

libfakekey

Generated by Doxygen 1.9.4

1 Module Index	1
1.1 Modules	1
2 File Index	3
2.1 File List	3
3 Module Documentation	5
3.1 FakeKey -	5
3.1.1 Detailed Description	6
3.1.2 Function Documentation	6
3.1.2.1 fakekey_init()	6
3.1.2.2 fakekey_press()	6
3.1.2.3 fakekey_press_keysym()	7
3.1.2.4 fakekey_release()	7
3.1.2.5 fakekey_reload_keysyms()	7
3.1.2.6 fakekey_repeat()	8
3.1.2.7 fakekey_send_keyevent()	8
4 File Documentation	9
4.1 fakekey.h	9
Index	11

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

FakeKey -	5
---------------------	---

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

fakekey.h	9
-------------------------------------	---

Chapter 3

Module Documentation

3.1 FakeKey -

yada yada yada

Typedefs

- typedef struct [FakeKey](#) **FakeKey**
Opaque structure used for all operations.

Enumerations

- enum [FakeKeyModifier](#)
enumerated types for #mb_pixbuf_img_transform

Functions

- [FakeKey](#) * [fakekey_init](#) (Display *xdpy)
Initiates FakeKey.
- int [fakekey_press](#) ([FakeKey](#) *fk, const unsigned char *utf8_char_in, int len_bytes, int modifiers)
Sends a Keypress to the server for the supplied UTF8 character.
- void [fakekey_repeat](#) ([FakeKey](#) *fk)
Repeats a press of the currently held key (from [fakekey_press](#))
- void [fakekey_release](#) ([FakeKey](#) *fk)
Releases the currently held key (from [fakekey_press](#))
- int [fakekey_reload_keysyms](#) ([FakeKey](#) *fk)
Resyncs the internal list of keysyms with the server.
- int [fakekey_press_keysym](#) ([FakeKey](#) *fk, KeySym keysym, int flags)
[fakekey_press](#) but with an X keysym rather than a UTF8 Char.
- int [fakekey_send_keyevent](#) ([FakeKey](#) *fk, KeyCode keycode, Bool is_press, int modifiers)

3.1.1 Detailed Description

yada yada yada

Always remember to release held keys

3.1.2 Function Documentation

3.1.2.1 fakekey_init()

```
FakeKey * fakekey_init (
    Display * xcpy )
```

Initiates FakeKey.

Parameters

<i>xcpy</i>	X Display connection.
-------------	-----------------------

Returns

new [FakeKey](#) reference on success, NULL on fail.

3.1.2.2 fakekey_press()

```
int fakekey_press (
    FakeKey * fk,
    const unsigned char * utf8_char_in,
    int len_bytes,
    int modifiers )
```

Sends a Keypress to the server for the supplied UTF8 character.

Parameters

<i>fk</i>	FakeKey refernce from fakekey_init
<i>utf8_char↔ _in</i>	Pointer to a single UTF8 Character data.
<i>len_bytes</i>	Lenth in bytes of character, or -1 in ends with 0
<i>modifiers</i>	OR'd list of FakeKeyModifier modifiers keys to press with the key.

Returns

3.1.2.3 fakekey_press_keysym()

```
int fakekey_press_keysym (
    FakeKey * fk,
    KeySym keysym,
    int flags )
```

[fakekey_press](#) but with an X keysym rather than a UTF8 Char.

Parameters

<i>fk</i>	FakeKey refernce from fakekey_init
<i>keysym</i>	X Keysym to send
<i>flags</i>	

Returns

3.1.2.4 fakekey_release()

```
void fakekey_release (
    FakeKey * fk )
```

Releases the currently held key (from [fakekey_press](#))

Parameters

<i>fk</i>	FakeKey refernce from fakekey_init
-----------	--

3.1.2.5 fakekey_reload_keysyms()

```
int fakekey_reload_keysyms (
    FakeKey * fk )
```

Resyns the internal list of keysyms with the server.

Should be called if a MappingNotify event is recieved.

Parameters

<i>fk</i>	FakeKey refernce from fakekey_init
-----------	--

Returns**3.1.2.6 fakekey_repeat()**

```
void fakekey_repeat (
    FakeKey * fk )
```

Repeats a press of the currently held key (from [fakekey_press](#))

Parameters

<i>fk</i>	FakeKey refernce from fakekey_init
-----------	--

3.1.2.7 fakekey_send_keyevent()

```
int fakekey_send_keyevent (
    FakeKey * fk,
    KeyCode keycode,
    Bool is_press,
    int modifiers )
```

Parameters

<i>fk</i>	FakeKey refernce from fakekey_init
<i>keycode</i>	X Keycode to send
<i>is_press</i>	Is this a press (or release)
<i>modifiers</i>	

Returns

Chapter 4

File Documentation

4.1 fakekey.h

```
00001 #ifndef _HAVE_LIBFAKEKEY_H
00002 #define _HAVE_LIBFAKEKEY_H
00003
00004 #include <stdio.h>
00005 #include <stdlib.h>
00006 #include <X11/X.h>
00007 #include <X11/Xlib.h>
00008 #include <X11/Xlibint.h>
00009 #include <X11/Xutil.h>
00010 #include <X11/cursorfont.h>
00011 #include <X11/keysymdef.h>
00012 #include <X11/keysym.h>
00013 #include <X11/extensions/XTest.h>
00014 #include <X11/Xos.h>
00015 #include <X11/Xproto.h>
00016
00017 #ifdef __cplusplus
00018 extern "C" {
00019 #endif
00020
00035 typedef struct FakeKey FakeKey;
00036
00042 typedef enum
00043 {
00044     FAKEKEYMOD_SHIFT    = (1<<1),
00045     FAKEKEYMOD_CONTROL  = (1<<2),
00046     FAKEKEYMOD_ALT      = (1<<3),
00047     FAKEKEYMOD_META     = (1<<4)
00048 } FakeKeyModifier;
00050
00058 FakeKey*
00059 fakekey_init(Display *xdpy);
00060
00061
00074 int
00075 fakekey_press(FakeKey          *fk,
00076               const unsigned char *utf8_char_in,
00077               int                 len_bytes,
00078               int                 modifiers);
00079
00085 void
00086 fakekey_repeat(FakeKey *fk);
00087
00088
00094 void
00095 fakekey_release(FakeKey *fk);
00096
00105 int
00106 fakekey_reload_keysyms(FakeKey *fk);
00107
00117 int
00118 fakekey_press_keysym(FakeKey *fk,
00119                      KeySym   keysym,
00120                      int       flags);
00121
00131 int
00132 fakekey_send_keyevent(FakeKey *fk,
```

```
00133         KeyCode  keycode,  
00134         Bool      is_press,  
00135         int        modifiers);  
00136  
00139 #ifdef __cplusplus  
00140 }  
00141 #endif  
00142  
00143 #endif /* _HAVE_LIBFAKEKEY_H */
```

Index

- FakeKey -, [5](#)
 - fakekey_init, [6](#)
 - fakekey_press, [6](#)
 - fakekey_press_keysym, [7](#)
 - fakekey_release, [7](#)
 - fakekey_reload_keysyms, [7](#)
 - fakekey_repeat, [8](#)
 - fakekey_send_keyevent, [8](#)
- fakekey.h, [9](#)
- fakekey_init
 - FakeKey -, [6](#)
- fakekey_press
 - FakeKey -, [6](#)
- fakekey_press_keysym
 - FakeKey -, [7](#)
- fakekey_release
 - FakeKey -, [7](#)
- fakekey_reload_keysyms
 - FakeKey -, [7](#)
- fakekey_repeat
 - FakeKey -, [8](#)
- fakekey_send_keyevent
 - FakeKey -, [8](#)