

# Efreet Reference Manual

## 0.0.3

Generated by Doxygen 1.5.1

Wed Mar 28 00:04:28 2007



# Contents

<b>1</b>	<b>The Efreet Library</b>	<b>1</b>
1.1	Introduction . . . . .	1
<b>2</b>	<b>Efreet Module Index</b>	<b>3</b>
2.1	Efreet Modules . . . . .	3
<b>3</b>	<b>Efreet Data Structure Index</b>	<b>5</b>
3.1	Efreet Data Structures . . . . .	5
<b>4</b>	<b>Efreet File Index</b>	<b>7</b>
4.1	Efreet File List . . . . .	7
<b>5</b>	<b>Efreet Module Documentation</b>	<b>9</b>
5.1	Efreet_Base: The XDG Base Directory Specification . . . . .	9
5.2	Efreet_Desktop: The FDO Desktop Entry . . . . .	11
5.3	Efreet_Icon: The FDO Icon Theme . . . . .	17
5.4	Efreet_Menu: The FDO Desktop Menu Specification . . . . .	21
5.5	Efreet_Private: Private methods and defines . . . . .	24
<b>6</b>	<b>Efreet Data Structure Documentation</b>	<b>29</b>
6.1	_Efreet_Cache_Fill Struct Reference . . . . .	29
6.2	_Efreet_Cache_Fill_Dir Struct Reference . . . . .	30
6.3	_Efreet_Cache_Search Struct Reference . . . . .	31
6.4	_Efreet_Cache_Search_List Struct Reference . . . . .	32
6.5	Efreet_Desktop Struct Reference . . . . .	33
6.6	Efreet_Desktop_Command Struct Reference . . . . .	36
6.7	Efreet_Desktop_Command_File Struct Reference . . . . .	37
6.8	Efreet_Desktop_Type_Info Struct Reference . . . . .	38
6.9	Efreet_Icon Struct Reference . . . . .	39
6.10	Efreet_Icon_Point Struct Reference . . . . .	41

6.11	Efreet_Icon_Theme Struct Reference . . . . .	42
6.12	Efreet_Icon_Theme_Directory Struct Reference . . . . .	44
6.13	Efreet_Ini Struct Reference . . . . .	46
6.14	Efreet_Menu Struct Reference . . . . .	47
6.15	Efreet_Menu_App_Dir Struct Reference . . . . .	49
6.16	Efreet_Menu_Desktop Struct Reference . . . . .	50
6.17	Efreet_Menu_Filter Struct Reference . . . . .	51
6.18	Efreet_Menu_Filter_Op Struct Reference . . . . .	52
6.19	Efreet_Menu_Internal Struct Reference . . . . .	53
6.20	Efreet_Menu_Layout Struct Reference . . . . .	57
6.21	Efreet_Menu_Move Struct Reference . . . . .	59
6.22	Efreet_Xml Struct Reference . . . . .	60
6.23	Efreet_Xml_Attribute Struct Reference . . . . .	61
<b>7</b>	<b>Efreet File Documentation</b>	<b>63</b>
7.1	lib/efreet.c File Reference . . . . .	63
7.2	lib/Efreet.h File Reference . . . . .	66
7.3	lib/efreet_base.c File Reference . . . . .	68
7.4	lib/efreet_base.h File Reference . . . . .	70
7.5	lib/efreet_desktop.c File Reference . . . . .	71
7.6	lib/efreet_desktop.h File Reference . . . . .	74
7.7	lib/efreet_icon.c File Reference . . . . .	77
7.8	lib/efreet_icon.h File Reference . . . . .	79
7.9	lib/efreet_ini.c File Reference . . . . .	81
7.10	lib/efreet_ini.h File Reference . . . . .	83
7.11	lib/efreet_menu.c File Reference . . . . .	85
7.12	lib/efreet_menu.h File Reference . . . . .	89
7.13	lib/efreet_private.h File Reference . . . . .	91
7.14	lib/efreet_utils.c File Reference . . . . .	95
7.15	lib/efreet_utils.h File Reference . . . . .	98
7.16	lib/efreet_xml.c File Reference . . . . .	100
7.17	lib/efreet_xml.h File Reference . . . . .	102

# Chapter 1

## The Efreet Library

### 1.1 Introduction

Efreet is a library designed to help apps work several of the Freedesktop.org standards regarding Icons, Desktop files and Menus. To that end it implements the following specifications:

- XDG Base Directory Specification
- Icon Theme Specification
- Desktop Entry Specification
- Desktop Menu Specification



## Chapter 2

# Efreet Module Index

### 2.1 Efreet Modules

Here is a list of all modules:

Efreet_Base: The XDG Base Directory Specification . . . . .	9
Efreet_Desktop: The FDO Desktop Entry . . . . .	11
Efreet_Icon: The FDO Icon Theme . . . . .	17
Efreet_Menu: The FDO Desktop Menu Specification . . . . .	21
Efreet_Private: Private methods and defines . . . . .	24





## Chapter 3

# Efreet Data Structure Index

### 3.1 Efreet Data Structures

Here are the data structures with brief descriptions:

<b>_Efreet_Cache_Fill</b> . . . . .	29
<b>_Efreet_Cache_Fill_Dir</b> . . . . .	30
<b>_Efreet_Cache_Search</b> . . . . .	31
<b>_Efreet_Cache_Search_List</b> . . . . .	32
<b>Efreet_Desktop</b> (Parsed representation of a .desktop file ) . . . . .	33
<b>Efreet_Desktop_Command</b> . . . . .	36
<b>Efreet_Desktop_Command_File</b> . . . . .	37
<b>Efreet_Desktop_Type_Info</b> . . . . .	38
<b>Efreet_Icon</b> (Contains all the information about a given icon ) . . . . .	39
<b>Efreet_Icon_Point</b> (Stores an x, y point ) . . . . .	41
<b>Efreet_Icon_Theme</b> (All of the known information about a given theme ) . . . . .	42
<b>Efreet_Icon_Theme_Directory</b> (Contains all the information about a sub-directory of a theme ) . . . . .	44
<b>Efreet_Ini</b> (Contains all the information about an ini file ) . . . . .	46
<b>Efreet_Menu</b> . . . . .	47
<b>Efreet_Menu_App_Dir</b> . . . . .	49
<b>Efreet_Menu_Desktop</b> . . . . .	50
<b>Efreet_Menu_Filter</b> . . . . .	51
<b>Efreet_Menu_Filter_Op</b> . . . . .	52
<b>Efreet_Menu_Internal</b> . . . . .	53
<b>Efreet_Menu_Layout</b> . . . . .	57
<b>Efreet_Menu_Move</b> . . . . .	59
<b>Efreet_Xml</b> (Contains the XML tree for a given XML document ) . . . . .	60
<b>Efreet_Xml_Attribute</b> (Contains information about a given XML attribute ) . . . . .	61



## Chapter 4

# Efreet File Index

### 4.1 Efreet File List

Here is a list of all files with brief descriptions:

lib/ <b>efreet.c</b> . . . . .	63
lib/ <b>Efreet.h</b> (The file that must be included by any project wishing to use Efreet. <b>Efreet.h</b> (p. 66) provides all of the necessary headers and includes to work with Efreet ) . . . . .	66
lib/ <b>efreet_base.c</b> . . . . .	68
lib/ <b>efreet_base.h</b> (Contains the methods used to support the FDO base directory spec- ification ) . . . . .	70
lib/ <b>efreet_desktop.c</b> . . . . .	71
lib/ <b>efreet_desktop.h</b> (Contains the structures and methods used to support the FDO desktop entry specification ) . . . . .	74
lib/ <b>efreet_icon.c</b> . . . . .	77
lib/ <b>efreet_icon.h</b> (Contains the structures and methods used to support the FDO icon theme specification ) . . . . .	79
lib/ <b>efreet_ini.c</b> . . . . .	81
lib/ <b>efreet_ini.h</b> . . . . .	83
lib/ <b>efreet_menu.c</b> . . . . .	85
lib/ <b>efreet_menu.h</b> (Contains the structures and methods to support the Desktop Menu Specification ) . . . . .	89
lib/ <b>efreet_private.h</b> (Contains methods and defines that are private to the Efreet im- plementaion ) . . . . .	91
lib/ <b>efreet_utils.c</b> . . . . .	95
lib/ <b>efreet_utils.h</b> . . . . .	98
lib/ <b>efreet_xml.c</b> . . . . .	100
lib/ <b>efreet_xml.h</b> . . . . .	102



# Chapter 5

## Efreet Module Documentation

### 5.1 Efreet\_Base: The XDG Base Directory Specification

#### 5.1.1 Detailed Description

functions

#### Functions

- `const char * efreet_cache_home_get (void)`  
*Retrieves the XDG Cache Home directory.*
- `Ecore_List * efreet_config_dirs_get (void)`  
*Returns the Ecore\_List of preference ordered extra config directories.*
- `const char * efreet_config_home_get (void)`  
*Retrieves the XDG Config Home directory.*
- `Ecore_List * efreet_data_dirs_get (void)`  
*Returns the Ecore\_List of preference ordered extra data directories.*
- `const char * efreet_data_home_get (void)`  
*Retrieves the XDG Data Home directory.*

#### 5.1.2 Function Documentation

##### 5.1.2.1 `const char* efreet_cache_home_get (void)`

Retrieves the XDG Cache Home directory.

#### Returns:

Returns the XDG Cache Home directory

**5.1.2.2 Ecore\_List\* efreet\_config\_dirs\_get (void)**

Returns the Ecore\_List of preference ordered extra config directories.

**Returns:**

Returns the Ecore\_List of preference ordered extra config directories

**Note:**

The returned list is static inside Efreet. If you add/remove from the list then the next call to **efreet\_config\_dirs\_get()** (p.10) will return your modified values. DO NOT free this list.

**5.1.2.3 const char\* efreet\_config\_home\_get (void)**

Retrieves the XDG Config Home directory.

**Returns:**

Returns the XDG Config Home directory

**5.1.2.4 Ecore\_List\* efreet\_data\_dirs\_get (void)**

Returns the Ecore\_List of preference ordered extra data directories.

**Returns:**

Returns the Ecore\_List of preference ordered extra data directories

**Note:**

The returned list is static inside Efreet. If you add/remove from the list then the next call to **efreet\_data\_dirs\_get()** (p.10) will return your modified values. DO NOT free this list.

**5.1.2.5 const char\* efreet\_data\_home\_get (void)**

Retrieves the XDG Data Home directory.

**Returns:**

Returns the XDG Data Home directory

## 5.2 Efreet \_Desktop: The FDO Desktop Entry

### 5.2.1 Detailed Description

Specification functions and structures

#### Data Structures

- struct **Efreet \_Desktop**  
*a parsed representation of a .desktop file*

#### Typedefs

- typedef **Efreet \_Desktop** **Efreet \_Desktop**
- typedef void(\*) **Efreet \_Desktop\_Command\_Cb** (void \*data, **Efreet \_Desktop** \*desktop, char \*command, int remaining)
- typedef int(\*) **Efreet \_Desktop\_Progress\_Cb** (void \*data, **Efreet \_Desktop** \*desktop, char \*uri, long int total, long int current)
- typedef enum **Efreet \_Desktop\_Type** **Efreet \_Desktop\_Type**
- typedef void \*(\*) **Efreet \_Desktop\_Type\_Free\_Cb** (void \*data)
- typedef void \*(\*) **Efreet \_Desktop\_Type\_Parse\_Cb** (**Efreet \_Desktop** \*desktop, **Efreet \_Ini** \*ini)
- typedef void(\*) **Efreet \_Desktop\_Type\_Save\_Cb** (**Efreet \_Desktop** \*desktop, **Efreet \_Ini** \*ini)

#### Functions

- void **efreet \_desktop\_category\_add** (**Efreet \_Desktop** \*desktop, const char \*category)  
*add a category to a desktop*
- unsigned int **efreet \_desktop\_category\_count\_get** (**Efreet \_Desktop** \*desktop)  
*Retrieves the number of categories the given desktop belongs too.*
- int **efreet \_desktop\_category\_del** (**Efreet \_Desktop** \*desktop, const char \*category)  
*removes a category from a desktop*
- int **efreet \_desktop\_command\_get** (**Efreet \_Desktop** \*desktop, Ecore\_List \*files, **Efreet \_Desktop\_Command\_Cb** func, void \*data)  
*Get a command to use to execute a desktop entry.*
- int **efreet \_desktop\_command\_progress\_get** (**Efreet \_Desktop** \*desktop, Ecore\_List \*files, **Efreet \_Desktop\_Command\_Cb** cb\_command, **Efreet \_Desktop\_Progress\_Cb** cb\_prog, void \*data)  
*Get a command to use to execute a desktop entry, and receive progress updates for downloading of remote URI's passed in.*

- **Efreet\_Desktop \* efreet\_desktop\_empty\_new** (const char \*file)  
*Creates a new empty **Efreet\_Desktop** (p. 33) structure or NULL on failure.*
- void **efreet\_desktop\_environment\_set** (const char \*environment)
- void **efreet\_desktop\_exec** (**Efreet\_Desktop** \*desktop, **Ecore\_List** \*files, void \*data)  
*Parses the desktop exec line and returns an **Ecore\_Exe**.*
- void **efreet\_desktop\_free** (**Efreet\_Desktop** \*desktop)
- **Efreet\_Desktop \* efreet\_desktop\_get** (const char \*file)  
*Gets a reference to an **Efreet\_Desktop** (p. 33) structure representing the contents of file or NULL if file is not a valid .desktop file.*
- int **efreet\_desktop\_save** (**Efreet\_Desktop** \*desktop)  
*Saves any changes made to desktop back to the file on the filesystem.*
- int **efreet\_desktop\_save\_as** (**Efreet\_Desktop** \*desktop, const char \*file)  
*Saves desktop to file.*
- int **efreet\_desktop\_type\_add** (const char \*type, **Efreet\_Desktop\_Type\_Parse\_Cb** parse\_func, **Efreet\_Desktop\_Type\_Save\_Cb** save\_func, **Efreet\_Desktop\_Type\_Free\_Cb** free\_func)  
*Adds the given type to the list of types in the system.*
- void \* **efreet\_desktop\_type\_data\_get** (**Efreet\_Desktop** \*desktop)  
*get type specific data for custom desktop types*

## Variables

- int **EFREET\_DESKTOP\_TYPE\_APPLICATION**
- int **EFREET\_DESKTOP\_TYPE\_DIRECTORY**
- int **EFREET\_DESKTOP\_TYPE\_LINK**

## 5.2.2 Typedef Documentation

### 5.2.2.1 typedef struct Efreet\_Desktop Efreet\_Desktop

**Efreet\_Desktop** (p. 33)

### 5.2.2.2 typedef void(\*) Efreet\_Desktop\_Command\_Cb(void \*data, Efreet\_Desktop \*desktop, char \*command, int remaining)

A callback used with **efreet\_desktop\_command\_get()** (p. 14)

### 5.2.2.3 typedef int(\*) Efreet\_Desktop\_Progress\_Cb(void \*data, Efreet\_Desktop \*desktop, char \*uri, long int total, long int current)

A callback used to get download progress of remote uris



#### 5.2.2.4 typedef enum Efreet \_\_Desktop \_\_Type Efreet \_\_Desktop \_\_Type

Efreet \_\_Desktop \_\_Type

#### 5.2.2.5 typedef void\*(\*) Efreet \_\_Desktop \_\_Type \_\_Free \_\_Cb(void \*data)

A callback used to free data for custom types

#### 5.2.2.6 typedef void\*(\*) Efreet \_\_Desktop \_\_Type \_\_Parse \_\_Cb(Efreet \_\_Desktop \*desktop, Efreet \_\_Ini \*ini)

A callback used to parse data for custom types

#### 5.2.2.7 typedef void(\*) Efreet \_\_Desktop \_\_Type \_\_Save \_\_Cb(Efreet \_\_Desktop \*desktop, Efreet \_\_Ini \*ini)

A callback used to save data for custom types

### 5.2.3 Function Documentation

#### 5.2.3.1 void efreet \_\_desktop \_\_category \_\_add (Efreet \_\_Desktop \* *desktop*, const char \* *category*)

add a category to a desktop

**Parameters:**

*desktop*,: the desktop

*category*,: the category name

#### 5.2.3.2 unsigned int efreet \_\_desktop \_\_category \_\_count \_\_get (Efreet \_\_Desktop \* *desktop*)

Retrieves the number of categories the given *desktop* belongs too.

**Parameters:**

*desktop*,: The desktop to work with

**Returns:**

Returns the number of categories assigned to this desktop

#### 5.2.3.3 int efreet \_\_desktop \_\_category \_\_del (Efreet \_\_Desktop \* *desktop*, const char \* *category*)

removes a category from a desktop

**Parameters:**

*desktop*,: the desktop  
*category*,: the category name

**Returns:**

1 if the desktop had his category listed, 0 otherwise

**5.2.3.4** `int efreet_desktop_command_get (Efreet_Desktop * desktop, Ecore_List * files, Efreet_Desktop_Command_Cb func, void * data)`

Get a command to use to execute a desktop entry.

**Parameters:**

*desktop*,: the desktop entry  
*files*,: an ecore list of file names to execute, as either absolute paths, relative paths, or uris  
*func*,: a callback to call for each prepared command line  
*data*,: user data passed to the callback

**Returns:**

Returns 1 on success or 0 on failure

**5.2.3.5** `int efreet_desktop_command_progress_get (Efreet_Desktop * desktop, Ecore_List * files, Efreet_Desktop_Command_Cb cb_command, Efreet_Desktop_Progress_Cb cb_progress, void * data)`

Get a command to use to execute a desktop entry, and receive progress updates for downloading of remote URI's passed in.

**Parameters:**

*desktop*,: the desktop entry  
*files*,: an ecore list of file names to execute, as either absolute paths, relative paths, or uris  
*cb\_command*,: a callback to call for each prepared command line  
*cb\_progress*,: a callback to get progress for the downloads  
*data*,: user data passed to the callback

**Returns:**

Returns 1 on success or 0 on failure

**5.2.3.6** `Efreet_Desktop* efreet_desktop_empty_new (const char * file)`

Creates a new empty **Efreet\_Desktop** (p. 33) structure or NULL on failure.

**Parameters:**

*file*,: The file to create the **Efreet\_Desktop** (p. 33) from

**Returns:**

Returns a new empty **Efreet\_Desktop** on success, NULL on failure

**5.2.3.7** void efreet \_desktop \_environment \_set (const char \* *environment*)

**5.2.3.8** void efreet \_desktop \_exec (Efreet \_Desktop \* *desktop*, Ecore \_List \* *files*, void \* *data*)

Parses the *desktop* exec line and returns an Ecore \_Exe.

**Parameters:**

*desktop*,: The desktop file to work with

*files*,: The files to be substituted into the exec line data: The data pointer to pass

**Returns:**

Returns the Ecore \_Exe for *desktop*

**5.2.3.9** void efreet \_desktop \_free (Efreet \_Desktop \* *desktop*)

**5.2.3.10** Efreet \_Desktop\* efreet \_desktop \_get (const char \* *file*)

Gets a reference to an **Efreet \_Desktop** (p. 33) structure representing the contents of *file* or NULL if *file* is not a valid .desktop file.

**Parameters:**

*file*,: The file to get the **Efreet \_Desktop** (p. 33) from

**Returns:**

Returns a reference to a cached **Efreet \_Desktop** (p. 33) on success, NULL on failure. This reference should not be freed.

**5.2.3.11** int efreet \_desktop \_save (Efreet \_Desktop \* *desktop*)

Saves any changes made to *desktop* back to the file on the filesystem.

**Parameters:**

*desktop*,: The desktop file to save

**Returns:**

Returns 1 on success or 0 on failure

**5.2.3.12** int efreet \_desktop \_save \_as (Efreet \_Desktop \* *desktop*, const char \* *file*)

Saves *desktop* to *file*.

**Parameters:**

*desktop*,: The desktop file to save

*file*,: The filename to save as

**Returns:**

Returns 1 on success or 0 on failure

**5.2.3.13** `int efreet_desktop_type_add (const char * type, Efreet_Desktop_Type_Parse_Cb parse_func, Efreet_Desktop_Type_Save_Cb save_func, Efreet_Desktop_Type_Free_Cb free_func)`

Adds the given type to the list of types in the system.

**Parameters:**

*type*,: The type to add to the list of matching types  
*parse\_func*,: a function to parse out custom fields  
*save\_func*,: a function to save data returned from *parse\_func*  
*free\_func*,: a function to free data returned from *parse\_func*

**Returns:**

Returns the id of the new type

**5.2.3.14** `void* efreet_desktop_type_data_get (Efreet_Desktop * desktop)`

get type specific data for custom desktop types

**Parameters:**

*desktop* the desktop

**Returns:**

type specific data, or NULL if there is none

## 5.2.4 Variable Documentation

**5.2.4.1** `int EFREET_DESKTOP_TYPE_APPLICATION`

**5.2.4.2** `int EFREET_DESKTOP_TYPE_DIRECTORY`

**5.2.4.3** `int EFREET_DESKTOP_TYPE_LINK`

## 5.3 Efreet\_Icon: The FDO Icon Theme

### 5.3.1 Detailed Description

Specification functions and structures

#### Data Structures

- struct **Efreet\_Icon**  
*Contains all the information about a given icon.*
- struct **Efreet\_Icon\_Point**  
*Stores an x, y point.*
- struct **Efreet\_Icon\_Theme**  
*contains all of the known information about a given theme*
- struct **Efreet\_Icon\_Theme\_Directory**  
*Contains all the information about a sub-directory of a theme.*

#### Typedefs

- typedef **Efreet\_Icon** Efreet\_Icon
- typedef **Efreet\_Icon\_Point** Efreet\_Icon\_Point
- typedef enum **Efreet\_Icon\_Size\_Type** Efreet\_Icon\_Size\_Type
- typedef **Efreet\_Icon\_Theme** Efreet\_Icon\_Theme
- typedef enum **Efreet\_Icon\_Theme\_Context** Efreet\_Icon\_Theme\_Context
- typedef **Efreet\_Icon\_Theme\_Directory** Efreet\_Icon\_Theme\_Directory

#### Enumerations

- enum **Efreet\_Icon\_Size\_Type** { EFREET\_ICON\_SIZE\_TYPE\_NONE, EFREET\_ICON\_SIZE\_TYPE\_FIXED, EFREET\_ICON\_SIZE\_TYPE\_SCALABLE, EFREET\_ICON\_SIZE\_TYPE\_THRESHOLD }
- enum **Efreet\_Icon\_Theme\_Context** { EFREET\_ICON\_THEME\_CONTEXT\_NONE, EFREET\_ICON\_THEME\_CONTEXT\_ACTIONS, EFREET\_ICON\_THEME\_CONTEXT\_DEVICES, EFREET\_ICON\_THEME\_CONTEXT\_FILESYSTEMS, EFREET\_ICON\_THEME\_CONTEXT\_MIMETYPES }

#### Functions

- const char \* **efreet\_icon\_dir\_get** (void)  
*Returns the user icon directory.*
- void **efreet\_icon\_extension\_add** (const char \*ext)  
*Adds the given extension to the list of possible icon extensions.*

- **Efreet\_Icon \* efreet\_icon\_find** (const char \*theme\_name, const char \*icon, const char \*size)  
*Retrieves all of the information about the given icon.*
- const char \* **efreet\_icon\_path\_find** (const char \*theme, const char \*icon, const char \*size)  
*Retrives the path to the given icon.*
- **Efreet\_Icon\_Theme \* efreet\_icon\_theme\_find** (const char \*theme\_name)  
*Tries to get the icon theme structure for the given theme name.*
- **Ecore\_List \* efreet\_icon\_theme\_list\_get** (void)  
*Retrieves all of the non-hidden icon themes available on the system. The returned list must be freed. Do not free the list data.*

## 5.3.2 Typedef Documentation

### 5.3.2.1 typedef struct Efreet\_Icon Efreet\_Icon

**Efreet\_Icon** (p. 39)

### 5.3.2.2 typedef struct Efreet\_Icon\_Point Efreet\_Icon\_Point

**Efreet\_Icon\_Point**

### 5.3.2.3 typedef enum Efreet\_Icon\_Size\_Type Efreet\_Icon\_Size\_Type

**Efreet\_Icon\_Size\_Type**

### 5.3.2.4 typedef struct Efreet\_Icon\_Theme Efreet\_Icon\_Theme

**Efreet\_Icon\_Theme** (p. 42)

### 5.3.2.5 typedef enum Efreet\_Icon\_Theme\_Context Efreet\_Icon\_Theme\_Context

**Efreet\_Icon\_Theme\_Context**

### 5.3.2.6 typedef struct Efreet\_Icon\_Theme\_Directory Efreet\_Icon\_Theme\_Directory

**Efreet\_Icon\_Theme\_Directory** (p. 44)

### 5.3.3 Enumeration Type Documentation

#### 5.3.3.1 enum Efreet\_Icon\_Size\_Type

The possible size types for an icon directory

**Enumerator:**

*EFREET\_ICON\_SIZE\_TYPE\_NONE*  
*EFREET\_ICON\_SIZE\_TYPE\_FIXED*  
*EFREET\_ICON\_SIZE\_TYPE\_SCALABLE*  
*EFREET\_ICON\_SIZE\_TYPE\_THRESHOLD*

#### 5.3.3.2 enum Efreet\_Icon\_Theme\_Context

The possible contexts for an icon directory

**Enumerator:**

*EFREET\_ICON\_THEME\_CONTEXT\_NONE*  
*EFREET\_ICON\_THEME\_CONTEXT\_ACTIONS*  
*EFREET\_ICON\_THEME\_CONTEXT\_DEVICES*  
*EFREET\_ICON\_THEME\_CONTEXT\_FILESYSTEMS*  
*EFREET\_ICON\_THEME\_CONTEXT\_MIMETYPES*

### 5.3.4 Function Documentation

#### 5.3.4.1 const char\* efreet\_icon\_dir\_get (void)

Returns the user icon directory.

**Returns:**

Returns the user icon directory

#### 5.3.4.2 void efreet\_icon\_extension\_add (const char \* ext)

Adds the given extension to the list of possible icon extensions.

**Parameters:**

*ext*,: The extension to add to the list of checked extensions

**Returns:**

Returns no value.

#### 5.3.4.3 **Efreet\_Icon\*** **efreet\_icon\_find** (**const char \*** *theme\_name*, **const char \*** *icon*, **const char \*** *size*)

Retrieves all of the information about the given icon.

##### Parameters:

*theme\_name*,: The icon theme to look for

*icon*,: The icon to look for

*size*; The icon size to look for

##### Returns:

Returns the **Efreet\_Icon** (p. 39) structure representing this icon or NULL if the icon is not found

#### 5.3.4.4 **const char\*** **efreet\_icon\_path\_find** (**const char \*** *theme*, **const char \*** *icon*, **const char \*** *size*)

Retrives the path to the given icon.

##### Parameters:

*theme*,: The icon theme to look for

*icon*,: The icon to look for

*size*,: The icon size to look for

##### Returns:

Returns the path to the given icon or NULL if none found

#### 5.3.4.5 **Efreet\_Icon\_Theme\*** **efreet\_icon\_theme\_find** (**const char \*** *theme\_name*)

Tries to get the icon theme structure for the given theme name.

##### Parameters:

*theme\_name*,: The theme to look for

##### Returns:

Returns the icon theme related to the given theme name or NULL if none exists.

#### 5.3.4.6 **Ecore\_List\*** **efreet\_icon\_theme\_list\_get** (**void**)

Retrieves all of the non-hidden icon themes available on the system. The returned list must be freed. Do not free the list data.

##### Returns:

Returns a list of **Efreet\_Icon** (p. 39) structs for all the non-hidden icon themes



## 5.4 Efreet\_Menu: The FDO Desktop Menu Specification

### 5.4.1 Detailed Description

functions and structures

#### Data Structures

- struct **Efreet\_Menu**

#### Typedefs

- typedef **Efreet\_Menu** **Efreet\_Menu**
- typedef enum **Efreet\_Menu\_Entry\_Type** **Efreet\_Menu\_Entry\_Type**

#### Enumerations

- enum **Efreet\_Menu\_Entry\_Type** { **EFREET\_MENU\_ENTRY\_MENU**, **EFREET\_MENU\_ENTRY\_DESKTOP**, **EFREET\_MENU\_ENTRY\_SEPARATOR**, **EFREET\_MENU\_ENTRY\_HEADER** }

#### Functions

- int **efreet\_menu\_desktop\_insert** (**Efreet\_Menu** \*menu, **Efreet\_Desktop** \*desktop, int pos)  
*Insert a desktop element in a menu structure. Only accepts desktop files in default directories.*
- void **efreet\_menu\_dump** (**Efreet\_Menu** \*menu, const char \*indent)  
*Dumps the contents of the menu to the command line.*
- void **efreet\_menu\_free** (**Efreet\_Menu** \*menu)
- **Efreet\_Menu** \* **efreet\_menu\_get** (void)  
*Creates the default menu representation.*
- int **efreet\_menu\_kde\_legacy\_init** (void)  
*Initialize legacy kde support. This function blocks while the kde-config script is run.*
- **Efreet\_Menu** \* **efreet\_menu\_parse** (const char \*path)  
*Parses the given .menu file and creates the menu representation.*
- int **efreet\_menu\_save** (**Efreet\_Menu** \*menu, const char \*path)  
*Saves the menu to file.*

### 5.4.2 Typedef Documentation

#### 5.4.2.1 typedef struct **Efreet\_Menu** **Efreet\_Menu**

**Efreet\_Menu** (p. 47)

#### 5.4.2.2 typedef enum Efreet\_Menu\_Entry\_Type Efreet\_Menu\_Entry\_Type

Efreet\_Menu\_Entry\_Type

### 5.4.3 Enumeration Type Documentation

#### 5.4.3.1 enum Efreet\_Menu\_Entry\_Type

The type of entry

**Enumerator:**

***EFREET\_MENU\_ENTRY\_MENU***  
***EFREET\_MENU\_ENTRY\_DESKTOP***  
***EFREET\_MENU\_ENTRY\_SEPARATOR***  
***EFREET\_MENU\_ENTRY\_HEADER***

### 5.4.4 Function Documentation

#### 5.4.4.1 int efreet\_menu\_desktop\_insert (Efreet\_Menu \* *menu*, Efreet\_Desktop \* *desktop*, int *pos*)

Insert a desktop element in a menu structure. Only accepts desktop files in default directories.

**Parameters:**

***menu*,:** The menu to work with  
***desktop*,:** The desktop to insert

**Returns:**

Returns 1 on success, 0 on failure

#### 5.4.4.2 void efreet\_menu\_dump (Efreet\_Menu \* *menu*, const char \* *indent*)

Dumps the contents of the menu to the command line.

**Parameters:**

***menu*,:** The menu to work with  
***indent*,:** The indent level to print the menu at

**Returns:**

Returns no value

**5.4.4.3 void efreet \_menu \_free (Efreet \_Menu \* *menu*)****5.4.4.4 Efreet \_Menu\* efreet \_menu \_get (void)**

Creates the default menu representation.

**Returns:**

Returns the **Efreet \_Menu \_Internal** (p. 53) representation of the default menu or NULL if none found

**5.4.4.5 int efreet \_menu \_kde \_legacy \_init (void)**

Initialize legacy kde support. This function blocks while the kde-config script is run.

**Returns:**

Returns no value

**5.4.4.6 Efreet \_Menu\* efreet \_menu \_parse (const char \* *path*)**

Parses the given .menu file and creates the menu representation.

**Parameters:**

*path*,: The path of the menu to load

**Returns:**

Returns the **Efreet \_Menu \_Internal** (p. 53) representation on success or NULL on failure

**5.4.4.7 int efreet \_menu \_save (Efreet \_Menu \* *menu*, const char \* *path*)**

Saves the menu to file.

**Parameters:**

*menu*,: The menu to work with

*path*,: The path where the menu should be saved

**Returns:**

Returns 1 on success, 0 on failure

## 5.5 Efreet\_Private: Private methods and defines

### Data Structures

- struct **Efreet\_Desktop\_Command**
- struct **Efreet\_Desktop\_Command\_File**

### Defines

- #define **\_\_UNUSED\_\_**
- #define **FREE(x)** { free(x); x = NULL; }
- #define **IF\_FREE(x)** { if (x) FREE(x) }
- #define **IF\_FREE\_DLIST(x)** { if (x) ecore\_dlist\_destroy(x); x = NULL; }
- #define **IF\_FREE\_HASH(x)** { if (x) ecore\_hash\_destroy(x); x = NULL; }
- #define **IF\_FREE\_LIST(x)** { if (x) ecore\_list\_destroy(x); x = NULL; }
- #define **IF\_RELEASE(x)** { if (x) ecore\_string\_release(x); x = NULL; }
- #define **NEW(x, c)** calloc(c, sizeof(x))
- #define **PATH\_MAX** 4096

### Typedefs

- typedef **Efreet\_Desktop\_Command** **Efreet\_Desktop\_Command**
- typedef **Efreet\_Desktop\_Command\_File** **Efreet\_Desktop\_Command\_File**
- typedef enum **Efreet\_Desktop\_Command\_Flag** **Efreet\_Desktop\_Command\_Flag**

### Enumerations

- enum **Efreet\_Desktop\_Command\_Flag** { **EFREET\_DESKTOP\_EXEC\_FLAG\_FULLPATH** = 0x0001, **EFREET\_DESKTOP\_EXEC\_FLAG\_URI** = 0x0002, **EFREET\_DESKTOP\_EXEC\_FLAG\_DIR** = 0x0004, **EFREET\_DESKTOP\_EXEC\_FLAG\_FILE** = 0x0008 }

### Functions

- size\_t **efreet\_array\_cat** (char \*buffer, size\_t size, const char \*strs[])
- int **efreet\_base\_init** (void)
- void **efreet\_base\_shutdown** (void)
- Ecore\_List \* **efreet\_default\_dirs\_get** (const char \*user\_dir, Ecore\_List \*system\_dirs, const char \*suffix)

*Creates the list of directories based on the user dir, system dirs and given suffix.*

- int **efreet\_desktop\_init** (void)
- int **efreet\_desktop\_shutdown** (void)
- const char \* **efreet\_home\_dir\_get** (void)
- int **efreet\_icon\_init** (void)
- void **efreet\_icon\_shutdown** (void)
- int **efreet\_ini\_init** (void)
- int **efreet\_ini\_shutdown** (void)

- `const char * efreet_lang_country_get` (void)
- `const char * efreet_lang_get` (void)
- `const char * efreet_lang_modifier_get` (void)
- `int efreet_menu_init` (void)

*Initializes the Efreet Menu system.*

- `void efreet_menu_shutdown` (void)

*Shuts down the Efreet menu system.*

- `int efreet_util_init` (void)
- `void efreet_util_shutdown` (void)

## 5.5.1 Define Documentation

### 5.5.1.1 `#define __UNUSED__`

A flag to mark a function parameter as unused

### 5.5.1.2 `#define FREE(x) { free(x); x = NULL; }`

Free x and set to NULL

### 5.5.1.3 `#define IF_FREE(x) { if (x) FREE(x) }`

If x is set, free x and set to NULL

### 5.5.1.4 `#define IF_FREE_DLIST(x) { if (x) ecore_dlist_destroy(x); x = NULL; }`

If x is a valid pointer destroy x and set to NULL

### 5.5.1.5 `#define IF_FREE_HASH(x) { if (x) ecore_hash_destroy(x); x = NULL; }`

If x is a valid pointer destroy x and set to NULL

### 5.5.1.6 `#define IF_FREE_LIST(x) { if (x) ecore_list_destroy(x); x = NULL; }`

If x is a valid pointer destroy x and set to NULL

### 5.5.1.7 `#define IF_RELEASE(x) { if (x) ecore_string_release(x); x = NULL; }`

If x is set, `ecore_string_release` x and set to NULL

### 5.5.1.8 `#define NEW(x, c) calloc(c, sizeof(x))`

Allocate and zero out c structures of type x

### 5.5.1.9 `#define PATH_MAX 4096`

Convenience define to set the maximim path length

## 5.5.2 Typedef Documentation

### 5.5.2.1 `typedef struct Efreet_Desktop_Command Efreet_Desktop_Command`

### 5.5.2.2 `typedef struct Efreet_Desktop_Command_File Efreet_Desktop_Command_File`

### 5.5.2.3 `typedef enum Efreet_Desktop_Command_Flag Efreet_Desktop_Command_Flag`

## 5.5.3 Enumeration Type Documentation

### 5.5.3.1 `enum Efreet_Desktop_Command_Flag`

Enumerator:

```
EFREET_DESKTOP_EXEC_FLAG_FULLPATH
EFREET_DESKTOP_EXEC_FLAG_URI
EFREET_DESKTOP_EXEC_FLAG_DIR
EFREET_DESKTOP_EXEC_FLAG_FILE
```

## 5.5.4 Function Documentation

### 5.5.4.1 `size_t efreet_array_cat (char * buffer, size_t size, const char * strs[])`

### 5.5.4.2 `int efreet_base_init (void)`

### 5.5.4.3 `void efreet_base_shutdown (void)`

### 5.5.4.4 `Ecore_List* efreet_default_dirs_get (const char * user_dir, Ecore_List * system_dirs, const char * suffix)`

Creates the list of directories based on the user dir, system dirs and given suffix.

Parameters:

*user\_dir*,: The user directory to work with  
*system\_dirs*,: The system directories to work with  
*suffix*,: The path suffix to add

Returns:

Returns the list of directories

**5.5.4.5** `int efreet__desktop__init (void)`

**5.5.4.6** `int efreet__desktop__shutdown (void)`

**5.5.4.7** `const char* efreet__home__dir__get (void)`

**5.5.4.8** `int efreet__icon__init (void)`

**5.5.4.9** `void efreet__icon__shutdown (void)`

**5.5.4.10** `int efreet__ini__init (void)`

**5.5.4.11** `int efreet__ini__shutdown (void)`

**5.5.4.12** `const char* efreet__lang__country__get (void)`

**5.5.4.13** `const char* efreet__lang__get (void)`

**5.5.4.14** `const char* efreet__lang__modifier__get (void)`

**5.5.4.15** `int efreet__menu__init (void)`

Initializes the Efreet Menu system.

**Returns:**

Returns 1 on success, 0 on failure

**5.5.4.16** `void efreet__menu__shutdown (void)`

Shuts down the Efreet menu system.

**Returns:**

Returns no value

**5.5.4.17** `int efreet__util__init (void)`

**5.5.4.18** `void efreet__util__shutdown (void)`



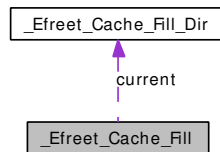


## Chapter 6

# Efreet Data Structure Documentation

### 6.1 `_Efreet_Cache_Fill` Struct Reference

Collaboration diagram for `_Efreet_Cache_Fill`:



#### Data Fields

- `Efreet_Cache_Fill_Dir * current`
- `Ecore_List * dirs`
- `DIR * files`

#### 6.1.1 Field Documentation

**6.1.1.1** `Efreet_Cache_Fill_Dir* _Efreet_Cache_Fill::current`

**6.1.1.2** `Ecore_List* _Efreet_Cache_Fill::dirs`

**6.1.1.3** `DIR* _Efreet_Cache_Fill::files`

The documentation for this struct was generated from the following file:

- `lib/efreet_utils.c`

## 6.2 `_Efreet_Cache_Fill_Dir` Struct Reference

### Data Fields

- `char * file_id`
- `char * path`

### 6.2.1 Field Documentation

**6.2.1.1** `char* _Efreet_Cache_Fill_Dir::file_id`

**6.2.1.2** `char* _Efreet_Cache_Fill_Dir::path`

The documentation for this struct was generated from the following file:

- `lib/efreet_utils.c`

## 6.3 \_\_Efreet\_\_Cache\_\_Search Struct Reference

### Data Fields

- const char \* **what1**
- const char \* **what2**

### 6.3.1 Field Documentation

**6.3.1.1** const char\* \_\_Efreet\_\_Cache\_\_Search::what1

**6.3.1.2** const char\* \_\_Efreet\_\_Cache\_\_Search::what2

The documentation for this struct was generated from the following file:

- lib/efreet\_utils.c

## 6.4 `_Efreet_Cache_Search_List` Struct Reference

### Data Fields

- `Ecore_List * list`
- `const char * what`

### 6.4.1 Field Documentation

**6.4.1.1** `Ecore_List* _Efreet_Cache_Search_List::list`

**6.4.1.2** `const char* _Efreet_Cache_Search_List::what`

The documentation for this struct was generated from the following file:

- `lib/efreet_utils.c`

## 6.5 Efreet\_Desktop Struct Reference

```
#include <efreet_desktop.h>
```

### 6.5.1 Detailed Description

a parsed representation of a .desktop file

**Efreet\_Desktop** (p. 33)

#### Data Fields

- `Ecore_List * categories`
- `char * comment`
- `char * exec`
- `char * generic_name`
- `unsigned char hidden:1`
- `char * icon`
- `double load_time`
- `Ecore_List * mime_types`
- `char * name`
- `unsigned char no_display:1`
- `Ecore_List * not_show_in`
- `Ecore_List * only_show_in`
- `char * orig_path`
- `char * path`
- `unsigned char startup_notify:1`
- `char * startup_wm_class`
- `unsigned char terminal:1`
- `char * try_exec`
- `int type`
- `void * type_data`
- `char * url`
- `double version`
- `Ecore_Hash * x`

### 6.5.2 Field Documentation

#### 6.5.2.1 `Ecore_List* Efreet_Desktop::categories`

Categories in which item should be shown

#### 6.5.2.2 `char* Efreet_Desktop::comment`

Tooltip for the entry

#### 6.5.2.3 `char* Efreet_Desktop::exec`

Program to execute

**6.5.2.4 char\* Efreet\_\_Desktop::generic\_\_name**

Generic name of the application

**6.5.2.5 unsigned char Efreet\_\_Desktop::hidden**

User delete the item

**6.5.2.6 char\* Efreet\_\_Desktop::icon**

Icon to display in file manager, menus, etc

**6.5.2.7 double Efreet\_\_Desktop::load\_\_time**

when the .desktop was loaded from disk

**6.5.2.8 Ecore\_List\* Efreet\_\_Desktop::mime\_\_types**

The mime types supported by this app

**6.5.2.9 char\* Efreet\_\_Desktop::name**

Specific name of the application

**6.5.2.10 unsigned char Efreet\_\_Desktop::no\_\_display**

Don't display this application in menus

**6.5.2.11 Ecore\_List\* Efreet\_\_Desktop::not\_\_show\_\_in**

list of environments that shouldn't display the icon

**6.5.2.12 Ecore\_List\* Efreet\_\_Desktop::only\_\_show\_\_in**

list of environments that should display the icon

**6.5.2.13 char\* Efreet\_\_Desktop::orig\_\_path**

original path to .desktop file

**6.5.2.14 char\* Efreet\_\_Desktop::path**

Working directory to run app in

**6.5.2.15 unsigned char Efreet\_\_Desktop::startup\_\_notify**

The startup notify settings of the app

**6.5.2.16 char\* Efreet \_\_Desktop::startup\_wm\_class**

If specified will map at least one window with the given string as it's WM class or WM name

**6.5.2.17 unsigned char Efreet \_\_Desktop::terminal**

Does the program run in a terminal

**6.5.2.18 char\* Efreet \_\_Desktop::try\_exec**

Binary to determine if app is installed

**6.5.2.19 int Efreet \_\_Desktop::type**

type of desktop file

**6.5.2.20 void\* Efreet \_\_Desktop::type\_data**

Type specific data for custom types

**6.5.2.21 char\* Efreet \_\_Desktop::url**

URL to access if type is EFREET\_TYPE\_LINK

**6.5.2.22 double Efreet \_\_Desktop::version**

version of spec file conforms to

**6.5.2.23 Ecore\_Hash\* Efreet \_\_Desktop::x**

Keep track of all user extensions, keys that begin with X-

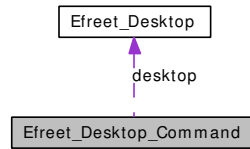
The documentation for this struct was generated from the following file:

- lib/efreet \_\_desktop.h

## 6.6 Efreet\_Desktop\_Command Struct Reference

```
#include <efreet_private.h>
```

Collaboration diagram for Efreet\_Desktop\_Command:



### Data Fields

- **Efreet\_Desktop\_Command\_Cb** `cb_command`
- **Efreet\_Desktop\_Progress\_Cb** `cb_progress`
- `void * data`
- **Efreet\_Desktop** \* `desktop`
- `Ecore_List * files`
- **Efreet\_Desktop\_Command\_Flag** `flags`
- `int num_pending`

### 6.6.1 Field Documentation

**6.6.1.1** `Efreet_Desktop_Command_Cb Efreet_Desktop_Command::cb_command`

**6.6.1.2** `Efreet_Desktop_Progress_Cb Efreet_Desktop_Command::cb_progress`

**6.6.1.3** `void* Efreet_Desktop_Command::data`

**6.6.1.4** `Efreet_Desktop* Efreet_Desktop_Command::desktop`

**6.6.1.5** `Ecore_List* Efreet_Desktop_Command::files`

list of `Efreet_Desktop_Command_File` (p.37)

**6.6.1.6** `Efreet_Desktop_Command_Flag Efreet_Desktop_Command::flags`

**6.6.1.7** `int Efreet_Desktop_Command::num_pending`

The documentation for this struct was generated from the following file:

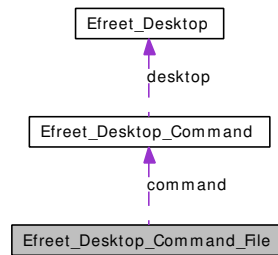
- `lib/efreet_private.h`



## 6.7 Efreet\_Desktop\_Command\_File Struct Reference

```
#include <efreet_private.h>
```

Collaboration diagram for Efreet\_Desktop\_Command\_File:



### Data Fields

- `Efreet_Desktop_Command * command`
- `char * dir`
- `char * file`
- `char * fullpath`
- `int pending`
- `char * uri`

#### 6.7.1 Field Documentation

**6.7.1.1** `Efreet_Desktop_Command* Efreet_Desktop_Command_File::command`

**6.7.1.2** `char* Efreet_Desktop_Command_File::dir`

**6.7.1.3** `char* Efreet_Desktop_Command_File::file`

**6.7.1.4** `char* Efreet_Desktop_Command_File::fullpath`

**6.7.1.5** `int Efreet_Desktop_Command_File::pending`

**6.7.1.6** `char* Efreet_Desktop_Command_File::uri`

The documentation for this struct was generated from the following file:

- `lib/efreet_private.h`

## 6.8 Efreet\_Desktop\_Type\_Info Struct Reference

### Data Fields

- `Efreet_Desktop_Type_Free_Cb free_func`
- `int id`
- `Efreet_Desktop_Type_Parse_Cb parse_func`
- `Efreet_Desktop_Type_Save_Cb save_func`
- `char * type`

### 6.8.1 Field Documentation

**6.8.1.1** `Efreet_Desktop_Type_Free_Cb Efreet_Desktop_Type_Info::free_func`

**6.8.1.2** `int Efreet_Desktop_Type_Info::id`

**6.8.1.3** `Efreet_Desktop_Type_Parse_Cb Efreet_Desktop_Type_Info::parse_func`

**6.8.1.4** `Efreet_Desktop_Type_Save_Cb Efreet_Desktop_Type_Info::save_func`

**6.8.1.5** `char* Efreet_Desktop_Type_Info::type`

The documentation for this struct was generated from the following file:

- `lib/efreet_desktop.c`

## 6.9 Efreet\_Icon Struct Reference

```
#include <efreet_icon.h>
```

### 6.9.1 Detailed Description

Contains all the information about a given icon.

**Efreet\_Icon** (p. 39)

### Data Fields

- `Ecore_List * attach_points`
- `struct {  
    int x0  
    int x1  
    int y0  
    int y1  
} embedded_text_rectangle`
- `unsigned char has_embedded_text_rectangle:1`
- `char * name`
- `char * path`
- `unsigned int ref_count`

### 6.9.2 Field Documentation

#### 6.9.2.1 `Ecore_List* Efreet_Icon::attach_points`

List of points to be used as anchor points for emblems/overlays

#### 6.9.2.2 `struct { ... } Efreet_Icon::embedded_text_rectangle`

Rectangle where text can be displayed on the icon

#### 6.9.2.3 `unsigned char Efreet_Icon::has_embedded_text_rectangle`

Was the embedded rectangle set

#### 6.9.2.4 `char* Efreet_Icon::name`

Translated UTF8 string that can be used for the icon name

#### 6.9.2.5 `char* Efreet_Icon::path`

Full path to the icon

**6.9.2.6 unsigned int Efreet\_Icon::ref\_count**

References to this icon

**6.9.2.7 int Efreet\_Icon::x0**

x0 position

**6.9.2.8 int Efreet\_Icon::x1**

x1 position

**6.9.2.9 int Efreet\_Icon::y0**

y0 position

**6.9.2.10 int Efreet\_Icon::y1**

y1 position

The documentation for this struct was generated from the following file:

- lib/efreet\_icon.h

## 6.10 Efreet\_Icon\_Point Struct Reference

```
#include <efreet_icon.h>
```

### 6.10.1 Detailed Description

Stores an x, y point.

Efreet\_Point

#### Data Fields

- int x
- int y

### 6.10.2 Field Documentation

#### 6.10.2.1 int Efreet\_Icon\_Point::x

x coord

#### 6.10.2.2 int Efreet\_Icon\_Point::y

y coord

The documentation for this struct was generated from the following file:

- lib/efreet\_icon.h

## 6.11 Efreet\_Icon\_Theme Struct Reference

```
#include <efreet_icon.h>
```

### 6.11.1 Detailed Description

contains all of the known information about a given theme

**Efreet\_Icon\_Theme** (p. 42)

### Data Fields

- char \* **comment**
- Ecore\_List \* **directories**
- char \* **example\_icon**
- unsigned char **fake**:1
- unsigned char **hidden**:1
- Ecore\_Hash \* **icon\_cache**
- Ecore\_List \* **inherits**
- double **last\_cache\_check**
- struct {  
    const char \* **internal**  
    const char \* **name**  
} **name**
- struct {  
    int **count**  
    void \* **path**  
} **paths**
- unsigned char **valid**:1

### 6.11.2 Field Documentation

#### 6.11.2.1 char\* Efreet\_Icon\_Theme::comment

String describing the theme

#### 6.11.2.2 int Efreet\_Icon\_Theme::count

The number of path's

#### 6.11.2.3 Ecore\_List\* Efreet\_Icon\_Theme::directories

List of subdirectories for this theme

#### 6.11.2.4 char\* Efreet\_Icon\_Theme::example\_icon

Icon to use as an example of the theme

**6.11.2.5 unsigned char Efreet\_Icon\_Theme::fake**

This isn't a real theme but the user has tried to query from it. We create the fake one to give us the theme cache.

**6.11.2.6 unsigned char Efreet\_Icon\_Theme::hidden**

Should this theme be hidden from users

**6.11.2.7 Ecore\_Hash\* Efreet\_Icon\_Theme::icon\_cache**

Cache of the icon data

**6.11.2.8 Ecore\_List\* Efreet\_Icon\_Theme::inherits**

Icon themes we inherit from

**6.11.2.9 const char\* Efreet\_Icon\_Theme::internal**

The internal theme name

**6.11.2.10 double Efreet\_Icon\_Theme::last\_cache\_check**

Last time the cache was checked

**6.11.2.11 struct { ... } Efreet\_Icon\_Theme::name**

The different names for the theme

**6.11.2.12 const char\* Efreet\_Icon\_Theme::name**

The user visible name

**6.11.2.13 void\* Efreet\_Icon\_Theme::path**

The paths

**6.11.2.14 struct { ... } Efreet\_Icon\_Theme::paths**

The paths to this theme

**6.11.2.15 unsigned char Efreet\_Icon\_Theme::valid**

Have we seen an index for this theme

The documentation for this struct was generated from the following file:

- lib/efreet\_icon.h

## 6.12 Efreet\_Icon\_Theme\_Directory Struct Reference

```
#include <efreet_icon.h>
```

### 6.12.1 Detailed Description

Contains all the information about a sub-directory of a theme.

**Efreet\_Icon\_Theme\_Directory** (p. 44)

### Data Fields

- **Efreet\_Icon\_Theme\_Context** context
- char \* **name**
- struct {
  - unsigned int **max**
  - unsigned int **min**
  - unsigned int **normal**
  - unsigned int **threshold** } **size**
- **Efreet\_Icon\_Size\_Type** type

### 6.12.2 Field Documentation

#### 6.12.2.1 Efreet\_Icon\_Theme\_Context Efreet\_Icon\_Theme\_Directory::context

The type of icons in the dir

#### 6.12.2.2 unsigned int Efreet\_Icon\_Theme\_Directory::max

The maximum size for this directory

#### 6.12.2.3 unsigned int Efreet\_Icon\_Theme\_Directory::min

The minimum size for this directory

#### 6.12.2.4 char\* Efreet\_Icon\_Theme\_Directory::name

The directory name

#### 6.12.2.5 unsigned int Efreet\_Icon\_Theme\_Directory::normal

The size for this directory

#### 6.12.2.6 struct { ... } Efreet\_Icon\_Theme\_Directory::size

The size settings for the icon theme



**6.12.2.7 unsigned int Efreet\_Icon\_Theme\_Directory::threshold**

Size difference threshold

**6.12.2.8 Efreet\_Icon\_Size\_Type Efreet\_Icon\_Theme\_Directory::type**

The size type for the icons

The documentation for this struct was generated from the following file:

- lib/efreet\_icon.h

## 6.13 Efreet\_Ini Struct Reference

```
#include <efreet_ini.h>
```

### 6.13.1 Detailed Description

Contains all the information about an ini file.

**Efreet\_Ini** (p. 46)

#### Data Fields

- `Ecore_Hash * data`
- `Ecore_Hash * section`

### 6.13.2 Field Documentation

#### 6.13.2.1 `Ecore_Hash* Efreet_Ini::data`

Hash of string => (Hash of string => string)

#### 6.13.2.2 `Ecore_Hash* Efreet_Ini::section`

currently selected section

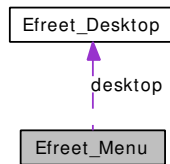
The documentation for this struct was generated from the following file:

- `lib/efreet_ini.h`

## 6.14 Efreet\_\_Menu Struct Reference

```
#include <efreet_menu.h>
```

Collaboration diagram for Efreet\_\_Menu:



### 6.14.1 Detailed Description

**Efreet\_\_Menu** (p. 47) Stores information on a entry in the menu

#### Data Fields

- **Efreet\_\_Desktop \* desktop**
- **Ecore\_List \* entries**
- **const char \* icon**
- **const char \* id**
- **const char \* name**
- **Efreet\_\_Menu\_Entry\_Type type**

### 6.14.2 Field Documentation

#### 6.14.2.1 Efreet\_\_Desktop\* Efreet\_\_Menu::desktop

The desktop we refer too

#### 6.14.2.2 Ecore\_List\* Efreet\_\_Menu::entries

The menu items

#### 6.14.2.3 const char\* Efreet\_\_Menu::icon

Icon for this entry

#### 6.14.2.4 const char\* Efreet\_\_Menu::id

File-id for desktop and relative name for menu

#### 6.14.2.5 const char\* Efreet\_\_Menu::name

Name this entry should show

#### 6.14.2.6 Efreet\_Menu\_Entry\_Type Efreet\_Menu::type

The documentation for this struct was generated from the following file:

- lib/efreet\_menu.h

## 6.15 Efreet\_Menu\_App\_Dir Struct Reference

### 6.15.1 Detailed Description

Holds information on an app dir

#### Data Fields

- unsigned int **legacy**:1
- char \* **path**
- char \* **prefix**

### 6.15.2 Field Documentation

#### 6.15.2.1 unsigned int Efreet\_Menu\_App\_Dir::legacy

is this a legacy dir

#### 6.15.2.2 char\* Efreet\_Menu\_App\_Dir::path

directory path

#### 6.15.2.3 char\* Efreet\_Menu\_App\_Dir::prefix

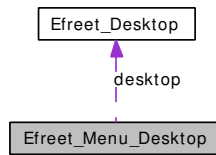
If it's legacy it can have a prefix

The documentation for this struct was generated from the following file:

- lib/efreet\_menu.c

## 6.16 Efreet\_Menu\_Desktop Struct Reference

Collaboration diagram for Efreet\_Menu\_Desktop:



### 6.16.1 Detailed Description

**Efreet\_Menu\_Desktop** (p. 50) Stores information on a desktop for the menu

#### Data Fields

- unsigned char **allocated**:1
- **Efreet\_Desktop** \* **desktop**
- const char \* **id**

### 6.16.2 Field Documentation

#### 6.16.2.1 unsigned char Efreet\_Menu\_Desktop::allocated

If this desktop has been allocated

#### 6.16.2.2 Efreet\_Desktop\* Efreet\_Menu\_Desktop::desktop

The desktop we refer too

#### 6.16.2.3 const char\* Efreet\_Menu\_Desktop::id

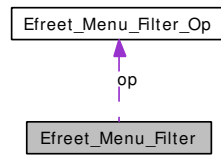
The desktop file id

The documentation for this struct was generated from the following file:

- lib/efreet\_menu.c

## 6.17 Efreet\_Menu\_Filter Struct Reference

Collaboration diagram for Efreet\_Menu\_Filter:



### 6.17.1 Detailed Description

**Efreet\_Menu\_Filter** (p. 51) Stores information on a filter

#### Data Fields

- `Efreet_Menu_Filter_Op * op`
- `Efreet_Menu_Filter_Type type`

### 6.17.2 Field Documentation

#### 6.17.2.1 `Efreet_Menu_Filter_Op* Efreet_Menu_Filter::op`

The filter operations

#### 6.17.2.2 `Efreet_Menu_Filter_Type Efreet_Menu_Filter::type`

The type of filter

The documentation for this struct was generated from the following file:

- `lib/efreet_menu.c`

## 6.18 Efreet\_Menu\_Filter\_Op Struct Reference

### 6.18.1 Detailed Description

**Efreet\_Menu\_Filter\_Op** (p. 52) Contains information on a filter operation

#### Data Fields

- unsigned char **all**:1
- Ecore\_List \* **categories**
- Ecore\_List \* **filenames**
- Ecore\_List \* **filters**
- **Efreet\_Menu\_Filter\_Op\_Type** type

### 6.18.2 Field Documentation

#### 6.18.2.1 unsigned char Efreet\_Menu\_Filter\_Op::all

Applies to all .desktop files

#### 6.18.2.2 Ecore\_List\* Efreet\_Menu\_Filter\_Op::categories

The categories this op applies too

#### 6.18.2.3 Ecore\_List\* Efreet\_Menu\_Filter\_Op::filenames

The filenames this op applies too

#### 6.18.2.4 Ecore\_List\* Efreet\_Menu\_Filter\_Op::filters

Child filters

#### 6.18.2.5 Efreet\_Menu\_Filter\_Op\_Type Efreet\_Menu\_Filter\_Op::type

The type of operation

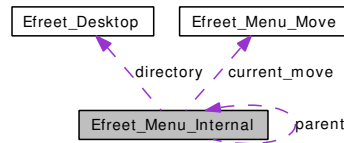
The documentation for this struct was generated from the following file:

- lib/efreet\_menu.c



## 6.19 Efreet \_Menu \_Internal Struct Reference

Collaboration diagram for Efreet \_Menu \_Internal:



### 6.19.1 Detailed Description

**Efreet \_Menu \_Internal** (p. 53) Contains the information about a menu

#### Data Fields

- Ecore\_List \* **app\_dirs**
- Ecore\_List \* **app\_pool**
- Ecore\_List \* **applications**
- **Efreet \_Menu \_Move** \* **current\_move**
- Ecore\_List \* **default\_layout**
- unsigned char **deleted**:1
- Ecore\_DList \* **directories**
- **Efreet \_Desktop** \* **directory**
- Ecore\_Hash \* **directory\_cache**
- Ecore\_DList \* **directory\_dirs**
- struct {
  - char \* **name**
  - char \* **path**
 } **file**
- Ecore\_List \* **filters**
- char **in\_line**
- char **inline\_alias**
- char **inline\_header**
- char **inline\_limit**
- Ecore\_List \* **layout**
- Ecore\_List \* **moves**
- struct {
  - const char \* **internal**
  - const char \* **name**
 } **name**
- unsigned char **only\_unallocated**:1
- **Efreet \_Menu \_Internal** \* **parent**
- unsigned char **seen\_allocated**:1
- unsigned char **seen\_deleted**:1
- char **show\_empty**
- Ecore\_List \* **sub\_menus**

## 6.19.2 Field Documentation

### 6.19.2.1 `Ecore_List* Efreet_Menu_Internal::app_dirs`

.desktop application directories

### 6.19.2.2 `Ecore_List* Efreet_Menu_Internal::app_pool`

application pool

### 6.19.2.3 `Ecore_List* Efreet_Menu_Internal::applications`

applications in this menu

### 6.19.2.4 `Efreet_Menu_Move* Efreet_Menu_Internal::current_move`

The current move

### 6.19.2.5 `Ecore_List* Efreet_Menu_Internal::default_layout`

Default layout

### 6.19.2.6 `unsigned char Efreet_Menu_Internal::deleted`

The menu is deleted

### 6.19.2.7 `Ecore_DList* Efreet_Menu_Internal::directories`

All the directories set in the menu file

### 6.19.2.8 `Efreet_Desktop* Efreet_Menu_Internal::directory`

The directory

### 6.19.2.9 `Ecore_Hash* Efreet_Menu_Internal::directory_cache`

.directory dirs

### 6.19.2.10 `Ecore_DList* Efreet_Menu_Internal::directory_dirs`

.directory file directories

### 6.19.2.11 `struct { ... } Efreet_Menu_Internal::file`

The menu file information

**6.19.2.12   Ecore \_List\* Efreet \_Menu \_Internal::filters**

Include and Exclude filters

**6.19.2.13   char Efreet \_Menu \_Internal::in \_line**

Whether this meny can be inlined

**6.19.2.14   char Efreet \_Menu \_Internal::inline \_alias**

Whether we should use the menu name when inlining

**6.19.2.15   char Efreet \_Menu \_Internal::inline \_header**

Whether we should use the header name when this menu is inlined

**6.19.2.16   char Efreet \_Menu \_Internal::inline \_limit**

Number of elements which triggers inline

**6.19.2.17   const char\* Efreet \_Menu \_Internal::internal**

The menu name

**6.19.2.18   Ecore \_List\* Efreet \_Menu \_Internal::layout**

This menus layout

**6.19.2.19   Ecore \_List\* Efreet \_Menu \_Internal::moves**

List of moves to be handled by the menu

**6.19.2.20   struct { ... } Efreet \_Menu \_Internal::name**

The names for this menu

**6.19.2.21   const char\* Efreet \_Menu \_Internal::name**

Name to use in the menus

**6.19.2.22   char\* Efreet \_Menu \_Internal::name**

The filename for this menu

**6.19.2.23   unsigned char Efreet \_Menu \_Internal::only \_unallocated**

Show only unallocated .desktops

**6.19.2.24** `Efreet_Menu_Internal* Efreet_Menu_Internal::parent`

Our parent menu

**6.19.2.25** `char* Efreet_Menu_Internal::path`

The base file path

**6.19.2.26** `unsigned char Efreet_Menu_Internal::seen_allocated`

have we set the only\_unallocated

**6.19.2.27** `unsigned char Efreet_Menu_Internal::seen_deleted`

Have we seen the deleted item yet

**6.19.2.28** `char Efreet_Menu_Internal::show_empty`

Whether to show empty menus

**6.19.2.29** `Ecore_List* Efreet_Menu_Internal::sub_menus`

Our sub menus

The documentation for this struct was generated from the following file:

- `lib/efreet_menu.c`

## 6.20 Efreet \_Menu \_Layout Struct Reference

### 6.20.1 Detailed Description

**Efreet \_Menu \_Layout** (p. 57) Stores information on a layout

#### Data Fields

- char **in \_line**
- char **inline \_alias**
- char **inline \_header**
- char **inline \_limit**
- char \* **name**
- char **show \_empty**
- **Efreet \_Menu \_Layout \_Type** type

### 6.20.2 Field Documentation

#### 6.20.2.1 char Efreet \_Menu \_Layout::in \_line

Whether this meny can be inlined

#### 6.20.2.2 char Efreet \_Menu \_Layout::inline \_alias

Whether we should use the menu name when inlining

#### 6.20.2.3 char Efreet \_Menu \_Layout::inline \_header

Whether we should use the header name when this menu is inlined

#### 6.20.2.4 char Efreet \_Menu \_Layout::inline \_limit

Number of elements which triggers inline

#### 6.20.2.5 char\* Efreet \_Menu \_Layout::name

The name of the element

#### 6.20.2.6 char Efreet \_Menu \_Layout::show \_empty

Whether to show empty menus

#### 6.20.2.7 Efreet \_Menu \_Layout \_Type Efreet \_Menu \_Layout::type

The type of layout

The documentation for this struct was generated from the following file:

- `lib/efreet__menu.c`

## 6.21 Efreet\_\_Menu\_\_Move Struct Reference

### 6.21.1 Detailed Description

**Efreet\_\_Menu\_\_Move** (p. 59) Info on a menu movement

#### Data Fields

- `char * new__name`
- `char * old__name`

### 6.21.2 Field Documentation

#### 6.21.2.1 `char* Efreet__Menu__Move::new__name`

The menu path to move too

#### 6.21.2.2 `char* Efreet__Menu__Move::old__name`

The menu path to move from

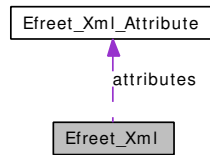
The documentation for this struct was generated from the following file:

- `lib/efreet__menu.c`

## 6.22 Efreet\_Xml Struct Reference

```
#include <efreet_xml.h>
```

Collaboration diagram for Efreet\_Xml:



### 6.22.1 Detailed Description

Contains the XML tree for a given XML document.

**Efreet\_Xml** (p. 60)

#### Data Fields

- **Efreet\_Xml\_Attribute\*\* attributes**
- **Ecore\_DList\* children**
- **const char\* tag**
- **char\* text**

### 6.22.2 Field Documentation

#### 6.22.2.1 Efreet\_Xml\_Attribute\*\* Efreet\_Xml::attributes

The attributes for this node

#### 6.22.2.2 Ecore\_DList\* Efreet\_Xml::children

Child nodes

#### 6.22.2.3 const char\* Efreet\_Xml::tag

The tag for this node

#### 6.22.2.4 char\* Efreet\_Xml::text

The XML text for this node

The documentation for this struct was generated from the following file:

- **lib/efreet\_xml.h**



## 6.23 Efreet\_Xml\_Attribute Struct Reference

```
#include <efreet_xml.h>
```

### 6.23.1 Detailed Description

Contains information about a given XML attribute.

Efreet\_Xml\_Attributes

### Data Fields

- `const char * key`
- `const char * value`

### 6.23.2 Field Documentation

#### 6.23.2.1 `const char* Efreet_Xml_Attribute::key`

The attribute key

#### 6.23.2.2 `const char* Efreet_Xml_Attribute::value`

The attribute value

The documentation for this struct was generated from the following file:

- `lib/efreet_xml.h`



## Chapter 7

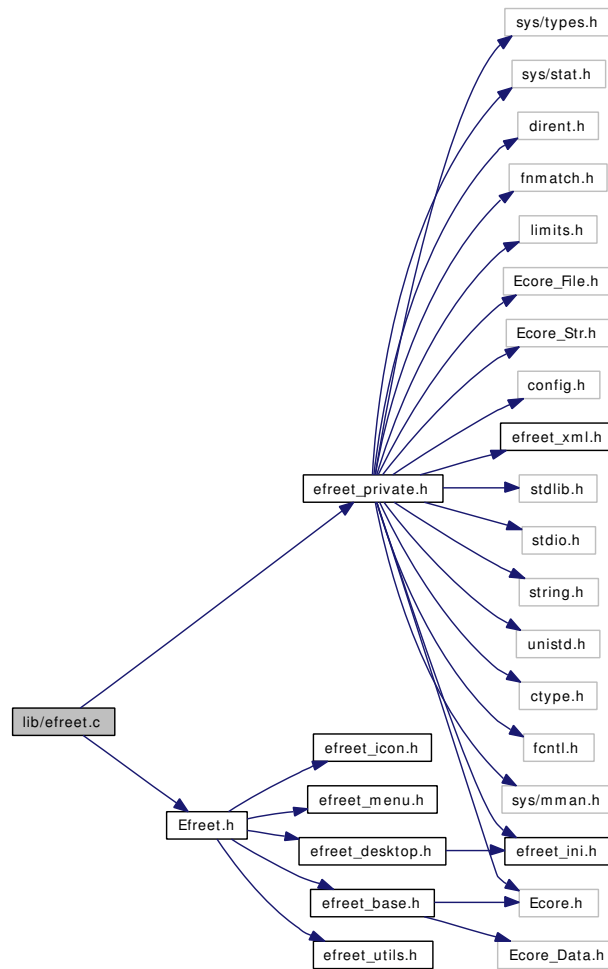
# Efreet File Documentation

### 7.1 lib/efreet.c File Reference

```
#include "Efreet.h"
```

```
#include "efreet_private.h"
```

Include dependency graph for efreet.c:



## Functions

- `size_t efreet_array_cat (char *buffer, size_t size, const char *strs[])`
- `int efreet_init (void)`

*Initializes the Efreet system.*

- `const char * efreet_lang_country_get (void)`
- `const char * efreet_lang_get (void)`
- `const char * efreet_lang_modifier_get (void)`
- `int efreet_shutdown (void)`

*Shuts down Efreet if a balanced number of init/shutdown calls have been made.*

### 7.1.1 Function Documentation

#### 7.1.1.1 `int efreet_init (void)`

Initializes the Efreet system.

**Returns:**

Returns  $> 0$  if the initialization was successful, 0 otherwise

**7.1.1.2 int efreet\_shutdown (void)**

Shuts down Efreet if a balanced number of init/shutdown calls have been made.

**Returns:**

Returns the number of times the init function as been called minus the corresponding init call.

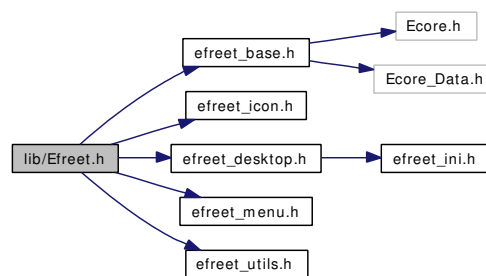
## 7.2 lib/Efreet.h File Reference

### 7.2.1 Detailed Description

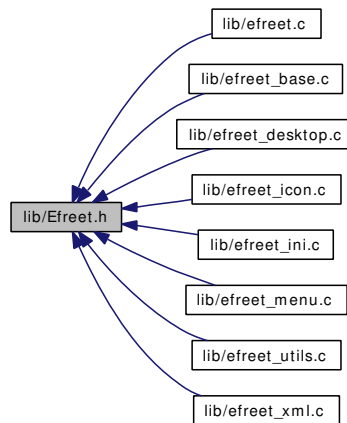
The file that must be included by any project wishing to use Efreet. **Efreet.h** (p. 66) provides all of the necessary headers and includes to work with Efreet.

```
#include "efreet_base.h"
#include "efreet_icon.h"
#include "efreet_desktop.h"
#include "efreet_menu.h"
#include "efreet_utils.h"
```

Include dependency graph for Efreet.h:



This graph shows which files directly or indirectly include this file:



## Functions

- `int efreet_init (void)`  
*Initializes the Efreet system.*
- `int efreet_shutdown (void)`  
*Shuts down Efreet if a balanced number of init/shutdown calls have been made.*

## 7.2.2 Function Documentation

### 7.2.2.1 int efreet\_init (void)

Initializes the Efreet system.

**Returns:**

Returns > 0 if the initialization was successful, 0 otherwise

### 7.2.2.2 int efreet\_shutdown (void)

Shuts down Efreet if a balanced number of init/shutdown calls have been made.

**Returns:**

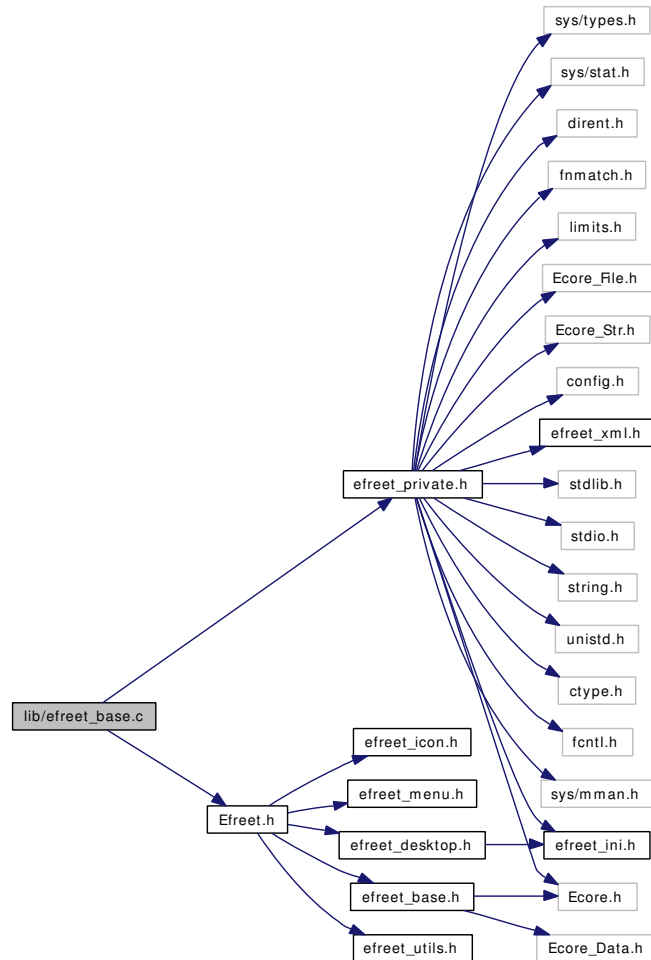
Returns the number of times the init function as been called minus the corresponding init call.

## 7.3 lib/efreet\_base.c File Reference

```
#include "Efreet.h"
```

```
#include "efreet_private.h"
```

Include dependency graph for efreet\_base.c:



## Functions

- `int efreet_base_init (void)`
- `void efreet_base_shutdown (void)`
- `const char * efreet_cache_home_get (void)`  
*Retrieves the XDG Cache Home directory.*
- `Ecore_List * efreet_config_dirs_get (void)`  
*Returns the Ecore\_List of preference ordered extra config directories.*
- `const char * efreet_config_home_get (void)`  
*Retrieves the XDG Config Home directory.*



- `Ecore_List * efreet_data_dirs_get (void)`  
*Returns the Ecore\_List of preference ordered extra data directories.*
- `const char * efreet_data_home_get (void)`  
*Retrieves the XDG Data Home directory.*
- `const char * efreet_home_dir_get (void)`

## 7.4 lib/efreet\_base.h File Reference

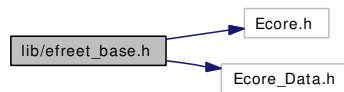
### 7.4.1 Detailed Description

Contains the methods used to support the FDO base directory specification.

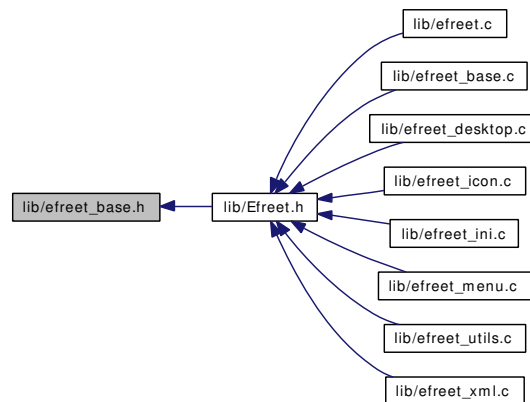
```
#include <Ecore.h>
```

```
#include <Ecore_Data.h>
```

Include dependency graph for efreet\_base.h:



This graph shows which files directly or indirectly include this file:



## Functions

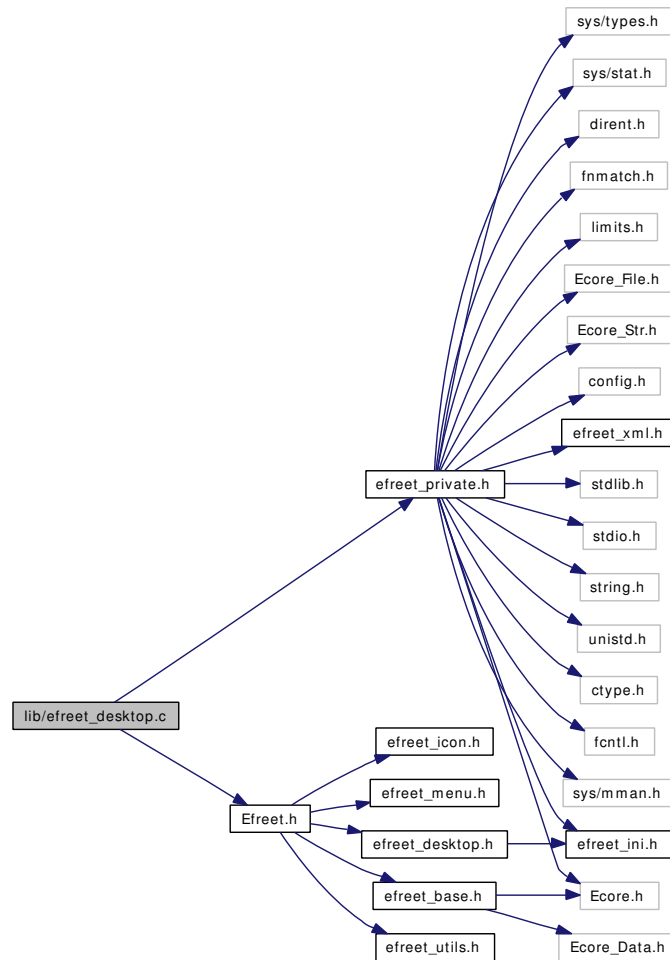
- `const char * efreet_cache_home_get (void)`  
*Retrieves the XDG Cache Home directory.*
- `Ecore_List * efreet_config_dirs_get (void)`  
*Returns the Ecore\_List of preference ordered extra config directories.*
- `const char * efreet_config_home_get (void)`  
*Retrieves the XDG Config Home directory.*
- `Ecore_List * efreet_data_dirs_get (void)`  
*Returns the Ecore\_List of preference ordered extra data directories.*
- `const char * efreet_data_home_get (void)`  
*Retrieves the XDG Data Home directory.*

## 7.5 lib/efreet\_\_desktop.c File Reference

```
#include "Efreet.h"
```

```
#include "efreet_private.h"
```

Include dependency graph for efreet\_\_desktop.c:



### Data Structures

- struct `Efreet_Desktop_Type_Info`

### Defines

- `#define` `DESKTOP_VERSION` 1.0

### Typedefs

- `typedef` `Efreet_Desktop_Type_Info` `Efreet_Desktop_Type_Info`

## Functions

- void **efreet\_desktop\_category\_add** (Efreet\_Desktop \*desktop, const char \*category)  
*add a category to a desktop*
- unsigned int **efreet\_desktop\_category\_count\_get** (Efreet\_Desktop \*desktop)  
*Retrieves the number of categories the given desktop belongs too.*
- int **efreet\_desktop\_category\_del** (Efreet\_Desktop \*desktop, const char \*category)  
*removes a category from a desktop*
- int **efreet\_desktop\_command\_get** (Efreet\_Desktop \*desktop, Ecore\_List \*files, Efreet\_Desktop\_Command\_Cb func, void \*data)  
*Get a command to use to execute a desktop entry.*
- int **efreet\_desktop\_command\_progress\_get** (Efreet\_Desktop \*desktop, Ecore\_List \*files, Efreet\_Desktop\_Command\_Cb cb\_command, Efreet\_Desktop\_Progress\_Cb cb\_progress, void \*data)  
*Get a command to use to execute a desktop entry, and receive progress updates for downloading of remote URI's passed in.*
- **Efreet\_Desktop \*****efreet\_desktop\_empty\_new** (const char \*file)  
*Creates a new empty **Efreet\_Desktop** (p. 33) structure or NULL on failure.*
- void **efreet\_desktop\_environment\_set** (const char \*environment)
- void **efreet\_desktop\_exec** (Efreet\_Desktop \*desktop, Ecore\_List \*files, void \*data)  
*Parses the desktop exec line and returns an Ecore\_Exe.*
- void **efreet\_desktop\_free** (Efreet\_Desktop \*desktop)
- **Efreet\_Desktop \*****efreet\_desktop\_get** (const char \*file)  
*Gets a reference to an **Efreet\_Desktop** (p. 33) structure representing the contents of file or NULL if file is not a valid .desktop file.*
- int **efreet\_desktop\_init** (void)
- int **efreet\_desktop\_save** (Efreet\_Desktop \*desktop)  
*Saves any changes made to desktop back to the file on the filesystem.*
- int **efreet\_desktop\_save\_as** (Efreet\_Desktop \*desktop, const char \*file)  
*Saves desktop to file.*
- int **efreet\_desktop\_shutdown** (void)
- int **efreet\_desktop\_type\_add** (const char \*type, Efreet\_Desktop\_Type\_Parse\_Cb parse\_func, Efreet\_Desktop\_Type\_Save\_Cb save\_func, Efreet\_Desktop\_Type\_Free\_Cb free\_func)  
*Adds the given type to the list of types in the system.*
- void \* **efreet\_desktop\_type\_data\_get** (Efreet\_Desktop \*desktop)  
*get type specific data for custom desktop types*

## Variables

- int EFREET\_DESKTOP\_TYPE\_APPLICATION = 0
- int EFREET\_DESKTOP\_TYPE\_DIRECTORY = 0
- int EFREET\_DESKTOP\_TYPE\_LINK = 0

### 7.5.1 Define Documentation

7.5.1.1 #define DESKTOP\_VERSION 1.0

### 7.5.2 Typedef Documentation

7.5.2.1 typedef struct Efreet\_Desktop\_Type\_Info Efreet\_Desktop\_Type\_Info

## 7.6 lib/efreet\_desktop.h File Reference

### 7.6.1 Detailed Description

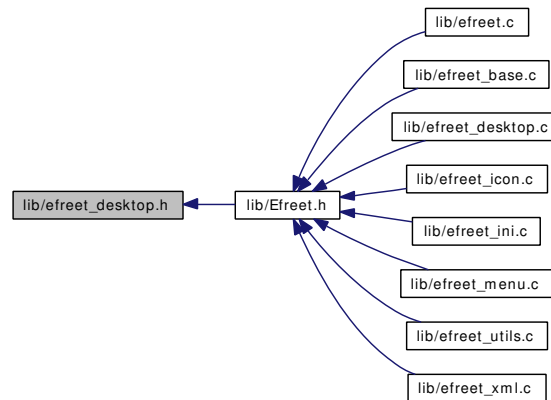
Contains the structures and methods used to support the FDO desktop entry specification.

```
#include "efreet_ini.h"
```

Include dependency graph for efreet\_desktop.h:



This graph shows which files directly or indirectly include this file:



### Data Structures

- struct **Efreet\_Desktop**  
*a parsed representation of a .desktop file*

### Typedefs

- typedef **Efreet\_Desktop** **Efreet\_Desktop**
- typedef void(\*) **Efreet\_Desktop\_Command\_Cb** (void \*data, **Efreet\_Desktop** \*desktop, char \*command, int remaining)
- typedef int(\*) **Efreet\_Desktop\_Progress\_Cb** (void \*data, **Efreet\_Desktop** \*desktop, char \*uri, long int total, long int current)
- typedef enum **Efreet\_Desktop\_Type** **Efreet\_Desktop\_Type**
- typedef void \*(\*) **Efreet\_Desktop\_Type\_Free\_Cb** (void \*data)
- typedef void \*(\*) **Efreet\_Desktop\_Type\_Parse\_Cb** (**Efreet\_Desktop** \*desktop, **Efreet\_Ini** \*ini)
- typedef void(\*) **Efreet\_Desktop\_Type\_Save\_Cb** (**Efreet\_Desktop** \*desktop, **Efreet\_Ini** \*ini)

## Functions

- void **efreet \_desktop \_category \_add** (**Efreet \_Desktop** \*desktop, const char \*category)  
*add a category to a desktop*
- unsigned int **efreet \_desktop \_category \_count \_get** (**Efreet \_Desktop** \*desktop)  
*Retrieves the number of categories the given desktop belongs too.*
- int **efreet \_desktop \_category \_del** (**Efreet \_Desktop** \*desktop, const char \*category)  
*removes a category from a desktop*
- int **efreet \_desktop \_command \_get** (**Efreet \_Desktop** \*desktop, **Ecore \_List** \*files, **Efreet \_Desktop \_Command \_Cb** func, void \*data)  
*Get a command to use to execute a desktop entry.*
- int **efreet \_desktop \_command \_progress \_get** (**Efreet \_Desktop** \*desktop, **Ecore \_List** \*files, **Efreet \_Desktop \_Command \_Cb** cb\_command, **Efreet \_Desktop \_Progress \_Cb** cb\_prog, void \*data)  
*Get a command to use to execute a desktop entry, and receive progress updates for downloading of remote URI's passed in.*
- **Efreet \_Desktop** \* **efreet \_desktop \_empty \_new** (const char \*file)  
*Creates a new empty **Efreet \_Desktop** (p. 33) structure or NULL on failure.*
- void **efreet \_desktop \_environment \_set** (const char \*environment)
- void **efreet \_desktop \_exec** (**Efreet \_Desktop** \*desktop, **Ecore \_List** \*files, void \*data)  
*Parses the desktop exec line and returns an **Ecore \_Exe**.*
- void **efreet \_desktop \_free** (**Efreet \_Desktop** \*desktop)
- **Efreet \_Desktop** \* **efreet \_desktop \_get** (const char \*file)  
*Gets a reference to an **Efreet \_Desktop** (p. 33) structure representing the contents of file or NULL if file is not a valid .desktop file.*
- int **efreet \_desktop \_save** (**Efreet \_Desktop** \*desktop)  
*Saves any changes made to desktop back to the file on the filesystem.*
- int **efreet \_desktop \_save \_as** (**Efreet \_Desktop** \*desktop, const char \*file)  
*Saves desktop to file.*
- int **efreet \_desktop \_type \_add** (const char \*type, **Efreet \_Desktop \_Type \_Parse \_Cb** parse\_func, **Efreet \_Desktop \_Type \_Save \_Cb** save\_func, **Efreet \_Desktop \_Type \_Free \_Cb** free\_func)  
*Adds the given type to the list of types in the system.*
- void \* **efreet \_desktop \_type \_data \_get** (**Efreet \_Desktop** \*desktop)  
*get type specific data for custom desktop types*

## Variables

- int EFREET\_DESKTOP\_TYPE\_APPLICATION
- int EFREET\_DESKTOP\_TYPE\_DIRECTORY
- int EFREET\_DESKTOP\_TYPE\_LINK

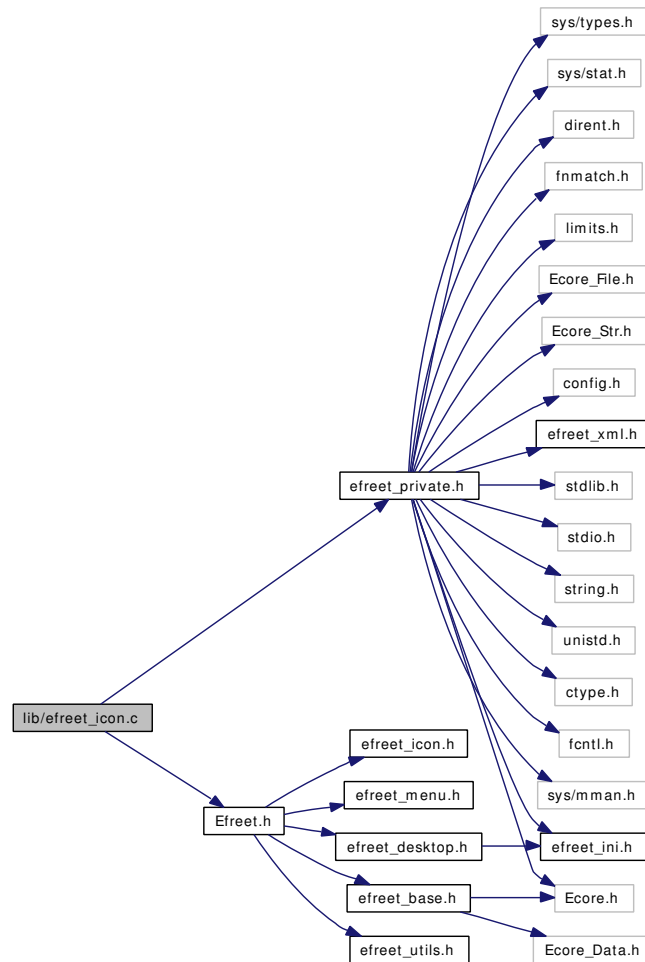


## 7.7 lib/efreet\_icon.c File Reference

```
#include "Efreet.h"
```

```
#include "efreet_private.h"
```

Include dependency graph for efreet\_icon.c:



### Defines

- `#define NO_MATCH_KEY ((char *)0xdeadbeef)`

### Functions

- `const char * efreet_icon_dir_get (void)`  
*Returns the user icon directory.*
- `void efreet_icon_extension_add (const char *ext)`  
*Adds the given extension to the list of possible icon extensions.*

- **Efreet\_Icon \* efreet\_icon\_find** (const char \*theme\_name, const char \*icon, const char \*size)

*Retrieves all of the information about the given icon.*

- **int efreet\_icon\_init** (void)
- **const char \* efreet\_icon\_path\_find** (const char \*theme, const char \*icon, const char \*size)

*Retrives the path to the given icon.*

- **void efreet\_icon\_shutdown** (void)
- **Efreet\_Icon\_Theme \* efreet\_icon\_theme\_find** (const char \*theme\_name)

*Tries to get the icon theme structure for the given theme name.*

- **Ecore\_List \* efreet\_icon\_theme\_list\_get** (void)

*Retrieves all of the non-hidden icon themes available on the system. The returned list must be freed. Do not free the list data.*

## Variables

- **Ecore\_List \* efreet\_icon\_extensions** = NULL

### 7.7.1 Define Documentation

- 7.7.1.1 **#define NO\_MATCH\_KEY** ((char \*)0xdeadbeef)

### 7.7.2 Variable Documentation

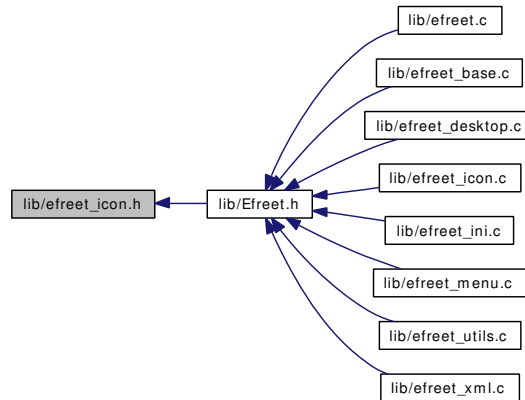
- 7.7.2.1 **Ecore\_List\* efreet\_icon\_extensions** = NULL

## 7.8 lib/efreet\_icon.h File Reference

### 7.8.1 Detailed Description

Contains the structures and methods used to support the FDO icon theme specification.

This graph shows which files directly or indirectly include this file:



### Data Structures

- struct **Efreet\_Icon**  
*Contains all the information about a given icon.*
- struct **Efreet\_Icon\_Point**  
*Stores an x, y point.*
- struct **Efreet\_Icon\_Theme**  
*contains all of the known information about a given theme*
- struct **Efreet\_Icon\_Theme\_Directory**  
*Contains all the information about a sub-directory of a theme.*

### Typedefs

- typedef **Efreet\_Icon** Efreet\_Icon
- typedef **Efreet\_Icon\_Point** Efreet\_Icon\_Point
- typedef enum **Efreet\_Icon\_Size\_Type** Efreet\_Icon\_Size\_Type
- typedef **Efreet\_Icon\_Theme** Efreet\_Icon\_Theme
- typedef enum **Efreet\_Icon\_Theme\_Context** Efreet\_Icon\_Theme\_Context
- typedef **Efreet\_Icon\_Theme\_Directory** Efreet\_Icon\_Theme\_Directory

### Enumerations

- enum **Efreet\_Icon\_Size\_Type** { EFREET\_ICON\_SIZE\_TYPE\_NONE, EFREET\_ICON\_SIZE\_TYPE\_FIXED, EFREET\_ICON\_SIZE\_TYPE\_SCALABLE, EFREET\_ICON\_SIZE\_TYPE\_THRESHOLD }

- enum **Efreet\_Icon\_Theme\_Context** {  
    **EFREET\_ICON\_THEME\_CONTEXT\_NONE**,                   **EFREET\_ICON\_**  
    **THEME\_CONTEXT\_ACTIONS**, **EFREET\_ICON\_THEME\_CONTEXT\_**  
    **DEVICES**, **EFREET\_ICON\_THEME\_CONTEXT\_FILESYSTEMS**,  
    **EFREET\_ICON\_THEME\_CONTEXT\_MIMETYPES** }

## Functions

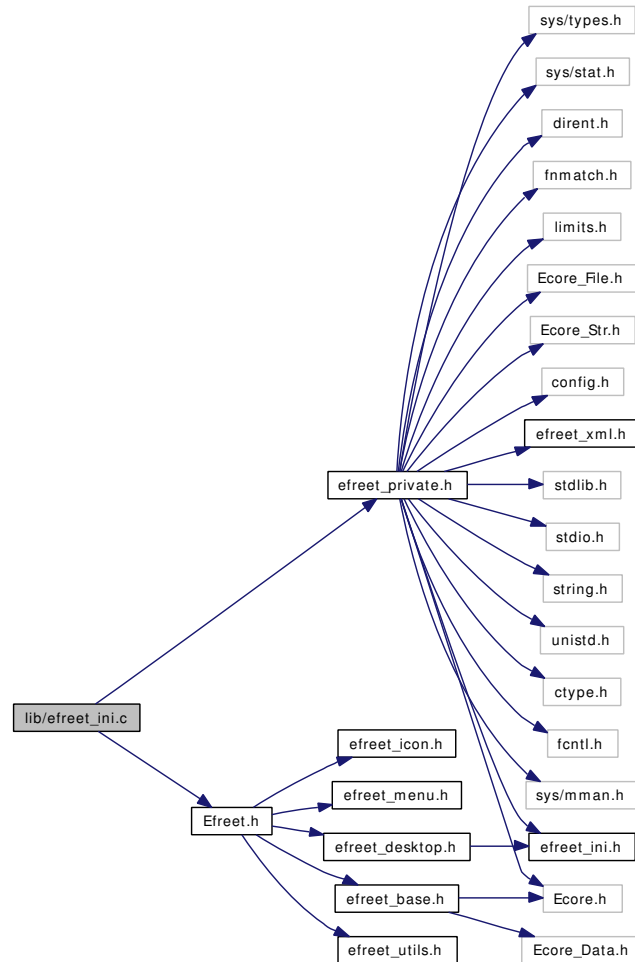
- const char \* **efreet\_icon\_dir\_get** (void)  
    *Returns the user icon directory.*
- void **efreet\_icon\_extension\_add** (const char \*ext)  
    *Adds the given extension to the list of possible icon extensions.*
- **Efreet\_Icon** \* **efreet\_icon\_find** (const char \*theme\_name, const char \*icon, const char \*size)  
    *Retrieves all of the information about the given icon.*
- const char \* **efreet\_icon\_path\_find** (const char \*theme, const char \*icon, const char \*size)  
    *Retrives the path to the given icon.*
- **Efreet\_Icon\_Theme** \* **efreet\_icon\_theme\_find** (const char \*theme\_name)  
    *Tries to get the icon theme structure for the given theme name.*
- **Ecore\_List** \* **efreet\_icon\_theme\_list\_get** (void)  
    *Retrieves all of the non-hidden icon themes available on the system. The returned list must be freed. Do not free the list data.*

## 7.9 lib/efreet \_ini.c File Reference

```
#include "Efreet.h"
```

```
#include "efreet_private.h"
```

Include dependency graph for efreet \_ini.c:



## Functions

- unsigned int **efreet \_ini \_boolean \_get** (Efreet \_Ini \*ini, const char \*key)
- void **efreet \_ini \_boolean \_set** (Efreet \_Ini \*ini, const char \*key, unsigned int value)
- double **efreet \_ini \_double \_get** (Efreet \_Ini \*ini, const char \*key)
- void **efreet \_ini \_double \_set** (Efreet \_Ini \*ini, const char \*key, double value)
- void **efreet \_ini \_free** (Efreet \_Ini \*ini)
- int **efreet \_ini \_init** (void)
- int **efreet \_ini \_int \_get** (Efreet \_Ini \*ini, const char \*key)
- void **efreet \_ini \_int \_set** (Efreet \_Ini \*ini, const char \*key, int value)
- const char \* **efreet \_ini \_localestring \_get** (Efreet \_Ini \*ini, const char \*key)
- void **efreet \_ini \_localestring \_set** (Efreet \_Ini \*ini, const char \*key, const char \*value)
- Efreet \_Ini \* **efreet \_ini \_new** (const char \*file)

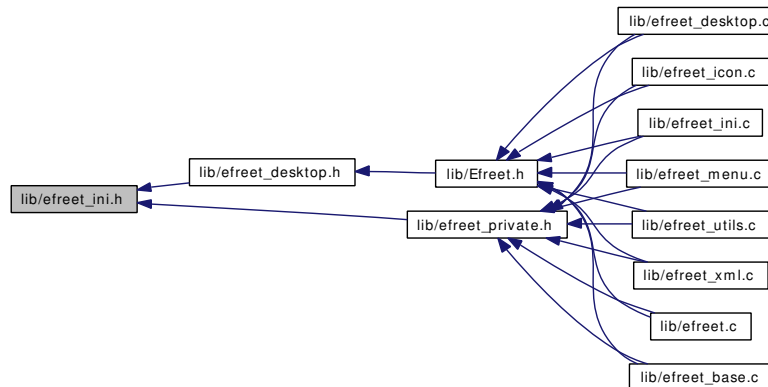
- int efreet\_ini\_save (Efreet\_Ini \*ini, const char \*file)
- void efreet\_ini\_section\_add (Efreet\_Ini \*ini, const char \*section)
- int efreet\_ini\_section\_set (Efreet\_Ini \*ini, const char \*section)
- int efreet\_ini\_shutdown (void)
- const char \* efreet\_ini\_string\_get (Efreet\_Ini \*ini, const char \*key)
- void efreet\_ini\_string\_set (Efreet\_Ini \*ini, const char \*key, const char \*value)

## 7.9.1 Function Documentation

- 7.9.1.1 unsigned int efreet\_ini\_boolean\_get (Efreet\_Ini \* *ini*, const char \* *key*)
- 7.9.1.2 void efreet\_ini\_boolean\_set (Efreet\_Ini \* *ini*, const char \* *key*, unsigned int *value*)
- 7.9.1.3 double efreet\_ini\_double\_get (Efreet\_Ini \* *ini*, const char \* *key*)
- 7.9.1.4 void efreet\_ini\_double\_set (Efreet\_Ini \* *ini*, const char \* *key*, double *value*)
- 7.9.1.5 void efreet\_ini\_free (Efreet\_Ini \* *ini*)
- 7.9.1.6 int efreet\_ini\_int\_get (Efreet\_Ini \* *ini*, const char \* *key*)
- 7.9.1.7 void efreet\_ini\_int\_set (Efreet\_Ini \* *ini*, const char \* *key*, int *value*)
- 7.9.1.8 const char\* efreet\_ini\_localestring\_get (Efreet\_Ini \* *ini*, const char \* *key*)
- 7.9.1.9 void efreet\_ini\_localestring\_set (Efreet\_Ini \* *ini*, const char \* *key*, const char \* *value*)
- 7.9.1.10 Efreet\_Ini\* efreet\_ini\_new (const char \* *file*)
- 7.9.1.11 int efreet\_ini\_save (Efreet\_Ini \* *ini*, const char \* *file*)
- 7.9.1.12 void efreet\_ini\_section\_add (Efreet\_Ini \* *ini*, const char \* *section*)
- 7.9.1.13 int efreet\_ini\_section\_set (Efreet\_Ini \* *ini*, const char \* *section*)
- 7.9.1.14 const char\* efreet\_ini\_string\_get (Efreet\_Ini \* *ini*, const char \* *key*)
- 7.9.1.15 void efreet\_ini\_string\_set (Efreet\_Ini \* *ini*, const char \* *key*, const char \* *value*)

## 7.10 lib/efreet \_ini.h File Reference

This graph shows which files directly or indirectly include this file:



### Data Structures

- struct **Efreet \_Ini**  
*Contains all the information about an ini file.*

### Typedefs

- typedef **Efreet \_Ini** **Efreet \_Ini**

### Functions

- unsigned int **efreet \_ini \_boolean \_get** (**Efreet \_Ini** \*ini, const char \*key)
- void **efreet \_ini \_boolean \_set** (**Efreet \_Ini** \*ini, const char \*key, unsigned int value)
- double **efreet \_ini \_double \_get** (**Efreet \_Ini** \*ini, const char \*key)
- void **efreet \_ini \_double \_set** (**Efreet \_Ini** \*ini, const char \*key, double value)
- void **efreet \_ini \_free** (**Efreet \_Ini** \*ini)
- int **efreet \_ini \_int \_get** (**Efreet \_Ini** \*ini, const char \*key)
- void **efreet \_ini \_int \_set** (**Efreet \_Ini** \*ini, const char \*key, int value)
- const char \* **efreet \_ini \_localestring \_get** (**Efreet \_Ini** \*ini, const char \*key)
- void **efreet \_ini \_localestring \_set** (**Efreet \_Ini** \*ini, const char \*key, const char \*value)
- **Efreet \_Ini** \* **efreet \_ini \_new** (const char \*file)
- int **efreet \_ini \_save** (**Efreet \_Ini** \*ini, const char \*path)
- void **efreet \_ini \_section \_add** (**Efreet \_Ini** \*ini, const char \*section)
- int **efreet \_ini \_section \_set** (**Efreet \_Ini** \*ini, const char \*section)
- const char \* **efreet \_ini \_string \_get** (**Efreet \_Ini** \*ini, const char \*key)
- void **efreet \_ini \_string \_set** (**Efreet \_Ini** \*ini, const char \*key, const char \*value)

## 7.10.1 Typedef Documentation

### 7.10.1.1 typedef struct Efreet\_Ini Efreet\_Ini

Efreet\_Ini (p. 46)

## 7.10.2 Function Documentation

7.10.2.1 unsigned int efreet\_ini\_boolean\_get (Efreet\_Ini \* *ini*, const char \* *key*)

7.10.2.2 void efreet\_ini\_boolean\_set (Efreet\_Ini \* *ini*, const char \* *key*, unsigned int *value*)

7.10.2.3 double efreet\_ini\_double\_get (Efreet\_Ini \* *ini*, const char \* *key*)

7.10.2.4 void efreet\_ini\_double\_set (Efreet\_Ini \* *ini*, const char \* *key*, double *value*)

7.10.2.5 void efreet\_ini\_free (Efreet\_Ini \* *ini*)

7.10.2.6 int efreet\_ini\_int\_get (Efreet\_Ini \* *ini*, const char \* *key*)

7.10.2.7 void efreet\_ini\_int\_set (Efreet\_Ini \* *ini*, const char \* *key*, int *value*)

7.10.2.8 const char\* efreet\_ini\_localestring\_get (Efreet\_Ini \* *ini*, const char \* *key*)

7.10.2.9 void efreet\_ini\_localestring\_set (Efreet\_Ini \* *ini*, const char \* *key*, const char \* *value*)

7.10.2.10 Efreet\_Ini\* efreet\_ini\_new (const char \* *file*)

7.10.2.11 int efreet\_ini\_save (Efreet\_Ini \* *ini*, const char \* *path*)

7.10.2.12 void efreet\_ini\_section\_add (Efreet\_Ini \* *ini*, const char \* *section*)

7.10.2.13 int efreet\_ini\_section\_set (Efreet\_Ini \* *ini*, const char \* *section*)

7.10.2.14 const char\* efreet\_ini\_string\_get (Efreet\_Ini \* *ini*, const char \* *key*)

7.10.2.15 void efreet\_ini\_string\_set (Efreet\_Ini \* *ini*, const char \* *key*, const char \* *value*)

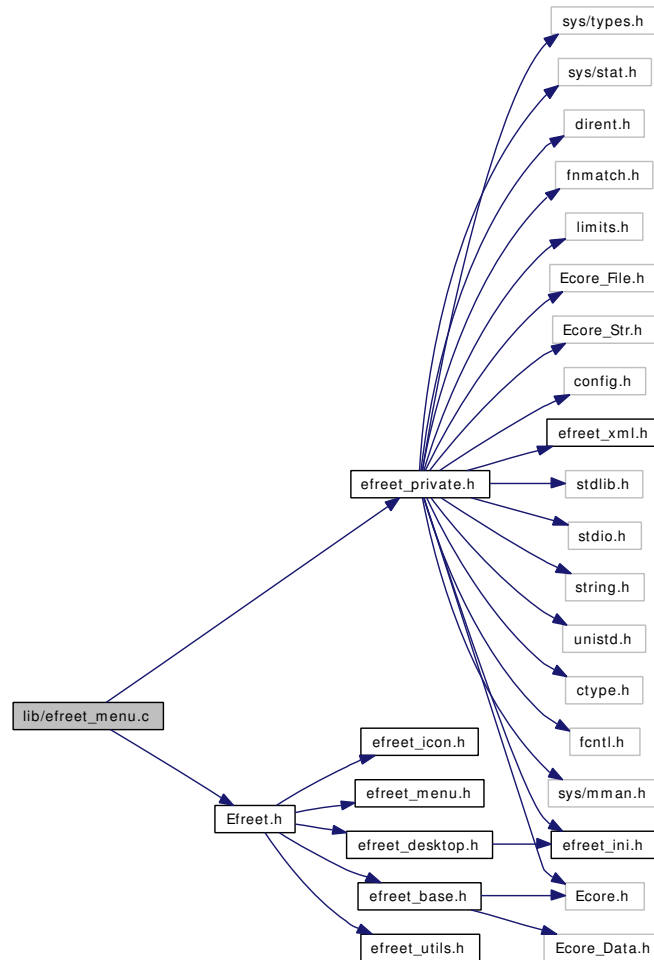


## 7.11 lib/efreet\_\_menu.c File Reference

```
#include "Efreet.h"
```

```
#include "efreet_private.h"
```

Include dependency graph for efreet\_\_menu.c:



### Data Structures

- struct **Efreet\_Menu\_App\_Dir**
- struct **Efreet\_Menu\_Desktop**
- struct **Efreet\_Menu\_Filter**
- struct **Efreet\_Menu\_Filter\_Op**
- struct **Efreet\_Menu\_Internal**
- struct **Efreet\_Menu\_Layout**
- struct **Efreet\_Menu\_Move**

### Typedefs

- typedef **Efreet\_Menu\_App\_Dir** Efreet\_Menu\_App\_Dir

- typedef **Efreet\_Menu\_Desktop** Efreet\_Menu\_Desktop
- typedef **Efreet\_Menu\_Filter** Efreet\_Menu\_Filter
- typedef **Efreet\_Menu\_Filter\_Op** Efreet\_Menu\_Filter\_Op
- typedef enum **Efreet\_Menu\_Filter\_Op\_Type** Efreet\_Menu\_Filter\_Op\_Type
- typedef enum **Efreet\_Menu\_Filter\_Type** Efreet\_Menu\_Filter\_Type
- typedef **Efreet\_Menu\_Internal** Efreet\_Menu\_Internal
- typedef **Efreet\_Menu\_Layout** Efreet\_Menu\_Layout
- typedef enum **Efreet\_Menu\_Layout\_Type** Efreet\_Menu\_Layout\_Type
- typedef **Efreet\_Menu\_Move** Efreet\_Menu\_Move

## Enumerations

- enum **Efreet\_Menu\_Filter\_Op\_Type** { EFREET\_MENU\_FILTER\_OP\_OR, EFREET\_MENU\_FILTER\_OP\_AND, EFREET\_MENU\_FILTER\_OP\_NOT }
- enum **Efreet\_Menu\_Filter\_Type** { EFREET\_MENU\_FILTER\_INCLUDE, EFREET\_MENU\_FILTER\_EXCLUDE }
- enum **Efreet\_Menu\_Layout\_Type** { EFREET\_MENU\_LAYOUT\_MENUNAME, EFREET\_MENU\_LAYOUT\_FILENAME, EFREET\_MENU\_LAYOUT\_SEPARATOR, EFREET\_MENU\_LAYOUT\_MERGE }

## Functions

- **Ecore\_List \*efreet\_default\_dirs\_get** (const char \*user\_dir, Ecore\_List \*system\_dirs, const char \*suffix)  
*Creates the list of directories based on the user dir, system dirs and given suffix.*
- **int efreet\_menu\_desktop\_insert** (Efreet\_Menu \*menu, Efreet\_Desktop \*desktop, int pos)  
*Insert a desktop element in a menu structure. Only accepts desktop files in default directories.*
- **void efreet\_menu\_dump** (Efreet\_Menu \*menu, const char \*indent)  
*Dumps the contents of the menu to the command line.*
- **void efreet\_menu\_free** (Efreet\_Menu \*entry)
- **Efreet\_Menu \*efreet\_menu\_get** (void)  
*Creates the default menu representation.*
- **int efreet\_menu\_init** (void)  
*Initializes the Efreet Menu system.*
- **int efreet\_menu\_kde\_legacy\_init** (void)  
*Initialize legacy kde support. This function blocks while the kde-config script is run.*
- **Efreet\_Menu \*efreet\_menu\_parse** (const char \*path)  
*Parses the given .menu file and creates the menu representation.*
- **int efreet\_menu\_save** (Efreet\_Menu \*menu, const char \*path)  
*Saves the menu to file.*

- void **efreet\_menu\_shutdown** (void)  
*Shuts down the Efreet menu system.*

## Variables

- **Ecore\_List \* efreet\_menu\_kde\_legacy\_dirs** = NULL

### 7.11.1 Typedef Documentation

#### 7.11.1.1 typedef struct Efreet\_Menu\_App\_Dir Efreet\_Menu\_App\_Dir

**Efreet\_Menu\_App\_Dir** (p. 49)

#### 7.11.1.2 typedef struct Efreet\_Menu\_Desktop Efreet\_Menu\_Desktop

**Efreet\_Menu\_Desktop** (p. 50)

#### 7.11.1.3 typedef struct Efreet\_Menu\_Filter Efreet\_Menu\_Filter

**Efreet\_Menu\_Filter** (p. 51)

#### 7.11.1.4 typedef struct Efreet\_Menu\_Filter\_Op Efreet\_Menu\_Filter\_Op

**Efreet\_Menu\_Filter\_Op** (p. 52)

#### 7.11.1.5 typedef enum Efreet\_Menu\_Filter\_Op\_Type Efreet\_Menu\_Filter\_Op\_Type

**Efreet\_Menu\_Filter\_Op\_Type**

#### 7.11.1.6 typedef enum Efreet\_Menu\_Filter\_Type Efreet\_Menu\_Filter\_Type

**Efreet\_Menu\_Filter\_Type**

#### 7.11.1.7 typedef struct Efreet\_Menu\_Internal Efreet\_Menu\_Internal

**Efreet\_Menu\_Internal** (p. 53)

#### 7.11.1.8 typedef struct Efreet\_Menu\_Layout Efreet\_Menu\_Layout

**Efreet\_Menu\_Layout** (p. 57)

#### 7.11.1.9 typedef enum Efreet\_Menu\_Layout\_Type Efreet\_Menu\_Layout\_Type

**Efreet\_Menu\_Layout\_Type**

#### 7.11.1.10 typedef struct Efreet\_Menu\_Move Efreet\_Menu\_Move

Efreet\_Menu\_Move (p. 59)

### 7.11.2 Enumeration Type Documentation

#### 7.11.2.1 enum Efreet\_Menu\_Filter\_Op\_Type

The type of operations we can perform with a filter

Enumerator:

*EFREET\_MENU\_FILTER\_OP\_OR*  
*EFREET\_MENU\_FILTER\_OP\_AND*  
*EFREET\_MENU\_FILTER\_OP\_NOT*

#### 7.11.2.2 enum Efreet\_Menu\_Filter\_Type

The type of filter

Enumerator:

*EFREET\_MENU\_FILTER\_INCLUDE*  
*EFREET\_MENU\_FILTER\_EXCLUDE*

#### 7.11.2.3 enum Efreet\_Menu\_Layout\_Type

The type of layout

Enumerator:

*EFREET\_MENU\_LAYOUT\_MENUNAME*  
*EFREET\_MENU\_LAYOUT\_FILENAME*  
*EFREET\_MENU\_LAYOUT\_SEPARATOR*  
*EFREET\_MENU\_LAYOUT\_MERGE*

### 7.11.3 Variable Documentation

#### 7.11.3.1 Ecore\_List\* efreet\_menu\_kde\_legacy\_dirs = NULL

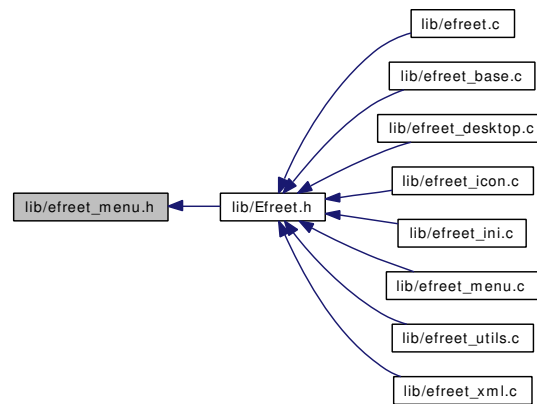
The directories to use for KDELegacy entries

## 7.12 lib/efreet\_menu.h File Reference

### 7.12.1 Detailed Description

Contains the structures and methods to support the Desktop Menu Specification.

This graph shows which files directly or indirectly include this file:



### Data Structures

- struct **Efreet\_Menu**

### Typedefs

- typedef **Efreet\_Menu** **Efreet\_Menu**
- typedef enum **Efreet\_Menu\_Entry\_Type** **Efreet\_Menu\_Entry\_Type**

### Enumerations

- enum **Efreet\_Menu\_Entry\_Type** { **EFREET\_MENU\_ENTRY\_MENU**, **EFREET\_MENU\_ENTRY\_DESKTOP**, **EFREET\_MENU\_ENTRY\_SEPARATOR**, **EFREET\_MENU\_ENTRY\_HEADER** }

### Functions

- int **efreet\_menu\_desktop\_insert** (**Efreet\_Menu** \*menu, **Efreet\_Desktop** \*desktop, int pos)  
*Insert a desktop element in a menu structure. Only accepts desktop files in default directories.*
- void **efreet\_menu\_dump** (**Efreet\_Menu** \*menu, const char \*indent)  
*Dumps the contents of the menu to the command line.*
- void **efreet\_menu\_free** (**Efreet\_Menu** \*menu)
- **Efreet\_Menu** \* **efreet\_menu\_get** (void)  
*Creates the default menu representation.*

- `int efreet__menu__kde__legacy__init (void)`  
*Initialize legacy kde support. This function blocks while the kde-config script is run.*
- `Efreet__Menu * efreet__menu__parse (const char *path)`  
*Parses the given .menu file and creates the menu representation.*
- `int efreet__menu__save (Efreet__Menu *menu, const char *path)`  
*Saves the menu to file.*

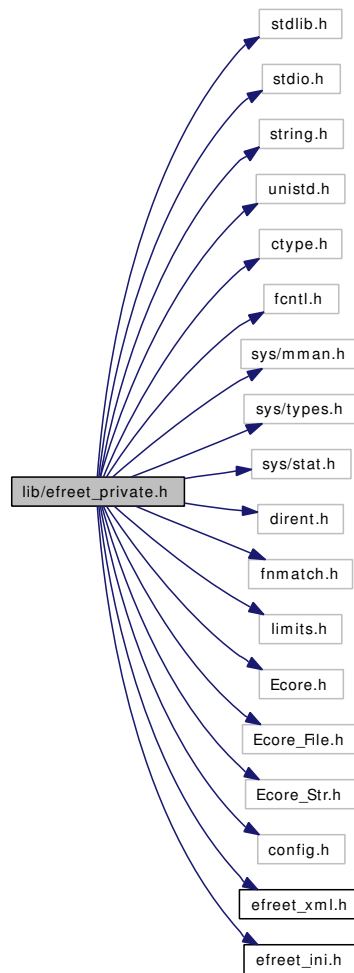
## 7.13 lib/efreet\_private.h File Reference

### 7.13.1 Detailed Description

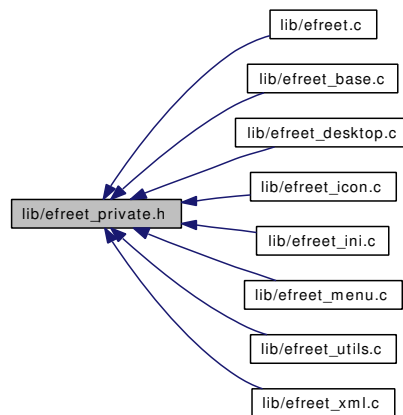
Contains methods and defines that are private to the Efreet implementaion.

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <ctype.h>
#include <fcntl.h>
#include <sys/mman.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <dirent.h>
#include <fnmatch.h>
#include <limits.h>
#include <Ecore.h>
#include <Ecore_File.h>
#include <Ecore_Str.h>
#include "config.h"
#include "efreet_xml.h"
#include "efreet_ini.h"
```

Include dependency graph for efreet\_private.h:



This graph shows which files directly or indirectly include this file:



## Data Structures

- struct `Efreet_Desktop_Command`



- struct **Efreet\_Desktop\_Command\_File**

## Defines

- #define **\_\_UNUSED**
- #define **FREE(x)** { free(x); x = NULL; }
- #define **IF\_FREE(x)** { if (x) FREE(x) }
- #define **IF\_FREE\_DLIST(x)** { if (x) ecore\_dlist\_destroy(x); x = NULL; }
- #define **IF\_FREE\_HASH(x)** { if (x) ecore\_hash\_destroy(x); x = NULL; }
- #define **IF\_FREE\_LIST(x)** { if (x) ecore\_list\_destroy(x); x = NULL; }
- #define **IF\_RELEASE(x)** { if (x) ecore\_string\_release(x); x = NULL; }
- #define **NEW(x, c)** calloc(c, sizeof(x))
- #define **PATH\_MAX** 4096

## Typedefs

- typedef **Efreet\_Desktop\_Command** **Efreet\_Desktop\_Command**
- typedef **Efreet\_Desktop\_Command\_File** **Efreet\_Desktop\_Command\_File**
- typedef enum **Efreet\_Desktop\_Command\_Flag** **Efreet\_Desktop\_Command\_Flag**

## Enumerations

- enum **Efreet\_Desktop\_Command\_Flag** { **EFREET\_DESKTOP\_EXEC\_FLAG\_FULLPATH** = 0x0001, **EFREET\_DESKTOP\_EXEC\_FLAG\_URI** = 0x0002, **EFREET\_DESKTOP\_EXEC\_FLAG\_DIR** = 0x0004, **EFREET\_DESKTOP\_EXEC\_FLAG\_FILE** = 0x0008 }

## Functions

- size\_t **efreet\_array\_cat** (char \*buffer, size\_t size, const char \*strs[])
- int **efreet\_base\_init** (void)
- void **efreet\_base\_shutdown** (void)
- Ecore\_List \* **efreet\_default\_dirs\_get** (const char \*user\_dir, Ecore\_List \*system\_dirs, const char \*suffix)

*Creates the list of directories based on the user dir, system dirs and given suffix.*

- int **efreet\_desktop\_init** (void)
- int **efreet\_desktop\_shutdown** (void)
- const char \* **efreet\_home\_dir\_get** (void)
- int **efreet\_icon\_init** (void)
- void **efreet\_icon\_shutdown** (void)
- int **efreet\_ini\_init** (void)
- int **efreet\_ini\_shutdown** (void)
- const char \* **efreet\_lang\_country\_get** (void)
- const char \* **efreet\_lang\_get** (void)
- const char \* **efreet\_lang\_modifier\_get** (void)
- int **efreet\_menu\_init** (void)

*Initializes the Efreet Menu system.*

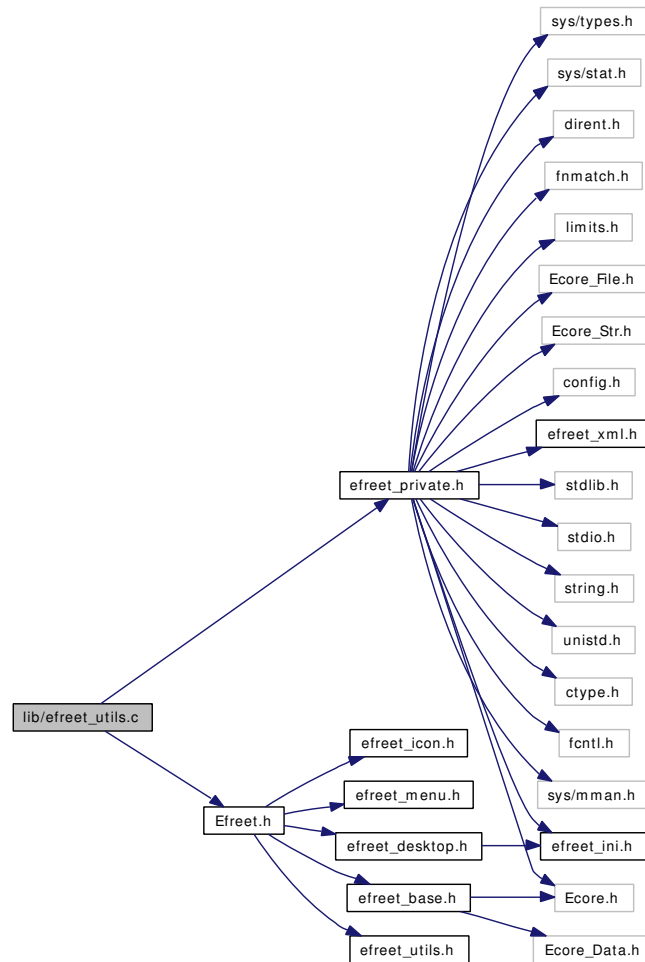
- void **efreet\_\_menu\_shutdown** (void)  
*Shuts down the Efreet menu system.*
- int **efreet\_\_util\_init** (void)
- void **efreet\_\_util\_shutdown** (void)

## 7.14 lib/efreet\_utils.c File Reference

```
#include "Efreet.h"
```

```
#include "efreet_private.h"
```

Include dependency graph for efreet\_utils.c:



### Data Structures

- struct `_Efreet_Cache_Fill`
- struct `_Efreet_Cache_Fill_Dir`
- struct `_Efreet_Cache_Search`
- struct `_Efreet_Cache_Search_List`

### Typedefs

- typedef `_Efreet_Cache_Fill Efreet_Cache_Fill`
- typedef `_Efreet_Cache_Fill_Dir Efreet_Cache_Fill_Dir`
- typedef `_Efreet_Cache_Search Efreet_Cache_Search`
- typedef `_Efreet_Cache_Search_List Efreet_Cache_Search_List`

## Functions

- `Ecore_List * efreet_util_desktop_comment_glob_list (const char *glob)`
- `Efreet_Desktop * efreet_util_desktop_exec_find (const char *exec)`
- `Ecore_List * efreet_util_desktop_exec_glob_list (const char *glob)`
- `Efreet_Desktop * efreet_util_desktop_file_id_find (const char *file_id)`
- `Efreet_Desktop * efreet_util_desktop_generic_name_find (const char *generic_name)`
- `Ecore_List * efreet_util_desktop_generic_name_glob_list (const char *glob)`
- `Ecore_List * efreet_util_desktop_mime_list (const char *mime)`
- `Efreet_Desktop * efreet_util_desktop_name_find (const char *name)`
- `Ecore_List * efreet_util_desktop_name_glob_list (const char *glob)`
- `Efreet_Desktop * efreet_util_desktop_wm_class_find (const char *wmname, const char *wmclass)`
- `int efreet_util_init (void)`
- `char * efreet_util_path_in_default (const char *section, const char *path)`
- `const char * efreet_util_path_to_file_id (const char *path)`
- `void efreet_util_shutdown (void)`

## 7.14.1 Typedef Documentation

7.14.1.1 typedef struct \_Efreet\_Cache\_Fill Efreet\_Cache\_Fill

7.14.1.2 typedef struct \_Efreet\_Cache\_Fill\_Dir Efreet\_Cache\_Fill\_Dir

7.14.1.3 typedef struct \_Efreet\_Cache\_Search Efreet\_Cache\_Search

7.14.1.4 typedef struct \_Efreet\_Cache\_Search\_List Efreet\_Cache\_Search\_List

## 7.14.2 Function Documentation

7.14.2.1 Ecore\_List\* efreet\_util\_desktop\_comment\_glob\_list (const char \* *glob*)

7.14.2.2 Efreet\_Desktop\* efreet\_util\_desktop\_exec\_find (const char \* *exec*)

7.14.2.3 Ecore\_List\* efreet\_util\_desktop\_exec\_glob\_list (const char \* *glob*)

7.14.2.4 Efreet\_Desktop\* efreet\_util\_desktop\_file\_id\_find (const char \* *file\_id*)

7.14.2.5 Efreet\_Desktop\* efreet\_util\_desktop\_generic\_name\_find (const char \* *generic\_name*)

7.14.2.6 Ecore\_List\* efreet\_util\_desktop\_generic\_name\_glob\_list (const char \* *glob*)

7.14.2.7 Ecore\_List\* efreet\_util\_desktop\_mime\_list (const char \* *mime*)

7.14.2.8 Efreet\_Desktop\* efreet\_util\_desktop\_name\_find (const char \* *name*)

7.14.2.9 Ecore\_List\* efreet\_util\_desktop\_name\_glob\_list (const char \* *glob*)

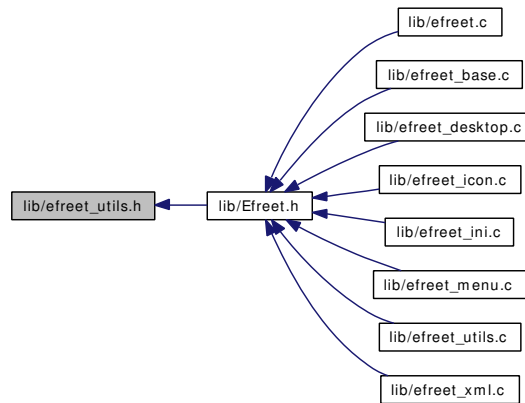
7.14.2.10 Efreet\_Desktop\* efreet\_util\_desktop\_wm\_class\_find (const char \* *wmname*, const char \* *wmclass*)

7.14.2.11 char\* efreet\_util\_path\_in\_default (const char \* *section*, const char \* *path*)

7.14.2.12 const char\* efreet\_util\_path\_to\_file\_id (const char \* *path*)

## 7.15 lib/efreet\_utils.h File Reference

This graph shows which files directly or indirectly include this file:



## Functions

- `Ecore_List * efreet_util_desktop_comment_glob_list (const char *glob)`
- `Efreet_Desktop * efreet_util_desktop_exec_find (const char *exec)`
- `Ecore_List * efreet_util_desktop_exec_glob_list (const char *glob)`
- `Efreet_Desktop * efreet_util_desktop_file_id_find (const char *file_id)`
- `Efreet_Desktop * efreet_util_desktop_generic_name_find (const char *generic_name)`
- `Ecore_List * efreet_util_desktop_generic_name_glob_list (const char *glob)`
- `Ecore_List * efreet_util_desktop_mime_list (const char *mime)`
- `Efreet_Desktop * efreet_util_desktop_name_find (const char *name)`
- `Ecore_List * efreet_util_desktop_name_glob_list (const char *glob)`
- `Efreet_Desktop * efreet_util_desktop_wm_class_find (const char *wmname, const char *wmclass)`
- `char * efreet_util_path_in_default (const char *section, const char *path)`
- `const char * efreet_util_path_to_file_id (const char *path)`

## 7.15.1 Function Documentation

7.15.1.1 `Ecore_List* efreet_util_desktop_comment_glob_list (const char * glob)`

7.15.1.2 `Efreedom* efreet_util_desktop_exec_find (const char * exec)`

7.15.1.3 `Ecore_List* efreet_util_desktop_exec_glob_list (const char * glob)`

7.15.1.4 `Efreedom* efreet_util_desktop_file_id_find (const char * file_id)`

7.15.1.5 `Efreedom* efreet_util_desktop_generic_name_find (const char * generic_name)`

7.15.1.6 `Ecore_List* efreet_util_desktop_generic_name_glob_list (const char * glob)`

7.15.1.7 `Ecore_List* efreet_util_desktop_mime_list (const char * mime)`

7.15.1.8 `Efreedom* efreet_util_desktop_name_find (const char * name)`

7.15.1.9 `Ecore_List* efreet_util_desktop_name_glob_list (const char * glob)`

7.15.1.10 `Efreedom* efreet_util_desktop_wm_class_find (const char * wmname, const char * wmclass)`

7.15.1.11 `char* efreet_util_path_in_default (const char * section, const char * path)`

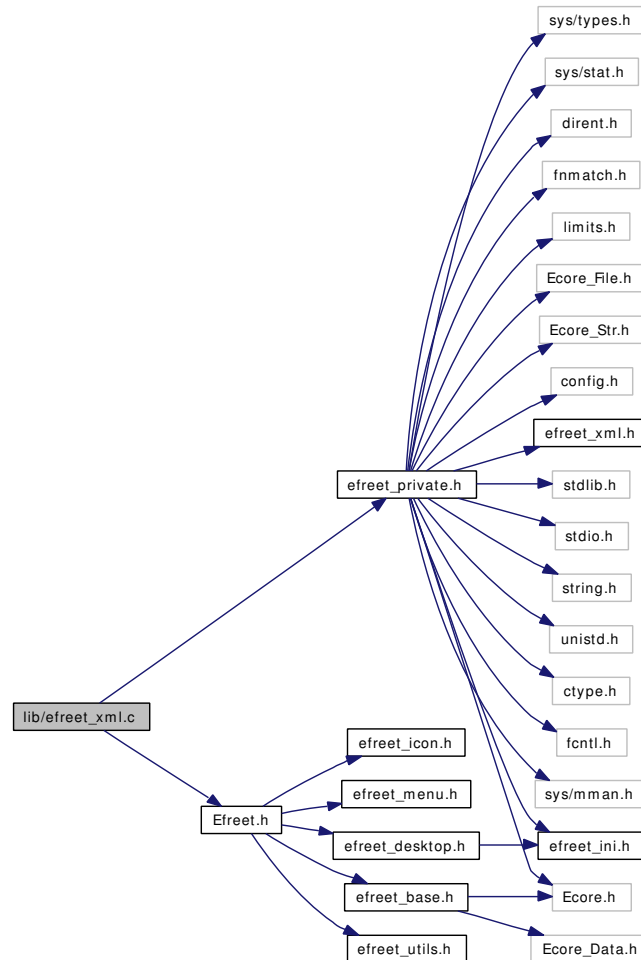
7.15.1.12 `const char* efreet_util_path_to_file_id (const char * path)`

## 7.16 lib/efreet\_xml.c File Reference

```
#include "Efreet.h"
```

```
#include "efreet_private.h"
```

Include dependency graph for efreet\_xml.c:



## Functions

- `const char * efreet_xml_attribute_get (Efreet_Xml *xml, const char *key)`

*Retrieves the value for the given attribute key.*

- `void efreet_xml_del (Efreet_Xml *xml)`
- `int efreet_xml_init (void)`
- `Efreet_Xml * efreet_xml_new (const char *file)`
- `int efreet_xml_shutdown (void)`



## 7.16.1 Function Documentation

### 7.16.1.1 `const char* efreet_xml_attribute_get (Efreet_Xml * xml, const char * key)`

Retrieves the value for the given attribute key.

#### Parameters:

*xml*,: The xml struct to work with

*key*,: The attribute key to look for

#### Returns:

Returns the value for the given key, or NULL if none found

### 7.16.1.2 `void efreet_xml_del (Efreet_Xml * xml)`

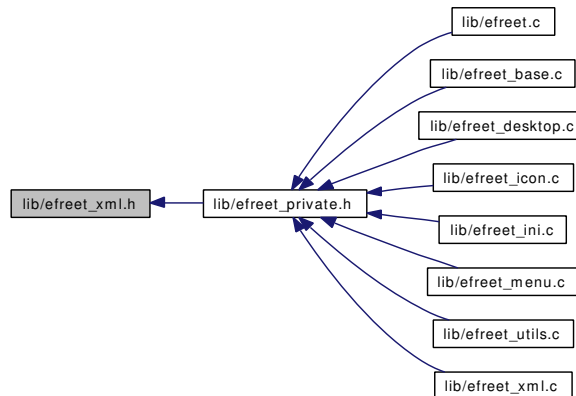
### 7.16.1.3 `int efreet_xml_init (void)`

### 7.16.1.4 `Efreet_Xml* efreet_xml_new (const char * file)`

### 7.16.1.5 `int efreet_xml_shutdown (void)`

## 7.17 lib/efreet\_xml.h File Reference

This graph shows which files directly or indirectly include this file:



### Data Structures

- struct **Efreet\_Xml**  
*Contains the XML tree for a given XML document.*
- struct **Efreet\_Xml\_Attribute**  
*Contains information about a given XML attribute.*

### Typedefs

- typedef **Efreet\_Xml** **Efreet\_Xml**
- typedef **Efreet\_Xml\_Attribute** **Efreet\_Xml\_Attribute**

### Functions

- const char \* **efreet\_xml\_attribute\_get** (**Efreet\_Xml** \*xml, const char \*key)  
*Retrieves the value for the given attribute key.*
- void **efreet\_xml\_del** (**Efreet\_Xml** \*xml)
- int **efreet\_xml\_init** (void)
- **Efreet\_Xml** \* **efreet\_xml\_new** (const char \*file)
- int **efreet\_xml\_shutdown** (void)

#### 7.17.1 Typedef Documentation

##### 7.17.1.1 typedef struct Efreet\_Xml Efreet\_Xml

**Efreet\_Xml** (p. 60)

### 7.17.1.2 typedef struct Efreet\_Xml\_Attribute Efreet\_Xml\_Attribute

Efreet\_Xml\_Attributes

## 7.17.2 Function Documentation

### 7.17.2.1 const char\* efreet\_xml\_attribute\_get (Efreet\_Xml \* *xml*, const char \* *key*)

Retrieves the value for the given attribute key.

#### Parameters:

*xml*,: The xml struct to work with

*key*,: The attribute key to look for

#### Returns:

Returns the value for the given key, or NULL if none found

### 7.17.2.2 void efreet\_xml\_del (Efreet\_Xml \* *xml*)

### 7.17.2.3 int efreet\_xml\_init (void)

### 7.17.2.4 Efreet\_Xml\* efreet\_xml\_new (const char \* *file*)

### 7.17.2.5 int efreet\_xml\_shutdown (void)

# Index

- `_Efreet_Cache_Fill`, 29
    - current, 29
    - dirs, 29
    - files, 29
  - `_Efreet_Cache_Fill_Dir`, 30
    - file\_id, 30
    - path, 30
  - `_Efreet_Cache_Search`, 31
    - what1, 31
    - what2, 31
  - `_Efreet_Cache_Search_List`, 32
    - list, 32
    - what, 32
  - `__UNUSED__`
    - `Efreet_Private`, 25
- all
  - `Efreet_Menu_Filter_Op`, 52
- allocated
  - `Efreet_Menu_Desktop`, 50
- app\_dirs
  - `Efreet_Menu_Internal`, 54
- app\_pool
  - `Efreet_Menu_Internal`, 54
- applications
  - `Efreet_Menu_Internal`, 54
- attach\_points
  - `Efreet_Icon`, 39
- attributes
  - `Efreet_Xml`, 60
- categories
  - `Efreet_Desktop`, 33
  - `Efreet_Menu_Filter_Op`, 52
- cb\_command
  - `Efreet_Desktop_Command`, 36
- cb\_progress
  - `Efreet_Desktop_Command`, 36
- children
  - `Efreet_Xml`, 60
- command
  - `Efreet_Desktop_Command_File`, 37
- comment
  - `Efreet_Desktop`, 33
  - `Efreet_Icon_Theme`, 42
- context
  - `Efreet_Icon_Theme_Directory`, 44
- count
  - `Efreet_Icon_Theme`, 42
- current
  - `_Efreet_Cache_Fill`, 29
- current\_move
  - `Efreet_Menu_Internal`, 54
- data
  - `Efreet_Desktop_Command`, 36
  - `Efreet_Ini`, 46
- default\_layout
  - `Efreet_Menu_Internal`, 54
- deleted
  - `Efreet_Menu_Internal`, 54
- desktop
  - `Efreet_Desktop_Command`, 36
  - `Efreet_Menu`, 47
  - `Efreet_Menu_Desktop`, 50
- DESKTOP\_VERSION
  - `efreet_desktop.c`, 73
- dir
  - `Efreet_Desktop_Command_File`, 37
- directories
  - `Efreet_Icon_Theme`, 42
  - `Efreet_Menu_Internal`, 54
- directory
  - `Efreet_Menu_Internal`, 54
- directory\_cache
  - `Efreet_Menu_Internal`, 54
- directory\_dirs
  - `Efreet_Menu_Internal`, 54
- dirs
  - `_Efreet_Cache_Fill`, 29
- efreet.c
  - `efreet_init`, 64
  - `efreet_shutdown`, 65
- Efreet.h
  - `efreet_init`, 67
  - `efreet_shutdown`, 67
- efreet\_array\_cat
  - `Efreet_Private`, 26
- `Efreet_Base`

- efreet\_cache\_home\_get, 9
- efreet\_config\_dirs\_get, 9
- efreet\_config\_home\_get, 10
- efreet\_data\_dirs\_get, 10
- efreet\_data\_home\_get, 10
- Efreet\_Base: The XDG Base Directory Specification, 9
- efreet\_base\_init
  - Efreet\_Private, 26
- efreet\_base\_shutdown
  - Efreet\_Private, 26
- Efreet\_Cache\_Fill
  - efreet\_utils.c, 97
- Efreet\_Cache\_Fill\_Dir
  - efreet\_utils.c, 97
- efreet\_cache\_home\_get
  - Efreet\_Base, 9
- Efreet\_Cache\_Search
  - efreet\_utils.c, 97
- Efreet\_Cache\_Search\_List
  - efreet\_utils.c, 97
- efreet\_config\_dirs\_get
  - Efreet\_Base, 9
- efreet\_config\_home\_get
  - Efreet\_Base, 10
- efreet\_data\_dirs\_get
  - Efreet\_Base, 10
- efreet\_data\_home\_get
  - Efreet\_Base, 10
- efreet\_default\_dirs\_get
  - Efreet\_Private, 26
- Efreet\_Desktop, 33
  - categories, 33
  - comment, 33
  - Efreet\_Desktop, 12
  - efreet\_desktop\_category\_add, 13
  - efreet\_desktop\_category\_count\_get, 13
  - efreet\_desktop\_category\_del, 13
  - Efreet\_Desktop\_Command\_Cb, 12
  - efreet\_desktop\_command\_get, 14
  - efreet\_desktop\_command\_progress\_get, 14
  - efreet\_desktop\_empty\_new, 14
  - efreet\_desktop\_environment\_set, 14
  - efreet\_desktop\_exec, 15
  - efreet\_desktop\_free, 15
  - efreet\_desktop\_get, 15
  - Efreet\_Desktop\_Progress\_Cb, 12
  - efreet\_desktop\_save, 15
  - efreet\_desktop\_save\_as, 15
  - Efreet\_Desktop\_Type, 12
  - efreet\_desktop\_type\_add, 15
  - EFREET\_DESKTOP\_TYPE\_APPLICATION, 16
  - efreet\_desktop\_type\_data\_get, 16
  - EFREET\_DESKTOP\_TYPE\_DIRECTORY, 16
  - Efreet\_Desktop\_Type\_Free\_Cb, 13
  - EFREET\_DESKTOP\_TYPE\_LINK, 16
  - Efreet\_Desktop\_Type\_Parse\_Cb, 13
  - Efreet\_Desktop\_Type\_Save\_Cb, 13
  - exec, 33
  - generic\_name, 33
  - hidden, 34
  - icon, 34
  - load\_time, 34
  - mime\_types, 34
  - name, 34
  - no\_display, 34
  - not\_show\_in, 34
  - only\_show\_in, 34
  - orig\_path, 34
  - path, 34
  - startup\_notify, 34
  - startup\_wm\_class, 34
  - terminal, 35
  - try\_exec, 35
  - type, 35
  - type\_data, 35
  - url, 35
  - version, 35
  - x, 35
- efreet\_desktop.c
  - DESKTOP\_VERSION, 73
  - Efreet\_Desktop\_Type\_Info, 73
- Efreet\_Desktop: The FDO Desktop Entry, 11
- efreet\_desktop\_category\_add
  - Efreet\_Desktop, 13
- efreet\_desktop\_category\_count\_get
  - Efreet\_Desktop, 13
- efreet\_desktop\_category\_del
  - Efreet\_Desktop, 13
- Efreet\_Desktop\_Command, 36
  - cb\_command, 36
  - cb\_progress, 36
  - data, 36
  - desktop, 36
  - Efreet\_Private, 26
  - files, 36
  - flags, 36
  - num\_pending, 36
- Efreet\_Desktop\_Command\_Cb
  - Efreet\_Desktop, 12
- Efreet\_Desktop\_Command\_File, 37
  - command, 37
  - dir, 37
  - Efreet\_Private, 26
  - file, 37

- fullpath, 37
- pending, 37
- uri, 37
- Efreedom\_Desktop\_Command\_Flag
  - Efreedom\_Private, 26
- efreet\_desktop\_command\_get
  - Efreedom\_Desktop, 14
- efreet\_desktop\_command\_progress\_get
  - Efreedom\_Desktop, 14
- efreet\_desktop\_empty\_new
  - Efreedom\_Desktop, 14
- efreet\_desktop\_environment\_set
  - Efreedom\_Desktop, 14
- efreet\_desktop\_exec
  - Efreedom\_Desktop, 15
- EFREET\_DESKTOP\_EXEC\_FLAG\_DIR
  - Efreedom\_Private, 26
- EFREET\_DESKTOP\_EXEC\_FLAG\_FILE
  - Efreedom\_Private, 26
- EFREET\_DESKTOP\_EXEC\_FLAG\_-FULLPATH
  - Efreedom\_Private, 26
- EFREET\_DESKTOP\_EXEC\_FLAG\_URI
  - Efreedom\_Private, 26
- efreet\_desktop\_free
  - Efreedom\_Desktop, 15
- efreet\_desktop\_get
  - Efreedom\_Desktop, 15
- efreet\_desktop\_init
  - Efreedom\_Private, 26
- Efreedom\_Desktop\_Progress\_Cb
  - Efreedom\_Desktop, 12
- efreet\_desktop\_save
  - Efreedom\_Desktop, 15
- efreet\_desktop\_save\_as
  - Efreedom\_Desktop, 15
- efreet\_desktop\_shutdown
  - Efreedom\_Private, 27
- Efreedom\_Desktop\_Type
  - Efreedom\_Desktop, 12
- efreet\_desktop\_type\_add
  - Efreedom\_Desktop, 15
- EFREET\_DESKTOP\_TYPE\_-APPLICATION
  - Efreedom\_Desktop, 16
- efreet\_desktop\_type\_data\_get
  - Efreedom\_Desktop, 16
- EFREET\_DESKTOP\_TYPE\_DIRECTORY
  - Efreedom\_Desktop, 16
- Efreedom\_Desktop\_Type\_Free\_Cb
  - Efreedom\_Desktop, 13
- Efreedom\_Desktop\_Type\_Info, 38
  - efreet\_desktop.c, 73
- free\_func, 38
- id, 38
- parse\_func, 38
- save\_func, 38
- type, 38
- EFREET\_DESKTOP\_TYPE\_LINK
  - Efreedom\_Desktop, 16
- Efreedom\_Desktop\_Type\_Parse\_Cb
  - Efreedom\_Desktop, 13
- Efreedom\_Desktop\_Type\_Save\_Cb
  - Efreedom\_Desktop, 13
- efreet\_home\_dir\_get
  - Efreedom\_Private, 27
- Efreedom\_Icon, 39
  - attach\_points, 39
  - Efreedom\_Icon, 18
  - efreet\_icon\_dir\_get, 19
  - efreet\_icon\_extension\_add, 19
  - efreet\_icon\_find, 19
  - efreet\_icon\_path\_find, 20
  - Efreedom\_Icon\_Point, 18
  - Efreedom\_Icon\_Size\_Type, 18, 19
  - EFREET\_ICON\_SIZE\_TYPE\_FIXED, 19
  - EFREET\_ICON\_SIZE\_TYPE\_NONE, 19
  - EFREET\_ICON\_SIZE\_TYPE\_-SCALABLE, 19
  - EFREET\_ICON\_SIZE\_TYPE\_-THRESHOLD, 19
  - Efreedom\_Icon\_Theme, 18
  - Efreedom\_Icon\_Theme\_Context, 18, 19
  - EFREET\_ICON\_THEME\_-CONTEXT\_ACTIONS, 19
  - EFREET\_ICON\_THEME\_-CONTEXT\_DEVICES, 19
  - EFREET\_ICON\_THEME\_-CONTEXT\_FILESYSTEMS, 19
  - EFREET\_ICON\_THEME\_-CONTEXT\_MIMETYPES, 19
  - EFREET\_ICON\_THEME\_-CONTEXT\_NONE, 19
  - Efreedom\_Icon\_Theme\_Directory, 18
  - efreet\_icon\_theme\_find, 20
  - efreet\_icon\_theme\_list\_get, 20
  - embedded\_text\_rectangle, 39
  - has\_embedded\_text\_rectangle, 39
  - name, 39
  - path, 39
  - ref\_count, 39
  - x0, 40
  - x1, 40
  - y0, 40
  - y1, 40
- efreet\_icon.c

- efreet\_icon\_extensions, 78
- NO\_MATCH\_KEY, 78
- Efreedom\_Icon: The FDO Icon Theme, 17
- efreet\_icon\_dir\_get
  - Efreedom\_Icon, 19
- efreet\_icon\_extension\_add
  - Efreedom\_Icon, 19
- efreet\_icon\_extensions
  - efreet\_icon.c, 78
- efreet\_icon\_find
  - Efreedom\_Icon, 19
- efreet\_icon\_init
  - Efreedom\_Private, 27
- efreet\_icon\_path\_find
  - Efreedom\_Icon, 20
- Efreedom\_Icon\_Point, 41
  - Efreedom\_Icon, 18
  - x, 41
  - y, 41
- efreet\_icon\_shutdown
  - Efreedom\_Private, 27
- Efreedom\_Icon\_Size\_Type
  - Efreedom\_Icon, 18, 19
- EFREET\_ICON\_SIZE\_TYPE\_FIXED
  - Efreedom\_Icon, 19
- EFREET\_ICON\_SIZE\_TYPE\_NONE
  - Efreedom\_Icon, 19
- EFREET\_ICON\_SIZE\_TYPE\_SCALABLE
  - Efreedom\_Icon, 19
- EFREET\_ICON\_SIZE\_TYPE\_-THRESHOLD
  - Efreedom\_Icon, 19
- Efreedom\_Icon\_Theme, 42
  - comment, 42
  - count, 42
  - directories, 42
  - Efreedom\_Icon, 18
  - example\_icon, 42
  - fake, 42
  - hidden, 43
  - icon\_cache, 43
  - inherits, 43
  - internal, 43
  - last\_cache\_check, 43
  - name, 43
  - path, 43
  - paths, 43
  - valid, 43
- Efreedom\_Icon\_Theme\_Context
  - Efreedom\_Icon, 18, 19
- EFREET\_ICON\_THEME\_CONTEXT\_-ACTIONS
  - Efreedom\_Icon, 19
- EFREET\_ICON\_THEME\_CONTEXT\_-DEVICES
  - Efreedom\_Icon, 19
- EFREET\_ICON\_THEME\_CONTEXT\_-FILESYSTEMS
  - Efreedom\_Icon, 19
- EFREET\_ICON\_THEME\_CONTEXT\_-MIMETYPES
  - Efreedom\_Icon, 19
- EFREET\_ICON\_THEME\_CONTEXT\_-NONE
  - Efreedom\_Icon, 19
- Efreedom\_Icon\_Theme\_Directory, 44
  - context, 44
  - Efreedom\_Icon, 18
  - max, 44
  - min, 44
  - name, 44
  - normal, 44
  - size, 44
  - threshold, 44
  - type, 45
- efreet\_icon\_theme\_find
  - Efreedom\_Icon, 20
- efreet\_icon\_theme\_list\_get
  - Efreedom\_Icon, 20
- Efreedom\_Ini, 46
  - data, 46
  - efreet\_ini.h, 84
  - section, 46
- efreet\_ini.c
  - efreet\_ini\_boolean\_get, 82
  - efreet\_ini\_boolean\_set, 82
  - efreet\_ini\_double\_get, 82
  - efreet\_ini\_double\_set, 82
  - efreet\_ini\_free, 82
  - efreet\_ini\_int\_get, 82
  - efreet\_ini\_int\_set, 82
  - efreet\_ini\_localestring\_get, 82
  - efreet\_ini\_localestring\_set, 82
  - efreet\_ini\_new, 82
  - efreet\_ini\_save, 82
  - efreet\_ini\_section\_add, 82
  - efreet\_ini\_section\_set, 82
  - efreet\_ini\_string\_get, 82
  - efreet\_ini\_string\_set, 82
- efreet\_ini.h
  - Efreedom\_Ini, 84
  - efreet\_ini\_boolean\_get, 84
  - efreet\_ini\_boolean\_set, 84
  - efreet\_ini\_double\_get, 84
  - efreet\_ini\_double\_set, 84
  - efreet\_ini\_free, 84
  - efreet\_ini\_int\_get, 84

- efreet\_ini\_int\_set, 84
- efreet\_ini\_localestring\_get, 84
- efreet\_ini\_localestring\_set, 84
- efreet\_ini\_new, 84
- efreet\_ini\_save, 84
- efreet\_ini\_section\_add, 84
- efreet\_ini\_section\_set, 84
- efreet\_ini\_string\_get, 84
- efreet\_ini\_string\_set, 84
- efreet\_ini\_boolean\_get
  - efreet\_ini.c, 82
  - efreet\_ini.h, 84
- efreet\_ini\_boolean\_set
  - efreet\_ini.c, 82
  - efreet\_ini.h, 84
- efreet\_ini\_double\_get
  - efreet\_ini.c, 82
  - efreet\_ini.h, 84
- efreet\_ini\_double\_set
  - efreet\_ini.c, 82
  - efreet\_ini.h, 84
- efreet\_ini\_free
  - efreet\_ini.c, 82
  - efreet\_ini.h, 84
- efreet\_ini\_init
  - Efreet\_Private, 27
- efreet\_ini\_int\_get
  - efreet\_ini.c, 82
  - efreet\_ini.h, 84
- efreet\_ini\_int\_set
  - efreet\_ini.c, 82
  - efreet\_ini.h, 84
- efreet\_ini\_localestring\_get
  - efreet\_ini.c, 82
  - efreet\_ini.h, 84
- efreet\_ini\_localestring\_set
  - efreet\_ini.c, 82
  - efreet\_ini.h, 84
- efreet\_ini\_new
  - efreet\_ini.c, 82
  - efreet\_ini.h, 84
- efreet\_ini\_save
  - efreet\_ini.c, 82
  - efreet\_ini.h, 84
- efreet\_ini\_section\_add
  - efreet\_ini.c, 82
  - efreet\_ini.h, 84
- efreet\_ini\_section\_set
  - efreet\_ini.c, 82
  - efreet\_ini.h, 84
- efreet\_ini\_shutdown
  - Efreet\_Private, 27
- efreet\_ini\_string\_get
  - efreet\_ini.c, 82
- efreet\_ini.h, 84
- efreet\_ini\_string\_set
  - efreet\_ini.c, 82
- efreet\_init
  - efreet.c, 64
  - Efreet.h, 67
- efreet\_lang\_country\_get
  - Efreet\_Private, 27
- efreet\_lang\_get
  - Efreet\_Private, 27
- efreet\_lang\_modifier\_get
  - Efreet\_Private, 27
- Efreet\_Menu, 47
  - desktop, 47
  - Efreet\_Menu, 21
  - efreet\_menu\_desktop\_insert, 22
  - efreet\_menu\_dump, 22
  - EFREET\_MENU\_ENTRY\_DESKTOP, 22
  - EFREET\_MENU\_ENTRY\_HEADER, 22
  - EFREET\_MENU\_ENTRY\_MENU, 22
  - EFREET\_MENU\_ENTRY\_SEPARATOR, 22
  - Efreet\_Menu\_Entry\_Type, 21, 22
  - efreet\_menu\_free, 22
  - efreet\_menu\_get, 23
  - efreet\_menu\_kde\_legacy\_init, 23
  - efreet\_menu\_parse, 23
  - efreet\_menu\_save, 23
  - entries, 47
  - icon, 47
  - id, 47
  - name, 47
  - type, 47
- efreet\_menu.c
  - Efreet\_Menu\_App\_Dir, 87
  - Efreet\_Menu\_Desktop, 87
  - Efreet\_Menu\_Filter, 87
  - EFREET\_MENU\_FILTER\_EXCLUDE, 88
  - EFREET\_MENU\_FILTER\_INCLUDE, 88
  - Efreet\_Menu\_Filter\_Op, 87
  - EFREET\_MENU\_FILTER\_OP\_AND, 88
  - EFREET\_MENU\_FILTER\_OP\_NOT, 88
  - EFREET\_MENU\_FILTER\_OP\_OR, 88
  - Efreet\_Menu\_Filter\_Op\_Type, 87, 88
  - Efreet\_Menu\_Filter\_Type, 87, 88
  - Efreet\_Menu\_Internal, 87
  - efreet\_menu\_kde\_legacy\_dirs, 88



- Efreet\_Menu\_Layout, 87
- EFREET\_MENU\_LAYOUT\_-
  - FILENAME, 88
- EFREET\_MENU\_LAYOUT\_-
  - MENUNAME, 88
- EFREET\_MENU\_LAYOUT\_MERGE, 88
- EFREET\_MENU\_LAYOUT\_-
  - SEPARATOR, 88
- Efreet\_Menu\_Layout\_Type, 87, 88
- Efreet\_Menu\_Move, 87
- Efreet\_Menu: The FDO Desktop Menu Specification, 21
- Efreet\_Menu\_App\_Dir, 49
  - efreet\_menu.c, 87
  - legacy, 49
  - path, 49
  - prefix, 49
- Efreet\_Menu\_Desktop, 50
  - allocated, 50
  - desktop, 50
  - efreet\_menu.c, 87
  - id, 50
- efreet\_menu\_desktop\_insert
  - Efreet\_Menu, 22
- efreet\_menu\_dump
  - Efreet\_Menu, 22
- EFREET\_MENU\_ENTRY\_DESKTOP
  - Efreet\_Menu, 22
- EFREET\_MENU\_ENTRY\_HEADER
  - Efreet\_Menu, 22
- EFREET\_MENU\_ENTRY\_MENU
  - Efreet\_Menu, 22
- EFREET\_MENU\_ENTRY\_SEPARATOR
  - Efreet\_Menu, 22
- Efreet\_Menu\_Entry\_Type
  - Efreet\_Menu, 21, 22
- Efreet\_Menu\_Filter, 51
  - efreet\_menu.c, 87
  - op, 51
  - type, 51
- EFREET\_MENU\_FILTER\_EXCLUDE
  - efreet\_menu.c, 88
- EFREET\_MENU\_FILTER\_INCLUDE
  - efreet\_menu.c, 88
- Efreet\_Menu\_Filter\_Op, 52
  - all, 52
  - categories, 52
  - efreet\_menu.c, 87
  - filenames, 52
  - filters, 52
  - type, 52
- EFREET\_MENU\_FILTER\_OP\_AND
  - efreet\_menu.c, 88
- EFREET\_MENU\_FILTER\_OP\_NOT
  - efreet\_menu.c, 88
- EFREET\_MENU\_FILTER\_OP\_OR
  - efreet\_menu.c, 88
- Efreet\_Menu\_Filter\_Op\_Type
  - efreet\_menu.c, 87, 88
- Efreet\_Menu\_Filter\_Type
  - efreet\_menu.c, 87, 88
- efreet\_menu\_free
  - Efreet\_Menu, 22
- efreet\_menu\_get
  - Efreet\_Menu, 23
- efreet\_menu\_init
  - Efreet\_Private, 27
- Efreet\_Menu\_Internal, 53
  - app\_dirs, 54
  - app\_pool, 54
  - applications, 54
  - current\_move, 54
  - default\_layout, 54
  - deleted, 54
  - directories, 54
  - directory, 54
  - directory\_cache, 54
  - directory\_dirs, 54
  - efreet\_menu.c, 87
  - file, 54
  - filters, 54
  - in\_line, 55
  - inline\_alias, 55
  - inline\_header, 55
  - inline\_limit, 55
  - internal, 55
  - layout, 55
  - moves, 55
  - name, 55
  - only\_unallocated, 55
  - parent, 55
  - path, 56
  - seen\_allocated, 56
  - seen\_deleted, 56
  - show\_empty, 56
  - sub\_menus, 56
- efreet\_menu\_kde\_legacy\_dirs
  - efreet\_menu.c, 88
- efreet\_menu\_kde\_legacy\_init
  - Efreet\_Menu, 23
- Efreet\_Menu\_Layout, 57
  - efreet\_menu.c, 87
  - in\_line, 57
  - inline\_alias, 57
  - inline\_header, 57
  - inline\_limit, 57
  - name, 57

- show\_empty, 57
- type, 57
- EFREET\_MENU\_LAYOUT\_FILENAME
  - efreet\_menu.c, 88
- EFREET\_MENU\_LAYOUT\_MENUNAME
  - efreet\_menu.c, 88
- EFREET\_MENU\_LAYOUT\_MERGE
  - efreet\_menu.c, 88
- EFREET\_MENU\_LAYOUT\_SEPARATOR
  - efreet\_menu.c, 88
- Efreedom\_Menu\_Layout\_Type
  - efreet\_menu.c, 87, 88
- Efreedom\_Menu\_Move, 59
  - efreet\_menu.c, 87
  - new\_name, 59
  - old\_name, 59
- efreet\_menu\_parse
  - Efreedom\_Menu, 23
- efreet\_menu\_save
  - Efreedom\_Menu, 23
- efreet\_menu\_shutdown
  - Efreedom\_Private, 27
- Efreedom\_Private
  - \_\_UNUSED\_\_, 25
  - efreet\_array\_cat, 26
  - efreet\_base\_init, 26
  - efreet\_base\_shutdown, 26
  - efreet\_default\_dirs\_get, 26
  - Efreedom\_Desktop\_Command, 26
  - Efreedom\_Desktop\_Command\_File, 26
  - Efreedom\_Desktop\_Command\_Flag, 26
  - EFREET\_DESKTOP\_EXEC\_FLAG\_-DIR, 26
  - EFREET\_DESKTOP\_EXEC\_FLAG\_-FILE, 26
  - EFREET\_DESKTOP\_EXEC\_FLAG\_-FULLPATH, 26
  - EFREET\_DESKTOP\_EXEC\_FLAG\_-URI, 26
  - efreet\_desktop\_init, 26
  - efreet\_desktop\_shutdown, 27
  - efreet\_home\_dir\_get, 27
  - efreet\_icon\_init, 27
  - efreet\_icon\_shutdown, 27
  - efreet\_ini\_init, 27
  - efreet\_ini\_shutdown, 27
  - efreet\_lang\_country\_get, 27
  - efreet\_lang\_get, 27
  - efreet\_lang\_modifier\_get, 27
  - efreet\_menu\_init, 27
  - efreet\_menu\_shutdown, 27
  - efreet\_util\_init, 27
  - efreet\_util\_shutdown, 27
  - FREE, 25
  - IF\_FREE, 25
  - IF\_FREE\_DLIST, 25
  - IF\_FREE\_HASH, 25
  - IF\_FREE\_LIST, 25
  - IF\_RELEASE, 25
  - NEW, 25
  - PATH\_MAX, 25
  - Efreedom\_Private: Private methods and defines, 24
  - efreet\_shutdown
    - efreet.c, 65
    - Efreedom.h, 67
  - efreet\_util\_desktop\_comment\_glob\_list
    - efreet\_utils.c, 97
    - efreet\_utils.h, 99
  - efreet\_util\_desktop\_exec\_find
    - efreet\_utils.c, 97
    - efreet\_utils.h, 99
  - efreet\_util\_desktop\_exec\_glob\_list
    - efreet\_utils.c, 97
    - efreet\_utils.h, 99
  - efreet\_util\_desktop\_file\_id\_find
    - efreet\_utils.c, 97
    - efreet\_utils.h, 99
  - efreet\_util\_desktop\_generic\_name\_find
    - efreet\_utils.c, 97
    - efreet\_utils.h, 99
  - efreet\_util\_desktop\_generic\_name\_glob\_list
    - efreet\_utils.c, 97
    - efreet\_utils.h, 99
  - efreet\_util\_desktop\_mime\_list
    - efreet\_utils.c, 97
    - efreet\_utils.h, 99
  - efreet\_util\_desktop\_name\_find
    - efreet\_utils.c, 97
    - efreet\_utils.h, 99
  - efreet\_util\_desktop\_name\_glob\_list
    - efreet\_utils.c, 97
    - efreet\_utils.h, 99
  - efreet\_util\_desktop\_wm\_class\_find
    - efreet\_utils.c, 97
    - efreet\_utils.h, 99
  - efreet\_util\_init
    - Efreedom\_Private, 27
  - efreet\_util\_path\_in\_default
    - efreet\_utils.c, 97
    - efreet\_utils.h, 99
  - efreet\_util\_path\_to\_file\_id
    - efreet\_utils.c, 97
    - efreet\_utils.h, 99
  - efreet\_util\_shutdown
    - Efreedom\_Private, 27
  - efreet\_utils.c
    - Efreedom\_Cache\_Fill, 97

- Efreet\_Cache\_Fill\_Dir, 97
- Efreet\_Cache\_Search, 97
- Efreet\_Cache\_Search\_List, 97
- efreet\_util\_desktop\_comment\_glob\_list, 97
- efreet\_util\_desktop\_exec\_find, 97
- efreet\_util\_desktop\_exec\_glob\_list, 97
- efreet\_util\_desktop\_file\_id\_find, 97
- efreet\_util\_desktop\_generic\_name\_find, 97
- efreet\_util\_desktop\_generic\_name\_-glob\_list, 97
- efreet\_util\_desktop\_mime\_list, 97
- efreet\_util\_desktop\_name\_find, 97
- efreet\_util\_desktop\_name\_glob\_list, 97
- efreet\_util\_desktop\_wm\_class\_find, 97
- efreet\_util\_path\_in\_default, 97
- efreet\_util\_path\_to\_file\_id, 97
- efreet\_utils.h
  - efreet\_util\_desktop\_comment\_glob\_list, 99
  - efreet\_util\_desktop\_exec\_find, 99
  - efreet\_util\_desktop\_exec\_glob\_list, 99
  - efreet\_util\_desktop\_file\_id\_find, 99
  - efreet\_util\_desktop\_generic\_name\_find, 99
  - efreet\_util\_desktop\_generic\_name\_-glob\_list, 99
  - efreet\_util\_desktop\_mime\_list, 99
  - efreet\_util\_desktop\_name\_find, 99
  - efreet\_util\_desktop\_name\_glob\_list, 99
  - efreet\_util\_desktop\_wm\_class\_find, 99
  - efreet\_util\_path\_in\_default, 99
  - efreet\_util\_path\_to\_file\_id, 99
- Efreet\_Xml, 60
  - attributes, 60
  - children, 60
  - efreet\_xml.h, 102
  - tag, 60
  - text, 60
- efreet\_xml.c
  - efreet\_xml\_attribute\_get, 101
  - efreet\_xml\_del, 101
  - efreet\_xml\_init, 101
  - efreet\_xml\_new, 101
  - efreet\_xml\_shutdown, 101
- efreet\_xml.h
  - Efreet\_Xml, 102
  - Efreet\_Xml\_Attribute, 102
  - efreet\_xml\_attribute\_get, 103
  - efreet\_xml\_del, 103
  - efreet\_xml\_init, 103
  - efreet\_xml\_new, 103
  - efreet\_xml\_shutdown, 103
- Efreet\_Xml\_Attribute, 61
  - efreet\_xml.h, 102
  - key, 61
  - value, 61
- efreet\_xml\_attribute\_get
  - efreet\_xml.c, 101
  - efreet\_xml.h, 103
- efreet\_xml\_del
  - efreet\_xml.c, 101
  - efreet\_xml.h, 103
- efreet\_xml\_init
  - efreet\_xml.c, 101
  - efreet\_xml.h, 103
- efreet\_xml\_new
  - efreet\_xml.c, 101
  - efreet\_xml.h, 103
- efreet\_xml\_shutdown
  - efreet\_xml.c, 101
  - efreet\_xml.h, 103
- embedded\_text\_rectangle
  - Efreet\_Icon, 39
- entries
  - Efreet\_Menu, 47
- example\_icon
  - Efreet\_Icon\_Theme, 42
- exec
  - Efreet\_Desktop, 33
- fake
  - Efreet\_Icon\_Theme, 42
- file
  - Efreet\_Desktop\_Command\_File, 37
  - Efreet\_Menu\_Internal, 54
- file\_id
  - \_Efreet\_Cache\_Fill\_Dir, 30
- filenames
  - Efreet\_Menu\_Filter\_Op, 52
- files
  - \_Efreet\_Cache\_Fill, 29
  - Efreet\_Desktop\_Command, 36
- filters
  - Efreet\_Menu\_Filter\_Op, 52
  - Efreet\_Menu\_Internal, 54
- flags
  - Efreet\_Desktop\_Command, 36
- FREE
  - Efreet\_Private, 25
- free\_func
  - Efreet\_Desktop\_Type\_Info, 38
- fullpath
  - Efreet\_Desktop\_Command\_File, 37
- generic\_name
  - Efreet\_Desktop, 33

- has\_embedded\_text\_rectangle
  - Efreet\_Icon, 39
- hidden
  - Efreet\_Desktop, 34
  - Efreet\_Icon\_Theme, 43
- icon
  - Efreet\_Desktop, 34
  - Efreet\_Menu, 47
- icon\_cache
  - Efreet\_Icon\_Theme, 43
- id
  - Efreet\_Desktop\_Type\_Info, 38
  - Efreet\_Menu, 47
  - Efreet\_Menu\_Desktop, 50
- IF\_FREE
  - Efreet\_Private, 25
- IF\_FREE\_DLIST
  - Efreet\_Private, 25
- IF\_FREE\_HASH
  - Efreet\_Private, 25
- IF\_FREE\_LIST
  - Efreet\_Private, 25
- IF\_RELEASE
  - Efreet\_Private, 25
- in\_line
  - Efreet\_Menu\_Internal, 55
  - Efreet\_Menu\_Layout, 57
- inherits
  - Efreet\_Icon\_Theme, 43
- inline\_alias
  - Efreet\_Menu\_Internal, 55
  - Efreet\_Menu\_Layout, 57
- inline\_header
  - Efreet\_Menu\_Internal, 55
  - Efreet\_Menu\_Layout, 57
- inline\_limit
  - Efreet\_Menu\_Internal, 55
  - Efreet\_Menu\_Layout, 57
- internal
  - Efreet\_Icon\_Theme, 43
  - Efreet\_Menu\_Internal, 55
- key
  - Efreet\_Xml\_Attribute, 61
- last\_cache\_check
  - Efreet\_Icon\_Theme, 43
- layout
  - Efreet\_Menu\_Internal, 55
- legacy
  - Efreet\_Menu\_App\_Dir, 49
- lib/efreet.c, 63
- lib/Efreet.h, 66
- lib/efreet\_base.c, 68
- lib/efreet\_base.h, 70
- lib/efreet\_desktop.c, 71
- lib/efreet\_desktop.h, 74
- lib/efreet\_icon.c, 77
- lib/efreet\_icon.h, 79
- lib/efreet\_ini.c, 81
- lib/efreet\_ini.h, 83
- lib/efreet\_menu.c, 85
- lib/efreet\_menu.h, 89
- lib/efreet\_private.h, 91
- lib/efreet\_utils.c, 95
- lib/efreet\_utils.h, 98
- lib/efreet\_xml.c, 100
- lib/efreet\_xml.h, 102
- list
  - \_Efreet\_Cache\_Search\_List, 32
- load\_time
  - Efreet\_Desktop, 34
- max
  - Efreet\_Icon\_Theme\_Directory, 44
- mime\_types
  - Efreet\_Desktop, 34
- min
  - Efreet\_Icon\_Theme\_Directory, 44
- moves
  - Efreet\_Menu\_Internal, 55
- name
  - Efreet\_Desktop, 34
  - Efreet\_Icon, 39
  - Efreet\_Icon\_Theme, 43
  - Efreet\_Icon\_Theme\_Directory, 44
  - Efreet\_Menu, 47
  - Efreet\_Menu\_Internal, 55
  - Efreet\_Menu\_Layout, 57
- NEW
  - Efreet\_Private, 25
- new\_name
  - Efreet\_Menu\_Move, 59
- no\_display
  - Efreet\_Desktop, 34
- NO\_MATCH\_KEY
  - efreet\_icon.c, 78
- normal
  - Efreet\_Icon\_Theme\_Directory, 44
- not\_show\_in
  - Efreet\_Desktop, 34
- num\_pending
  - Efreet\_Desktop\_Command, 36
- old\_name
  - Efreet\_Menu\_Move, 59

- only\_show\_in
  - Efreet\_Desktop, 34
- only\_unallocated
  - Efreet\_Menu\_Internal, 55
- op
  - Efreet\_Menu\_Filter, 51
- orig\_path
  - Efreet\_Desktop, 34
- parent
  - Efreet\_Menu\_Internal, 55
- parse\_func
  - Efreet\_Desktop\_Type\_Info, 38
- path
  - \_Efreet\_Cache\_Fill\_Dir, 30
  - Efreet\_Desktop, 34
  - Efreet\_Icon, 39
  - Efreet\_Icon\_Theme, 43
  - Efreet\_Menu\_App\_Dir, 49
  - Efreet\_Menu\_Internal, 56
- PATH\_MAX
  - Efreet\_Private, 25
- paths
  - Efreet\_Icon\_Theme, 43
- pending
  - Efreet\_Desktop\_Command\_File, 37
- prefix
  - Efreet\_Menu\_App\_Dir, 49
- ref\_count
  - Efreet\_Icon, 39
- save\_func
  - Efreet\_Desktop\_Type\_Info, 38
- section
  - Efreet\_Ini, 46
- seen\_allocated
  - Efreet\_Menu\_Internal, 56
- seen\_deleted
  - Efreet\_Menu\_Internal, 56
- show\_empty
  - Efreet\_Menu\_Internal, 56
  - Efreet\_Menu\_Layout, 57
- size
  - Efreet\_Icon\_Theme\_Directory, 44
- startup\_notify
  - Efreet\_Desktop, 34
- startup\_wm\_class
  - Efreet\_Desktop, 34
- sub\_menus
  - Efreet\_Menu\_Internal, 56
- tag
  - Efreet\_Xml, 60
- terminal
  - Efreet\_Desktop, 35
- text
  - Efreet\_Xml, 60
- threshold
  - Efreet\_Icon\_Theme\_Directory, 44
- try\_exec
  - Efreet\_Desktop, 35
- type
  - Efreet\_Desktop, 35
  - Efreet\_Desktop\_Type\_Info, 38
  - Efreet\_Icon\_Theme\_Directory, 45
  - Efreet\_Menu, 47
  - Efreet\_Menu\_Filter, 51
  - Efreet\_Menu\_Filter\_Op, 52
  - Efreet\_Menu\_Layout, 57
- type\_data
  - Efreet\_Desktop, 35
- uri
  - Efreet\_Desktop\_Command\_File, 37
- url
  - Efreet\_Desktop, 35
- valid
  - Efreet\_Icon\_Theme, 43
- value
  - Efreet\_Xml\_Attribute, 61
- version
  - Efreet\_Desktop, 35
- what
  - \_Efreet\_Cache\_Search\_List, 32
- what1
  - \_Efreet\_Cache\_Search, 31
- what2
  - \_Efreet\_Cache\_Search, 31
- x
  - Efreet\_Desktop, 35
  - Efreet\_Icon\_Point, 41
- x0
  - Efreet\_Icon, 40
- x1
  - Efreet\_Icon, 40
- y
  - Efreet\_Icon\_Point, 41
- y0
  - Efreet\_Icon, 40
- y1
  - Efreet\_Icon, 40