the license.

Table of Contents

Introduction 1
Installation 1
    Debian forky (testing) and sid (unstable) 1
    Other Debian-based Systems 1
    Other UNIX Systems 2
Using Theme-D-Intr without Installing 3
Example Programs 3
Writing Programs using Theme-D-Intr 4
Distributing Programs using Theme-D-Intr 7
Help with callback types 7

Introduction

Theme-D-Intr is a software library allowing the use of the introspected GObject modules in Theme-D.

Installation

Debian forky (testing) and sid (unstable)

If you use the Synaptic Package Manager install the packages `theme-d-intr` and `theme-d-intr-dev` . If you want to install the example programs install one or both of the packages `theme-d-intr-gtk3-examples` and `theme-d-intr-gtk4-examples`. In order to install the packages from the command line give command

```
sudo apt install theme-d-intr theme-d-intr-dev
```

and optionally one or both of the commands

```
sudo apt install theme-d-intr-gtk3-examples
sudo apt install theme-d-intr-gtk4-examples
```

Other Debian-based Systems

- Install Guile 3.0 is you don't have it already.

- Install Theme-D version 7.3.2 is you don't have it already. See Theme-D User Guide [https://www.iki.fi/tohoyn/theme-d/theme-d-user-guide-20260505-1.pdf] for instructions.
- Install G-Golf. See http://www.iki.fi/tohoyn/g-golf-debian/.
- Create a directory for the build and download packages `theme-d-intr-0.20.5.tar.xz` and `theme-d-intr_0.20.5-2.debian.tar.xz` there.
- Execute the following commands

```
tar xvf theme-d-intr-0.20.4.tar.xz
ln -s theme-d-intr-0.20.4.tar.xz theme-d-intr_0.20.4.orig.tar.xz
cd theme-d-intr-0.20.4
tar xvf ../theme-d-intr_0.20.4-1.debian.tar.xz
dpkg-buildpackage --no-sign -b
```

  in the directory.

- Give command `cd ..` and install the packages with commands

```
sudo dpkg -i theme-d-intr_0.20.5-2_ARCH.deb
sudo dpkg -i theme-d-intr-dev_0.20.5-2_all.deb
```

  The architecture ARCH can be computed by

```
dpkg-architecture -q DEB_HOST_ARCH
```

  You can install the example programs by installing one or both of the packages `theme-d-intr-gtk4-examples_0.20.5-2_all.deb` and `theme-d-intr-gtk3-examples_0.20.5-2_all.deb`.

- Ensure that your Guile installation file `~/.guile` contains the following lines:

```
(use-modules (oop goops))
(default-duplicate-binding-handler
  '(merge-generics replace warn-override-core warn last))
```

Other UNIX Systems

- Install the G-Golf library. See https://www.gnu.org/software/g-golf/install.html [https://www.gnu.org/software/g-golf/install.html] .
- Unpack package `theme-d-intr-0.20.5.tar.xz` into some directory.
- Change to the subdirectory `theme-d-intr-0.20.5` and give commands

```
./configure
make
sudo make install-complete
```

- Ensure that your Guile installation file `~/.guile` contains the following lines:

```
(use-modules (oop goops))
(default-duplicate-binding-handler
  '(merge-generics replace warn-override-core warn last))
```

Using Theme-D-Intr without Installing

It is possible to use the software without installing it. First give the following commands in the subdirectory `theme-d-intr-0.20.5`:

```
./configure
make
```

Modify the file `~/.guile` as in a normal installation (see the previous section). Now you can launch the uninstalled environment by giving command

```
./uninstalled-env bash
```

in subdirectory `meta`. You also have to take care that the Theme-D module search path (option `-m` or `-M` for the Theme-D compiler and linker) contains the directory `xxx/theme-d-intr-0.20.5/theme-d-code`. See Theme-D User Guide for a description of the module search path.

Example Programs

It is recommended that you copy the example programs into your home directory for building them. If you have installed Theme-D-Intr normally go to directory `/usr/share/doc/theme-d-intr-gtk4-examples`. If you use Theme-D-Intr without installed it go to the subdirectory `examples` in the source code. Give command

```
cp -avi intr-gtk4-examples MYDIR
```

where `MYDIR` is the directory where you want to install the programs. Then, go to the directory `/usr/share/doc/theme-d-intr-gtk3-examples` or `examples` and give command

```
cp -avi intr-gtk3-examples MYDIR
```

The example programs `appwindow`, `hello`, `hello2`, `hello3`, `calc`, `calc2`, `custom-layout-example`, `flag-demo`, and `theme-d-intr-demo` for GTK 4 are located in directory `/usr/share/doc/theme-d-intr-gtk4-examples/intr-gtk4-examples` and in directory `examples/intr-gtk4-examples` in the source package. You can build the example programs with command `make -f user.mk` and run them with command

```
GUILE_LOAD_PATH=../..:$GUILE_LOAD_PATH run-theme-d-program \
<program-name>
```

where `<program-name>` is the name of the target `.go` file of the program (`main.go` or `run-demo.go`). For program `hello` the command is

```
GUILE_LOAD_PATH=.:$GUILE_LOAD_PATH run-theme-d-program hello.go
```

Example program `intr-gtk4-examples/calc2` is linked modularly and it shall be run with command

```
run-theme-d-program-m -g ../..: main.go
```

or in uninstalled mode

```
run-theme-d-program-m -g ../..:../../../theme-d-code: main.go
```

Before you run it you have to install its configuration files with command

```
./install-calc2.sh
```

If you have a Debian-based Linux operating system you have to ensure that packages `gir1.2-gtk-4.0` and `gir1.2-adw-1` are installed in your system.

The example programs `hello`, `hello2`, `hello3`, `calc`, and `theme-d-intr-demo` for GTK 3 are located in directory `/usr/share/doc/theme-d-intr-gtk3-examples/intr-gtk3-examples` and in directory `examples/intr-gtk3-examples` in the source package. You can build the example programs with command `make -f user.mk` and run them with command

```
GUILE_LOAD_PATH=../..:$GUILE_LOAD_PATH run-theme-d-program \
  <program-name>
```

where `<program-name>` is the name of the target `.go` file of the program (`main.go` or `run-demo.go`). For program `hello` the command is

```
GUILE_LOAD_PATH=.:$GUILE_LOAD_PATH run-theme-d-program hello.go
```

If you have a Debian-based Linux operating system you have to ensure that package `gir1.2-gtk-3.0` is installed in your system.

Writing Programs using Theme-D-Intr

It is recommended you create a new directory for your program. First you have to list the G-Golf classes and functions you use in file `intr-imports.scm`. Its format is:

```
(intr-entities
  (version namespace ver)
  ...
  (classes
    (namespace class)
    ...)
  (functions
```

```
    (namespace function)
    ...)
  (rejected-methods
    name ...)
  (overridden-functions
    (name (argument-type ...) result-type attributes)
    ...)
  (strip-boolean-result
    name ...)
  (ignore-slots
    (namespace class) slot-name ...)
    ...)
```

Namespace is the library where the definitions are imported, such as `Gtk`. Class names are given in format `MyClassName`. Note that the namespace is not included in the class or function name (`Widget` instead of `GtkWidget`). Methods belonging to a class are imported automatically when the class is imported and they must not be listed in the `functions` section. Rejected methods are generic function names for which we do not generate methods. There are two main reasons for rejecting a method:

- The method name overlaps a nongeneric function name, e.g. `append`.
- The methods of some name violate the covariance rule, see Theme-D Language Manual [https://www.iki.fi/tohoyn/theme-d/theme-d-language-manual-7.2.3.pdf].

The purpose of `overridden-functions` is to handle functions overridden in G-Golf. The argument types, the result type and the attributes are given in Theme-D format. The `version` specifies the version used for a namespace. The version must be enclosed in double quotes. The boolean function result of a function can be stripped by listing it under `strip-boolean-result`. See G-Golf documentation for this feature. Form `ignore-slots` can be used to prevent code generation for some specific slots. For example, `(ignore-slots (Gtk Label) xalign yalign)`. Here is an example import file:

```
(intr-entities
  (version Gtk "4.0")
  (classes
    (Gdk Display)
    (Gtk Widget)
    (Gtk CssProvider)
    (Gtk StyleContext)
    (Gtk Application)
    (Gtk ApplicationWindow)
    (Gtk Button)
    (Gtk Box)
    (Gtk ScrolledWindow)
    (Gtk TextBuffer)
    (Gtk TextView)
    (Gtk Entry)
    (Gtk EntryBuffer)
    (Gtk Label)
    (Gtk Separator)
    (Gtk TextTagTable)
    (Gtk TextTag)
    (Gtk TextMark)))
```

and here is another:

```
(intr-entities
  (version Gtk "3.0")
  (classes
    (Gtk Widget)
    (Gtk Window)
    (Gtk Button))
  (functions
    (Gtk init)
    (Gtk main)
    (Gtk main_quit))
  (rejected-methods
    append map get-style activate compare copy)
  (overridden-functions
    (gtk-container-child-get-property (<gtk-container>
      <gtk-widget> <string>)
      <object> nonpure)
    (gtk-container-child-set-property
      (<gtk-container> <gtk-widget> <string> <object>)
      <none> nonpure)))
```

The following three files are generated from the definition file:

- Interface file `_intr-imports.thi`
- Body file `_intr-imports.thb`
- Target module `_intr-imports-target.scm`

The program and modules using G-Golf should import the module `_intr-imports`. In case you have a multiple part program name, such as `(myexamples myprogram)`, module `_intr-imports` has a multiple part name too, e.g. `(myexamples _intr-imports)`. File `_intr-imports-target.scm` is used by the linked program to provide access to the introspected libraries.

The following commands are used to generate Theme-D-Intr files:

- `generate-intr-interface`
- `generate-intr-body`
- `generate-intr-target-module`

If you use a multiple part program name you must give the `-m` option for these commands. The option argument is the imports module name for the first two commands, e.g. `-m "(myexamples _intr-imports)"` and the setter module name for the fourth command, e.g. `-m "(myexamples _intr-imports-target)"`. If you want to generate accessor methods `NAMESPACE-CLASS-get-SLOT` and `NAMESPACE-CLASS-set-SLOT!` use option `--generate-accessors` for all of these programs. Note that you can access slots without accessor methods by using procedures `slot-ref` and `slot-set!`.

When you link a program using Theme-D-Intr you have to give the following options to the Theme-D linker:

- `-x "(g-golf)"`
- `-x "(guile-theme-d-intr support)"`
- `-x "<target-module>"`
- `--duplicates="merge-generics replace warn-override-core warn last"`

Here `<target-module>` is the Theme-D name of the generated target module. For a program with a single part name it is `(_intr-imports-target)`. If you use the extra support module give also option `-x "(guile-theme-d-intr support2-gtk3)"` or `-x "(guile-theme-d-intr support2-gtk4)"`.

You have to ensure that you have the introspection files for the external libraries your program uses (with G-Golf) installed in your system. If your program uses GTK and you have a Debian-based Linux system this can be ensured by having package `gir1.2-gtk-3.0` (for GTK 3.0) or `gir1.2-gtk-4.0` (for GTK 4.0) in your system.

See also the `user.mk` makefiles of the example programs. Example program `hello` has a single part program name and the other example programs multiple part program names.

When you run programs using Theme-D-Intr the environment variable `GUILE_LOAD_PATH` has to contain the root directory of your program. If you use a single part program name this is the directory of your program.

Distributing Programs using Theme-D-Intr

If you use a Debian-based operating system the following packages are required to run compiled (built) Theme-D programs using Theme-D-Intr:

- `theme-d-rte`
- `th-scheme-utilities`
- `libthemedsupport`
- `theme-d-intr`

If you have a non-Debian operating system see the Theme-D User Guide for the files required to distribute a compiled Theme-D program. In addition to those, you need file `support.scm` from the Theme-D-Intr source package. If you use files `support2-gtk3.scm` or `support2-gtk4.scm` you need distribute that, too. These files have to be installed in subdirectory `guile-theme-d-support` somewhere in the Guile library search path. In UNIX systems this path usually contains directory `/usr/share/guile/site/3.0/` or `/usr/share/guile/site/2.2/`.
If your application uses modular linking you also have to distribute files `__intf_support.go` and `__impl_support.go` into subdirectory `theme-d-intr` of the Theme-D target library directory. You can obtain this directory with command

```
get-theme-d-config-var target-lib-dir
```

Help with callback types

You can use command `describe-cb-type` to get the Theme-D name and Theme-D procedure type of a callback type. The syntax of the command is

```
describe-cb-type library callback-name [version]
```

Argument `library` is the library where the callback is defined. Argument `callback-name` is the name of the callback type. Optional argument `version` is the version of the library. Here is an example:

```
describe-cb-type Gtk CustomMeasureFunc 4.0
```