Guile-GNOME: Libgnome

version 2.16.2, updated 9 December 2011

Kjartan Maraas
Malcolm Tredinnick
many others

Short Contents

1 Overview

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

Libgnome historically was a staging ground for code meant to go elsewhere. In the modern environment, it has largely been replaced, its functionality having been pushed into GLib, GTK+, and other lower-level libraries.

However, as of 2008 it still has some utility. The help functions are useful. `(gnome gnome)` wraps those parts that have not been replaced yet.

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

2 gnome-help

Display application and GNOME system help.

2.1 Overview

These functions provide a way to display help files that are either installed as part of the main GNOME installation or that are specific to the current application.

2.2 Usage

`gnome-help-display` (*file_name* `mchars`) (*link_id* `mchars`) ⇒ (*ret* `bool`) [Function]

Displays the help file specified by *file-name* at location *link-id* in the preferred help browser of the user.

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.

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).

`gnome-help-display-with-doc-id` (*program* `<gnome-program>`) (*doc_id* `mchars`) (*file_name* `mchars`) (*link_id* `mchars`) ⇒ (*ret* `bool`) [Function]

Displays the help file specified by *file-name* at location *link-id* within the *doc-id* domain in the preferred help browser of the user. Most of the time, you want to call `gnome-help-display` instead.

This function will display the help through creating a "ghelp" URI, by looking for *file-name* in the applications installed help location (found by `<gnome-file-domain-app-help>`) and its app_id. The resulting URI is roughly in the form "ghelp:appid/file_name?link_id". If a matching file cannot be found, '`#f`' is returned and *error* is set.

Please note that this only displays application help. To display help files from the global GNOME domain, you will want to use `gnome-help-display-desktop`.

program The current application object, or '`#f`' for the default one.

doc-id The document identifier, or '`#f`' to default to the application ID (app_id) of the specified *program*.

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.

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).

`gnome-help-display-desktop` (*program* `<gnome-program>`) (*doc_id* `mchars`) (*file_name* `mchars`) (*link_id* `mchars`) ⇒ (*ret* `bool`) [Function]

Displays the GNOME system help file specified by *file-name* at location *link-id* in the preferred help browser of the user. This is done by creating a "ghelp" URI, by looking for *file-name* in the system help domain (`<gnome-file-domain-help>`) and it's app_id. This domain is determined when the library is compiled. If a matching file cannot be found, '`#f`' is returned and *error* is set.

Please note that this only displays system help. To display help files for an application, you will want to use `gnome-help-display`.

program The current application object, or '`#f`' for the default one.

doc-id The name of the help file relative to the system's help domain (`<gnome-file-domain-help>`).

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.

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).

`gnome-help-display-uri` (*help_uri* `mchars`) ⇒ (*ret* `bool`) [Function]

Displays *help-uri* in the user's preferred viewer. You should never need to call this function directly in code, since it is just a wrapper for `gnome-url-show` and consequently the viewer used to display the results depends upon the scheme of the URI (so it is not strictly a help-only function).

help-uri The URI to display.

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).

3 gnome-init

Functions used during initialisation of libgnome and other platform libraries.

3.1 Overview

3.2 Usage

`gnome-user-dir-get` ⇒ (*ret* `mchars`) [Function]

Retrieves the user-specific directory for GNOME apps to use ($HOME/.gnome2 is the usual GNOME 2 value).

ret An absolute path to the directory.

`gnome-user-private-dir-get` ⇒ (*ret* `mchars`) [Function]

Differs from `gnome-user-dir-get` in that the directory returned here will have had permissions of 0700 (rwx——) enforced when it was created. Of course, the permissions may have been altered since creation, so care still needs to be taken.

ret An absolute path to the user-specific private directory that GNOME apps can use.

`gnome-user-accels-dir-get` ⇒ (*ret* `mchars`) [Function]

Retrieves the user-specific directory that stores the keyboard shortcut files for each GNOME app. Note that most applications should be using GConf for storing this information, but it may be necessary to use the `gnome-user-accels-dir-get` directory for legacy applications.

ret The absolute path to the directory.

4 gnome-program

Initialize and retrieve information about a GNOME application.

4.1 Overview

4.2 Usage

`<gnome-program>` [Class]

Derives from `<gobject>`.

This class defines the following slots:

`app-id` ID string to use for this application

`app-version`
: Version of this application

`human-readable-name`
: Human readable name of this application

`gnome-path`
: Path in which to look for installed files

`gnome-prefix`
: Prefix where GNOME was installed

`gnome-libdir`
: Library prefix where GNOME was installed

`gnome-datadir`
: Data prefix where GNOME was installed

`gnome-sysconfdir`
: Configuration prefix where GNOME was installed

`app-prefix`
: Prefix where this application was installed

`app-libdir`
: Library prefix where this application was installed

`app-datadir`
: Data prefix where this application was installed

`app-sysconfdir`
: Configuration prefix where this application was installed

`create-directories`
: Create standard GNOME directories on startup

`enable-sound`
: Enable sound on startup

`espeaker` How to connect to esd

`popt-table`
: The table of options for popt

`popt-flags`
: The flags to use for popt

`popt-context`
: The popt context pointer that GnomeProgram is using

`goption-context`
: The goption context pointer that GnomeProgram is using

gnome-program-get ⇒ (*ret* `<gnome-program>`) [Function]

Retrieves an object that stored information about the application's state. Other functions assume this will always return a `<gnome-program>` object which (if not '`#f`') has already been initialized.

ret The application's `<gnome-program>` instance, or '`#f`' if it does not exist.

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

get-app-id [Method]

This function returns a pointer to a static string that the application has provided as an identifier. This is not meant as a human-readable identifier so much as a unique identifier for programs and libraries.

program The program object

ret Application ID string.

gnome-program-get-app-version (*self* `<gnome-program>`) ⇒ (*ret* `mchars`) [Function]

get-app-version [Method]

This function returns a pointer to a static string that the application has provided as a version number. This is not meant as a human-readable identifier so much as a unique identifier for programs and libraries.

program The application object

ret Application version string.

5 gnome-sound

Sound playing routines.

5.1 Overview

This module provides wrapper functions for playing sound samples. Currently it just wraps the esound daemon, but the API is flexible enough that other sound infrastructures can be included in the future.

These functions also allow for the fact that no sound may be supported on the current platform. So applications can safely call these functions to play sounds and they will just quietly return if no action is possible.

5.2 Usage

`gnome-sound-init` (*hostname* `mchars`) [Function]

Initialize the esd connection.

hostname Hostname where esd daemon resides.

`gnome-sound-shutdown` [Function]

Shuts down the gnome sound support.

`gnome-sound-play` (*filename* `mchars`) [Function]

Plays the audio stored in *filename*, if possible. Fail quietly if playing is not possible (due to missing sound support or for other reasons).

filename File containing the sound sample.

`gnome-sound-sample-load` (*sample_name* `mchars`) (*filename* `mchars`) [Function]
⇒ (*ret* `int`)

Loads the audio from *filename* and load it into the esd cache for later playing. Programs will rarely want to call this function directly. Use `gnome-sound-play` instead for fire and forget sound playing.

sample-name
The name of the sample.

filename The filename where the audio is stored.

ret The esound sample_id or '`-1`' if the sample was unable to be cached for esound.

6 gnome-url

Display urls using appropriate viewers.

6.1 Overview

A GNOME user can configure which viewers they wish to use to view certain protocols. Protocols can include http, ftp (where "view" might mean "download"), ghelp, etc. This module provides a means for application to display a url without having to worry about which viewer is going to ultimately handle the job.

6.2 Usage

`gnome-url-show` (*url* `mchars`) ⇒ (*ret* `bool`) [Function]

Once the input has been converted into a fully qualified url this function calls gnome_vfs_url_show. Any error codes returned by gnome-vfs will be wrapped in the error parameter. All errors comes from the '`GNOME_URL_ERROR`'% domain.

url The url or path to display. The path can be relative to the current working directory or the user's home directory. This function will convert it into a fully qualified url using the gnome_url_get_from_input function.

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).

7 Undocumented

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

7.1 (gnome gnome)

`gnome-program-init` *name version . properties* [Function]

7.2 (gnome gw libgnome)

`%gnome-program-init` [Variable]

`gnome-gconf-get-app-settings-relative` [Variable]

`gnome-gconf-get-gnome-libs-settings-relative` [Variable]

`gnome-program-get-human-readable-name` [Variable]

Type Index

Function Index