Indicate-0.7 Python API Documentation

Class: Indicate.Indicator

Subclass of: GObject.Object

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"
(self, indicator)
```

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

```
void                    "displayed"
(self, indicator, 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"
(self, indicator, arg1)
```

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

```
void                    "user-display"
(self, indicator, object)
void                    "show"
(self, indicator)
```

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

Constructors

```
                        new                                 ()
                        new_with_server                     (server)
```

Methods

```
void                    displayed                           (displayed)
bool                    get_displayed                       ()
int                     get_id                              ()
GLib.Variant            get_property                        (key)
```

GLib.Variant	**get_property_variant**	(key)
Indicate.Server	**get_server**	()
void	**hide**	(data)
bool	**is_visible**	()
Array<utf8>	**list_properties**	()
void	**modified**	(property, data)
void	**set_displayed**	(displayed)
void	**set_property**	(key, data)
void	**set_property_bool**	(key, value)
void	**set_property_int**	(key, value)
void	**set_property_time**	(key, time)
void	**set_property_variant**	(key, value)
void	**set_server**	(server)
void	**show**	(data)
void	**user_display**	(timestamp, data)

Class: Indicate.Listener

Subclass of: GObject.Object

Signals

void	**"indicator-servers-report"**	(self, listener)
void	**"server-removed"**	(self, listener, object, p0)
void	**"server-added"**	(self, listener, object, p0)
void	**"indicator-removed"**	(self, listener, object, p0)
void	**"indicator-modified"**	(self, listener, object, p0, p1)
void	**"indicator-added"**	(self, listener, object, p0)
void	**"server-count-changed"**	(self, listener, object, p0)

Constructors

	new	()

Methods

void	**display**	(server, indicator, timestamp)
void	**displayed**	(server, indicator, displayed)
void	**get_property**	(server, indicator, property, callback, data)
void	**get_property_bool**	(server, indicator, property, callback, data)
void	**get_property_int**	(server, indicator, property, callback, data)
void	**get_property_time**	(server, indicator, property, callback, data)
void	**get_property_variant**	

```
(server, indicator, property, callback, data)
Indicate.Listener          ref_default                    ()
    (static method)
bool                       server_check_interest
(server, interest)
void                       server_get_count
(server, callback, data)
void                       server_get_desktop
(server, callback, data)
void                       server_get_icon_theme
(server, callback, data)
GLib.List                  server_get_indicators          (server)
void                       server_get_menu
(server, callback, data)
void                       server_get_type
(server, callback, data)
void                       server_remove_interest
(server, interest)
void                       server_show_interest
(server, interest)
void                       set_default_max_indicators     (max)
void                       set_server_max_indicators      (server, max)
```

Class: Indicate.Server

Subclass of: GObject.Object

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

Properties

```
int                        "count"                        read/write
string                     "icon-theme"                   read/write
string                     "menu"                         read-only
string                     "desktop"                      read/write
string                     "path"                         construct-only
string                     "type"                         read/write
```

Signals

```
void                       "indicator-delete"
(self, server, object)
void                       "server-show"
(self, server, 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"
(self, server, object)
void                       "server-hide"
```

```
(self, server, arg1)
```

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

```
void                        "indicator-modified"
(self, server, arg1, 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"
(self, server, 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"
(self, server, 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"
(self, server, 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"
(self, server, 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"
(self, server, arg1)
```

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

Methods

```
void                        add_indicator                   (indicator)
bool                        check_interest                  (interest)
bool                        get_indicator_count             (count)
bool                        get_indicator_property
(id, property, value)
int                         get_max_indicators              ()
int                         get_next_id                     ()
string                      get_path                        ()
void                        hide                            ()
void                        indicator_added                 (id)
bool                        indicator_displayed
(sender, id, displayed)
void                        indicator_modified              (id, property)
void                        indicator_removed               (id)
void                        interest_added                  (interest)
void                        interest_removed                (interest)
```

void	**max_indicators_changed**	(max)
int	**max_indicators_get**	()
bool	**max_indicators_set**	(sender, max)
Indicate.Server	***ref_default***	*()*
(static method)		
void	**remove_indicator**	(indicator)
bool	**remove_interest**	
(sender, interest)		
void	**server_count_changed**	(count)
void	**server_display**	(timestamp)
void	**server_hide**	(type)
void	**server_show**	(type)
void	**set_count**	(count)
void	***set_dbus_object***	*(obj)*
(static method)		
void	**set_default**	()
void	**set_desktop_file**	(path)
void	**set_icon_theme**	(name)
void	**set_menu**	(menu)
void	**set_type**	(type)
void	**show**	()
bool	**show_indicator_to_user**	(id, timestamp)
bool	**show_interest**	
(sender, interest)		

Enum: Indicate.Interests

NONE	0
SERVER_DISPLAY	1
SERVER_SIGNAL	2
INDICATOR_DISPLAY	3
INDICATOR_SIGNAL	4
INDICATOR_COUNT	5
LAST	6