Guile-GNOME: libgnomeui

version 2.16.2, updated 9 December 2011

Kjartan Maraas
Anders Carlsson
many others

Short Contents

1 Overview 1
2 GnomeAuthentication 2
3 GnomeClient 3
4 GnomeDateEdit 9
5 GnomeMultiScreen 12
6 GnomePassword 13
7 gnome-vfs-util 17
8 Undocumented 18
Type Index 19
Function Index 20

1 Overview

`(gnome gnome-ui)` wraps some pieces of the libgnomeui library for Guile. It is a part of Guile-GNOME.

Libgnomeui historically was a staging ground for code meant to go into GTK. In the modern environment, it has largely been replaced, its functionality having been pushed into GTK+ itself.

However, as of 2008 it still has some utility as a kind of glue layer between lower-level modules, with the authentication manager, the GNOME-VFS GDK-Pixbuf integration, and the session manager code.

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

2 GnomeAuthentication

2.1 Overview

2.2 Usage

`gnome-authentication-manager-init` [Function]

This function checks for thread support and does a thread initialisation, if the support is available. Also sets the default sync and async gnome-vfs callbacks for various types of authentication.

Note: If you call this, and you use threads with gtk+, you must never hold the gdk lock while doing synchronous gnome-vfs calls. Otherwise an authentication callback presenting a dialog could try to grab the already held gdk lock, causing a deadlock.

Since 2.4

3 GnomeClient

Interactions with the session manager.

3.1 Overview

3.2 Usage

`<gnome-client>` [Class]

Derives from `<gtk-object>`.

This class defines no direct slots.

`save-yourself` (*arg0* `<gint>`) [Signal on `<gnome-client>`]
(*arg1* `<gnome-save-style>`) (*arg2* `<gboolean>`)
(*arg3* `<gnome-interact-style>`) (*arg4* `<gboolean>`) ⇒ `<gboolean>`

Called when either a "SaveYourself" or a "SaveYourselfPhase2" call is made by the session manager.

`die` [Signal on `<gnome-client>`]

Called when the session manager wants the client to shut down.

`save-complete` [Signal on `<gnome-client>`]

Called when the session manager has finished checkpointing all of the clients. Clients are then free to change their state.

`shutdown-cancelled` [Signal on `<gnome-client>`]

Called if the session manager had sent a "SaveYourself" to all clients in preparation for shutting down and the shutdown was then cancelled. A client can then continue running and change its state.

`connect` (*arg0* `<gboolean>`) [Signal on `<gnome-client>`]

Called once the client has been connected to the signal manager.

`disconnect` [Signal on `<gnome-client>`]

Called when the client is disconnected from the session manager.

`gnome-master-client` ⇒ (*ret* `<gnome-client>`) [Function]

Get the master session management client. This master client gets a client id, that may be specified by the '–sm-client-id' command line option. A master client will be generated by `gnome-program-init`. If possible the master client will contact the session manager after command-line parsing is finished (unless `gnome-client-disable-master-connection` was called). The master client will also set the SM_CLIENT_ID property on the client leader window of your application.

Additionally, the master client gets some static arguments set automatically (see `gnome-client-add-static-arg` for static arguments): `gnome-program-init` passes all the command line options which are recognised by gtk as static arguments to the master client.

ret Pointer to the master client

`gnome-client-get-config-prefix` (*self* `<gnome-client>`) [Function]
⇒ (*ret* `mchars`)
`get-config-prefix` [Method]

Get the config prefix for a client. This config prefix provides a suitable place to store any details about the state of the client which can not be described using the app's command line arguments (as set in the restart command). You may push the returned value using `gnome-config-push-prefix` and read or write any values you require.

client Pointer to GNOME session client object.

ret Config prefix. The returned string belongs to libgnomeui library and should NOT be freed by the caller.

`gnome-client-get-flags` (*self* `<gnome-client>`) [Function]
⇒ (*ret* `<gnome-client-flags>`)
`get-flags` [Method]

Determine the client's status with the session manager.,

client Pointer to GNOME session client object.

ret Various `<gnome-client-flag>` flags which have been or'd together.

`gnome-client-set-restart-style` (*self* `<gnome-client>`) [Function]
(*style* `<gnome-restart-style>`)
`set-restart-style` [Method]

Tells the session manager how the client should be restarted in future session. The options are given by the `<gnome-restart-style>` enum.

client Pointer to GNOME session client object.

style When to restart the client.

`gnome-client-set-priority` (*self* `<gnome-client>`) [Function]
(*priority* `unsigned-int`)
`set-priority` [Method]

The gnome-session manager restarts clients in order of their priorities in a similar way to the start up ordering in SysV. This function allows the app to suggest a position in this ordering. The value should be between 0 and 99. A default value of 50 is assigned to apps that do not provide a value. The user may assign a different priority.

client Pointer to GNOME session client object.

priority Position of client in session start up ordering.

`gnome-client-set-current-directory` (*self* `<gnome-client>`) [Function]
(*dir* `mchars`)
`set-current-directory` [Method]

Set the directory to be in when running shutdown, discard, restart, etc. commands.

client Pointer to GNOME session client object.

dir Directory path.

gnome-client-set-environment (*self* <gnome-client>) (*name* mchars) (*value* mchars) [Function]
set-environment [Method]

Set an environment variable to be placed in the client's environment prior to running restart, shutdown, discard, etc. commands.

client Pointer to GNOME session client object.

name Name of the environment variable

value Value of the environment variable

gnome-client-set-process-id (*self* <gnome-client>) (*pid* int) [Function]
set-process-id [Method]

The client should tell the session manager the result of `getpid`. However, GNOME does this automatically; so you do not need this function.

client Pointer to GNOME session client object.

pid PID to set as the client's PID.

gnome-client-set-program (*self* <gnome-client>) (*program* mchars) [Function]
set-program [Method]

Used to tell the session manager the name of your program. Set automatically; this function isn't needed.

client Pointer to GNOME session client object.

program Name of the program.

gnome-client-set-user-id (*self* <gnome-client>) (*id* mchars) [Function]
set-user-id [Method]

Tell the session manager the user's login name. GNOME does this automatically; no need to call the function.

client Pointer to GNOME session client object.

id Username.

gnome-client-save-any-dialog (*self* <gnome-client>) (*dialog* <gtk-dialog>) [Function]
save-any-dialog [Method]

May be called during a "save_youself" handler to request that a (modal) dialog is presented to the user. The session manager decides when the dialog is shown, but it will not be shown it unless the session manager is sending an interaction style of `<gnome-interact-any>`. A "Cancel Logout" button will be added during a shutdown.

client Pointer to `<gnome-client>` object.

dialog Pointer to GNOME dialog widget (a `<gtk-dialog>` widget).

gnome-client-save-error-dialog (*self* <gnome-client>) (*dialog* <gtk-dialog>) [Function]
save-error-dialog [Method]

May be called during a "save_youself" handler when an error has occurred during the save. The session manager decides when the dialog is shown, but it will not be shown it unless the session manager is sending an interaction style of `<gnome-interact-any>`. A "Cancel Logout" button will be added during a shutdown.

client Pointer to `<gnome-client>` object.

dialog Pointer to GNOME dialog widget (a `<gtk-dialog>` widget).

gnome-client-request-phase-2 (*self* `<gnome-client>`) [Function]
request-phase-2 [Method]

Request the session managaer to emit the "save_yourself" signal for a second time after all the clients in the session have ceased interacting with the user and entered an idle state. This might be useful if your app manages other apps and requires that they are in an idle state before saving its final data.

client A `<gnome-client>` object.

gnome-client-request-save (*self* `<gnome-client>`) [Function]
(*save_style* `<gnome-save-style>`) (*shutdown* `bool`)
(*interact_style* `<gnome-interact-style>`) (*fast* `bool`) (*global* `bool`)
request-save [Method]

Request the session manager to save the session in some way. The arguments correspond with the arguments passed to the "save_yourself" signal handler.

The save_style indicates whether the save should affect data accessible to other users (`<gnome-save-global>`) or only the state visible to the current user (`<gnome-save-local>`) or both. Setting shutdown to '`#t`' will initiate a logout. The interact_style specifies which kinds of interaction will be available. Setting fast to '`#t`' will limit the save to setting the session manager properties plus any essential data. Setting the value of global to '`#t`' will request that all the other apps in the session do a save as well. A global save is mandatory when doing a shutdown.

client Pointer to GNOME session client object.

save-style Save style to request.

shutdown Whether to log out of the session.

interact-style
Whether to allow user interaction.

fast Minimize activity to save as soon as possible.

global Request that all other apps in the session also save their state.

gnome-client-flush (*self* `<gnome-client>`) [Function]
flush [Method]

This will force the underlying connection to the session manager to be flushed. This is useful if you have some pending changes that you want to make sure get committed.

client A `<gnome-client>` instance.

gnome-client-new ⇒ (*ret* `<gnome-client>`) [Function]

Allocates memory for a new GNOME session management client object. After allocating, the client tries to connect to a session manager. You probably want to use `gnome-master-client` instead.

ret Pointer to a newly allocated GNOME session management client object.

`gnome-client-new-without-connection` ⇒ (*ret* `<gnome-client>`) [Function]

Allocates memory for a new GNOME session management client object. You probably want to use `gnome-master-client` instead.

ret Pointer to a newly allocated GNOME session management client object.

`gnome-client-connect` (*self* `<gnome-client>`) [Function]
`connect` [Method]

Causes the client to connect to the session manager. Usually happens automatically; no need to call this function.

client A `<gnome-client>` instance.

`gnome-client-disconnect` (*self* `<gnome-client>`) [Function]
`disconnect` [Method]

Disconnect the client from the session manager.

client A `<gnome-client>` instance.

`gnome-client-set-id` (*self* `<gnome-client>`) (*id* `mchars`) [Function]
`set-id` [Method]

Set the client's session management ID; must be done before connecting to the session manager. There is usually no reason to call this function.

client A `<gnome-client>` instance.

id Session management ID.

`gnome-client-get-id` (*self* `<gnome-client>`) ⇒ (*ret* `mchars`) [Function]
`get-id` [Method]

Returns session management ID

client A `<gnome-client>` instance.

ret Session management ID for this client; '`#f`' if not connected to a session manager.

`gnome-client-get-previous-id` (*self* `<gnome-client>`) [Function]
⇒ (*ret* `mchars`)
`get-previous-id` [Method]

Get the session management ID from the previous session.

client A `<gnome-client>` instance.

ret Pointer to the session management ID the client had in the last session, or '`#f`' if it was not in a previous session.

`gnome-client-get-desktop-id` (*self* `<gnome-client>`) [Function]
⇒ (*ret* `mchars`)
`get-desktop-id` [Method]

Get the client ID of the desktop's current instance, i.e. if you consider the desktop as a whole as a session managed app, this returns its session ID using a GNOME extension to session management. May return '`#f`' for apps not running under a recent version of gnome-session; apps should handle that case.

client A `<gnome-client>` instance.

ret Session ID of GNOME desktop instance, or '`#f`' if none.

4 GnomeDateEdit

4.1 Overview

4.2 Usage

`<gnome-date-edit>` [Class]

Derives from `<gtk-hbox>`.

This class defines the following slots:

`time` The time currently selected

`dateedit-flags`
: Flags for how DateEdit looks

`lower-hour`
: Lower hour in the time popup selector

`upper-hour`
: Upper hour in the time popup selector

`initial-time`
: The initial time

`time-changed` [Signal on `<gnome-date-edit>`]

`date-changed` [Signal on `<gnome-date-edit>`]

`gnome-date-edit-new` (*the_time* `long`) (*show_time* `bool`) (*use_24_format* `bool`) ⇒ (*ret* `<gtk-widget>`) [Function]

Creates a new `<gnome-date-edit>` widget which can be used to provide an easy to use way for entering dates and times. If *the-time* is 0 then current time is used.

the-time date and time to be displayed on the widget

show-time whether time should be displayed

use-24-format
: whether 24-hour format is desired for the time display.

ret a new `<gnome-date-edit>` widget.

`gnome-date-edit-new-flags` (*the_time* `long`) (*flags* `<gnome-date-edit-flags>`) ⇒ (*ret* `<gtk-widget>`) [Function]

Creates a new `<gnome-date-edit>` widget with the specified flags. If *the-time* is 0 then current time is used.

the-time The initial time for the date editor.

flags A bitmask of GnomeDateEditFlags values.

ret the newly-created date editor widget.

gnome-date-edit-construct (*self* `<gnome-date-edit>`) (*the_time* `long`) (*flags* `<gnome-date-edit-flags>`) [Function]

construct [Method]

For language bindings and subclassing only

gde The `<gnome-date-edit>` object to construct

the-time The initial time for the date editor.

flags A bitmask of GnomeDateEditFlags values.

gnome-date-edit-set-time (*self* `<gnome-date-edit>`) (*the_time* `long`) [Function]

set-time [Method]

Changes the displayed date and time in the GnomeDateEdit widget to be the one represented by *the-time*. If *the-time* is 0 then current time is used.

gde the GnomeDateEdit widget

the-time The time and date that should be set on the widget

gnome-date-edit-get-time (*self* `<gnome-date-edit>`) ⇒ (*ret* `long`) [Function]

get-time [Method]

gde The GnomeDateEdit widget

ret the time entered in the GnomeDateEdit widget

gnome-date-edit-set-popup-range (*self* `<gnome-date-edit>`) (*low_hour* `int`) (*up_hour* `int`) [Function]

set-popup-range [Method]

Sets the range of times that will be provide by the time popup selectors.

gde The GnomeDateEdit widget

low-hour low boundary for the time-range display popup.

up-hour upper boundary for the time-range display popup.

gnome-date-edit-set-flags (*self* `<gnome-date-edit>`) (*flags* `<gnome-date-edit-flags>`) [Function]

set-flags [Method]

Changes the display flags on an existing date editor widget.

gde The date editor widget whose flags should be changed.

flags The new bitmask of GnomeDateEditFlags values.

gnome-date-edit-get-flags (*self* `<gnome-date-edit>`) ⇒ (*ret* `int`) [Function]

get-flags [Method]

Queries the display flags on a date editor widget.

gde The date editor whose flags should be queried.

ret The current display flags for the specified date editor widget.

`gnome-date-edit-get-initial-time` (*self* `<gnome-date-edit>`) [Function]
⇒ (*ret* `long`)

`get-initial-time` [Method]

Queries the initial time that was set using the `<gnome-date-edit-set-time>` or during creation

gde The date editor whose initial time should be queried

ret The initial time in seconds (standard time_t format)

5 GnomeMultiScreen

5.1 Overview

5.2 Usage

`gnome-url-show-on-screen` (*url* `mchars`) (*screen* `<gdk-screen>`) ⇒ (*ret* `bool`) [Function]

Like `gnome-url-show`, but ensures that the viewer application appears on *screen*.

url The url to display. Should begin with the protocol to use (e.g. "http:", "ghelp:", etc)

screen a `<gdk-screen>`.

error Used to store any errors that result from trying to display the *url*.

ret '`#t`' if everything went fine, '`#f`' otherwise (in which case *error* will contain the actual error).

Since 2.6

`gnome-help-display-on-screen` (*file_name* `mchars`) (*link_id* `mchars`) (*screen* `<gdk-screen>`) ⇒ (*ret* `bool`) [Function]

Like `gnome-help-display`, but ensures that the help viewer application appears on *screen*.

file-name The name of the help document to display.

link-id Can be '`#f`'. If set, refers to an anchor or section id within the requested document.

screen a `<gdk-screen>`.

error A `<g-error>` instance that will hold the specifics of any error which occurs during processing, or '`#f`'

ret '`#t`' on success, '`#f`' otherwise (in which case *error* will contain the actual error).

Since 2.6

`gnome-help-display-uri-on-screen` (*help_uri* `mchars`) (*screen* `<gdk-screen>`) ⇒ (*ret* `bool`) [Function]

Like `gnome-help-display-uri`, but ensures that the help viewer application appears on *screen*.

help-uri The URI to display.

screen a `<gdk-screen>`.

error A `<g-error>` instance that will hold the specifics of any error which occurs during processing, or '`#f`'

ret '`#t`' on success, '`#f`' otherwise (in which case *error* will contain the actual error).

Since 2.6

6 GnomePassword

6.1 Overview

6.2 Usage

`<gnome-password-dialog>` [Class]

Derives from `<gtk-dialog>`.

This class defines the following slots:

`show-username`
: show-username

`show-domain`
: show-domain

`show-password`
: show-password

`show-new-password`
: show-new-password

`show-new-password-quality`
: show-new-password-quality

`show-userpass-buttons`
: show-userpass-buttons

`show-remember`
: show-remember

`readonly-username`
: readonly-username

`readonly-domain`
: readonly-domain

`anonymous`
: anonymous

`remember-mode`
: remember-mode

`message`
: message

`message-markup`
: message-markup

`username`
: username

`domain`
: domain

`password`
: password

`new-password`
: new-password

`gnome-password-dialog-anon-selected` [Function]
(*self* `<gnome-password-dialog>`) ⇒ (*ret* `bool`)

`anon-selected` [Method]

Checks whether anonymous support is set to `#t` and the radio button for connecting as anonymous user is active.

password-dialog
A `<gnome-password-dialog>`

ret `#t` if anonymous support is set and the radio button is active, `#f` otherwise.

`gnome-password-dialog-get-domain` [Function]
(*self* `<gnome-password-dialog>`) ⇒ (*ret* `mchars`)

`get-domain` [Method]

Gets the domain name from the password dialog.

password-dialog
A `<gnome-password-dialog>`

ret The domain name, a char*.

Since 2.4

`gnome-password-dialog-get-password` [Function]
(*self* `<gnome-password-dialog>`) ⇒ (*ret* `mchars`)

`get-password` [Method]

Gets the password from the password dialog.

password-dialog
A `<gnome-password-dialog>`

ret The password, a char*.

Since 2.4

`gnome-password-dialog-get-username` [Function]
(*self* `<gnome-password-dialog>`) ⇒ (*ret* `mchars`)

`get-username` [Method]

Gets the username from the password dialog.

password-dialog
A `<gnome-password-dialog>`

ret The username, a char*.

Since 2.4

`gnome-password-dialog-new` (*dialog_title* `mchars`) (*message* `mchars`) [Function]
(*username* `mchars`) (*password* `mchars`) (*readonly_username* `bool`)
⇒ (*ret* `<gtk-widget>`)

Creates a new password dialog with an optional title, message, username, password etc. The user will be given the option to save the password for this session only or store it permanently in her keyring.

dialog-title
The title of the dialog

message Message text for the dialog

username The username to be used in the dialog

password Password to be used

readonly-username
Boolean value that controls whether the user can edit the username or not

ret A new password dialog.

Since 2.4

`gnome-password-dialog-run-and-block` [Function]
(*self* `<gnome-password-dialog>`) ⇒ (*ret* `bool`)

`run-and-block` [Method]

Gets the user input from PasswordDialog.

password-dialog
A `<gnome-password-dialog>`

ret '`#t`' if "Connect" button is pressed. '`#f`' if "Cancel" button is pressed.

Since 2.4

`gnome-password-dialog-set-domain` [Function]
(*self* `<gnome-password-dialog>`) (*domain* `mchars`)

`set-domain` [Method]

Sets the domain field in the password dialog to *domain*.

password-dialog
A `<gnome-password-dialog>`

domain The domain that should be set

Since 2.4

`gnome-password-dialog-set-password` [Function]
(*self* `<gnome-password-dialog>`) (*password* `mchars`)

`set-password` [Method]

Sets the password in the password dialog.

password-dialog
A `<gnome-password-dialog>`

password The password that should be set

Since 2.4

`gnome-password-dialog-set-remember` [Function]
(*self* `<gnome-password-dialog>`)
(*remember* `<gnome-password-dialog-remember>`)

`set-remember` [Method]

Based on the value of `<gnome-password-dialog-remember>`, sets the state of the check buttons to remember password for the session and save password to keyring .

password-dialog
A `<gnome-password-dialog>`.

remember A `<gnome-password-dialog-remember>`.

Since 2.6

gnome-password-dialog-set-username (*self* `<gnome-password-dialog>`) (*username* `mchars`) [Function]

set-username [Method]

Sets the username in the password dialog.

password-dialog
A `<gnome-password-dialog>`

username The username that should be set

Since 2.4

7 gnome-vfs-util

7.1 Overview

7.2 Usage

`gnome-gdk-pixbuf-new-from-uri` (*uri* `mchars`) [Function]
⇒ (*ret* `<gdk-pixbuf>`)

Loads a GdkPixbuf from the image file *uri* points to, scaling it to the desired size. If you pass -1 for *width* or *height* then the value specified in the file will be used.

When preserving aspect ratio, if both height and width are set the size is picked such that the scaled image fits in a width * height rectangle.

uri the uri of an image

ret The loaded pixbuf, or NULL on error

Since 2.14

8 Undocumented

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

8.1 (gnome gw libgnomeui)

`gnome-authentication-manager-dialog-is-visible` [Variable]

`gnome-client-get-global-config-prefix` [Variable]

`gnome-client-set-global-config-prefix` [Variable]

`gnome-gdk-pixbuf-new-from-uri-at-scale` [Variable]

`gnome-help-display-desktop-on-screen` [Function]

`gnome-help-display-with-doc-id-on-screen` [Function]

`gnome-password-dialog-get-remember` [Variable]

`gnome-password-dialog-set-readonly-domain` [Variable]

`gnome-password-dialog-set-readonly-username` [Variable]

`gnome-password-dialog-set-show-domain` [Variable]

`gnome-password-dialog-set-show-password` [Variable]

`gnome-password-dialog-set-show-remember` [Variable]

`gnome-password-dialog-set-show-username` [Variable]

`gnome-password-dialog-set-show-userpass-buttons` [Variable]

Type Index

<gnome-client> 3
<gnome-date-edit> 9
<gnome-password-dialog> 13

Function Index

A

anon-selected 14

C

connect 7
connect on <gnome-client> 3
construct 10

D

date-changed on <gnome-date-edit> 9
die on <gnome-client> 3
disconnect 7
disconnect on <gnome-client> 3

F

flush 6

G

get-config-prefix 4
get-desktop-id 7
get-domain 14
get-flags 4, 10
get-id 7
get-initial-time 11
get-password 14
get-previous-id 7
get-time 10
get-username 14
gnome-authentication-manager-init 2
gnome-client-connect 7
gnome-client-disconnect 7
gnome-client-flush 6
gnome-client-get-config-prefix 4
gnome-client-get-desktop-id 7
gnome-client-get-flags 4
gnome-client-get-id 7
gnome-client-get-previous-id 7
gnome-client-new 6
gnome-client-new-without-connection 7
gnome-client-request-phase-2 6
gnome-client-request-save 6
gnome-client-save-any-dialog 5
gnome-client-save-error-dialog 5
gnome-client-set-current-directory 4
gnome-client-set-environment 5
gnome-client-set-id 7
gnome-client-set-priority 4
gnome-client-set-process-id 5
gnome-client-set-program 5
gnome-client-set-restart-style 4
gnome-client-set-user-id 5
gnome-date-edit-construct 10
gnome-date-edit-get-flags 10
gnome-date-edit-get-initial-time 11
gnome-date-edit-get-time 10
gnome-date-edit-new 9
gnome-date-edit-new-flags 9
gnome-date-edit-set-flags 10
gnome-date-edit-set-popup-range 10
gnome-date-edit-set-time 10
gnome-gdk-pixbuf-new-from-uri 17
gnome-help-display-desktop-on-screen 18
gnome-help-display-on-screen 12
gnome-help-display-uri-on-screen 12
gnome-help-display-with-doc-id-on-screen 18
gnome-master-client 3
gnome-password-dialog-anon-selected 14
gnome-password-dialog-get-domain 14
gnome-password-dialog-get-password 14
gnome-password-dialog-get-username 14
gnome-password-dialog-new 14
gnome-password-dialog-run-and-block 15
gnome-password-dialog-set-domain 15
gnome-password-dialog-set-password 15
gnome-password-dialog-set-remember 15
gnome-password-dialog-set-username 16
gnome-url-show-on-screen 12

R

request-phase-2 6
request-save 6
run-and-block 15

S

save-any-dialog 5
save-complete on <gnome-client> 3
save-error-dialog 5
save-yourself on <gnome-client> 3
set-current-directory 4
set-domain 15
set-environment 5
set-flags 10
set-id 7
set-password 15
set-popup-range 10
set-priority 4
set-process-id 5
set-program 5
set-remember 15
set-restart-style 4
set-time 10
set-user-id 5
set-username 16
shutdown-cancelled on <gnome-client> 3

T

time-changed on <gnome-date-edit> 9