

USER DOCUMENTATION OF GdgObject.GdgObject V0.1.1

Table of content :

- 1) Object [_me] as class(GdgObject.GdgObject)
 - 1.1) Function : _me.__init__
 - 1.2) Function : _me.destroy
 - 1.3) Function : _me._print_debug
 - 1.5) Function : _me.main
 - 1.6) Function : _me.run_main_loop
 - 1.7) Function : _me.exit_main_loop
 - 1.8) Function : _me.set_language
 - 1.9) Function : _me.refresh_GUI
 - 1.10) Function : _me.restore_GUI
 - 1.11) Function : _me.string
 - 1.12) Function : _me.get_name
 - 1.13) Function : _me.get_author
 - 1.14) Function : _me.get_version
 - 1.15) Function : _me.get_description
 - 1.16) Function : _me.get_path
 - 1.17) Function : _me.get_icon
 - 1.18) Function : _me.get_sound
 - 1.19) Function : _me.get_lib
 - 1.20) Function : _me.gui
 - 1.21) Function : _me.insert_funct
 - 1.22) Function : _me.get_funct
 - 1.23) Function : _me.get_var
 - 1.24) Function : _me.set_var
 - 1.25) Function : _me.get_param
 - 1.26) Function : _me.set_param
 - 1.27) Function : _me.speak
 - 1.28) Function : _me.speak_free
 - 1.29) Function : _me.speak_notification
 - 1.30) Function : _me.wait_permission
 - 1.31) Function : _me.insert_menu
 - 1.32) Function : _me.remove_menu
- 1.33) Object [_me.download] as class(GdgDownload.GdgDownload)
 - 1.33.1) Function : _me.download.check_server_available
 - 1.33.2) Function : _me.download.download_url
- 1.34) Object [_me.notify] as class(GdgObject.GdgNotify)
 - 1.34.1) Function : _me.notify.start
 - 1.34.2) Function : _me.notify.stop
 - 1.34.3) Function : _me.notify.set_active
 - 1.34.4) Function : _me.notify.set_delay
 - 1.34.5) Function : _me.notify.check_now
- 1.35) Object [_me.gui('gui name')] as class(GdgObject.GdgGUI)
 - 1.35.1) Function : _me.gui('gui name').showed
 - 1.35.2) Function : _me.gui('gui name').show
 - 1.35.3) Function : _me.gui('gui name').show_from_main
 - 1.35.4) Function : _me.gui('gui name').hide
 - 1.35.5) Function : _me.gui('gui name').hide_from_main
 - 1.35.6) Function : _me.gui('gui name').refresh
- 1.36) Object [_me.voice_recognition] as class(GdgObject.GdgVRManager)
 - 1.36.1) Function : _me.voice_recognition.set_dict
 - 1.36.2) Function : _me.voice_recognition.set_rules
 - 1.36.3) Function : _me.voice_recognition.get_run
 - 1.36.4) Function : _me.voice_recognition.push_dict
 - 1.36.5) Function : _me.voice_recognition.pop_dict
 - 1.36.6) Function : _me.voice_recognition.clear_dict

1) Object [_me] as class(GdgObject.GdgObject)

Class which create and control a gadget.

Objects in this class:

"download" as instance of class(GdgDownload):

Object which manages the download of files.

"notify" as instance of class(GdgNotify):

Object which create a notification controller.

"gui('gui name')" as instance of class(GdgGUI):

Object which create and control a GUI.

Possibles values for 'gui name':
 'widget'
 'conf'

"voice_recognition" as instance of class(GdgVRManager):
 Object which manages the voice recognition.

Additional functions and ressources:

Create a GTK dialog box window:
 dialog = GdgDialog(title, message)

Create a GTK dialog box window with a feed back:
 dialog = GdgDialogQuestion(title, message, funct_pt)

Close a gadget
 _me.framework.close_gadget(gadget_name)

Exclude a gadget from the framework
 _me.framework.exclude_gadget(gadget_name)

Start a gadget
 _me.framework.start_gadget(gadget_path)

Restart a gadget
 _me.framework.restart_gadget(gadget_name)

1.1) Function : _me._init_

Constructor of class.

Parameters:
 "path" as string : Path of the TGF file

1.2) Function : _me.destroy

This function destroy all objects and close the TGF file.

1.3) Function : _me._print_debug

To print a formatted debug message.

Parameters:
 "message" as string : Message to print

1.5) Function : _me.main

Main function of the gadget.
 This function is overloaded with the code found in the
 'main.pyp' file of the TFG file.

1.6) Function : _me.run_main_loop

Function to run a blocking loop. The loop is stopped when
 the function 'exit_main_loop' is executed from an other
 thread.

1.7) Function : _me.exit_main_loop

Function to stop the blocking loop run by the function
 'run_main_loop'.

1.8) Function : _me.set_language

Set the language of the gadget.

Parameters:
 "lang_country" as string : Language of the gadget in
 "lang_country" format.
 See TGFParse.

1.9) Function : _me.refresh_GUI

Refresh the graphical user interfaces of the gadget.

1.10) Function : _me.restore_GUI

Restore the context of the graphical user interfaces.

Parameters:

"threads_protected" as boolean : Indicate if a GTK threads protection is needed.

Comments:

If this function is loaded from a thread, you must set the 'threads_protected' value to 'True'.

1.11) Function : `_me.string`

Get a string from the strings dictionary.

Parameters:

"msg_name" as string : Name of the message

Returns:

A string.

Comments:

If the entry in the dictionary is a list of strings, the returned value is a random value from this list.

1.12) Function : `_me.get_name`

Get the name of the gadget.

Returns:

A string

1.13) Function : `_me.get_author`

Get the author of the gadget.

Returns:

A string

1.14) Function : `_me.get_version`

Get the version of the gadget.

Returns:

A string

1.15) Function : `_me.get_description`

Get the description of the gadget.

Returns:

A string

1.16) Function : `_me.get_path`

Get a path from the 'paths' dictionary.

Parameters:

"path_name" as string : Name of the path

Returns:

A string.

Comments:

Possible values of "path_name" are:

'tgf' : Path of the TGF source file
'root' : Root path of the decompressed TGF

'data'	: Path of 'data'
'sounds'	: Path of 'sounds'
'pictures'	: Path of 'pictures'
'GUI'	: Path of the graphical user interfaces
'libs'	: Path of the python libraries
'behaviors'	: Path of 'behaviors'

1.17) Function : `_me.get_icon`

Get the path of an icon.

Parameters:

"icon_name" as string : Name of the icon

Returns:

The file path of the icon as string.

1.18) Function : `_me.get_sound`

Get the path of a sound.

Parameters:

"sound_name" as string : Name of the sound

Returns:

The file path of the sound as string.

1.19) Function : `_me.get_lib`

Get the path of a library.

Parameters:

"lib_name" as string : Name of the library

Returns:

The path of the library as string.

1.20) Function : `_me.gui`

Get a graphical user interface of the gadget.

Parameters:

"window_name" as string : Name of the GUI

Returns:

The GUI object as GdgGUI

1.21) Function : `_me.insert_func`

Insert a function.

Parameters:

"func_name" as string : Name of the function

"value" as pointer of function : Target function

1.22) Function : `_me.get_func`

Get the pointer of a function.

Parameters:

"func_name" as string : Name of the function

Returns:

Pointer as function.

1.23) Function : `_me.get_var`

Get the value of a variable.

Parameters:

"var_name" as string : Name of the variable

Returns:

The value of the variable.

Possible values of "var_name" are:

'language'	: Current language of the gadget in "lang_country" format. See TGFParse
'pitch'	: Current TTS pitch
'notified'	: Current state of notifications
'notify_delay'	: Delay between 2 notifications
'menu_active'	: Current state of the menu insertion
'name_to_read'	: Current translation of the gadget name
'speaker'	: Current TTS speaker

1.24) Function : `_me.set_var`

Set the value of a variable.

Parameters:

"var_name" as string : Name of the variable

"value" as untyped : Value of the variable

Possible values of "var_name" are:

'language'	: Current language of the gadget in "lang_country" format. See TGFParse
'pitch'	: Current TTS pitch
'notified'	: Current state of notifications
'notify_delay'	: Delay between 2 notifications
'menu_active'	: Current state of the menu insertion
'name_to_read'	: Current translation of the gadget name
'speaker'	: Current TTS speaker

1.25) Function : `_me.get_param`

Get the value of a parameter.

Parameters:

"param_name" as string : Name of the parameter

Returns:

The value of the parameter.

1.26) Function : `_me.set_param`

Set the value of a parameter.

Parameters:

"param_name" as string : Name of the parameter

"value" as untyped : Value of the parameter

1.27) Function : `_me.speak`

Speak a text.

Parameters:

"text" as string : Text to speak

Comments:

Do not use it in the 'notify_actuate' function of 'notify.pyp'.

1.28) Function : `_me.speak_free`

Speak a text in asynchronous mode.

Parameters:
"text" as string : Text to speak

Comments:
Do not use it in the 'notify_actuate' function of 'notify.pyp'.

1.29) Function : `_me.speak_notification`

Speak a text from the notification part.

Parameters:
"text" as string : Text to speak

Comment:
Only use it in the 'notify_actuate' function of 'notify.pyp'.

1.30) Function : `_me.wait_permission`

Wait a permission to continue.

1.31) Function : `_me.insert_menu`

Insert an item in the 'Options' menu of the manager.

Parameters:
"name" as string : Name of the item
"funct" as pointer of function : Function to launch on item event
"menu_type" as string : Menu item type
<'image', 'check', 'separator'>
"active_state" as boolean : State of checkbox for 'check' type
"stock_id" as GTK Stock ID type : GTK Stock id for 'image' type

1.32) Function : `_me.remove_menu`

Remove manager menu items created by this gadget.

1.33) Object [`_me.download`] as class(`GdgDownload.GdgDownload`)

Class which manages the download of files.

1.33.1) Function : `_me.download.check_server_available`

Check if a server is available.

Parameters:
"url" as string : url of the server
"timeout" as integer : timeout in seconds

Returns:
A boolean.

1.33.2) Function : `_me.download.download_url`

Download an url and return the content in a string.

Parameters:
"url" as string : url of the file
"timeout" as integer : timeout in seconds

Returns:
A tuple(boolean, string)

1.34) Object [`_me.notify`] as class(`GdgObject.GdgNotify`)

Class which create a notification controler.

1.34.1) Function : `_me.notify.start`

Start the loop of (check - actuate) notification.

Comments:

The loop is started on the initialization of the class.

1.34.2) Function : `_me.notify.stop`

Stop the loop of (check - actuate) notification.

1.34.3) Function : `_me.notify.set_active`

Set the state of the notification loop.

Parameters:

"value" as boolean : State

1.34.4) Function : `_me.notify.set_delay`

Set the delay between two cycles of the loop.

Parameters:

"value" as integer : Delay in seconds

1.34.5) Function : `_me.notify.check_now`

Run a cycle of (check - actuate) now.

1.35) Object [`_me.gui('gui name')`] as class(`GdgObject.GdgGUI`)

Class which create and control a GUI.

1.35.1) Function : `_me.gui('gui name').showed`

Indicate if the GUI is showed or not.

Returns:

A boolean

1.35.2) Function : `_me.gui('gui name').show`

Create an instance of the window and show it.

1.35.3) Function : `_me.gui('gui name').show_from_main`

Create an instance of the window and show it, from 'main.py'.

1.35.4) Function : `_me.gui('gui name').hide`

Destroy the instance of the window.

1.35.5) Function : `_me.gui('gui name').hide_from_main`

Destroy the instance of the window, from 'main.py'.

1.35.6) Function : `_me.gui('gui name').refresh`

Refresh the window.

1.36) Object [`_me.voice_recognition`] as class(`GdgObject.GdgVRManager`)

Class which manages the voice recognition

1.36.1) Function : `_me.voice_recognition.set_dict`

Set a word dictionary.

Parameters:

"dict_path" as string : path of the dictionary

1.36.2) Function : _me.voice_recognition.set_rules

Set a rules.

Parameters:

"rules" as list of list : rules

1.36.3) Function : _me.voice_recognition.get_run

Get the run state of the voice recognition.

Returns:

A boolean.

1.36.4) Function : _me.voice_recognition.push_dict

Push the current word dictionary, and the current rules.

1.36.5) Function : _me.voice_recognition.pop_dict

Pop the last saved word dictionary, and the last saved rules.

1.36.6) Function : _me.voice_recognition.clear_dict

Clear the current word dictionary, and the current rules.