Guile-GNOME: Libglade

version 2.16.2, updated 9 December 2011

James Henstridge

Short Contents

1 Overview

`(gnome glade)` wraps the libglade interface XML interface definition library for Guile. It is a part of Guile-GNOME.

Glade files are XML descriptions of a GTK+ widget hierarchy. They are usually created with the Glade user interface editor. A simple example might look like this:

```
(define xml-buf "<?xml version=\"1.0\" standalone=\"no\"?>
<!DOCTYPE glade-interface SYSTEM \"http://glade.gnome.org/glade-2.0.dtd\">

<glade-interface>

<widget class=\"GtkWindow\" id=\"window1\">
  <property name=\"title\" translatable=\"yes\">window1</property>
  <child>
    <widget class=\"GtkLabel\" id=\"label1\">
      <property name=\"label\" translatable=\"yes\">Hello world!</property>
    </widget>
  </child>
</widget>

</glade-interface>")

(define glade-xml (glade-xml-new-from-buffer xml-buf))
(define main-window (get-widget glade-xml "window1"))
(show-all main-window)
(g-main-loop-run (g-main-loop-new))
```

Of course, in practice you are more likely to load the XML from a file; in that case you would use `glade-xml-new`.

Also interesting are the `signal-connect` and `signal-autoconnect` implementations; read on for more details.

See the documentation for `(gnome gobject)` for more information on Guile-GNOME.

2 GladeXML

Allows dynamic loading of user interfaces from XML descriptions.

2.1 Overview

This object represents an 'instantiation' of an XML interface description. When one of these objects is created, the XML file is read, and the interface is created. The GladeXML object then provides an interface for accessing the widgets in the interface by the names assigned to them inside the XML description.

The GladeXML object can also be used to connect handlers to the named signals in the description. Libglade also provides an interface by which it can look up the signal handler names in the program's symbol table and automatically connect as many handlers up as it can that way.

2.2 Usage

`<glade-xml>` [Class]

Derives from `<gobject>`.

This class defines no direct slots.

`glade-xml-new` (*fname* `mchars`) (*root* `mchars`) (*domain* `mchars`) [Function]
⇒ (*ret* `<glade-xml>`)

Creates a new GladeXML object (and the corresponding widgets) from the XML file *fname*. Optionally it will only build the interface from the widget node *root* (if it is not '`#f`'). This feature is useful if you only want to build say a toolbar or menu from the XML file, but not the window it is embedded in. Note also that the XML parse tree is cached to speed up creating another GladeXML object for the same file

fname the XML file name.

root the widget node in *fname* to start building from (or '`#f`')

domain the translation domain for the XML file (or '`#f`' for default)

ret the newly created GladeXML object, or NULL on failure.

`glade-xml-new-from-buffer` (*buffer* `mchars`) (*root* `mchars`) [Function]
(*domain* `mchars`) ⇒ (*ret* `<glade-xml>`)

Creates a new GladeXML object (and the corresponding widgets) from the buffer *buffer*. Optionally it will only build the interface from the widget node *root* (if it is not '`#f`'). This feature is useful if you only want to build say a toolbar or menu from the XML document, but not the window it is embedded in.

buffer the memory buffer containing the XML document.

size the size of the buffer.

root the widget node in *buffer* to start building from (or '`#f`')

domain the translation domain to use for this interface (or '`#f`')

ret the newly created GladeXML object, or NULL on failure.

glade-xml-signal-connect (*self* `<glade-xml>`) (*handlername* `mchars`) (*proc* `scm`) [Function]

signal-connect [Method]

In the glade interface descriptions, signal handlers are specified for widgets by name. This function allows you to connect a Scheme function to all signals in the GladeXML file with the given signal handler name.

self the GladeXML object

handlername
the signal handler name

func the signal handler function

glade-xml-signal-autoconnect (*self* `<glade-xml>`) (*module* `scm`) [Function]

signal-autoconnect [Method]

This function is a variation of `glade-xml-signal-autoconnect`. It will bind signal handers to the result of evaluating each signal handler name within *module*. So for example, if you have ten widgets defining a handler as the string `"on-clicked"`, you might do this:

```
(define (on-clicked . args)
  (display "Doing me thing...\n"))
(signal-autoconnect glade-xml (current-module))
;; now the on-clicked handlers are connected
```

self the GladeXML object.

module A scheme module in which to evaluate the signal handlers definitions.

glade-xml-get-widget (*self* `<glade-xml>`) (*name* `mchars`) ⇒ (*ret* `<gtk-widget>`) [Function]

get-widget [Method]

This function is used to get a pointer to the GtkWidget corresponding to *name* in the interface description. You would use this if you have to do anything to the widget after loading.

self the GladeXML object.

name the name of the widget.

ret the widget matching *name*, or '`#f`' if none exists.

glade-xml-get-widget-prefix (*self* `<glade-xml>`) (*name* `mchars`) ⇒ (*ret* `glist-of`) [Function]

get-widget-prefix [Method]

This function is used to get a list of pointers to the GtkWidget(s) with names that start with the string *name* in the interface description. You would use this if you have to do something to all of these widgets after loading.

self the GladeXML object.

name the name of the widget.

ret A list of the widget that match *name* as the start of their name, or '`#f`' if none exists.

glade-get-widget-name (*widget* `<gtk-widget>`) ⇒ (*ret* `mchars`) [Function]

Used to get the name of a widget that was generated by a GladeXML object.

widget the widget

ret the name of the widget.

glade-get-widget-tree (*widget* `<gtk-widget>`) [Function]
⇒ (*ret* `<glade-xml>`)

This function is used to get the GladeXML object that built this widget.

widget the widget

ret the GladeXML object that built this widget.

3 Undocumented

The following symbols, if any, have not been properly documented.

3.1 (gnome gw libglade)

`glade-xml-relative-file` [Variable]

Type Index

Function Index