NAME

File::BaseDir – Use the Freedesktop.org base directory specification

SYNOPSIS

```
use File::BaseDir qw/xdg_data_files/;
for ( xdg_data_files('mime/globs') ) {
        # do something
}
```

DESCRIPTION

This module can be used to find directories and files as specified by the Freedesktop.org Base Directory Specification. This specifications gives a mechanism to locate directories for configuration, application data and cache data. It is suggested that desktop applications for e.g. the Gnome, KDE or Xfce platforms follow this layout. However, the same layout can just as well be used for non-GUI applications.

This module forked from File::MimeInfo.

This module follows version 0.6 of BaseDir specification.

EXPORT

None by default, but all methods can be exported on demand. Also the groups ":lookup" and ":vars" are defined. The ":vars" group contains all routines with a "xdg_" prefix; the ":lookup" group contains the routines to locate files and directories.

METHODS

`new()`

Simple constructor to allow Object Oriented use of this module.

Lookup

The following methods are used to lookup files and folders in one of the search paths.

`data_home(@PATH)`

Takes a list of file path elements and returns a new path by appending them to the data home directory. The new path does not need to exist. Use this when writing user specific application data.

Example:

```
# data_home is: /home/USER/.local/share
$path = $bd->data_home('Foo', 'Bar', 'Baz');
# returns: /home/USER/.local/share/Foo/Bar/Baz
```

`data_dirs(@PATH)`

Looks for directories specified by `@PATH` in the data home and other data directories. Returns (possibly empty) list of readable directories. In scalar context only the first directory found is returned. Use this to lookup application data.

`data_files(@PATH)`

Looks for files specified by `@PATH` in the data home and other data directories. Only returns files that are readable. In scalar context only the first file found is returned. Use this to lookup application data.

`config_home(@PATH)`

Takes a list of path elements and appends them to the config home directory returning a new path. The new path does not need to exist. Use this when writing user specific configuration.

`config_dirs(@PATH)`

Looks for directories specified by `@PATH` in the config home and other config directories. Returns (possibly empty) list of readable directories. In scalar context only the first directory found is returned. Use this to lookup configuration.

`config_files(@PATH)`

Looks for files specified by `@PATH` in the config home and other config directories. Returns a (possibly empty) list of files that are readable. In scalar context only the first file found is returned. Use this to lookup configuration.

`cache_home(@PATH)`
: Takes a list of path elements and appends them to the cache home directory returning a new path. The new path does not need to exist.

Variables

The following methods only returns the value of one of the XDG variables.

`xdg_data_home`
: Returns either `$ENV{XDG_DATA_HOME}` or it's default value. Default is *$HOME/.local/share*.

`xdg_data_dirs`
: Returns either `$ENV{XDG_DATA_DIRS}` or it's default value as list. Default is */usr/local/share*, */usr/share*.

`xdg_config_home`
: Returns either `$ENV{XDG_CONFIG_HOME}` or it's default value. Default is *$HOME/.config*.

`xdg_config_dirs`
: Returns either `$ENV{XDG_CONFIG_DIRS}` or it's default value as list. Default is */etc/xdg*.

`xdg_cache_home`
: Returns either `$ENV{XDG_CACHE_HOME}` or it's default value. Default is *$HOME/.cache*.

NON-UNIX PLATFORMS

The use of File::Spec ensures that all paths are returned in the appropriate form for the current platform. On Windows this module will try to set `$HOME` to a sensible value if it is not defined yet. On other platforms one can use e.g. File::HomeDir to set `$HOME` before loading File::BaseDir.

Please note that the specification is targeting Unix platforms only and will only have limited relevance on other platforms. Any platform dependent behavior in this module should be considered an extension of the spec.

BACKWARDS COMPATIBILITY

The methods `xdg_data_files()` and `xdg_config_files()` are exported for backwards compatibility with version 0.02. They are identical to `data_files()` and `config_files()` respectively but without the `wantarray` behavior.

BUGS

Please mail the author if you encounter any bugs.

AUTHOR

Jaap Karssenberg || Pardus [Larus] <pardus@cpan.org>

Currently being maintained by Kim Ryan

SEE ALSO

<http://www.freedesktop.org/wiki/Specifications/basedir–spec>