

# Indicate-0.7 C API Documentation

## Class: IndicateIndicator



Subclass of: [GObject](#)

The indicator object represents a single item that is shared over the indicator bus. This could be something like one IM, one e-mail or a single system update. It should be accessed only through its accessors.

### Signals

```
void                    "hide"
(IndicateIndicator *indicator)
```

Emitted every time this indicator is hidden. This is mostly used by #IndicateServer. Typically this results in an emission of #IndicateServer::indicator-removed.

```
void                    "displayed"
(IndicateIndicator *indicator, gboolean arg1)
```

This is the signal that the indicator has been displayed, or hidden by a listener. In most cases, the signal will be that it has been displayed as most folks don't go hiding it later.

```
void                    "modified"
(IndicateIndicator *indicator, gchar* arg1)
```

Emitted every time an indicator property is changed. This is mostly used by #IndicateServer. Typically this results in an emission of #IndicateServer::indicator-modified.

```
void                    "user-display"
(IndicateIndicator *indicator, guint object)
void                    "show"
(IndicateIndicator *indicator)
```

Emitted every time this indicator is shown. This is mostly used by #IndicateServer. Typically this results in an emission of #IndicateServer::indicator-added.

### Constructors

```
IndicateIndicator*    indicate_indicator_new          ()
IndicateIndicator*    indicate_indicator_new_with_server
(IndicateServer *server)
```

### Methods

```
void                    indicate_indicator_displayed
(IndicateIndicator *self, gboolean displayed)
gboolean                indicate_indicator_get_displayed
```

```

(IndicateIndicator *self)
guint indicate_indicator_get_id
(IndicateIndicator *self)
GVariant\* indicate_indicator_get_property
(IndicateIndicator *self, gchar *key)
GVariant\* indicate_indicator_get_property_variant
(IndicateIndicator *self, gchar *key)
IndicateServer\* indicate_indicator_get_server
(IndicateIndicator *self)
void indicate_indicator_hide
(IndicateIndicator *self, gpointer data)
gboolean indicate_indicator_is_visible
(IndicateIndicator *self)
gchar** indicate_indicator_list_properties
(IndicateIndicator *self)
void indicate_indicator_modified
(IndicateIndicator *self, gchar *property, gpointer data)
void indicate_indicator_set_displayed
(IndicateIndicator *self, gboolean displayed)
void indicate_indicator_set_property
(IndicateIndicator *self, gchar *key, GVariant *data)
void indicate_indicator_set_property_bool
(IndicateIndicator *self, gchar *key, gboolean value)
void indicate_indicator_set_property_int
(IndicateIndicator *self, gchar *key, gint value)
void indicate_indicator_set_property_time
(IndicateIndicator *self, gchar *key, GTimeVal *time)
void indicate_indicator_set_property_variant
(IndicateIndicator *self, gchar *key, GVariant *value)
void indicate_indicator_set_server
(IndicateIndicator *self, IndicateServer *server)
void indicate_indicator_show
(IndicateIndicator *self, gpointer data)
void indicate_indicator_user_display
(IndicateIndicator *self, guint timestamp, gpointer data)

```

## Class: IndicateListener



Subclass of: [GObject](#)

### Signals

```

void "indicator-servers-report"
(IndicateListener *listener)
void "server-removed"
(IndicateListener *listener, unknown_{http://www.gtk.org/introspection/core/10}type
object, gchar* p0)
void "server-added"
(IndicateListener *listener, unknown_{http://www.gtk.org/introspection/core/10}type
object, gchar* p0)
void "indicator-removed"
(IndicateListener *listener, unknown_{http://www.gtk.org/introspection/core/10}type
object, unknown_{http://www.gtk.org/introspection/core/10}type p0)
void "indicator-modified"
(IndicateListener *listener, unknown_{http://www.gtk.org/introspection/core/10}type
object, unknown_{http://www.gtk.org/introspection/core/10}type p0, gchar* p1)
void "indicator-added"

```

```
(IndicateListener *listener, unknown_{http://www.gtk.org/introspection/core/10}type
object, unknown_{http://www.gtk.org/introspection/core/10}type p0)
void                                "server-count-changed"
(IndicateListener *listener, unknown_{http://www.gtk.org/introspection/core/10}type
object, guint p0)
```

## Constructors

[IndicateListener\\*](#) [indicate\\_listener\\_new](#) [\(\)](#)

## Methods

```
void                                indicate_listener_display
(IndicateListener *self, IndicateListenerServer *server, IndicateListenerIndicator
*indicator, guint timestamp)
void                                indicate_listener_displayed
(IndicateListener *self, IndicateListenerServer *server, IndicateListenerIndicator
*indicator, gboolean displayed)
void                                indicate_listener_get_property
(IndicateListener *self, IndicateListenerServer *server, IndicateListenerIndicator
*indicator, gchar *property, gpointer *callback, gpointer data)
void                                indicate_listener_get_property_bool
(IndicateListener *self, IndicateListenerServer *server, IndicateListenerIndicator
*indicator, gchar *property, gpointer *callback, gpointer data)
void                                indicate_listener_get_property_int
(IndicateListener *self, IndicateListenerServer *server, IndicateListenerIndicator
*indicator, gchar *property, gpointer *callback, gpointer data)
void                                indicate_listener_get_property_time
(IndicateListener *self, IndicateListenerServer *server, IndicateListenerIndicator
*indicator, gchar *property, gpointer *callback, gpointer data)
void                                indicate_listener_get_property_variant
(IndicateListener *self, IndicateListenerServer *server, IndicateListenerIndicator
*indicator, gchar *property, gpointer *callback, gpointer data)
GList\*                            indicate_listener_get_servers
(IndicateListener *self)
IndicateListener\*                indicate_listener_ref_default      ()
gboolean                            indicate_listener_server_check_interest
(IndicateListener *self, IndicateListenerServer *server, IndicateInterests
interest)
void                                indicate_listener_server_get_count
(IndicateListener *self, IndicateListenerServer *server, gpointer *callback,
gpointer data)
void                                indicate_listener_server_get_desktop
(IndicateListener *self, IndicateListenerServer *server, gpointer *callback,
gpointer data)
void                                indicate_listener_server_get_icon_theme
(IndicateListener *self, IndicateListenerServer *server, gpointer *callback,
gpointer data)
GList\*                            indicate_listener_server_get_indicators
(IndicateListener *self, IndicateListenerServer *server)
void                                indicate_listener_server_get_menu
(IndicateListener *self, IndicateListenerServer *server, gpointer *callback,
gpointer data)
void                                indicate_listener_server_get_type
(IndicateListener *self, IndicateListenerServer *server, gpointer *callback,
gpointer data)
void                                indicate_listener_server_remove_interest
(IndicateListener *self, IndicateListenerServer *server, IndicateInterests
interest)
void                                indicate_listener_server_show_interest
(IndicateListener *self, IndicateListenerServer *server, IndicateInterests
interest)
```

```

void                indicate_listener_set_default_max_indicators
(IndicateListener *self, gint max)
void                indicate_listener_set_server_max_indicators
(IndicateListener *self, IndicateListenerServer *server, gint max)

```

## Class: IndicateServer



Subclass of: [GObject](#)

This is the object that represents the overall connection between this application and DBus. It acts as the proxy for incoming DBus calls and also sends the appropriate signals on DBus for events happening on other objects locally. It provides some settings that effect how the application as a whole is perceived by listeners of the indicator protocol.

### Properties

|        |                     |                       |
|--------|---------------------|-----------------------|
| guint  | <b>"count"</b>      | <i>read/write</i>     |
| gchar* | <b>"icon-theme"</b> | <i>read/write</i>     |
| gchar* | <b>"menu"</b>       | <i>read-only</i>      |
| gchar* | <b>"desktop"</b>    | <i>read/write</i>     |
| gchar* | <b>"path"</b>       | <i>construct-only</i> |
| gchar* | <b>"type"</b>       | <i>read/write</i>     |

### Signals

```

void                "indicator-delete"
(IndicateServer *server, guint object)
void                "server-show"
(IndicateServer *server, gchar* arg1)

```

Emitted when a server comes onto DBus by being shown. This is typically when listeners start reacting to the application's indicators. This results in a signal on DBus.

```

void                "indicator-new"
(IndicateServer *server, guint object)
void                "server-hide"
(IndicateServer *server, gchar* arg1)

```

Emitted when a server removes itself from DBus. This results in a signal on DBus.

```

void                "indicator-modified"
(IndicateServer *server, guint arg1, gchar* arg2)

```

Emitted every time that a property on an indicator changes and it is visible to the world. This results in a signal on DBus.

```

void                "max-indicators-changed"
(IndicateServer *server, gint arg1)

```

Emitted when a listener either specifies their max number to be higher, or at all. The default is -1 or infinite.

```

void                "server-display"

```

```
(IndicateServer *server, guint arg1)
```

Emitted when a listener signals that the server itself should be displayed. This signal is caused by a user clicking on the application item in the Messaging Menu. This signal is emitted by DBus.

```
void                                "interest-removed"
(IndicateServer *server, guint arg1)
```

Emitted when a listener signals that they are no longer interested in this server for a particular reason. This signal is emitted by DBus. @note This signal is also emitted after a timeout when the object is created with @arg1 set to #INDICATOR\_INTREST\_NONE if no one has shown any interest in the server.

```
void                                "interest-added"
(IndicateServer *server, guint arg1)
```

Emitted when a listener signals that they are interested in this server for a particular reason. This signal is emitted by DBus.

```
void                                "server-count-changed"
(IndicateServer *server, guint arg1)
```

Emitted when the count property of the server changes to a new value.

## Methods

```
void                                indicate_server_add_indicator
(IndicateServer *self, IndicateIndicator *indicator)
gboolean                            indicate_server_check_interest
(IndicateServer *self, IndicateInterests interest)
gboolean                            indicate_server_get_indicator_count
(IndicateServer *self, guint *count)
gboolean                            indicate_server_get_indicator_property
(IndicateServer *self, guint id, gchar *property, GVariant *value)
gint                                indicate_server_get_max_indicators
(IndicateServer *self)
guint                                indicate_server_get_next_id
(IndicateServer *self)
gchar*                              indicate_server_get_path
(IndicateServer *self)
void                                indicate_server_hide
(IndicateServer *self)
void                                indicate_server_indicator_added
(IndicateServer *self, guint id)
gboolean                            indicate_server_indicator_displayed
(IndicateServer *self, gchar *sender, guint id, gboolean displayed)
void                                indicate_server_indicator_modified
(IndicateServer *self, guint id, gchar *property)
void                                indicate_server_indicator_removed
(IndicateServer *self, guint id)
void                                indicate_server_interest_added
(IndicateServer *self, IndicateInterests interest)
void                                indicate_server_interest_removed
(IndicateServer *self, IndicateInterests interest)
void                                indicate_server_max_indicators_changed
(IndicateServer *self, gint max)
gint                                indicate_server_max_indicators_get
(IndicateServer *self)
gboolean                            indicate_server_max_indicators_set
(IndicateServer *self, gchar *sender, gint max)
```

```

IndicateServer*          indicate_server_ref_default          ( )
void          indicate_server_remove_indicator
(IndicateServer *self, IndicateIndicator *indicator)
gboolean      indicate_server_remove_interest
(IndicateServer *self, gchar *sender, IndicateInterests interest)
void          indicate_server_server_count_changed
(IndicateServer *self, guint count)
void          indicate_server_server_display
(IndicateServer *self, guint timestamp)
void          indicate_server_server_hide
(IndicateServer *self, gchar *type)
void          indicate_server_server_show
(IndicateServer *self, gchar *type)
void          indicate_server_set_count
(IndicateServer *self, guint count)
void          indicate_server_set_dbus_object          (gchar *obj)
void          indicate_server_set_default
(IndicateServer *self)
void          indicate_server_set_desktop_file
(IndicateServer *self, gchar *path)
void          indicate_server_set_icon_theme
(IndicateServer *self, gchar *name)
void          indicate_server_set_menu
(IndicateServer *self, DbusmenuServer *menu)
void          indicate_server_set_type
(IndicateServer *self, gchar *type)
void          indicate_server_show
(IndicateServer *self)
gboolean      indicate_server_show_indicator_to_user
(IndicateServer *self, guint id, guint timestamp)
gboolean      indicate_server_show_interest
(IndicateServer *self, gchar *sender, IndicateInterests interest)

```