NAME

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

VERSION

version 0.09

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.

CONSTRUCTOR

new

```
 my $bd = File::BaseDir−>new;
```

Simple constructor to allow calling functions as object oriented methods.

FUNCTIONS

None of these are exported by default, but all functions can be exported by request. Also the groups `:lookup` and `:vars` are defined. The `:vars` group contains all the routines with a `xdg_` prefix. The `:lookup` group contains the routines to locate files and directories.

data_home

```
 my $path = data_home(@path);
 my $path = $bd−>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

```
 # :lookup
 my $dir = data_dirs(@path);
 my $dir = $bd−>data_dirs(@path);
 my @dirs = data_dirs(@path);
 my @dirs = $bd−>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

```
 # :lookup
 my $file = data_files(@path);
 my $file = $bd−>data_files(@path);
 my @files = data_files(@path);
 my @files = $bd−>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

```
 # :lookup
 my $dir = config_home(@path);
 my $dir = $bd->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

```
 # :lookup
 my $dir = config_dirs(@path);
 my $dir = $bd->config_dirs(@path);
 my @dirs = config_dirs(@path);
 my @dirs = $bd->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

```
 # :lookup
 my $file = config_files(@path);
 my $file = $bd->config_files(@path);
 my @files = config_files(@path);
 my @files = $bd->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

```
 # :lookup
 my $dir = cache_home(@path);
 my $dir = $bd->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.

xdg_data_home

```
 # :var
 my $dir = xdg_data_home;
 my $dir = $bd->xdg_data_home;
```

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

xdg_data_dirs

```
 # :var
 my @dirs = xdg_data_dirs;
 my @dirs = $bd->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

```
 # :var
 my $dir = xdg_config_home;
 my $dir = $bd->xdg_config_home;
```

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

xdg_config_dirs

```
# :var
my @dirs = xdg_config_dirs;
my @dirs = $bd->xdg_config_dirs;
```

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

xdg_cache_home

```
# :var
my $dir = xdg_cache_home;
my $dir = $bd->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 their native formats regardless of platform. On Windows this module will use the native environment variables, rather than the default on UNIX (which is traditionally `$HOME`).

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.

AUTHORS

- Jaap Karssenberg || Pardus [Larus] <pardus@cpan.org>
- Graham Ollis <plicease@cpan.org>

COPYRIGHT AND LICENSE