NAME

Glib::Utils − Miscellaneous utility functions

SYNOPSIS

```
use Glib;
Glib::set_application_name (Glib::get_real_name."'s Cool Program");

print "app name is ".Glib::get_application_name()."\n";
```

DESCRIPTION

Here are some useful miscellaneous utilities. GLib is a portability library, providing portable utility functions for C programs. As such, most of these functions seem to violate the Glib binding principle of not duplicating functionality that Perl already provides, but there's a distinction for each one, i swear. The functions for dealing with user information are provided on all GLib-supported platforms, not just where POSIX (which provides similar information) is available, and even work on platforms where `%ENV` may not include the expected information. Also, the "application name" referred to by (set|get)_application_name is a human readable name, distinct from the actual program name provided by Perl's own `$0`.

METHODS

string = Glib::get_application_name

Get the human-readable application name set by `set_application_name`.

Since: glib 2.2

Glib::set_application_name ($application_name)

- `$application_name` (string)

Set the human-readable application name.

Since: glib 2.2

string = Glib::get_home_dir

Find the current user's home directory, by system−dependent/appropriate means.

list = Glib::get_language_names

Computes a list of applicable locale names, which can be used to e.g. construct locale-dependent filenames or search paths. The returned list is sorted from most desirable to least desirable and always contains the default locale "C".

Since: glib 2.6

string = Glib::get_real_name

Get the current user's real name.

string = Glib::strerror ($err)

- `$err` (integer)

Return a string describing the given errno value, like "No such file or directory" for ENOENT. This is translated into the user's preferred language and is a utf8 wide-char string (unlike a $! string (perlvar) or POSIX::strerror (POSIX) which are locale codeset bytes).

string = Glib::strsignal ($signum)

- `$signum` (integer)

Return a string describing the given signal number, like "Segmentation violation" for SIGSEGV. This is translated into the user's preferred language and is a utf8 wide-char string.

list = Glib::get_system_config_dirs

Returns an ordered list of base directories in which to access system-wide configuration information.

Since: glib 2.6

list = Glib::get_system_data_dirs

Returns an ordered list of base directories in which to access system-wide application data.

Since: glib 2.6

string = Glib::get_tmp_dir

Get the temp dir as appropriate for the current system. See the GLib docs for info on how it works.

string = Glib::get_user_cache_dir

Gets the base directory in which to store non-essential, cached data specific to particular user.

Since: glib 2.6

string = Glib::get_user_config_dir

Gets the base directory in which to store user-specific application configuration information such as user preferences and settings.

Since: glib 2.6

string = Glib::get_user_data_dir

Get the base directory for application data such as icons that is customized for a particular user.

Since: glib 2.6

string = Glib::get_user_name

Get the current user's name by whatever system-dependent means necessary.

string = Glib::get_user_special_dir ($directory)

- `$directory` (Glib::UserDirectory)

Returns the full path of a special directory using its logical id.

Since: glib 2.14

ENUMS AND FLAGS

enum Glib::UserDirectory

- 'desktop' / 'G_USER_DIRECTORY_DESKTOP'
- 'documents' / 'G_USER_DIRECTORY_DOCUMENTS'
- 'download' / 'G_USER_DIRECTORY_DOWNLOAD'
- 'music' / 'G_USER_DIRECTORY_MUSIC'
- 'pictures' / 'G_USER_DIRECTORY_PICTURES'
- 'public−share' / 'G_USER_DIRECTORY_PUBLIC_SHARE'
- 'templates' / 'G_USER_DIRECTORY_TEMPLATES'
- 'videos' / 'G_USER_DIRECTORY_VIDEOS'

SEE ALSO

Glib

COPYRIGHT