

Give Reference Manual

Generated by Doxygen 1.5.3

Sat May 17 16:27:32 2008

Contents

1	Give	1
2	Give Class Index	5
2.1	Give Class List	5
3	Give File Index	7
3.1	Give File List	7
4	Give Page Index	9
4.1	Give Related Pages	9
5	Give Class Documentation	11
5.1	_give_bluetooth Struct Reference	11
5.2	_give_bluetooth_private Struct Reference	12
5.3	_give_bluetoothClass Struct Reference	13
5.4	_give_drag_and_drop Struct Reference	14
5.5	_give_drag_and_drop_private Struct Reference	15
5.6	_give_drag_and_dropClass Struct Reference	17
5.7	_give_ebook Struct Reference	18
5.8	_give_ebook_private Struct Reference	19
5.9	_give_ebookClass Struct Reference	20
5.10	_GiveStatusBarPlugin Struct Reference	21
5.11	_GiveStatusBarPluginClass Struct Reference	22
5.12	_GiveStatusBarPluginPrivate Struct Reference	23
6	Give File Documentation	25
6.1	documentation.txt File Reference	25
6.2	src/give_bluetooth.c File Reference	26
6.3	src/give_bluetooth.h File Reference	31
6.4	src/give_drag_and_drop.c File Reference	34

6.5	src/give_drag_and_drop.h File Reference	48
6.6	src/give_ebook.c File Reference	51
6.7	src/give_ebook.h File Reference	58
6.8	src/give_statusbar_plugin.c File Reference	61
6.9	src/give_statusbar_plugin.h File Reference	73
7	Give Page Documentation	77
7.1	Todo List	77

Chapter 1

Give

Statusbar plugin for easy drag & drop data sending via Bluetooth.

Version:

0.8.0

Author:

Dénes Mátételki <denes.matetelki@gmail.com>

Date:

2008

Plot summary

After analyzing the data exchange services of the current devices used in mobile communication, came the idea of an application, which uses the possibilities of Bluetooth. Due to the survey, most of the mobile devices has Bluetooth capabilities, but no WLAN or 3G.

The idea is to accomplish the following use-cases: Two people meet with devices of Bluetooth capabilities so WLAN or internet is not needed, since at most places, for example at a railway station, free wlan connection is not always present, and most of the devices don't have 3G adapters. At the device of the data exchange initiative person, the application is running, however it is not necessary for the data receiver's device, to have the application installed.

The target device is the N800 series of Nokia, which is a linux-based internet tablet and has Bluetooth and WLAN adapters. The data exchange initiative person owns a device of this kind, with the application, Give.

The project focuses on two services:

- Sending data, providing compatibility with as much application as possible, by using standards. Applications provided by the vendor have high priority.
- Sharing internet access to devices which has Bluetooth adapter but no 3G or WLAN capabilities.

The data exchange in this case is not a server-client or a sharing type, the data provider initiates the transaction, "gives" the information to the destination.

The name comes from here, Give.



Figure 1.1: A screenshot of the menu of Give

Services

- Sending text **OK**
- Sending file(s) **OK**
- Sending calendar event **OK** with dates, compiled with `-enable-dnd=yes`
- Sending business card **OK**
- E-mail sending **TODO** patch modest to produce complete e-mail at drag & drop
- Sharing internet access. **TODO** The device doesn't act as a server, just client.

The application has been accomplished as a statusbar-plugin, represented by an icon in the top-right status line, which can always be seen. The choice come down to this because an icon, visible all the time, always available for a drag & drop destination.

So for the sake of the fastest data sending, data objects are selected with drag & drop, by dropping the data object to the application icon, to by-pass the time spent in context-sensitive menus.

Objects

The project separated into four projects.

- The project itself is a hildon-statusbar-plugin, which means it is an icon, in the statusbar area which interacts with user actions and shows the some state of system. The icon is a GTK+ drag & drop destination, and the actions to be taken upon a data drop is described in the drag & drop object.
- The drag & drop object examines the recieved data, converts it into file or files, according to its nature. For example, a contact will be send as a VCard, a calendar event as an VCal event, and so on. For VCard data manipulation, an EBook connection has been defined. The data transfer has been done with the Bluetooth object. Also, the state of the device's Bluetooth is checked, and if it's turned off, a dialog appears, asking for permission of the user to turn it on.
- The EBook interacts with the Evolution Data Server's EBook component, for data queries, and VCard representations of the contacts.
- The Bluetooth object is responsible for the data transfer. It uses a convenient D-Bus call, which invokes the Hildon Connectivity's Bluetooth routines. They pops up a dialog to select a visible Bluetooth device as the destination of the sending, doing the data transfer and handling the success or failure cases.

Object relations

The objects are not derived from each other, they are encapsulated.

The `_GiveStatusbarPlugin` derives from `StatusBarItem` which derives directly from `GObject`, like `_give_drag_and_drop`, `_give_bluetooth` and `_give_ebook`.

The encapsulation is like the following:

```
give_statusbar_plugin
+----give_drag_and_drop
      +----give_bluetooth
      +----give_ebook
```

Installation

```
./autogen.sh
./configure --enable-ltc=yes --enable-ltf=yes
make
make install
```

The two configure options are: enable log to console and enable log to file.

Keywords

Nokia Internet Tablet, maemo, Bluetooth, drag & drop, IP over Bluetooth, PAN.

Screenshot

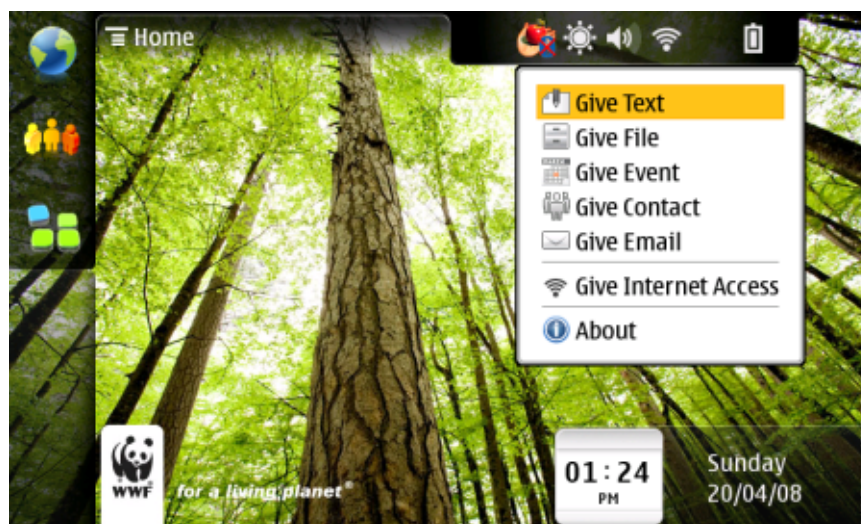


Figure 1.2: A screenshot of the menu of Give

Todos left

- Patch modest.
- Solve the pand error.
- Look for memory leaks.

Known bugs

- Selecting Control Panel -> Panels -> Statusbar -> selecting Give's checkbox off, then close Panels menu. Open again and the desktop crashes. Object registration issue?

Chapter 2

Give Class Index

2.1 Give Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

_give_bluetooth	11
_give_bluetooth_private (Struct of private members of give_bluetooth)	12
_give_bluetoothClass	13
_give_drag_and_drop	14
_give_drag_and_drop_private (Struct of private members of give_bluetooth)	15
_give_drag_and_dropClass	17
_give_ebook	18
_give_ebook_private (Struct of private members of give_ebook)	19
_give_ebookClass	20
_GiveStatusBarPlugin	21
_GiveStatusBarPluginClass	22
_GiveStatusBarPluginPrivate (Struct of private members of GiveStatusBarPlugin)	23

Chapter 3

Give File Index

3.1 Give File List

Here is a list of all files with brief descriptions:

src/give_bluetooth.c	26
src/give_bluetooth.h	31
src/give_drag_and_drop.c	34
src/give_drag_and_drop.h	48
src/give_ebook.c	51
src/give_ebook.h	58
src/give_statusbar_plugin.c	61
src/give_statusbar_plugin.h	73

Chapter 4

Give Page Index

4.1 Give Related Pages

Here is a list of all related documentation pages:

Todo List	77
---------------------	--------------------

Chapter 5

Give Class Documentation

5.1 `_give_bluetooth` Struct Reference

```
#include <give_bluetooth.h>
```

Public Attributes

- GObject [parent_object](#)

5.1.1 Detailed Description

Definition at line 41 of file `give_bluetooth.h`.

5.1.2 Member Data Documentation

5.1.2.1 `GObject _give_bluetooth::parent_object`

Definition at line 42 of file `give_bluetooth.h`.

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

- [src/give_bluetooth.h](#)

5.2 `_give_bluetooth_private` Struct Reference

Struct of private members of [give_bluetooth](#).

Public Attributes

- `DBusGProxy * dbus_proxy`

5.2.1 Detailed Description

Struct of private members of [give_bluetooth](#).

Definition at line 44 of file `give_bluetooth.c`.

5.2.2 Member Data Documentation

5.2.2.1 `DBusGProxy* _give_bluetooth_private::dbus_proxy`

Bridge to CONBTDIALOGS for sending files.

Definition at line 47 of file `give_bluetooth.c`.

Referenced by `give_bluetooth_finalize()`, `give_bluetooth_init()`, and `give_bluetooth_send_file()`.

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

- `src/give_bluetooth.c`

5.3 `_give_bluetoothClass` Struct Reference

```
#include <give_bluetooth.h>
```

Public Attributes

- GObjectClass [parent_class](#)

5.3.1 Detailed Description

Definition at line 45 of file `give_bluetooth.h`.

5.3.2 Member Data Documentation

5.3.2.1 GObjectClass `_give_bluetoothClass::parent_class`

Definition at line 46 of file `give_bluetooth.h`.

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

- [src/give_bluetooth.h](#)

5.4 `_give_drag_and_drop` Struct Reference

```
#include <give_drag_and_drop.h>
```

Public Attributes

- GObject [parent_object](#)

5.4.1 Detailed Description

Definition at line 46 of file `give_drag_and_drop.h`.

5.4.2 Member Data Documentation

5.4.2.1 `GObject _give_drag_and_drop::parent_object`

Definition at line 47 of file `give_drag_and_drop.h`.

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

- [src/give_drag_and_drop.h](#)

5.5 `_give_drag_and_drop_private` Struct Reference

Struct of private members of `give_bluetooth`.

Public Attributes

- gboolean `bt_is_on`
- GtkWidget * `destination`
- gulong `handler_id`
- DBusGProxy * `dbus_proxy`
- `give_bluetooth` * `give_bt_object`
- `give_ebook` * `give_ebook_object`

5.5.1 Detailed Description

Struct of private members of `give_bluetooth`.

Definition at line 53 of file `give_drag_and_drop.c`.

5.5.2 Member Data Documentation

5.5.2.1 gboolean `_give_drag_and_drop_private::bt_is_on`

The Bluetooth adapter is ON or OFF.

Definition at line 56 of file `give_drag_and_drop.c`.

Referenced by `give_drag_and_drop_drag_recieved()`, and `give_drag_and_drop_have_bt()`.

5.5.2.2 GtkWidget* `_give_drag_and_drop_private::destination`

The destination widget for d&d. It's the button of the plugin.

Definition at line 58 of file `give_drag_and_drop.c`.

Referenced by `give_drag_and_drop_finalize()`, and `give_drag_and_drop_set_dest()`.

5.5.2.3 gulong `_give_drag_and_drop_private::handler_id`

ID of the callback listening to the "drag_data_received" signal.

Definition at line 60 of file `give_drag_and_drop.c`.

Referenced by `give_drag_and_drop_finalize()`, and `give_drag_and_drop_set_dest()`.

5.5.2.4 DBusGProxy* `_give_drag_and_drop_private::dbus_proxy`

Proxy to BLUEZ_DBUS for turning on Bluetooth if necessary.

Definition at line 62 of file `give_drag_and_drop.c`.

Referenced by `give_drag_and_drop_finalize()`, `give_drag_and_drop_init()`, and `give_drag_and_drop_turn_bt_on_dialog()`.

5.5.2.5 `give_bluetooth* _give_drag_and_drop_private::give_bt_object`

Encapsulated `give_bluetooth` object, for sending files.

Definition at line 64 of file `give_drag_and_drop.c`.

Referenced by `give_drag_and_drop_finalize()`, `give_drag_and_drop_init()`, and `give_drag_and_drop_send()`.

5.5.2.6 `give_ebook* _give_drag_and_drop_private::give_ebook_object`

Encapsulated `give_ebook` object for contact queries.

Definition at line 66 of file `give_drag_and_drop.c`.

Referenced by `give_drag_and_drop_create_dialog()`, `give_drag_and_drop_finalize()`, `give_drag_and_drop_init()`, and `give_drag_and_drop_run_dialog()`.

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

- `src/give_drag_and_drop.c`

5.6 `_give_drag_and_dropClass` Struct Reference

```
#include <give_drag_and_drop.h>
```

Public Attributes

- GObjectClass [parent_class](#)

5.6.1 Detailed Description

Definition at line 50 of file `give_drag_and_drop.h`.

5.6.2 Member Data Documentation

5.6.2.1 GObjectClass `_give_drag_and_dropClass::parent_class`

Definition at line 51 of file `give_drag_and_drop.h`.

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

- [src/give_drag_and_drop.h](#)

5.7 `_give_ebook` Struct Reference

```
#include <give_ebook.h>
```

Public Attributes

- GObject [parent_object](#)

5.7.1 Detailed Description

Definition at line 39 of file `give_ebook.h`.

5.7.2 Member Data Documentation

5.7.2.1 `GObject _give_ebook::parent_object`

Definition at line 40 of file `give_ebook.h`.

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

- [src/give_ebook.h](#)

5.8 `_give_ebook_private` Struct Reference

Struct of private members of [give_ebook](#).

Public Attributes

- EBook * [ebook](#)
- gboolean [ready](#)

5.8.1 Detailed Description

Struct of private members of [give_ebook](#).

Definition at line 38 of file `give_ebook.c`.

5.8.2 Member Data Documentation

5.8.2.1 EBook* `_give_ebook_private::ebook`

An EBook which contains the contacts.

Definition at line 41 of file `give_ebook.c`.

Referenced by `give_ebook_get_contact_name()`, `give_ebook_get_contact_vcard()`, and `give_ebook_init()`.

5.8.2.2 gboolean `_give_ebook_private::ready`

The EBook is opened or not.

Definition at line 43 of file `give_ebook.c`.

Referenced by `give_ebook_get_contact_name()`, `give_ebook_get_contact_vcard()`, and `give_ebook_open_response()`.

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

- `src/give_ebook.c`

5.9 `_give_ebookClass` Struct Reference

```
#include <give_ebook.h>
```

Public Attributes

- GObjectClass [parent_class](#)

5.9.1 Detailed Description

Definition at line 43 of file `give_ebook.h`.

5.9.2 Member Data Documentation

5.9.2.1 GObjectClass `_give_ebookClass::parent_class`

Definition at line 44 of file `give_ebook.h`.

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

- [src/give_ebook.h](#)

5.10 `_GiveStatusBarPlugin` Struct Reference

```
#include <give_statusbar_plugin.h>
```

Public Attributes

- `StatusBarItem` [parent](#)

5.10.1 Detailed Description

Definition at line 47 of file `give_statusbar_plugin.h`.

5.10.2 Member Data Documentation

5.10.2.1 `StatusBarItem _GiveStatusBarPlugin::parent`

Definition at line 49 of file `give_statusbar_plugin.h`.

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

- `src/give_statusbar_plugin.h`

5.11 `_GiveStatusbarPluginClass` Struct Reference

```
#include <give_statusbar_plugin.h>
```

Public Attributes

- StatusbarItemClass [parent_class](#)

5.11.1 Detailed Description

Definition at line 52 of file `give_statusbar_plugin.h`.

5.11.2 Member Data Documentation

5.11.2.1 StatusbarItemClass `_GiveStatusbarPluginClass::parent_class`

Definition at line 54 of file `give_statusbar_plugin.h`.

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

- [src/give_statusbar_plugin.h](#)

5.12 `_GiveStatusbarPluginPrivate` Struct Reference

Struct of private members of `GiveStatusbarPlugin`.

Public Attributes

- `GtkWidget * button`
- `GtkWidget * image`
- `give_drag_and_drop * dnd`
- `GConfClient * gc_client`
- `guint notify_uid`
- `guint state`
- `gint pand_pid`

5.12.1 Detailed Description

Struct of private members of `GiveStatusbarPlugin`.

Definition at line 56 of file `give_statusbar_plugin.c`.

5.12.2 Member Data Documentation

5.12.2.1 `GtkWidget* _GiveStatusbarPluginPrivate::button`

Button in the statusbar area. Clicking on it invokes the Give menu.

Definition at line 59 of file `give_statusbar_plugin.c`.

Referenced by `give_statusbar_plugin_init()`.

5.12.2.2 `GtkWidget* _GiveStatusbarPluginPrivate::image`

The icon of the plugin in `_GiveStatusbarPluginPrivate::button`. It is set to a be a d&d destination.

Definition at line 62 of file `give_statusbar_plugin.c`.

Referenced by `give_statusbar_plugin_init()`, and `give_statusbar_plugin_update_status()`.

5.12.2.3 `give_drag_and_drop* _GiveStatusbarPluginPrivate::dnd`

Encapsulated `give_drag_and_drop` object.

Definition at line 65 of file `give_statusbar_plugin.c`.

Referenced by `give_statusbar_plugin_finalize()`, `give_statusbar_plugin_init()`, `give_statusbar_plugin_pand_dialog()`, and `give_statusbar_plugin_update_status()`.

5.12.2.4 `GConfClient* _GiveStatusbarPluginPrivate::gc_client`

GConf connection to monitor the state of the Bluetooth adapter

Definition at line 67 of file `give_statusbar_plugin.c`.

Referenced by `give_statusbar_plugin_finalize()`, and `give_statusbar_plugin_init()`.

5.12.2.5 `guint _GiveStatusbarPluginPrivate::notify_uid`

ID of the handler listening of the change of gconf key [BT_KEY](#).

Definition at line 69 of file `give_statusbar_plugin.c`.

Referenced by `give_statusbar_plugin_finalize()`, and `give_statusbar_plugin_init()`.

5.12.2.6 `guint _GiveStatusbarPluginPrivate::state`

The of the Bluetooth defined in [bt_state](#).

Definition at line 71 of file `give_statusbar_plugin.c`.

Referenced by `give_statusbar_plugin_bt_changed()`, `give_statusbar_plugin_clicked()`, `give_statusbar_plugin_init()`, `give_statusbar_plugin_pand_dialog()`, `give_statusbar_plugin_start_pand()`, `give_statusbar_plugin_stop_pand()`, and `give_statusbar_plugin_update_status()`.

5.12.2.7 `gint _GiveStatusbarPluginPrivate::pand_pid`

The Process ID of pand.

Definition at line 73 of file `give_statusbar_plugin.c`.

Referenced by `give_statusbar_plugin_bt_changed()`, `give_statusbar_plugin_init()`, `give_statusbar_plugin_start_pand()`, and `give_statusbar_plugin_stop_pand()`.

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

- [src/give_statusbar_plugin.c](#)

Chapter 6

Give File Documentation

6.1 `documentation.txt` File Reference

6.2 src/give_ bluetooth.c File Reference

```
#include "give_bluetooth.h"
#include <dbus/dbus.h>
#include <dbus/dbus-glib.h>
#include <conbtdialogs-dbus.h>
```

Include dependency graph for give_ bluetooth.c:

Classes

- struct [_give_bluetooth_private](#)
Struct of private members of give_bluetooth.

Defines

- #define [GIVE_BLUETOOTH_GET_PRIVATE](#)(obj)

Typedefs

- typedef struct [_give_bluetooth_private](#) [give_bluetooth_private](#)

Functions

Private functions

- GType [give_bluetooth_get_type](#) (void)
Registers the give_bluetooth type statically.
- static void [give_bluetooth_class_init](#) ([give_bluetoothClass](#) *klass)
Overrides the class finalize function and adds the private structure.
- static void [give_bluetooth_init](#) ([give_bluetooth](#) *self)
Connects to the DBus system bus.
- static void [give_bluetooth_finalize](#) (GObject *self)
Unreferences the private [_give_bluetooth_private::dbus_proxy](#).

Public functions

- [give_bluetooth](#) * [give_bluetooth_new](#) (void)
Creates a new instance of give_bluetooth.
- void [give_bluetooth_send_file](#) ([give_bluetooth](#) *self, gchar **files)
Invokes the send file procedure of CONBTDIALOGS.

6.2.1 Detailed Description

The Bluetooth object is responsible for the data transfer. It uses a convenient D-Bus call, which invokes the Hildon Connectivity's Bluetooth routines. They pops up a dialog to select a visible Bluetooth device as the destination of the sending, doing the data transfer and handling the success or failure cases.

Definition in file [give_bluetooth.c](#).

6.2.2 Define Documentation

6.2.2.1 #define GIVE_BLUETOOTH_GET_PRIVATE(obj)

Value:

```
(G_TYPE_INSTANCE_GET_PRIVATE ((obj), \
                               GIVE_BLUETOOTH_TYPE, \
                               give_bluetooth_private));
```

Definition at line 36 of file [give_bluetooth.c](#).

Referenced by [give_bluetooth_finalize\(\)](#), [give_bluetooth_init\(\)](#), and [give_bluetooth_send_file\(\)](#).

6.2.3 Typedef Documentation

6.2.3.1 typedef struct _give_bluetooth_private give_bluetooth_private

Definition at line 34 of file [give_bluetooth.c](#).

6.2.4 Function Documentation

6.2.4.1 static void give_bluetooth_class_init (give_bluetoothClass * *klass*) [static]

Overrides the class finalize function and adds the private structure.

Sets [give_bluetooth_finalize](#) as the class' finalize function.

Registers the [give_bluetooth_private](#) structure as private member.

Parameters:

klass a [give_bluetoothClass](#) class.

Definition at line 116 of file [give_bluetooth.c](#).

References [give_bluetooth_finalize\(\)](#).

Referenced by [give_bluetooth_get_type\(\)](#).

Here is the call graph for this function:

Here is the caller graph for this function:

6.2.4.2 static void give_bluetooth_finalize (GObject * *self*) [static]

Unreferences the private `_give_bluetooth_private::dbus_proxy`.

Parameters:

self a GObject.

Definition at line 177 of file `give_bluetooth.c`.

References `_give_bluetooth_private::dbus_proxy`, `GIVE_BLUETOOTH_GET_PRIVATE`, and `GIVE_BLUETOOTH_IS_OBJECT`.

Referenced by `give_bluetooth_class_init()`.

Here is the caller graph for this function:

6.2.4.3 GType give_bluetooth_get_type (void)

Registers the `give_bluetooth` type statically.

Sets class init, and instance init functions.

The class finalize function has been set in the class init function.

Returns:

The new registered GType, with the name of "give_bluetooth".

Definition at line 80 of file `give_bluetooth.c`.

References `give_bluetooth_class_init()`, and `give_bluetooth_init()`.

Here is the call graph for this function:

6.2.4.4 static void give_bluetooth_init (give_bluetooth * *self*) [static]

Connects to the DBus system bus.

- Connects to the DBus system bus.
- Sets the `_give_bluetooth_private::dbus_proxy` to connect to CONBTDIALOGS which is used to send files via Bluetooth.

Parameters:

self a `give_bluetooth` object.

Definition at line 136 of file give_bluetooth.c.

References `_give_bluetooth_private::dbus_proxy`, `GIVE_BLUETOOTH_GET_PRIVATE`, and `GIVE_BLUETOOTH_IS_OBJECT`.

Referenced by `give_bluetooth_get_type()`.

Here is the caller graph for this function:

6.2.4.5 `give_bluetooth * give_bluetooth_new (void)`

Creates a new instance of `give_bluetooth`.

Returns:

A new `give_bluetooth` object.

Definition at line 198 of file give_bluetooth.c.

References `GIVE_BLUETOOTH_TYPE`.

Referenced by `give_drag_and_drop_init()`.

Here is the caller graph for this function:

6.2.4.6 `void give_bluetooth_send_file (give_bluetooth * self, gchar ** files)`

Invokes the send file procedure of `CONBTDIALOGS`.

The job is handled completely by the `CONBTDIALOGS_SEND_FILE_REQ` method, which pops up the device discovery dialog, send the given files and handles the result.

Parameters:

self A `give_bluetooth` object.

files A stringlist of filepaths, ended with `NULL`. The list need to be freed.

Definition at line 220 of file give_bluetooth.c.

References `_give_bluetooth_private::dbus_proxy`, `GIVE_BLUETOOTH_GET_PRIVATE`, and `GIVE_BLUETOOTH_IS_OBJECT`.

Referenced by `give_drag_and_drop_send()`.

Here is the caller graph for this function:

6.3 src/give_ bluetooth.h File Reference

```
#include <glib-object.h>
```

Include dependency graph for give_ bluetooth.h:

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

Classes

- struct [_give_ bluetooth](#)
- struct [_give_ bluetoothClass](#)

Defines

- #define [GIVE_ BLUETOOTH_ TYPE](#) (give_ bluetooth_ get_ type ())
- #define [GIVE_ BLUETOOTH\(obj\)](#)
- #define [GIVE_ BLUETOOTH_ CLASS\(klass\)](#)
- #define [GIVE_ BLUETOOTH_ IS_ OBJECT\(obj\)](#)
- #define [GIVE_ BLUETOOTH_ IS_ CLASS\(klass\)](#)

Typedefs

- typedef typedefG_ BEGIN_ DECLS struct [_give_ bluetooth](#) give_ bluetooth
- typedef struct [_give_ bluetoothClass](#) give_ bluetoothClass

Functions

- GType [give_ bluetooth_ get_ type](#) (void) G_ GNUC_ CONST
- [give_ bluetooth * give_ bluetooth_ new](#) (void)
- void [give_ bluetooth_ send_ file](#) ([give_ bluetooth *self](#), gchar **files)

6.3.1 Define Documentation

6.3.1.1 #define GIVE_ BLUETOOTH(obj)

Value:

```
(G_ TYPE_ CHECK_ INSTANCE_ CAST ((obj), \
                                GIVE_ BLUETOOTH_ TYPE, give_ bluetooth))
```

Definition at line 30 of file give_ bluetooth.h.

6.3.1.2 #define GIVE__BLUETOOTH__CLASS(klass)**Value:**

```
(G_TYPE_CHECK_CLASS_CAST ((klass), \
                           GIVE_BLUETOOTH_TYPE, \
                           give_bluetoothClass))
```

Definition at line 32 of file give_bluetooth.h.

6.3.1.3 #define GIVE__BLUETOOTH__IS__CLASS(klass)**Value:**

```
(G_TYPE_CHECK_CLASS_TYPE ((klass), \
                           GIVE_BLUETOOTH_TYPE))
```

Definition at line 37 of file give_bluetooth.h.

6.3.1.4 #define GIVE__BLUETOOTH__IS__OBJECT(obj)**Value:**

```
(G_TYPE_CHECK_INSTANCE_TYPE ((obj), \
                              GIVE_BLUETOOTH_TYPE))
```

Definition at line 35 of file give_bluetooth.h.

Referenced by give_bluetooth_finalize(), give_bluetooth_init(), and give_bluetooth_send_file().

6.3.1.5 #define GIVE__BLUETOOTH__TYPE (give_bluetooth_get_type ())

Definition at line 29 of file give_bluetooth.h.

Referenced by give_bluetooth_new().

6.3.2 Typedef Documentation**6.3.2.1** typedef typedefG__BEGIN_DECLS struct __give_bluetooth give_bluetooth

Definition at line 26 of file give_bluetooth.h.

6.3.2.2 typedef struct __give_bluetoothClass give_bluetoothClass

Definition at line 27 of file give_bluetooth.h.

6.3.3 Function Documentation

6.3.3.1 GType give__bluetooth__get__type (void)

6.3.3.2 give__bluetooth* give__bluetooth__new (void)

6.3.3.3 void give__bluetooth__send__file (give__bluetooth * *self*, gchar ** *files*)

6.4 src/give_drag_and_drop.c File Reference

```
#include "give_drag_and_drop.h"
#include "give_bluetooth.h"
#include "give_ebook.h"
#include <stdio.h>
#include <string.h>
#include <sys/stat.h>
#include <dbus/dbus-glib.h>
#include <locale.h>
#include <libintl.h>
```

Include dependency graph for give_drag_and_drop.c:

Classes

- struct `_give_drag_and_drop_private`
Struct of private members of `give_bluetooth`.

Defines

- #define `_(String) gettext(String)`
- #define `GIVE_DRAG_AND_DROP_GET_PRIVATE(obj)`
- #define `TEMP_DIR "/tmp"`

Bluez

- #define `BLUEZ_DBUS_SERVICE "org.bluez"`
- #define `BLUEZ_DBUS_PATH "/org/bluez/hci0"`
- #define `BLUEZ_DBUS_INTERFACE "org.bluez.Adapter"`
- #define `BLUEZ_DBUS_SET_MODE "SetMode"`
- #define `BLUEZ_DBUS_MESSAGE "connectable"`

Accepted d&d types

- #define `PLAIN_TEXT_TYPE "text/plain"`
- #define `URI_LIST_TYPE "text/uri-list"`
- #define `CALENDAR_EVENT_TYPE "application/x-e-calendar-event"`
- #define `CONTACT_LIST_TYPE "x-osso-contact-list"`
- #define `TREE_ROW_TYPE "GTK_TREE_MODEL_ROW"`
- #define `TREE_AS_STRING_TYPE "text/tree-path-as-string-list"`
- #define `MODEST_EMAIL_TYPE "text/modest-email"`

Typedefs

- typedef struct `_give_drag_and_drop_private` `give_drag_and_drop_private`

Enumerations

- enum {
[PLAIN_TEXT](#), [URI_LIST](#), [CALENDAR_EVENT](#), [CONTACT_LIST](#),
[TREE_ROW](#), [TREE_AS_STRING](#), [MODEST_EMAIL](#) }

Functions

Private functions

- GType [give_drag_and_drop_get_type](#) (void)
Registers the [give_drag_and_drop](#) type statically.
- static void [give_drag_and_drop_class_init](#) ([give_drag_and_dropClass](#) *klass)
Overrides the class finalize function and adds the private structure.
- static void [give_drag_and_drop_init](#) ([give_drag_and_drop](#) *self)
Connects to D_{Bus} and instantiates [give_bluetooth](#) and [give_ebook](#).
- static void [give_drag_and_drop_finalize](#) (GObject *self)
Unsets the widget as d_{Ed} destination, unrefs objects.
- [give_drag_and_drop](#) * [give_drag_and_drop_new](#) (GtkWidget *widget)
Creates a new instance of [give_drag_and_drop](#).
- static void [give_drag_and_drop_set_dest](#) ([give_drag_and_drop](#) *self, GtkWidget *widget)
Sets the parameter widget a d_{Ed} destination.
- void [give_drag_and_drop_have_bt](#) ([give_drag_and_drop](#) *self, gboolean setting)
Sets [_give_drag_and_drop_private::bt_is_on](#).
- gboolean [give_drag_and_drop_turn_bt_on_dialog](#) ([give_drag_and_drop](#) *self)
Confirmation dialog whether to turn on the Bluetooth adapter or not.

Public functions

- static GtkWidget * [give_drag_and_drop_create_dialog](#) ([give_drag_and_drop](#) *self, const guint info, const gchar *text, guint *file_count, GtkWidget **entry)
Creates a confirmation dialog with whether to send the data or not.
- static int [give_drag_and_drop_run_dialog](#) ([give_drag_and_drop](#) *self, GtkWidget *dialog, const guint info, const gchar *text, const guint file_count, gchar ***files, const GtkWidget *entry)
Runs the dialog created by [give_drag_and_drop_create_dialog](#).
- static void [give_drag_and_drop_send](#) ([give_drag_and_drop](#) *self, const guint info, const gchar *text)
Creates a confirmation dialog and send the files.
- static void [give_drag_and_drop_drag_recieved](#) (GtkWidget *widget, GdkDragContext *context, gint x, gint y, GtkSelectionData *selection_data, guint info, guint32 time, gpointer user_data)
Handles the "drag_data_received" signal.

Variables

- const GtkTargetEntry `drop_types` []
Accepted d&d types.

6.4.1 Detailed Description

The drag & drop object examines the recieved data, converts it into file or files, according to its nature. For example, a contact will be send as a VCard, a calendar event as an VCal event, and so on. For VCard data manipulation, an EBook connection has been defined. The data transfer has been done with the Bluetooth object. Also, the state of the device's Bluetooth is checked, and if it's turned off, a dialog appears, asking for permission of the user to turn it on.

Definition in file `give_drag_and_drop.c`.

6.4.2 Define Documentation

6.4.2.1 `#define _ (String) gettext(String)`

Definition at line 43 of file `give_drag_and_drop.c`.

Referenced by `give_drag_and_drop_create_dialog()`, `give_drag_and_drop_turn_bt_on_dialog()`, `give_statusbar_plugin_about_dialog()`, `give_statusbar_plugin_clicked()`, `give_statusbar_plugin_item_clicked()`, `give_statusbar_plugin_pand_dialog()`, `give_statusbar_plugin_start_pand()`, and `give_statusbar_plugin_stop_pand()`.

6.4.2.2 `#define BLUEZ_DBUS_INTERFACE "org.bluez.Adapter"`

Definition at line 76 of file `give_drag_and_drop.c`.

Referenced by `give_drag_and_drop_init()`.

6.4.2.3 `#define BLUEZ_DBUS_MESSAGE "connectable"`

Definition at line 78 of file `give_drag_and_drop.c`.

Referenced by `give_drag_and_drop_turn_bt_on_dialog()`.

6.4.2.4 `#define BLUEZ_DBUS_PATH "/org/bluez/hci0"`

Definition at line 75 of file `give_drag_and_drop.c`.

Referenced by `give_drag_and_drop_init()`.

6.4.2.5 `#define BLUEZ_DBUS_SERVICE "org.bluez"`

Definition at line 74 of file `give_drag_and_drop.c`.

Referenced by `give_drag_and_drop_init()`.

6.4.2.6 #define BLUEZ_DBUS_SET_MODE "SetMode"

Definition at line 77 of file give_drag_and_drop.c.

Referenced by give_drag_and_drop_turn_bt_on_dialog().

6.4.2.7 #define CALENDAR_EVENT_TYPE "application/x-e-calendar-event"

Definition at line 88 of file give_drag_and_drop.c.

6.4.2.8 #define CONTACT_LIST_TYPE "x-osso-contact-list"

Definition at line 89 of file give_drag_and_drop.c.

6.4.2.9 #define GIVE_DRAG_AND_DROP_GET_PRIVATE(obj)

Value:

```
(G_TYPE_INSTANCE_GET_PRIVATE ((obj), \
                               GIVE_DRAG_AND_DROP_TYPE, \
                               give_drag_and_drop_private));
```

Definition at line 47 of file give_drag_and_drop.c.

Referenced by give_drag_and_drop_create_dialog(), give_drag_and_drop_drag_recieved(), give_drag_and_drop_finalize(), give_drag_and_drop_have_bt(), give_drag_and_drop_init(), give_drag_and_drop_run_dialog(), give_drag_and_drop_send(), give_drag_and_drop_set_dest(), and give_drag_and_drop_turn_bt_on_dialog().

6.4.2.10 #define MODEST_EMAIL_TYPE "text/modest-email"

Definition at line 93 of file give_drag_and_drop.c.

6.4.2.11 #define PLAIN_TEXT_TYPE "text/plain"

Definition at line 86 of file give_drag_and_drop.c.

6.4.2.12 #define TEMP_DIR "/tmp"

Definition at line 69 of file give_drag_and_drop.c.

Referenced by give_drag_and_drop_run_dialog().

6.4.2.13 #define TREE_AS_STRING_TYPE "text/tree-path-as-string-list"

Definition at line 92 of file give_drag_and_drop.c.

6.4.2.14 #define TREE_ROW_TYPE "GTK_TREE_MODEL_ROW"

Definition at line 91 of file give_drag_and_drop.c.

6.4.2.15 #define URI_LIST_TYPE "text/uri-list"

Definition at line 87 of file give_drag_and_drop.c.

6.4.3 Typedef Documentation

6.4.3.1 typedef struct _give_drag_and_drop_private give_drag_and_drop_private

Definition at line 45 of file give_drag_and_drop.c.

6.4.4 Enumeration Type Documentation

6.4.4.1 anonymous enum

Enumerator:

```
PLAIN_TEXT
URI_LIST
CALENDAR_EVENT
CONTACT_LIST
TREE_ROW
TREE_AS_STRING
MODEST_EMAIL
```

Definition at line 99 of file give_drag_and_drop.c.

6.4.5 Function Documentation

6.4.5.1 static void give_drag_and_drop_class_init (give_drag_and_dropClass **klass*) [static]

Overrides the class finalize function and adds the private structure.

Sets [give_drag_and_drop_finalize](#) as the class' finalize function.

Registers the [give_drag_and_drop_private](#) structure as private member.

Parameters:

klass a [give_drag_and_dropClass](#).

Definition at line 214 of file give_drag_and_drop.c.

References [give_drag_and_drop_finalize\(\)](#).

Referenced by [give_drag_and_drop_get_type\(\)](#).

Here is the call graph for this function:

Here is the caller graph for this function:

```
6.4.5.2 static GtkWidget * give_drag_and_drop_create_dialog
(give_drag_and_drop * self, const guint info, const gchar * text, guint *
file_count, GtkWidget ** entry) [static]
```

Creates a confirmation dialog with whether to send the data or not.

The created dialog will be run by [give_drag_and_drop_run_dialog](#).

The dialog contains the filenames.

- Text: default: note.txt, changeable.
- File: The filenames.
- Calendar event: default: summary field, with .vcs extension, changeable.
- Contacts: The names of contacts.

Todo

Look for memory leaks.

Parameters:

self A [give_drag_and_drop](#) object.

info The type of the dragged data.

text The dragged data.

file_count Number of files or contacs processed. Used later for memory allocation.

entry Pointer to the file name entry in the text and calendar ceses. Used by [give_drag_and_drop_run_dialog](#) to get the value.

Returns:

A GtkWidget, the created dialog.

Definition at line 405 of file [give_drag_and_drop.c](#).

References [_](#), [CALENDAR_EVENT](#), [CONTACT_LIST](#), [GIVE_DRAG_AND_DROP_GET_PRIVATE](#), [GIVE_DRAG_AND_DROP_IS_OBJECT](#), [give_ebook_get_contact_name\(\)](#), [_give_drag_and_drop_private::give_ebook_object](#), [PLAIN_TEXT](#), and [URI_LIST](#).

Referenced by [give_drag_and_drop_send\(\)](#).

Here is the call graph for this function:

Here is the caller graph for this function:

6.4.5.3 `static void give_drag_and_drop_drag_recieved (GtkWidget * widget, GdkDragContext * context, gint x, gint y, GtkSelectionData * selection_data, guint info, guint32 time, gpointer user_data) [static]`

Handles the "drag_data_received" signal.

- Turn on the Bluetooth adapter if it's off with [give_drag_and_drop_turn_bt_on_dialog](#).
- Invokes [give_drag_and_drop_send](#) to do the job.

Parameters:

widget The widget which got the data. It's the button of the plugin.
context ???
x ???
y ???
selection_data The dragged data.
info The type of the dragged data.
time ???
user_data ???

Todo

Investigate whether the got data needs to freed or not. probably yes, since the data is copied (GDK_ACTION_COPY)

Definition at line 857 of file give_drag_and_drop.c.

References [_give_drag_and_drop_private::bt_is_on](#), [GIVE_DRAG_AND_DROP_GET_PRIVATE](#), [GIVE_DRAG_AND_DROP_IS_OBJECT](#), [give_drag_and_drop_send\(\)](#), [give_drag_and_drop_turn_bt_on_dialog\(\)](#), [MODEST_EMAIL](#), [TREE_AS_STRING](#), and [TREE_ROW](#).

Referenced by [give_drag_and_drop_set_dest\(\)](#).

Here is the call graph for this function:

Here is the caller graph for this function:

6.4.5.4 `static void give_drag_and_drop_finalize (GObject * self) [static]`

Unsets the widget as d&d destination, unrefs objects.

- Unreferences [_give_drag_and_drop_private::dbus_proxy](#).
- Unsets the Give button not to be a d&d destination anymore.
- Disconnects the "drag_data_received" signal with it's handler: [give_drag_and_drop_drag_recieved](#).

- Unreferences `_give_drag_and_drop_private::give_bt_object`.
- Unreferences `_give_drag_and_drop_private::give_ebook_object`.

Definition at line 283 of file `give_drag_and_drop.c`.

References `_give_drag_and_drop_private::dbus_proxy`, `_give_drag_and_drop_private::destination`, `_give_drag_and_drop_private::give_bt_object`, `GIVE_DRAG_AND_DROP_GET_PRIVATE`, `GIVE_DRAG_AND_DROP_IS_OBJECT`, `_give_drag_and_drop_private::give_ebook_object`, and `_give_drag_and_drop_private::handler_id`.

Referenced by `give_drag_and_drop_class_init()`.

Here is the caller graph for this function:

6.4.5.5 GType `give_drag_and_drop_get_type (void)`

Registers the `give_drag_and_drop` type statically.

Sets class init, and instance init functions.

The class finalize function has been set in the class init function.

Returns:

The new registered GType, with the name of "give_drag_and_drop".

Definition at line 177 of file `give_drag_and_drop.c`.

References `give_drag_and_drop_class_init()`, and `give_drag_and_drop_init()`.

Here is the call graph for this function:

6.4.5.6 `void give_drag_and_drop_have_bt (give_drag_and_drop * self, gboolean setting)`

Sets `_give_drag_and_drop_private::bt_is_on`.

Parameters:

self A `give_drag_and_drop` object.

setting Sets the private member `_give_drag_and_drop_private::bt_is_on` according to setting.

Definition at line 371 of file `give_drag_and_drop.c`.

References `_give_drag_and_drop_private::bt_is_on`, `GIVE_DRAG_AND_DROP_GET_PRIVATE`, and `GIVE_DRAG_AND_DROP_IS_OBJECT`.

Referenced by `give_drag_and_drop_init()`, and `give_statusbar_plugin_update_status()`.

Here is the caller graph for this function:

6.4.5.7 static void give_drag_and_drop_init (give_drag_and_drop * *self*) [static]

Connects to Dbus and instantiates [give_bluetooth](#) and [give_ebook](#).

- Connects to BLUEZ_DBUS to turn on the Bluetooth adapter if needed.
- Instantiates [give_bluetooth](#) to send files.
- Instantiates [give_ebook](#) to get contact data.

Parameters:

self a [give_drag_and_drop](#) object.

Definition at line 234 of file [give_drag_and_drop.c](#).

References [BLUEZ_DBUS_INTERFACE](#), [BLUEZ_DBUS_PATH](#), [BLUEZ_DBUS_SERVICE](#), [_give_drag_and_drop_private::dbus_proxy](#), [give_bluetooth_new\(\)](#), [_give_drag_and_drop_private::give_bt_object](#), [GIVE_DRAG_AND_DROP_GET_PRIVATE](#), [give_drag_and_drop_have_bt\(\)](#), [GIVE_DRAG_AND_DROP_IS_OBJECT](#), [give_ebook_new\(\)](#), and [_give_drag_and_drop_private::give_ebook_object](#).

Referenced by [give_drag_and_drop_get_type\(\)](#).

Here is the call graph for this function:

Here is the caller graph for this function:

6.4.5.8 give_drag_and_drop * give_drag_and_drop_new (GtkWidget * *widget*)

Creates a new instance of [give_drag_and_drop](#).

Sets the widget parameter as a d&d destination with [give_drag_and_drop_set_dest](#).

Parameters:

widget A widget, it's the button of the plugin.

Returns:

A new [give_drag_and_drop](#).

Definition at line 314 of file [give_drag_and_drop.c](#).

References [give_drag_and_drop_set_dest\(\)](#), and [GIVE_DRAG_AND_DROP_TYPE](#).

Referenced by [give_statusbar_plugin_init\(\)](#).

Here is the call graph for this function:

Here is the caller graph for this function:

6.4.5.9 `static gint give_drag_and_drop_run_dialog (give_drag_and_drop * self, GtkWidget * dialog, const guint info, const gchar * text, const guint file_count, gchar *** files, const GtkWidget * entry)` [static]

Runs the dialog created by [give_drag_and_drop_create_dialog](#).

Allocates a stringlist containing the list of filenames to be send. The filenames are absolute since the Bluetooth sending call needs it.

The temporary files at the text, calendar event and contact cases are stored under [TEMP_DIR](#). They are created by printing the formatted data into the appropriate files.

Parameters:

self A [give_drag_and_drop](#) object.

dialog The dialog, created by [give_drag_and_drop_create_dialog](#).

info The type of the dragged data.

text The dragged data.

file_count Number of files or contacts processed. Used for memory allocation.

files Returned filenames, with absolute paths.

entry Pointer to the file name entry in the text and calendar cases. Used by [give_drag_and_drop_run_dialog](#) to get the value.

Returns:

1 on approval, 0 if the user cancels the send.

Definition at line 591 of file [give_drag_and_drop.c](#).

References [CALENDAR_EVENT](#), [CONTACT_LIST](#), [GIVE_DRAG_AND_DROP_GET_PRIVATE](#), [GIVE_DRAG_AND_DROP_IS_OBJECT](#), [give_ebook_get_contact_name\(\)](#), [give_ebook_get_contact_vcard\(\)](#), [_give_drag_and_drop_private::give_ebook_object](#), [PLAIN_TEXT](#), [TEMP_DIR](#), and [URI_LIST](#).

Referenced by [give_drag_and_drop_send\(\)](#).

Here is the call graph for this function:

Here is the caller graph for this function:

6.4.5.10 `static void give_drag_and_drop_send (give_drag_and_drop * self, const guint info, const gchar * text)` [static]

Creates a confirmation dialog and send the files.

Creates and runs a confirmation dialog with [give_drag_and_drop_create_dialog](#) and [give_drag_and_drop_run_dialog](#).

Send the files with the [give_bluetooth_send_file](#) method of the [give_bluetooth](#) object.

Parameters:

- self* A [give_drag_and_drop](#) object.
- info* The type of the dragged data.
- text* The dragged data.

Todo

Remove temporary files after sending.

Definition at line 741 of file [give_drag_and_drop.c](#).

References [give_bluetooth_send_file\(\)](#), [_give_drag_and_drop_private::give_bt_object](#), [give_drag_and_drop_create_dialog\(\)](#), [GIVE_DRAG_AND_DROP_GET_PRIVATE](#), [GIVE_DRAG_AND_DROP_IS_OBJECT](#), and [give_drag_and_drop_run_dialog\(\)](#).

Referenced by [give_drag_and_drop_drag_recieved\(\)](#).

Here is the call graph for this function:

Here is the caller graph for this function:

6.4.5.11 static void [give_drag_and_drop_set_dest](#) ([give_drag_and_drop](#) * *self*, [GtkWidget](#) * *widget*) [static]

Sets the parameter widget a d&d destination.

- Sets the parameter widget to be highlighted for the accepted d&d types, defined in [drop_types](#).
- Connects [give_drag_and_drop_drag_recieved](#) to "drag_data_received" signal and stores the handler ID so it can be disconnected.

Definition at line 338 of file [give_drag_and_drop.c](#).

References [_give_drag_and_drop_private::destination](#), [drop_types](#), [give_drag_and_drop_drag_recieved\(\)](#), [GIVE_DRAG_AND_DROP_GET_PRIVATE](#), [GIVE_DRAG_AND_DROP_IS_OBJECT](#), and [_give_drag_and_drop_private::handler_id](#).

Referenced by [give_drag_and_drop_new\(\)](#).

Here is the call graph for this function:

Here is the caller graph for this function:

6.4.5.12 gboolean give_drag_and_drop_turn_bt_on_dialog (give_drag_and_drop * self)

Confirmation dialog whether to turn on the Bluetooth adapter or not.

Parameters:

self A `give_drag_and_drop` object.

Definition at line 778 of file `give_drag_and_drop.c`.

References `_`, `BLUEZ_DBUS_MESSAGE`, `BLUEZ_DBUS_SET_MODE`, `_give_drag_and_drop_private::dbus_proxy`, `GIVE_DRAG_AND_DROP_GET_PRIVATE`, and `GIVE_DRAG_AND_DROP_IS_OBJECT`.

Referenced by `give_drag_and_drop_drag_recieved()`, and `give_statusbar_plugin_pand_dialog()`.

Here is the caller graph for this function:

6.4.6 Variable Documentation

6.4.6.1 const GtkTargetEntry drop_types[]

Initial value:

```
{
    { PLAIN_TEXT_TYPE,      0, PLAIN_TEXT },
    { URI_LIST_TYPE,       0, URI_LIST },
    { CALENDAR_EVENT_TYPE, 0, CALENDAR_EVENT },
    { CONTACT_LIST_TYPE,   0, CONTACT_LIST },
    { TREE_ROW_TYPE,       0, TREE_ROW },
    { TREE_AS_STRING_TYPE, 0, TREE_AS_STRING },
    { MODEST_EMAIL_TYPE,   0, MODEST_EMAIL }
}
```

Accepted d&d types.

- Text: [PLAIN_TEXT_TYPE](#).
- Files: [URI_LIST_TYPE](#).
- Calendar event: [CALENDAR_EVENT_TYPE](#).
- Contact: [CONTACT_LIST_TYPE](#).
- E-mail: [MODEST_EMAIL_TYPE](#).

Note:

`modest` is still under heavy hacking, so the mime type may change.

Definition at line 118 of file `give_drag_and_drop.c`.

Referenced by `give_drag_and_drop_set_dest()`.

6.5 src/give_drag_and_drop.h File Reference

```
#include <glib-object.h>
```

```
#include <gtk/gtk.h>
```

Include dependency graph for give_drag_and_drop.h:

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

Classes

- struct [_give_drag_and_drop](#)
- struct [_give_drag_and_dropClass](#)

Defines

- #define [GIVE_DRAG_AND_DROP_TYPE](#) (give_drag_and_drop_get_type ())
- #define [GIVE_DRAG_AND_DROP](#)(obj)
- #define [GIVE_DRAG_AND_DROP_CLASS](#)(klass)
- #define [GIVE_DRAG_AND_DROP_IS_OBJECT](#)(obj)
- #define [GIVE_DRAG_AND_DROP_IS_CLASS](#)(klass)

Typedefs

- typedef typedefG_BEGIN_DECLS struct [_give_drag_and_drop](#) give_drag_and_drop
- typedef struct [_give_drag_and_dropClass](#) give_drag_and_dropClass

Functions

- GType [give_drag_and_drop_get_type](#) (void) G_GNUC_CONST
- [give_drag_and_drop](#) * [give_drag_and_drop_new](#) (GtkWidget *widget)
- void [give_drag_and_drop_have_bt](#) ([give_drag_and_drop](#) *self, gboolean setting)
- gboolean [give_drag_and_drop_turn_bt_on_dialog](#) ([give_drag_and_drop](#) *self)

6.5.1 Define Documentation

6.5.1.1 #define GIVE_DRAG_AND_DROP(obj)

Value:

```
(G_TYPE_CHECK_INSTANCE_CAST ((obj), \
                               GIVE_DRAG_AND_DROP_TYPE, \
                               give_drag_and_drop))
```

Definition at line 32 of file give_drag_and_drop.h.

6.5.1.2 #define GIVE_DRAG_AND_DROP_CLASS(klass)**Value:**

```
(G_TYPE_CHECK_CLASS_CAST ((klass), \
                          GIVE_DRAG_AND_DROP_TYPE, \
                          give_drag_and_dropClass))
```

Definition at line 35 of file give_drag_and_drop.h.

6.5.1.3 #define GIVE_DRAG_AND_DROP_IS_CLASS(klass)**Value:**

```
(G_TYPE_CHECK_CLASS_TYPE ((klass), \
                          GIVE_DRAG_AND_DROP_TYPE))
```

Definition at line 42 of file give_drag_and_drop.h.

6.5.1.4 #define GIVE_DRAG_AND_DROP_IS_OBJECT(obj)**Value:**

```
(G_TYPE_CHECK_INSTANCE_TYPE ((obj), \
                              GIVE_DRAG_AND_DROP_TYPE))
```

Definition at line 39 of file give_drag_and_drop.h.

Referenced by give_drag_and_drop_create_dialog(), give_drag_and_drop_drag_recieved(), give_drag_and_drop_finalize(), give_drag_and_drop_have_bt(), give_drag_and_drop_init(), give_drag_and_drop_run_dialog(), give_drag_and_drop_send(), give_drag_and_drop_set_dest(), and give_drag_and_drop_turn_bt_on_dialog().

6.5.1.5 #define GIVE_DRAG_AND_DROP_TYPE (give_drag_and_drop_get_type ())

Definition at line 31 of file give_drag_and_drop.h.

Referenced by give_drag_and_drop_new().

6.5.2 Typedef Documentation**6.5.2.1 typedef typedefG_BEGIN_DECLS struct _give_drag_and_drop give_drag_and_drop**

Definition at line 28 of file give_drag_and_drop.h.

6.5.2.2 typedef struct _give_drag_and_dropClass give_drag_and_dropClass

Definition at line 29 of file give_drag_and_drop.h.

6.5.3 Function Documentation

6.5.3.1 GType give_drag_and_drop_get_type (void)

6.5.3.2 void give_drag_and_drop_have_bt (give_drag_and_drop * *self*,
gboolean *setting*)

6.5.3.3 give_drag_and_drop* give_drag_and_drop_new (GtkWidget * *widget*)

6.5.3.4 gboolean give_drag_and_drop_turn_bt_on_dialog
(give_drag_and_drop * *self*)

6.6 src/give_ebook.c File Reference

```
#include "give_ebook.h"
#include <string.h>
#include <libebook/e-book.h>
```

Include dependency graph for give_ebook.c:

Classes

- struct [_give_ebook_private](#)
Struct of private members of [give_ebook](#).

Defines

- #define [GIVE_EBOOK_GET_PRIVATE](#)(obj)

Typedefs

- typedef struct [_give_ebook_private](#) [give_ebook_private](#)

Functions

Private functions

- GType [give_ebook_get_type](#) (void)
Registers the [give_ebook](#) type statically.
- static void [give_ebook_class_init](#) ([give_ebookClass](#) *klass)
Overrides the class finalize function and adds the private structure.
- static void [give_ebook_init](#) ([give_ebook](#) *self)
Connects to and opens the default system addressbook.
- static void [give_ebook_finalize](#) (GObject *self)
Free private members.
- static void [give_ebook_open_response](#) (EBook *book, EBookStatus status, gpointer data)
Callback of asynchronous EBook opening.
- static void [give_ebook_status_to_string](#) (EBookStatus status, gchar *state_string)
Converts a EBookStatus to a string.

Public functions

- [give_ebook](#) * [give_ebook_new](#) (void)

Creates a new instance of `give_ebook`.

- `gchar * give_ebook_get_contact_name (give_ebook *self, const gchar *id)`
Get the contact's name, from the ID.
- `gchar * give_ebook_get_contact_vcard (give_ebook *self, const gchar *id)`
Get the string representation of the contact, from the ID.

6.6.1 Detailed Description

The EBook interacts with the Evolution Data Server's EBook component, for data queries, and VCard representations of the contacts.

Definition in file [give_ebook.c](#).

6.6.2 Define Documentation

6.6.2.1 #define GIVE_EBOOK_GET_PRIVATE(obj)

Value:

```
(G_TYPE_INSTANCE_GET_PRIVATE ((obj), \
                               GIVE_EBOOK_TYPE, \
                               give_ebook_private));
```

Definition at line 32 of file [give_ebook.c](#).

Referenced by [give_ebook_finalize\(\)](#), [give_ebook_get_contact_name\(\)](#), [give_ebook_get_contact_vcard\(\)](#), [give_ebook_init\(\)](#), and [give_ebook_open_response\(\)](#).

6.6.3 Typedef Documentation

6.6.3.1 typedef struct _give_ebook_private give_ebook_private

Definition at line 30 of file [give_ebook.c](#).

6.6.4 Function Documentation

6.6.4.1 static void give_ebook_class_init (give_ebookClass * *klass*) [static]

Overrides the class finalize function and adds the private structure.

Sets [give_ebook_finalize](#) as the class' finalize function.

Registers the [give_ebook_private](#) structure as private member.

Parameters:

klass a [give_ebookClass](#) class.

Definition at line 119 of file [give_ebook.c](#).

References [give_ebook_finalize\(\)](#).

Referenced by `give_ebook_get_type()`.

Here is the call graph for this function:

Here is the caller graph for this function:

6.6.4.2 `static void give_ebook_finalize (GObject * self) [static]`

Free private members.

Note:

No members needed to be closed jet.

Parameters:

self a GObject.

Note:

no need to close the ebook, nothing to be freed jet.

Definition at line 171 of file `give_ebook.c`.

References `GIVE_EBOOK_GET_PRIVATE`, and `GIVE_EBOOK_IS_OBJECT`.

Referenced by `give_ebook_class_init()`.

Here is the caller graph for this function:

6.6.4.3 `gchar * give_ebook_get_contact_name (give_ebook * self, const gchar * id)`

Get the contact's name, from the ID.

The the EBook is opened (`_give_ebook_private::ready = true`), gets an EContact with `e_book_get_contact` according to the ID, and queries the full name with invoking `e_contact_get (contact, E_CONTACT_FULL_NAME)`.

Parameters:

self A `give_ebook` object.

id The ID of the contact.

Returns:

The name of the contact.

Definition at line 347 of file give_ebook.c.

References `_give_ebook_private::ebook`, `GIVE_EBOOK_GET_PRIVATE`, `GIVE_EBOOK_IS_OBJECT`, and `_give_ebook_private::ready`.

Referenced by `give_drag_and_drop_create_dialog()`, and `give_drag_and_drop_run_dialog()`.

Here is the caller graph for this function:

6.6.4.4 `gchar * give_ebook_get_contact_vcard (give_ebook * self, const gchar * id)`

Get the string representation of the contact, from the ID.

The the EBook is opened (`_give_ebook_private::ready = true`), gets an EContact with `e_book_get_contact` according to the ID. The string representation available for EContact's parent object: E_VCard, so first the contact needs to be casted to E_VCard. Then the return string is generated with: `e_vcard_to_string`.

Parameters:

self A `give_ebook` object.

id The ID of the contact.

Returns:

String representation of a VCard.

Definition at line 391 of file give_ebook.c.

References `_give_ebook_private::ebook`, `GIVE_EBOOK_GET_PRIVATE`, `GIVE_EBOOK_IS_OBJECT`, and `_give_ebook_private::ready`.

Referenced by `give_drag_and_drop_run_dialog()`.

Here is the caller graph for this function:

6.6.4.5 `GType give_ebook_get_type (void)`

Registers the `give_ebook` type statically.

Sets class init, and instance init functions.

The class finalize function has been set in the class init function.

Returns:

The new registered GType, with the name of "give_ebook".

Definition at line 83 of file give_ebook.c.

References `give_ebook_class_init()`, and `give_ebook_init()`.

Here is the call graph for this function:

6.6.4.6 `static void give_ebook_init (give_ebook * self) [static]`

Connects to and opens the default system addressbook.

- Connects to the default system addressbook.
- Opens it asynchronously, by setting `give_ebook_open_response` as response handler.

Parameters:

self a `give_ebook` object.

Definition at line 139 of file `give_ebook.c`.

References `_give_ebook_private::ebook`, `GIVE_EBOOK_GET_PRIVATE`, `GIVE_EBOOK_IS_OBJECT`, and `give_ebook_open_response()`.

Referenced by `give_ebook_get_type()`.

Here is the call graph for this function:

Here is the caller graph for this function:

6.6.4.7 `give_ebook * give_ebook_new (void)`

Creates a new instance of `give_ebook`.

Returns:

A new `give_ebook` object.

Definition at line 193 of file `give_ebook.c`.

References `GIVE_EBOOK_TYPE`.

Referenced by `give_drag_and_drop_init()`.

Here is the caller graph for this function:

6.6.4.8 static void give_ebook_open_response (EBook * *book*, EBookStatus *status*, gpointer *data*) [static]

Callback of asynchronous EBook opening.

Converts the status to string with `give_ebook_status_to_string` and prints it. If the status is `E_BOOK_ERROR_OK` (success), sets the `_give_ebook_private::ready` private member to false or true, according to it.

Parameters:

book The opened EBook.

status The status returned by opening.

data User data passed to `give_ebook_open_response`. It is the `give_ebook_object`.

Definition at line 218 of file `give_ebook.c`.

References `GIVE_EBOOK_GET_PRIVATE`, `give_ebook_status_to_string()`, and `_give_ebook_private::ready`.

Referenced by `give_ebook_init()`.

Here is the call graph for this function:

Here is the caller graph for this function:

6.6.4.9 static void give_ebook_status_to_string (EBookStatus *status*, gchar * *state_string*) [static]

Converts a EBookStatus to a string.

Converts an emun to a readable string.

Parameters:

status An EBookStatus code.

state_string The return string, memory has not been allocated, it needs to be handled in the caller side.

Definition at line 253 of file `give_ebook.c`.

Referenced by `give_ebook_open_response()`.

Here is the caller graph for this function:

6.7 src/give_ebook.h File Reference

```
#include <glib-object.h>
```

Include dependency graph for give_ebook.h:

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

Classes

- struct [_give_ebook](#)
- struct [_give_ebookClass](#)

Defines

- #define [GIVE_EBOOK_TYPE](#) (give_ebook_get_type ())
- #define [GIVE_EBOOK](#)(obj)
- #define [GIVE_EBOOK_CLASS](#)(klass)
- #define [GIVE_EBOOK_IS_OBJECT](#)(obj)
- #define [GIVE_EBOOK_IS_CLASS](#)(klass)

Typedefs

- typedef typedefG_BEGIN_DECLS struct [_give_ebook](#) [give_ebook](#)
- typedef struct [_give_ebookClass](#) [give_ebookClass](#)

Functions

- GType [give_ebook_get_type](#) (void) G_GNUC_CONST
- [give_ebook](#) * [give_ebook_new](#) (void)
- gchar * [give_ebook_get_contact_name](#) ([give_ebook](#) *self, const gchar *id)
- gchar * [give_ebook_get_contact_vcard](#) ([give_ebook](#) *self, const gchar *id)

6.7.1 Define Documentation

6.7.1.1 #define GIVE_EBOOK(obj)

Value:

```
(G_TYPE_CHECK_INSTANCE_CAST ((obj), \
                               GIVE_EBOOK_TYPE, give_ebook))
```

Definition at line 30 of file give_ebook.h.

6.7.1.2 #define GIVE_EBOOK_CLASS(klass)

Value:

```
(G_TYPE_CHECK_CLASS_CAST ((klass), \
                           GIVE_EBOOK_TYPE, give_ebookClass))
```

Definition at line 32 of file give_ebook.h.

6.7.1.3 #define GIVE_EBOOK_IS_CLASS(klass)

Value:

```
(G_TYPE_CHECK_CLASS_TYPE ((klass), \
                           GIVE_EBOOK_TYPE))
```

Definition at line 36 of file give_ebook.h.

6.7.1.4 #define GIVE_EBOOK_IS_OBJECT(obj)

Value:

```
(G_TYPE_CHECK_INSTANCE_TYPE ((obj), \
                              GIVE_EBOOK_TYPE))
```

Definition at line 34 of file give_ebook.h.

Referenced by give_ebook_finalize(), give_ebook_get_contact_name(), give_ebook_get_contact_vcard(), and give_ebook_init().

6.7.1.5 #define GIVE_EBOOK_TYPE (give_ebook_get_type ())

Definition at line 29 of file give_ebook.h.

Referenced by give_ebook_new().

6.7.2 Typedef Documentation

6.7.2.1 typedef typedefG_BEGIN_DECLS struct _give_ebook give_ebook

Definition at line 26 of file give_ebook.h.

6.7.2.2 typedef struct _give_ebookClass give_ebookClass

Definition at line 27 of file give_ebook.h.

6.7.3 Function Documentation

6.7.3.1 `gchar* give_ebook_get_contact_name (give_ebook * self, const gchar * id)`

6.7.3.2 `gchar* give_ebook_get_contact_vcard (give_ebook * self, const gchar * id)`

6.7.3.3 `GType give_ebook_get_type (void)`

6.7.3.4 `give_ebook* give_ebook_new (void)`

6.8 src/give_statusbar_plugin.c File Reference

```
#include "config.h"
#include "give_statusbar_plugin.h"
#include "give_drag_and_drop.h"
#include <locale.h>
#include <libintl.h>
#include <dbus/dbus-glib.h>
#include <gconf/gconf-client.h>
#include <hildon/hildon-banner.h>
```

Include dependency graph for give_statusbar_plugin.c:

Classes

- struct [_GiveStatusbarPluginPrivate](#)
Struct of private members of [GiveStatusbarPlugin](#).

Defines

- #define [_\(String\)](#) gettext(String)
- #define [GIVE_STATUSBAR_PLUGIN_GET_PRIVATE\(obj\)](#)
- #define [BT_DIR](#) "/system/osso/connectivity/BT"
- #define [BT_KEY](#) "/system/osso/connectivity/BT/enabled"
- #define [LOG_FILE](#) "/tmp/give_log.txt"

Typedefs

- typedef struct [_GiveStatusbarPluginPrivate](#) [GiveStatusbarPluginPrivate](#)

Enumerations

- enum [bt_state](#) { [BT_IS_OFF](#), [BT_IS_ON](#), [BT_PAND_IS_ACTIVE](#) }
The states of the plugin.
- enum [menu_items](#) {
[NOTE_ITEM](#), [FILE_ITEM](#), [APPOINTMENT_ITEM](#), [CONTACT_ITEM](#),
[EMAIL_ITEM](#), [ABOUT_ITEM](#) }
Enumeration of the menu of Give, except the Net access.

Functions

- [HD_DEFINE_PLUGIN](#) ([GiveStatusbarPlugin](#), [give_statusbar_plugin](#), [STATUSBAR_TYPE_ITEM](#))

Private functions

- static void [give_statusbar_plugin_class_init](#) ([GiveStatusbarPluginClass](#) *klass)
Overrides the class finalize function and adds the private structure.
- static void [give_statusbar_plugin_init](#) ([GiveStatusbarPlugin](#) *self)
Initialize the plugin.
- static void [give_statusbar_plugin_finalize](#) (GObject *self)
Unreferences the private members.
- static void [give_statusbar_plugin_popup_place](#) (GtkMenu *menu, gint *x, gint *y, gboolean *push_in, gpointer user_data)
Calculates the popup place of the Give menu.
- static void [give_statusbar_plugin_update_status](#) ([GiveStatusbarPlugin](#) *self)
Update the icon of plugin according to the Bluetooth status.
- static void [give_statusbar_plugin_clicked](#) (GtkButton *button, [GiveStatusbarPlugin](#) *self)
Builds up the popup menu of Give and show it.
- static void [give_statusbar_plugin_item_clicked](#) (GtkWidget *menu_item, guint item)
Handles the clicks in the popup menu.
- static void [give_statusbar_plugin_about_dialog](#) (void)
Shows the about dialog about Give.
- static void [give_statusbar_plugin_log_handler](#) (const gchar *log_domain, GLogLevelFlags log_level, const gchar *message, gpointer user_data)
Give's own log handler.
- static void [give_statusbar_plugin_bt_changed](#) (GConfClient *gc_client, guint cnxn_id, GConfEntry *entry, gpointer data)
Gets the Bluetooth adapter's state from GConf and sets the private [GiveStatusbarPluginPrivate::state](#) due to this.
- static void [give_statusbar_plugin_pand_dialog](#) (GtkMenuItem *menuitem, [GiveStatusbarPlugin](#) *self)
Creates and shows the stop/start pand dialog.
- static void [give_statusbar_plugin_start_pand](#) (GtkMenuItem *menuitem, [GiveStatusbarPlugin](#) *self)
Starts the pand daemon.
- static void [give_statusbar_plugin_stop_pand](#) ([GiveStatusbarPlugin](#) *self)
Kills the pand daemon.

6.8.1 Detailed Description

Statusbar plugin for easy drag & drop data sending via Bluetooth.

The icon is a d&d destination for specified types in [drop_types](#).

Pressing it invokes the Give menu and the icon itself represents the Bluetooth state [bt_state](#).

Definition in file [give_statusbar_plugin.c](#).

6.8.2 Define Documentation

6.8.2.1 `#define _(String) gettext(String)`

Definition at line 34 of file [give_statusbar_plugin.c](#).

6.8.2.2 `#define BT_DIR "/system/osso/connectivity/BT"`

Definition at line 48 of file [give_statusbar_plugin.c](#).

Referenced by [give_statusbar_plugin_finalize\(\)](#), and [give_statusbar_plugin_init\(\)](#).

6.8.2.3 `#define BT_KEY "/system/osso/connectivity/BT/enabled"`

boolean, TRUE if the Bluetooth adapter is on, FALSE otherwise.

Definition at line 50 of file [give_statusbar_plugin.c](#).

Referenced by [give_statusbar_plugin_bt_changed\(\)](#), and [give_statusbar_plugin_init\(\)](#).

6.8.2.4 `#define GIVE_STATUSBAR_PLUGIN_GET_PRIVATE(obj)`

Value:

```
(G_TYPE_INSTANCE_GET_PRIVATE ((obj), \
                               GIVE_TYPE_STATUSBAR_PLUGIN, \
                               GiveStatusbarPluginPrivate));
```

Definition at line 42 of file [give_statusbar_plugin.c](#).

Referenced by [give_statusbar_plugin_bt_changed\(\)](#), [give_statusbar_plugin_clicked\(\)](#), [give_statusbar_plugin_finalize\(\)](#), [give_statusbar_plugin_init\(\)](#), [give_statusbar_plugin_pand_dialog\(\)](#), [give_statusbar_plugin_start_pand\(\)](#), [give_statusbar_plugin_stop_pand\(\)](#), and [give_statusbar_plugin_update_status\(\)](#).

6.8.2.5 `#define LOG_FILE "/tmp/give_log.txt"`

Logfile used if compiled with `-enable-utf=yes`.

Definition at line 53 of file [give_statusbar_plugin.c](#).

Referenced by [give_statusbar_plugin_log_handler\(\)](#).

6.8.3 Typedef Documentation

6.8.3.1 typedef struct `_GiveStatusbarPluginPrivate` `GiveStatusbarPluginPrivate`

Definition at line 40 of file `give_statusbar_plugin.c`.

6.8.4 Enumeration Type Documentation

6.8.4.1 enum `bt_state`

The states of the plugin.

Enumerator:

```
BT_IS_OFF  
BT_IS_ON  
BT_PAND_IS_ACTIVE
```

Definition at line 81 of file `give_statusbar_plugin.c`.

6.8.4.2 enum `menu_items`

Enumeration of the menu of Give, except the Net access.

Enumerator:

```
NOTE_ITEM  
FILE_ITEM  
APPOINTMENT_ITEM  
CONTACT_ITEM  
EMAIL_ITEM  
ABOUT_ITEM
```

Definition at line 88 of file `give_statusbar_plugin.c`.

6.8.5 Function Documentation

6.8.5.1 static void `give_statusbar_plugin_about_dialog(void)` [static]

Shows the about dialog about Give.

Definition at line 564 of file `give_statusbar_plugin.c`.

References `_`.

Referenced by `give_statusbar_plugin_item_clicked()`.

Here is the caller graph for this function:

6.8.5.2 static void give_statusbar_plugin_bt_changed (GConfClient * *gc_client*, guint *cnxn_id*, GConfEntry * *entry*, gpointer *data*) [static]

Gets the Bluetooth adapter's state from GConf and sets the private `_GiveStatusbarPluginPrivate::state` due to this.

Parameters:

- gc_client* The GConfClient notifying us.
- cnxn_id* Connection ID from `gconf_client_notify_add()`. Not used.
- entry* A GConfEntry. Not used.
- data* User data from `gconf_client_notify_add()`. It's a `GiveStatusbarPlugin` object.

Definition at line 675 of file `give_statusbar_plugin.c`.

References `BT_IS_OFF`, `BT_IS_ON`, `BT_KEY`, `GIVE_IS_STATUSBAR_PLUGIN`, `GIVE_STATUSBAR_PLUGIN_GET_PRIVATE`, `give_statusbar_plugin_stop_pand()`, `give_statusbar_plugin_update_status()`, `_GiveStatusbarPluginPrivate::pand_pid`, and `_GiveStatusbarPluginPrivate::state`.

Referenced by `give_statusbar_plugin_init()`.

Here is the call graph for this function:

Here is the caller graph for this function:

6.8.5.3 static void give_statusbar_plugin_class_init (GiveStatusbarPluginClass * *klass*) [static]

Overrides the class finalize function and adds the private structure.

Sets `give_statusbar_plugin_finalize` as the class' finalize function.

Registers the `GiveStatusbarPluginPrivate` structure as private member.

Parameters:

- klass* a `GiveStatusbarPluginClass` class.

Definition at line 138 of file `give_statusbar_plugin.c`.

References `give_statusbar_plugin_finalize()`.

Here is the call graph for this function:

6.8.5.4 static void give_statusbar_plugin_clicked (GtkButton * *button*, GiveStatusbarPlugin * *self*) [static]

Builds up the popup menu of Give and show it.

Parameters:

- button* The button of the plugin.
- self* A [GiveStatusBarPlugin](#) object.

Definition at line 361 of file `give_statusbar_plugin.c`.

References `_`, `ABOUT_ITEM`, `APPOINTMENT_ITEM`, `BT_PAND_IS_ACTIVE`, `CONTACT_ITEM`, `EMAIL_ITEM`, `FILE_ITEM`, `GIVE_IS_STATUSBAR_PLUGIN`, `GIVE_STATUSBAR_PLUGIN_GET_PRIVATE`, `give_statusbar_plugin_item_clicked()`, `give_statusbar_plugin_pand_dialog()`, `give_statusbar_plugin_popup_place()`, `NOTE_ITEM`, and `_GiveStatusBarPluginPrivate::state`.

Referenced by `give_statusbar_plugin_init()`.

Here is the call graph for this function:

Here is the caller graph for this function:

6.8.5.5 `static void give_statusbar_plugin_finalize (GObject * self) [static]`

Unreferences the private members.

- Unreferences the private `_GiveStatusBarPluginPrivate::dnd` object.
- Stops listen to the changes in `BT_DIR`.
- Unreferences the GConf connection.

Parameters:

- self* a `GObject`.

Definition at line 240 of file `give_statusbar_plugin.c`.

References `BT_DIR`, `_GiveStatusBarPluginPrivate::dnd`, `_GiveStatusBarPluginPrivate::gc_client`, `GIVE_IS_STATUSBAR_PLUGIN`, `GIVE_STATUSBAR_PLUGIN_GET_PRIVATE`, and `_GiveStatusBarPluginPrivate::notify_uid`.

Referenced by `give_statusbar_plugin_class_init()`.

Here is the caller graph for this function:

6.8.5.6 `static void give_statusbar_plugin_init (GiveStatusBarPlugin * self) [static]`

Initialize the plugin.

- Sets the localization.

- Overwrites the glib log handler with [give_statusbar_plugin_log_handler](#).
- Creates the button of the plugin with image and callback.
- Makes the button a d&d destination.
- Listens to the state of the Bluetooth adapter with GConf.

Parameters:

self A [GiveStatusBarPlugin](#) object.

Definition at line 161 of file `give_statusbar_plugin.c`.

References `BT_DIR`, `BT_IS_OFF`, `BT_KEY`, `_GiveStatusBarPluginPrivate::button`, `_GiveStatusBarPluginPrivate::dnd`, `_GiveStatusBarPluginPrivate::gc_client`, `give_drag_and_drop_new()`, `GIVE_IS_STATUSBAR_PLUGIN`, `give_statusbar_plugin_bt_changed()`, `give_statusbar_plugin_clicked()`, `GIVE_STATUSBAR_PLUGIN_GET_PRIVATE`, `give_statusbar_plugin_log_handler()`, `_GiveStatusBarPluginPrivate::image`, `_GiveStatusBarPluginPrivate::notify_uid`, `_GiveStatusBarPluginPrivate::pand_pid`, and `_GiveStatusBarPluginPrivate::state`.

Here is the call graph for this function:

6.8.5.7 `static void give_statusbar_plugin_item_clicked (GtkWidget * menu_item, guint item)` [static]

Handles the clicks in the popup menu.

Invokes the programs for:

- Text: `osso_notes`.
- File: `osso_filemanager`.
- Calendar event: `dates`.
- Contact: `osso-addressbook`.
- E-mails: `ossoemail`.

Parameters:

menu_item MenuItem recieved the "activate" signal.

item Which menuitem got selected. Enumerated with [menu_items](#).

Definition at line 496 of file `give_statusbar_plugin.c`.

References `_`, `ABOUT_ITEM`, `APPOINTMENT_ITEM`, `CONTACT_ITEM`, `EMAIL_ITEM`, `FILE_ITEM`, `give_statusbar_plugin_about_dialog()`, and `NOTE_ITEM`.

Referenced by `give_statusbar_plugin_clicked()`.

Here is the call graph for this function:

Here is the caller graph for this function:

6.8.5.8 `static void give_statusbar_plugin_log_handler (const gchar * log_domain, GLogLevelFlags log_level, const gchar * message, gpointer user_data) [static]`

Give's own log handler.

- If Give is compiled with `-enable-ltc` it will print colorful log messages to the console.
- If Give is compiled with `-enable-ltf` it will print log messages to [LOG_FILE](#).

Parameters:

log_domain Not used.
log_level The nature of the log message..
message The log message.
user_data No used.

Definition at line 595 of file `give_statusbar_plugin.c`.

References [LOG_FILE](#).

Referenced by `give_statusbar_plugin_init()`.

Here is the caller graph for this function:

6.8.5.9 `static void give_statusbar_plugin_pand_dialog (GtkMenuItem * menuitem, GiveStatusbarPlugin * self) [static]`

Creates and shows the stop/start pand dialog.

According to [_GiveStatusbarPluginPrivate::state](#) it stops pand if it's running and starts if it's not.

Invokes the [_GiveStatusbarPluginPrivate::dnd](#) object's `give_drag_and_drop_turn_bt_on_dialog` if the Bluetooth is off.

Parameters:

menuitem The menuitem recieved the "activate" signal.
self A [GiveStatusbarPlugin](#) object.

Definition at line 746 of file `give_statusbar_plugin.c`.

References `_`, `BT_IS_OFF`, `BT_PAND_IS_ACTIVE`, `_GiveStatusbarPluginPrivate::dnd`, `give_drag_and_drop_turn_bt_on_dialog()`, `GIVE_IS_STATUSBAR_PLUGIN`, `GIVE_STATUSBAR_PLUGIN_GET_PRIVATE`, `give_statusbar_plugin_start_pand()`, `give_statusbar_plugin_stop_pand()`, and `_GiveStatusbarPluginPrivate::state`.

Referenced by `give_statusbar_plugin_clicked()`.

Here is the call graph for this function:

Here is the caller graph for this function:

6.8.5.10 `static void give_statusbar_plugin_popup_place (GtkMenu * menu, gint * x, gint * y, gboolean * push_in, gpointer user_data) [static]`

Calculates the popup place of the Give menu.

Parameters:

menu A GtkMenu.

x A address of the gint representing the horizontal position where the menu shall be drawn. This is an output parameter.

y A address of the gint representing the vertical position where the menu shall be drawn. This is an output parameter.

push_in This parameter controls how menus placed outside the monitor are handled. Not used.

user_data The data supplied by the user in the gtk_menu_popup() data parameter.

Definition at line 282 of file give_statusbar_plugin.c.

Referenced by give_statusbar_plugin_clicked().

Here is the caller graph for this function:

6.8.5.11 `static void give_statusbar_plugin_start_pand (GtkMenuItem * menuitem, GiveStatusbarPlugin * self) [static]`

Starts the pand daemon.

Stores the PID in `_GiveStatusbarPluginPrivate::pand_pid` so it can be killed later with `give_statusbar_plugin_stop_pand`.

Todo

The device don't act as a server just as a client.

Parameters:

menuitem The menuitem recieved the "activate" signal.

self A GiveStatusbarPlugin object.

Definition at line 818 of file give_statusbar_plugin.c.

References `_`, `BT_PAND_IS_ACTIVE`, `GIVE_IS_STATUSBAR_PLUGIN`, `GIVE_STATUSBAR_PLUGIN_GET_PRIVATE`, `give_statusbar_plugin_update_status()`, `_GiveStatusbarPluginPrivate::pand_pid`, and `_GiveStatusbarPluginPrivate::state`.

Referenced by `give_statusbar_plugin_pand_dialog()`.

Here is the call graph for this function:

Here is the caller graph for this function:

6.8.5.12 `static void give_statusbar_plugin_stop_pand (GiveStatusbarPlugin * self) [static]`

Kills the pand daemon.

The PID of the process is stored in `_GiveStatusbarPluginPrivate::notify_uid`.

Parameters:

self [GiveStatusbarPlugin](#).

Definition at line 863 of file `give_statusbar_plugin.c`.

References `_`, `BT_IS_ON`, `GIVE_IS_STATUSBAR_PLUGIN`, `GIVE_STATUSBAR_PLUGIN_GET_PRIVATE`, `give_statusbar_plugin_update_status()`, `_GiveStatusbarPluginPrivate::pand_pid`, and `_GiveStatusbarPluginPrivate::state`.

Referenced by `give_statusbar_plugin_bt_changed()`, and `give_statusbar_plugin_pand_dialog()`.

Here is the call graph for this function:

Here is the caller graph for this function:

6.8.5.13 `static void give_statusbar_plugin_update_status (GiveStatusbarPlugin * self) [static]`

Update the icon of plugin according to the Bluetooth status.

Parameters:

self A [GiveStatusbarPlugin](#) object.

Definition at line 316 of file `give_statusbar_plugin.c`.

References `BT_IS_OFF`, `BT_IS_ON`, `BT_PAND_IS_ACTIVE`, `_GiveStatusbarPluginPrivate::dnd`, `give_drag_and_drop_have_bt()`, `GIVE_IS_STATUSBAR_PLUGIN`, `GIVE_STATUSBAR_PLUGIN_GET_PRIVATE`, `_GiveStatusbarPluginPrivate::image`, and `_GiveStatusbarPluginPrivate::state`.

Referenced by `give_statusbar_plugin_bt_changed()`, `give_statusbar_plugin_start_pand()`, and `give_statusbar_plugin_stop_pand()`.

Here is the call graph for this function:

Here is the caller graph for this function:

6.8.5.14 HD_DEFINE_PLUGIN (GiveStatusbarPlugin, give_statusbar_plugin, STATUSBAR_TYPE_ITEM)

6.9 src/give_statusbar_plugin.h File Reference

```
#include <glib-object.h>
#include <libhildondesktop/hildon-status-bar-item.h>
#include <libhildondesktop/libhildondesktop.h>
Include dependency graph for give_statusbar_plugin.h:
```

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

Classes

- [struct _GiveStatusbarPlugin](#)
- [struct _GiveStatusbarPluginClass](#)

Defines

- [#define GIVE_TYPE_STATUSBAR_PLUGIN](#) (give_statusbar_plugin_get_type ())
- [#define GIVE_STATUSBAR_PLUGIN\(obj\)](#)
- [#define GIVE_STATUSBAR_PLUGIN_CLASS\(klass\)](#)
- [#define GIVE_IS_STATUSBAR_PLUGIN\(obj\)](#)
- [#define GIVE_IS_STATUSBAR_PLUGIN_CLASS\(klass\)](#)
- [#define GIVE_STATUSBAR_PLUGIN_GET_CLASS\(obj\)](#)

Typedefs

- [typedef typedefG_BEGIN_DECLS struct _GiveStatusbarPlugin GiveStatusbarPlugin](#)
- [typedef struct _GiveStatusbarPluginClass GiveStatusbarPluginClass](#)

Functions

- [GType give_statusbar_plugin_get_type](#) (void) [G_GNUC_CONST](#)

6.9.1 Define Documentation

6.9.1.1 [#define GIVE_IS_STATUSBAR_PLUGIN\(obj\)](#)

Value:

```
(G_TYPE_CHECK_INSTANCE_TYPE ((obj), \
                               GIVE_TYPE_STATUSBAR_PLUGIN))
```


Definition at line 39 of file give_statusbar_plugin.h.

Referenced by give_statusbar_plugin_bt_changed(), give_statusbar_plugin_clicked(), give_statusbar_plugin_finalize(), give_statusbar_plugin_init(), give_statusbar_plugin_pand_dialog(), give_statusbar_plugin_start_pand(), give_statusbar_plugin_stop_pand(), and give_statusbar_plugin_update_status().

6.9.1.2 #define GIVE_IS_STATUSBAR_PLUGIN_CLASS(klass)

Value:

```
(G_TYPE_CHECK_CLASS_TYPE ((klass), \
                          GIVE_TYPE_STATUSBAR_PLUGIN))
```

Definition at line 41 of file give_statusbar_plugin.h.

6.9.1.3 #define GIVE_STATUSBAR_PLUGIN(obj)

Value:

```
(G_TYPE_CHECK_INSTANCE_CAST ((obj), \
                              GIVE_TYPE_STATUSBAR_PLUGIN, \
                              GiveStatusbarPlugin))
```

Definition at line 33 of file give_statusbar_plugin.h.

6.9.1.4 #define GIVE_STATUSBAR_PLUGIN_CLASS(klass)

Value:

```
(G_TYPE_CHECK_CLASS_CAST ((klass), \
                           GIVE_TYPE_STATUSBAR_PLUGIN, \
                           GiveStatusbarPluginClass))
```

Definition at line 36 of file give_statusbar_plugin.h.

6.9.1.5 #define GIVE_STATUSBAR_PLUGIN_GET_CLASS(obj)

Value:

```
(G_TYPE_INSTANCE_GET_CLASS ((obj), \
                             GIVE_TYPE_STATUSBAR_PLUGIN, \
                             GiveStatusbarPluginClass))
```

Definition at line 43 of file give_statusbar_plugin.h.

6.9.1.6 #define GIVE_TYPE_STATUSBAR_PLUGIN (give_statusbar_plugin_get_type ())

Definition at line 32 of file give_statusbar_plugin.h.

6.9.2 Typedef Documentation

6.9.2.1 typedef typedefG_BEGIN_DECLS struct _GiveStatusbarPlugin GiveStatusbarPlugin

Definition at line 29 of file give_statusbar_plugin.h.

6.9.2.2 typedef struct _GiveStatusbarPluginClass GiveStatusbarPluginClass

Definition at line 30 of file give_statusbar_plugin.h.

6.9.3 Function Documentation

6.9.3.1 GType give_statusbar_plugin_get_type (void)

Chapter 7

Give Page Documentation

7.1 Todo List

Member [give_drag_and_drop_create_dialog](#) Look for memory leaks.

Member [give_drag_and_drop_send](#) Remove temporary files after sending.

Member [give_drag_and_drop_drag_recieved](#) Investigate whether the got data needs to freed or not. probably yes, since the data is copied (GDK_ACTION_COPY)

Member [give_statusbar_plugin_start_pand](#) The device don't act as a server just as a client.

Index

- - give_drag_and_drop.c, 36
 - give_statusbar_plugin.c, 63
- _GiveStatusbarPlugin, 21
 - parent, 21
- _GiveStatusbarPluginClass, 22
 - parent_class, 22
- _GiveStatusbarPluginPrivate, 23
 - button, 23
 - dnd, 23
 - gc_client, 23
 - image, 23
 - notify_uid, 24
 - pand_pid, 24
 - state, 24
- _give_bluetooth, 11
 - parent_object, 11
- _give_bluetoothClass, 13
 - parent_class, 13
- _give_bluetooth_private, 12
 - dbus_proxy, 12
- _give_drag_and_drop, 14
 - parent_object, 14
- _give_drag_and_dropClass, 17
 - parent_class, 17
- _give_drag_and_drop_private, 15
 - bt_is_on, 15
 - dbus_proxy, 15
 - destination, 15
 - give_bt_object, 16
 - give_ebook_object, 16
 - handler_id, 15
- _give_ebook, 18
 - parent_object, 18
- _give_ebookClass, 20
 - parent_class, 20
- _give_ebook_private, 19
 - ebook, 19
 - ready, 19
- ABOUT_ITEM
 - give_statusbar_plugin.c, 64
- APPOINTMENT_ITEM
 - give_statusbar_plugin.c, 64
- BLUEZ_DBUS_INTERFACE
 - give_drag_and_drop.c, 36
- BLUEZ_DBUS_MESSAGE
 - give_drag_and_drop.c, 36
- BLUEZ_DBUS_PATH
 - give_drag_and_drop.c, 37
- BLUEZ_DBUS_SERVICE
 - give_drag_and_drop.c, 37
- BLUEZ_DBUS_SET_MODE
 - give_drag_and_drop.c, 37
- BT_DIR
 - give_statusbar_plugin.c, 63
- BT_IS_OFF
 - give_statusbar_plugin.c, 64
- BT_IS_ON
 - give_statusbar_plugin.c, 64
- bt_is_on
 - _give_drag_and_drop_private, 15
- BT_KEY
 - give_statusbar_plugin.c, 63
- BT_PAND_IS_ACTIVE
 - give_statusbar_plugin.c, 64
- bt_state
 - give_statusbar_plugin.c, 64
- button
 - _GiveStatusbarPluginPrivate, 23
- CALENDAR_EVENT
 - give_drag_and_drop.c, 38
- CALENDAR_EVENT_TYPE
 - give_drag_and_drop.c, 37
- CONTACT_ITEM
 - give_statusbar_plugin.c, 64
- CONTACT_LIST
 - give_drag_and_drop.c, 38
- CONTACT_LIST_TYPE
 - give_drag_and_drop.c, 37
- dbus_proxy
 - _give_bluetooth_private, 12
 - _give_drag_and_drop_private, 15
- destination
 - _give_drag_and_drop_private, 15
- dnd
 - _GiveStatusbarPluginPrivate, 23

- documentation.txt, 25
- drop_types
 - give_drag_and_drop.c, 47
- ebook
 - _give_ebook_private, 19
- EMAIL_ITEM
 - give_statusbar_plugin.c, 64
- FILE_ITEM
 - give_statusbar_plugin.c, 64
- gc_client
 - _GiveStatusbarPluginPrivate, 23
- GIVE_BLUETOOTH
 - give_bluetooth.h, 32
- give_bluetooth
 - give_bluetooth.h, 33
- give_bluetooth.c
 - give_bluetooth_class_init, 27
 - give_bluetooth_finalize, 28
 - GIVE_BLUETOOTH_GET_PRIVATE, 27
 - give_bluetooth_get_type, 28
 - give_bluetooth_init, 29
 - give_bluetooth_new, 29
 - give_bluetooth_private, 27
 - give_bluetooth_send_file, 30
- give_bluetooth.h
 - GIVE_BLUETOOTH, 32
 - give_bluetooth, 33
 - GIVE_BLUETOOTH_CLASS, 32
 - give_bluetooth_get_type, 33
 - GIVE_BLUETOOTH_IS_CLASS, 32
 - GIVE_BLUETOOTH_IS_OBJECT, 32
 - give_bluetooth_new, 33
 - give_bluetooth_send_file, 33
 - GIVE_BLUETOOTH_TYPE, 32
 - give_bluetoothClass, 33
- GIVE_BLUETOOTH_CLASS
 - give_bluetooth.h, 32
- give_bluetooth_class_init
 - give_bluetooth.c, 27
- give_bluetooth_finalize
 - give_bluetooth.c, 28
- GIVE_BLUETOOTH_GET_PRIVATE
 - give_bluetooth.c, 27
- give_bluetooth_get_type
 - give_bluetooth.c, 28
 - give_bluetooth.h, 33
- give_bluetooth_init
 - give_bluetooth.c, 29
- GIVE_BLUETOOTH_IS_CLASS
 - give_bluetooth.h, 32
- GIVE_BLUETOOTH_IS_OBJECT
 - give_bluetooth.h, 32
- give_bluetooth_new
 - give_bluetooth.c, 29
 - give_bluetooth.h, 33
- give_bluetooth_private
 - give_bluetooth.c, 27
- give_bluetooth_send_file
 - give_bluetooth.c, 30
 - give_bluetooth.h, 33
- GIVE_BLUETOOTH_TYPE
 - give_bluetooth.h, 32
- give_bluetoothClass
 - give_bluetooth.h, 33
- give_bt_object
 - _give_drag_and_drop_private, 16
- GIVE_DRAG_AND_DROP
 - give_drag_and_drop.h, 49
- give_drag_and_drop
 - give_drag_and_drop.h, 50
- give_drag_and_drop.c
 - CALENDAR_EVENT, 38
 - CONTACT_LIST, 38
 - MODEST_EMAIL, 38
 - PLAIN_TEXT, 38
 - TREE_AS_STRING, 38
 - TREE_ROW, 38
 - URI_LIST, 38
- give_drag_and_drop.c
 - _, 36
 - BLUEZ_DBUS_INTERFACE, 36
 - BLUEZ_DBUS_MESSAGE, 36
 - BLUEZ_DBUS_PATH, 37
 - BLUEZ_DBUS_SERVICE, 37
 - BLUEZ_DBUS_SET_MODE, 37
 - CALENDAR_EVENT_TYPE, 37
 - CONTACT_LIST_TYPE, 37
 - drop_types, 47
 - give_drag_and_drop_class_init, 39
 - give_drag_and_drop_create_dialog, 39
 - give_drag_and_drop_drag_recieved, 40
 - give_drag_and_drop_finalize, 41
 - GIVE_DRAG_AND_DROP_GET_PRIVATE, 37
 - give_drag_and_drop_get_type, 42
 - give_drag_and_drop_have_bt, 42
 - give_drag_and_drop_init, 42
 - give_drag_and_drop_new, 43
 - give_drag_and_drop_private, 38
 - give_drag_and_drop_run_dialog, 44
 - give_drag_and_drop_send, 45
 - give_drag_and_drop_set_dest, 45
 - give_drag_and_drop_turn_bt_on_dialog, 46

- MODEST_EMAIL_TYPE, 37
- PLAIN_TEXT_TYPE, 37
- TEMP_DIR, 38
- TREE_AS_STRING_TYPE, 38
- TREE_ROW_TYPE, 38
- URI_LIST_TYPE, 38
- give_drag_and_drop.h
 - GIVE_DRAG_AND_DROP, 49
 - give_drag_and_drop, 50
 - GIVE_DRAG_AND_DROP_CLASS, 49
 - give_drag_and_drop_get_type, 50
 - give_drag_and_drop_have_bt, 50
 - GIVE_DRAG_AND_DROP_IS_-CLASS, 49
 - GIVE_DRAG_AND_DROP_IS_-OBJECT, 49
 - give_drag_and_drop_new, 50
 - give_drag_and_drop_turn_bt_on_-dialog, 50
 - GIVE_DRAG_AND_DROP_TYPE, 49
 - give_drag_and_dropClass, 50
- GIVE_DRAG_AND_DROP_CLASS
 - give_drag_and_drop.h, 49
- give_drag_and_drop_class_init
 - give_drag_and_drop.c, 39
- give_drag_and_drop_create_dialog
 - give_drag_and_drop.c, 39
- give_drag_and_drop_drag_recieved
 - give_drag_and_drop.c, 40
- give_drag_and_drop_finalize
 - give_drag_and_drop.c, 41
- GIVE_DRAG_AND_DROP_GET_-PRIVATE
 - give_drag_and_drop.c, 37
- give_drag_and_drop_get_type
 - give_drag_and_drop.c, 42
 - give_drag_and_drop.h, 50
- give_drag_and_drop_have_bt
 - give_drag_and_drop.c, 42
 - give_drag_and_drop.h, 50
- give_drag_and_drop_init
 - give_drag_and_drop.c, 42
- GIVE_DRAG_AND_DROP_IS_CLASS
 - give_drag_and_drop.h, 49
- GIVE_DRAG_AND_DROP_IS_OBJECT
 - give_drag_and_drop.h, 49
- give_drag_and_drop_new
 - give_drag_and_drop.c, 43
 - give_drag_and_drop.h, 50
- give_drag_and_drop_private
 - give_drag_and_drop.c, 38
- give_drag_and_drop_run_dialog
 - give_drag_and_drop.c, 44
- give_drag_and_drop_send
 - give_drag_and_drop.c, 45
- give_drag_and_drop_set_dest
 - give_drag_and_drop.c, 45
- give_drag_and_drop_turn_bt_on_dialog
 - give_drag_and_drop.c, 46
 - give_drag_and_drop.h, 50
- GIVE_DRAG_AND_DROP_TYPE
 - give_drag_and_drop.h, 49
- give_drag_and_dropClass
 - give_drag_and_drop.h, 50
- GIVE_EBOOK
 - give_ebook.h, 59
- give_ebook
 - give_ebook.h, 59
- give_ebook.c
 - give_ebook_class_init, 53
 - give_ebook_finalize, 53
 - give_ebook_get_contact_name, 54
 - give_ebook_get_contact_vcard, 54
 - GIVE_EBOOK_GET_PRIVATE, 52
 - give_ebook_get_type, 55
 - give_ebook_init, 55
 - give_ebook_new, 56
 - give_ebook_open_response, 56
 - give_ebook_private, 52
 - give_ebook_status_to_string, 57
- give_ebook.h
 - GIVE_EBOOK, 59
 - give_ebook, 59
 - GIVE_EBOOK_CLASS, 59
 - give_ebook_get_contact_name, 60
 - give_ebook_get_contact_vcard, 60
 - give_ebook_get_type, 60
 - GIVE_EBOOK_IS_CLASS, 59
 - GIVE_EBOOK_IS_OBJECT, 59
 - give_ebook_new, 60
 - GIVE_EBOOK_TYPE, 59
 - give_ebookClass, 60
- GIVE_EBOOK_CLASS
 - give_ebook.h, 59
- give_ebook_class_init
 - give_ebook.c, 53
- give_ebook_finalize
 - give_ebook.c, 53
- give_ebook_get_contact_name
 - give_ebook.c, 54
 - give_ebook.h, 60
- give_ebook_get_contact_vcard
 - give_ebook.c, 54
 - give_ebook.h, 60
- GIVE_EBOOK_GET_PRIVATE
 - give_ebook.c, 52
- give_ebook_get_type
 - give_ebook.c, 55

- give_ebook.h, 60
- give_ebook_init
 - give_ebook.c, 55
- GIVE_EBOOK_IS_CLASS
 - give_ebook.h, 59
- GIVE_EBOOK_IS_OBJECT
 - give_ebook.h, 59
- give_ebook_new
 - give_ebook.c, 56
 - give_ebook.h, 60
- give_ebook_object
 - _give_drag_and_drop_private, 16
- give_ebook_open_response
 - give_ebook.c, 56
- give_ebook_private
 - give_ebook.c, 52
- give_ebook_status_to_string
 - give_ebook.c, 57
- GIVE_EBOOK_TYPE
 - give_ebook.h, 59
- give_ebookClass
 - give_ebook.h, 60
- GIVE_IS_STATUSBAR_PLUGIN
 - give_statusbar_plugin.h, 74
- GIVE_IS_STATUSBAR_PLUGIN_CLASS
 - give_statusbar_plugin.h, 74
- GIVE_STATUSBAR_PLUGIN
 - give_statusbar_plugin.h, 74
- give_statusbar_plugin.c
 - ABOUT_ITEM, 64
 - APPOINTMENT_ITEM, 64
 - BT_IS_OFF, 64
 - BT_IS_ON, 64
 - BT_PAND_IS_ACTIVE, 64
 - CONTACT_ITEM, 64
 - EMAIL_ITEM, 64
 - FILE_ITEM, 64
 - NOTE_ITEM, 64
- give_statusbar_plugin_
 - _ , 63
 - BT_DIR, 63
 - BT_KEY, 63
 - bt_state, 64
 - give_statusbar_plugin_about_dialog, 64
 - give_statusbar_plugin_bt_changed, 65
 - give_statusbar_plugin_class_init, 65
 - give_statusbar_plugin_clicked, 66
 - give_statusbar_plugin_finalize, 67
 - GIVE_STATUSBAR_PLUGIN_GET_-PRIVATE, 63
 - give_statusbar_plugin_init, 67
 - give_statusbar_plugin_item_clicked, 68
 - give_statusbar_plugin_log_handler, 68
 - give_statusbar_plugin_pand_dialog, 69
 - give_statusbar_plugin_popup_place, 70
 - give_statusbar_plugin_start_pand, 70
 - give_statusbar_plugin_stop_pand, 71
 - give_statusbar_plugin_update_status, 72
 - GiveStatusbarPluginPrivate, 64
 - HD_DEFINE_PLUGIN, 72
 - LOG_FILE, 63
 - menu_items, 64
- give_statusbar_plugin.h
 - GIVE_IS_STATUSBAR_PLUGIN, 74
 - GIVE_IS_STATUSBAR_PLUGIN_CLASS, 74
 - GIVE_STATUSBAR_PLUGIN, 74
 - GIVE_STATUSBAR_PLUGIN_CLASS, 74
 - GIVE_STATUSBAR_PLUGIN_GET_-CLASS, 74
 - give_statusbar_plugin_get_type, 75
 - GIVE_TYPE_STATUSBAR_PLUGIN, 75
 - GiveStatusbarPlugin, 75
 - GiveStatusbarPluginClass, 75
- give_statusbar_plugin_about_dialog
 - give_statusbar_plugin.c, 64
- give_statusbar_plugin_bt_changed
 - give_statusbar_plugin.c, 65
- GIVE_STATUSBAR_PLUGIN_CLASS
 - give_statusbar_plugin.h, 74
- give_statusbar_plugin_class_init
 - give_statusbar_plugin.c, 65
- give_statusbar_plugin_clicked
 - give_statusbar_plugin.c, 66
- give_statusbar_plugin_finalize
 - give_statusbar_plugin.c, 67
- GIVE_STATUSBAR_PLUGIN_GET_-CLASS
 - give_statusbar_plugin.h, 74
- GIVE_STATUSBAR_PLUGIN_GET_-PRIVATE
 - give_statusbar_plugin.c, 63
- give_statusbar_plugin_get_type
 - give_statusbar_plugin.h, 75
- give_statusbar_plugin_init
 - give_statusbar_plugin.c, 67
- give_statusbar_plugin_item_clicked
 - give_statusbar_plugin.c, 68
- give_statusbar_plugin_log_handler
 - give_statusbar_plugin.c, 68
- give_statusbar_plugin_pand_dialog
 - give_statusbar_plugin.c, 69
- give_statusbar_plugin_popup_place
 - give_statusbar_plugin.c, 70
- give_statusbar_plugin_start_pand

- give_statusbar_plugin.c, 70
- give_statusbar_plugin_stop_pand
 - give_statusbar_plugin.c, 71
- give_statusbar_plugin_update_status
 - give_statusbar_plugin.c, 72
- GIVE_TYPE_STATUSBAR_PLUGIN
 - give_statusbar_plugin.h, 75
- GiveStatusbarPlugin
 - give_statusbar_plugin.h, 75
- GiveStatusbarPluginClass
 - give_statusbar_plugin.h, 75
- GiveStatusbarPluginPrivate
 - give_statusbar_plugin.c, 64
- handler_id
 - _give_drag_and_drop_private, 15
- HD_DEFINE_PLUGIN
 - give_statusbar_plugin.c, 72
- image
 - _GiveStatusbarPluginPrivate, 23
- LOG_FILE
 - give_statusbar_plugin.c, 63
- menu_items
 - give_statusbar_plugin.c, 64
- MODEST_EMAIL
 - give_drag_and_drop.c, 38
- MODEST_EMAIL_TYPE
 - give_drag_and_drop.c, 37
- NOTE_ITEM
 - give_statusbar_plugin.c, 64
- notify_uid
 - _GiveStatusbarPluginPrivate, 24
- pand_pid
 - _GiveStatusbarPluginPrivate, 24
- parent
 - _GiveStatusbarPlugin, 21
- parent_class
 - _GiveStatusbarPluginClass, 22
 - _give_bluetoothClass, 13
 - _give_drag_and_dropClass, 17
 - _give_ebookClass, 20
- parent_object
 - _give_bluetooth, 11
 - _give_drag_and_drop, 14
 - _give_ebook, 18
- PLAIN_TEXT
 - give_drag_and_drop.c, 38
- PLAIN_TEXT_TYPE
 - give_drag_and_drop.c, 37
- ready
 - _give_ebook_private, 19
- src/give_bluetooth.c, 26
- src/give_bluetooth.h, 31
- src/give_drag_and_drop.c, 34
- src/give_drag_and_drop.h, 48
- src/give_ebook.c, 51
- src/give_ebook.h, 58
- src/give_statusbar_plugin.c, 61
- src/give_statusbar_plugin.h, 73
- state
 - _GiveStatusbarPluginPrivate, 24
- TEMP_DIR
 - give_drag_and_drop.c, 38
- TREE_AS_STRING
 - give_drag_and_drop.c, 38
- TREE_AS_STRING_TYPE
 - give_drag_and_drop.c, 38
- TREE_ROW
 - give_drag_and_drop.c, 38
- TREE_ROW_TYPE
 - give_drag_and_drop.c, 38
- URI_LIST
 - give_drag_and_drop.c, 38
- URI_LIST_TYPE
 - give_drag_and_drop.c, 38