

# I-SIMPA Scripting Guide

1.1.4

Generated by Doxygen 1.6.1

Tue Jan 19 09:41:48 2010



# Contents

<b>1</b>	<b>I-SIMPA scripting guide</b>	<b>1</b>
<b>2</b>	<b>I-SIMPA adding fonctionnality</b>	<b>3</b>
2.1	Register events . . . . .	4
2.2	Register a new menu manager . . . . .	4
2.3	Define getmenu method . . . . .	5
2.4	Run python code on element update . . . . .	5
<b>3</b>	<b>Add data in projects tree</b>	<b>7</b>
3.1	Creation of the new element type . . . . .	8
3.1.1	Tree label . . . . .	9
3.1.2	Icon . . . . .	9
3.1.3	Modification event . . . . .	9
3.2	Register the new module . . . . .	10
<b>4</b>	<b>Namespace Index</b>	<b>11</b>
4.1	Namespace List . . . . .	11
<b>5</b>	<b>Class Index</b>	<b>13</b>
5.1	Class Hierarchy . . . . .	13
<b>6</b>	<b>Class Index</b>	<b>15</b>
6.1	Class List . . . . .	15
<b>7</b>	<b>Namespace Documentation</b>	<b>17</b>
7.1	uictrl Namespace Reference . . . . .	17
7.1.1	Detailed Description . . . . .	20
7.1.2	Enumeration Type Documentation . . . . .	20
7.1.2.1	element_type . . . . .	20
7.1.2.2	graph . . . . .	23

7.1.2.3	idevent	24
<b>8</b>	<b>Class Documentation</b>	<b>27</b>
8.1	uictrl::application Class Reference	27
8.1.1	Detailed Description	27
8.1.2	Member Function Documentation	28
8.1.2.1	clearlogdata	28
8.1.2.2	clearshellhisto	28
8.1.2.3	getcachedir	28
8.1.2.4	getdataarray	28
8.1.2.5	getlastcalculationpath	28
8.1.2.6	getlocale	28
8.1.2.7	getrootcore	28
8.1.2.8	getrootpreference	28
8.1.2.9	getrootreport	29
8.1.2.10	getrootscene	29
8.1.2.11	getuserinput	29
8.1.2.12	importscene	29
8.1.2.13	loadproject	30
8.1.2.14	newproject	30
8.1.2.15	register_event	30
8.1.2.16	register_menu_manager	30
8.1.2.17	reloadgroupsfrommodel	30
8.1.2.18	savelog	30
8.1.2.19	saveproject	31
8.1.2.20	saveshell	31
8.1.2.21	sendevent	31
8.2	uictrl::e_file Class Reference	32
8.2.1	Detailed Description	32
8.2.2	Member Function Documentation	32
8.2.2.1	buildfullpath	32
8.3	uictrl::Element Struct Reference	33
8.3.1	Detailed Description	33
8.4	uictrl::element Class Reference	34
8.4.1	Detailed Description	35
8.4.2	Member Function Documentation	35
8.4.2.1	appendfilsbytype	35

8.4.2.2	appendpropertybool	35
8.4.2.3	appendpropertycolor	36
8.4.2.4	appendpropertydecimal	36
8.4.2.5	appendpropertyentier	37
8.4.2.6	appendpropertyfont	37
8.4.2.7	appendpropertylist	37
8.4.2.8	appendpropertyposition	38
8.4.2.9	appendpropertytext	38
8.4.2.10	appenduserelement	39
8.4.2.11	childs	39
8.4.2.12	deleteallementbytype	39
8.4.2.13	deleteallementbytyper	39
8.4.2.14	deleteelementbyxmlid	39
8.4.2.15	getallementbytype	39
8.4.2.16	getboolconfig	40
8.4.2.17	getcolorconfig	40
8.4.2.18	getdecimalconfig	40
8.4.2.19	getelementbylibelle	40
8.4.2.20	getelementbytype	40
8.4.2.21	getentierconfig	40
8.4.2.22	getid	41
8.4.2.23	getinfos	41
8.4.2.24	getlistconfig	41
8.4.2.25	getmenu	41
8.4.2.26	getpositionconfig	41
8.4.2.27	getstringconfig	42
8.4.2.28	hasproperty	42
8.4.2.29	hide	42
8.4.2.30	modified	42
8.4.2.31	register_update_manager	42
8.4.2.32	setreadonlyallconfig	42
8.4.2.33	setreadonlyconfig	43
8.4.2.34	updateboolconfig	43
8.4.2.35	updatedecimalconfig	43
8.4.2.36	updateentierconfig	43
8.4.2.37	updatelistconfig	44

8.4.2.38	<a href="#">updatepositionconfig</a>	44
8.4.2.39	<a href="#">updatestringconfig</a>	44

# Chapter 1

## I-SIMPA scripting guide

There are two main ways to make script for I-SIMPA. The first way is aiming at append fonctionnality via the right-click on tree elements. The goal of the second way is to append element data to projects tree.

- [I-SIMPA adding fonctionnality](#)
- [Add data in projects tree](#)





## **Chapter 2**

# **I-SIMPA adding fonctionnality**

Adding popup menu fonctionnality need 4 steps :

- Make module sub-folder in UserScript folder and add `__ui_startup.py` file that import this module.
- [Register events](#) in menu manager constructor.
- [Register a new menu manager](#) object in `__init__.py` file.
- [Define getmenu method](#) that will append items in menu list structure.

## 2.1 Register events

Each python implemented function had an integer index called event type. Register the function give this new event type index. This index will be used later in the menu structure at the last step.

To register new event type call the method `uictrl::application::register_event`

### Sample

```
class manager:
    """
    This class make the user able to enable or disable a group of emitters wi
    th one click only
    """
    def __init__(self):
        """
        Constructor. Register the two new menu functions
        """
        self.enable_grp_sourcesid=uictrl.application.register_event(self.enable_g
        rp_sources)
        self.disable_grp_sourcesid=uictrl.application.register_event(self.disable
        _grp_sources)
    def set_grp_src_activation(self, idgrp, newstate):
        grpsrc=uictrl.element(idgrp)
        all_property=grpsrc.getallelementbytype(uictrl.element_type.ELEMENT_TYPE_
        SCENE_SOURCES_SOURCE_PROPRIETES)
        for prop in all_property:
            uictrl.element(prop).updateboolconfig("enable", newstate)
    def enable_grp_sources(self, idgrp):
        """
        Called by user interface when the user click on the enable menu item
        """
        self.set_grp_src_activation(idgrp, True)
    def disable_grp_sources(self, idgrp):
        """
        Called by user interface when the user click on the disable menu item
        """
        self.set_grp_src_activation(idgrp, False)
```

## 2.2 Register a new menu manager

To register a new menu manager call the function `uictrl::application::register_menu_manager`

### Sample

```
uictrl.application.register_menu_manager(uictrl.element_type.ELEMENT_TYPE_SCENE_S
    OURCES, manager())
```

- The first parameter `uictrl::element_type` indicate the associated element type with the manager.
- The second parameter is the instance of the manager.

## 2.3 Define getmenu method

When the user right click on an items I-SIMPA will call the getmenu function of all menu manager registered with the item element type.

This method must return true if you have modified the menu list data, false otherwise.

### Sample

```
def getmenu(self, typeel, idel, menu) :
    """
        Called by the user interface
        The list menu structure contains the current implemented functions.
    """
    submenu=[(uictrl._("Enable"), self.enable_grp_sourcesid), (uictrl._("Disabl
e"), self.disable_grp_sourcesid)]
    menu.insert(2, (uictrl._("All emitters"), submenu))
    menu.insert(2, ())
    return True
```

## 2.4 Run python code on element update

You can link python method with any application element, this method is call when this element will be updated by I-SIMPA or the user. The method parameters must be an element id.

### Warning:

Do not call `ui::element::Update` due to infinite loop. The `uictrl::element::register_update_manager` do this operation



## **Chapter 3**

### **Add data in projects tree**

Adding fields to projects is not sufficient.

You must take control of these fields to implement constraint and to adding more functionality.

Step to add data in trees :

- Append a folder in UserScript/
- [Creation of the new element type](#) in the UserScript/yourmod/\_\_\_init\_\_\_py
- [Register the new module](#) in \_\_\_project\_loading\_\_\_py

### 3.1 Creation of the new element type

The sample in this guide is aiming at linking projet with a new calculation core. First of all, you need to build the python class inherit from `uictrl::element` class and with the `uictrl::element_type::ELEMENT_TYPE_CORE_CORE` base id.

#### Sample

```
class mdf(uictrl.element):
    """
    Diffusion model calculation core.
    """
    def __init__(self, idel):
        uictrl.element.__init__(self, idel)

        if not self.hasproperty("exeName"): #Test if this is a new project initialisation
            #If this is a new project then we add properties
            #Add tetgen parameters
            self.appendfilsbytype(uictrl.element_type.ELEMENT_TYPE_CORE_CORE_CONF
            MAILLAGE)
            #Add frequencies selection
            self.appendfilsbytype(uictrl.element_type.ELEMENT_TYPE_CORE_CORE_BFRE
            QSELECTION)
            #Add configuration core
            coreconf=uictrl.element(self.appendfilsbytype(uictrl.element_type.ELE
            MENT_TYPE_CORE_CORE_CONFIG))
            #Append hidden config, used by I-SIMPA to find the core files and bin
            aries
            uictrl.element(self.appendpropertytext("modelName", "", "mesh.cbin", Tru
            e, True)).hide()
            uictrl.element(self.appendpropertytext("tetrameshFileName", "", "tetram
            esh.mbin", True, True)).hide()
            uictrl.element(self.appendpropertytext("exeName", "", "md.py")).hide()
            uictrl.element(self.appendpropertytext("corePath", "", "md\\")).hide()

            #User options
            coreconf.appendpropertylist("solver_mode", "Calculation mode", [{"Time"
            , "Static"}, [0,1]], 0, False, 1, True)
            coreconf.appendpropertybool("with_direct_sound", "Use direct sound", Tr
            ue, True)
            _("Calculation mode")
            _("Use direct sound")
            _("Time")
            _("Static")

        else:
            pass #Here in case of loading an existing project
```

In the `__init__` constructor you can add your mod's properties . But you have to test their existence because this constructor is also called when loading a project.

### 3.1.1 Tree label

By default the name shown in the tree is the class name. To set another label you must override the `gettree-label` function.

#### Sample

```
def gettreelabel(self):
    """
        Return label
    """
    return "Mdf"
```

### 3.1.2 Icon

There are two kind of icon :

- Built-in icon referenced by the `uictrl::graph` enumeration and declared by `element::geticonid(self)` that return graph id.
- Local declaration of icon. Declared by `element::geticonpath(self)` that return the icon path.

#### Sample

```
def geticonid(self, state_open):
    """
        Return tree icon Id
    """
    if state_open:
        return uictrl.graph.GRAPH_FOLDER_OPEN
```

### 3.1.3 Modification event

From the property itself to the highest parent the method `uictrl::element::modified` is automatically called when the user change the value of a property.

This is a usefull method to implement properties constraints.

In our sample, we use this method to disable time dependant properties when the user choose the static resolution method.

#### Sample

```
def modified(self, idelmodified):
    #In case of sub element modification this func is call by ui
    #We disable the time dependant parameters in case of static solver mode
    if uictrl.element(idelmodified).getinfos()["name"]=="solver_mode":
        elconf=uictrl.element(self.getelementbytype(uictrl.element_type.ELEMENT_TYPE_CORE_CORE_CONFIG))
        is_temporal=(elconf.getlistconfig("solver_mode")==0)
        elconf.setreadonlyconfig("duree_simulation",not is_temporal)
        elconf.setreadonlyconfig("pasdetemps",not is_temporal)
        uictrl.element.modified(self, idelmodified)
```

## 3.2 Register the new module

All files named "\_\_project\_loading\_\_.py" in the UserScript/ folder and sub folder is executed when the user create or load a project.

You need to create this file in your module folder to register your new module in new projects and in existing projects that doesn't contain your module.

In the source code check first if your module was not already inserted in the project. Then insert your module thanks to the `uictrl::element::appenduserelement` method.

You can use the already imported library `uictrl` under the name "ui"

### Sample

```
rootcore=ui.element(ui.application.getrootcore())
#Check if our mod has been already inserted
if rootcore.getelementbylibelle("mdf")==-1: #Then append our mod
    rootcore.appenduserelement(ui.element_type.ELEMENT_TYPE_CORE_CORE,"mdf","mdf"
    )
```



# Chapter 4

## Namespace Index

### 4.1 Namespace List

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

<a href="#">uictrl</a> (Python embedding of c++ class ) . . . . .	17
---	----



# Chapter 5

## Class Index

### 5.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

uictrl::application . . . . .	27
uictrl::Element . . . . .	33
uictrl::element . . . . .	34
uictrl::e_file . . . . .	32



# Chapter 6

## Class Index

### 6.1 Class List

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

<a href="#">uictl::application</a> (Python <a href="#">application</a> control class ) . . . . .	27
<a href="#">uictl::e_file</a> . . . . .	32
<a href="#">uictl::Element</a> (Alias) . . . . .	33
<a href="#">uictl::element</a> (Give control on a built-in(c++) or python implemented <a href="#">element</a> ) . . . . .	34



# Chapter 7

## Namespace Documentation

### 7.1 uictrl Namespace Reference

Python embedding of c++ class.

#### Classes

- class [application](#)  
*Python [application](#) control class.*
- class [e\\_file](#)
- struct [Element](#)  
*Alias.*
- class [element](#)  
*Give control on a built-in(c++) or python implemented [element](#).*

#### Enumerations

- enum [element\\_type](#) {  
ELEMENT\_TYPE\_CORE\_ROOT, ELEMENT\_TYPE\_RESULT\_ROOT, ELEMENT\_TYPE\_-  
SCENE\_ROOT, ELEMENT\_TYPE\_SCENE\_PROJET\_CONFIGURATION,  
ELEMENT\_TYPE\_SCENE\_GROUPESURFACES, ELEMENT\_TYPE\_SCENE\_-  
GROUPESURFACES\_GROUPE, ELEMENT\_TYPE\_SCENE\_GROUPESURFACES\_GROUPE\_-  
VERTEX, ELEMENT\_TYPE\_SCENE\_RECEPTEURSP,  
ELEMENT\_TYPE\_SCENE\_RECEPTEURSP\_RECEPTEUR, ELEMENT\_TYPE\_SCENE\_-  
RECEPTEURSP\_RECEPTEUR\_PROPRIETES, ELEMENT\_TYPE\_SCENE\_RECEPTEURSP\_-  
RECEPTEUR\_RENDU, ELEMENT\_TYPE\_SCENE\_RECEPTEURSS,  
ELEMENT\_TYPE\_SCENE\_RECEPTEURSS\_RECEPTEUR, ELEMENT\_TYPE\_SCENE\_-  
RECEPTEURSS\_RECEPTEUR\_PROPRIETES, ELEMENT\_TYPE\_SCENE\_RECEPTEURSS\_-  
RECEPTEUR\_RENDU, ELEMENT\_TYPE\_SCENE\_SOURCES,  
ELEMENT\_TYPE\_SCENE\_SOURCES\_SOURCE, ELEMENT\_TYPE\_SCENE\_SOURCES\_-  
SOURCE\_PROPRIETES, ELEMENT\_TYPE\_SCENE\_SOURCES\_SOURCE\_PUISSANCE,  
ELEMENT\_TYPE\_SCENE\_SOURCES\_SOURCE\_RENDU,

ELEMENT\_TYPE\_TEXT, ELEMENT\_TYPE\_COLOR, ELEMENT\_TYPE\_POSITION,  
 ELEMENT\_TYPE\_LIST,  
 ELEMENT\_TYPE\_INTEGER, ELEMENT\_TYPE\_FLOAT, ELEMENT\_TYPE\_BOOL,  
 ELEMENT\_TYPE\_SCENE\_PROJET\_RENDU,  
 ELEMENT\_TYPE\_SCENE\_PROJET\_RENDU\_ORIGINE, ELEMENT\_TYPE\_SCENE\_-  
 PROJET\_RENDU\_MODEL, ELEMENT\_TYPE\_SCENE\_BDD, ELEMENT\_TYPE\_SCENE\_-  
 BDD\_SPECTRUMS,  
 ELEMENT\_TYPE\_SCENE\_BDD\_SPECTRUMS\_USER, ELEMENT\_TYPE\_SCENE\_BDD\_-  
 SPECTRUMS\_APP, ELEMENT\_TYPE\_SCENE\_BDD\_MATERIAUX, ELEMENT\_TYPE\_-  
 SCENE\_BDD\_MATERIAUX\_APP,  
 ELEMENT\_TYPE\_SCENE\_BDD\_MATERIAUX\_APP\_GROUP, ELEMENT\_TYPE\_SCENE\_-  
 BDD\_MATERIAUX\_APP\_MATERIAU, ELEMENT\_TYPE\_SCENE\_BDD\_MATERIAUX\_-  
 USER, ELEMENT\_TYPE\_SCENE\_BDD\_MATERIAUX\_USER\_GROUP,  
 ELEMENT\_TYPE\_SCENE\_BDD\_MATERIAUX\_USER\_MATERIAU, ELEMENT\_TYPE\_-  
 SCENE\_BDD\_MATERIAUX\_PROPMATERIAU, ELEMENT\_TYPE\_SCENE\_BDD\_-  
 CATMATERIAL, ELEMENT\_TYPE\_MATERIAU\_APP,  
 ELEMENT\_TYPE\_MATERIAU\_USER, ELEMENT\_TYPE\_GAMMEFREQ\_APP, ELEMENT\_-  
 TYPE\_GAMMEFREQ\_USER, ELEMENT\_TYPE\_PROPERTY\_FREQ,  
 ELEMENT\_TYPE\_ROW, ELEMENT\_TYPE\_ROW\_BFREQ, ELEMENT\_TYPE\_ROW\_-  
 MATERIAU, ELEMENT\_TYPE\_SCENE\_ENCOMBREMENTS,  
 ELEMENT\_TYPE\_SCENE\_ENCOMBREMENTS\_ENCOMBREMENT, ELEMENT\_-  
 TYPE\_SCENE\_ENCOMBREMENTS\_ENCOMBREMENT\_PROPRIETES, ELEMENT\_-  
 TYPE\_SCENE\_ENCOMBREMENTS\_ENCOMBREMENT\_CUBOIDE, ELEMENT\_TYPE\_-  
 GAMMEABSORPTION,  
 ELEMENT\_TYPE\_CORE\_SPPS, ELEMENT\_TYPE\_CORE\_CORE\_CONFIG, ELEMENT\_-  
 TYPE\_CORE\_CORE\_CONFMAILLAGE, ELEMENT\_TYPE\_SCENE\_PROJET,  
 ELEMENT\_TYPE\_SCENE\_PROJET\_USERCONFIGURATION, ELEMENT\_TYPE\_SCENE\_-  
 PROJET\_RENDU\_PARTICULES, ELEMENT\_TYPE\_SCENE\_DONNEES, ELEMENT\_TYPE\_-  
 SCENE\_ENCOMBREMENTS\_ENCOMBREMENT\_RENDU,  
 ELEMENT\_TYPE\_SCENE\_PROJET\_ENVIRONNEMENTCONF, ELEMENT\_TYPE\_-  
 DRAWABLE, ELEMENT\_TYPE\_CORE\_CORE\_BFREQSELECTION, ELEMENT\_TYPE\_-  
 BOOL\_BFREQ,  
 ELEMENT\_TYPE\_REPORT\_FOLDER, ELEMENT\_TYPE\_REPORT\_PARTVISUALISATION,  
 ELEMENT\_TYPE\_REPORT\_RECEPTEURSSVISUALISATION, ELEMENT\_TYPE\_-  
 REPORT\_GABE,  
 ELEMENT\_TYPE\_REPORT\_GABE\_RECIP, ELEMENT\_TYPE\_TREE\_LIST, ELEMENT\_-  
 TYPE\_CORE\_TC, ELEMENT\_TYPE\_SCENE\_PROJET\_INFORMATION,  
 ELEMENT\_TYPE\_SCENE\_BDD\_MATERIAUX\_MATERIAU\_RENDER, ELEMENT\_TYPE\_-  
 FONT, ELEMENT\_TYPE\_CORE\_TLM, ELEMENT\_TYPE\_REPORT\_GABE\_GAP,  
 ELEMENT\_TYPE\_REPORT\_UNKNOWN, ELEMENT\_TYPE\_CORE\_SPPS\_OCTREE,  
 ELEMENT\_TYPE\_REPORT\_RPI, ELEMENT\_TYPE\_SCENE\_VOLUMES,  
 ELEMENT\_TYPE\_SCENE\_VOLUMES\_VOLUME, ELEMENT\_TYPE\_SCENE\_VOLUMES\_-  
 VOLUME\_RENDU, ELEMENT\_TYPE\_SCENE\_VOLUMES\_VOLUME\_PROPRIETES,  
 ELEMENT\_TYPE\_PYTHON\_EXTENSION,  
 ELEMENT\_TYPE\_ELEMENT, ELEMENT\_TYPE\_CORE\_CORE, ELEMENT\_TYPE\_-  
 REPORT\_FILE, ELEMENT\_TYPE\_REPORT\_RECEPTEURSSVISUALISATION\_STANDART,



```

ELEMENT_TYPE_REPORT_RECEPTEURSSVISUALISATION_GAIN,      ELEMENT_
TYPE_REPORT_RECEPTEURSSVISUALISATION_TR,      ELEMENT_TYPE_REPORT_
RECEPTEURSSVISUALISATION_EDT,      ELEMENT_TYPE_SCENE_RECEPTEURSS_
RECEPTEURCOUPE,

ELEMENT_TYPE_SCENE_RECEPTEURSS_RECEPTEURCOUPE_PROPRIETES,
ELEMENT_TYPE_SCENE_RECEPTEURSS_RECEPTEURCOUPE_RENDU,      ELEMENT_
TYPE_USER_PREFERENCE_NODE, ELEMENT_TYPE_USER_PREFERENCE_ITEM,
ELEMENT_TYPE_USER_PREFERENCE_ITEM_ISOTEMPLATE }
• enum graph {
GRAPH_FOLDER, GRAPH_ITEM, GRAPH_FOLDER_OPEN, GRAPH_FITTINGS_OPEN,
GRAPH_FITTINGS_CLOSE, GRAPH_FITTING_OPEN, GRAPH_FITTING_CLOSE, GRAPH_
PUNCTUAL_RECEIVERS_OPEN,
GRAPH_PUNCTUAL_RECEIVERS_CLOSE,      GRAPH_SURFACE_RECEIVERS_OPEN,
GRAPH_SURFACE_RECEIVERS_CLOSE, GRAPH_SOUND_SOURCES_OPEN,
GRAPH_SOUND_SOURCES_CLOSE, GRAPH_SURFACES_OPEN, GRAPH_SURFACES_
CLOSE, GRAPH_VOLUMES_CLOSE,
GRAPH_VOLUMES_OPEN, GRAPH_PROJECT_OPEN, GRAPH_PROJECT_CLOSE,
GRAPH_DATA_CLOSE,
GRAPH_DATA_OPEN,      GRAPH_USER_MATERIALS_CLOSE,      GRAPH_USER_
MATERIALS_OPEN, GRAPH_APPLICATION_MATERIALS_CLOSE,
GRAPH_APPLICATION_MATERIALS_OPEN, GRAPH_DATABASE_CLOSE, GRAPH_
DATABASE_OPEN, GRAPH_USER_SPECTRUMS_CLOSE,
GRAPH_USER_SPECTRUMS_OPEN, GRAPH_APPLICATION_SPECTRUMS_CLOSE,
GRAPH_APPLICATION_SPECTRUMS_OPEN, GRAPH_MATERIAL_CLOSE,
GRAPH_MATERIAL_OPEN,      GRAPH_STANDARTCORE_CLOSE,      GRAPH_
STANDARTCORE_OPEN, GRAPH_CORES_CLOSE,
GRAPH_CORES_OPEN, GRAPH_SPPSCORE_CLOSE, GRAPH_SPPSCORE_OPEN,
GRAPH_DISK_FOLDER_OPEN,
GRAPH_DISK_FOLDER_CLOSE, GRAPH_ENVIRONMENT, GRAPH_PROJECT_
AUTHOR, GRAPH_INFORMATION,
GRAPH_DISK_DEFAULT_FILE, GRAPH_DISK_GABE, GRAPH_DISK_RS, GRAPH_
DISK_PARTICLE,
GRAPH_EL_CONFIGURATION, GRAPH_EL_3D_DISPLAY, GRAPH_EL_POSITION,
GRAPH_EL_TRIANGLE,
GRAPH_SPECTRUM, GRAPH_ORIGIN, GRAPH_TETMESH_PARAMETERS, GRAPH_
RENDERING_FOLDER_CLOSE,
GRAPH_RENDERING_FOLDER_OPEN,      GRAPH_ROOT_MATERIALS_OPEN,
GRAPH_ROOT_MATERIALS_CLOSE, GRAPH_ROOT_SPECTRUMS_OPEN,
GRAPH_ROOT_SPECTRUMS_CLOSE,      GRAPH_PUNCTUAL_RECEIVER_OPEN,
GRAPH_PUNCTUAL_RECEIVER_CLOSE, GRAPH_SURFACE_RECEIVER_OPEN,
GRAPH_SURFACE_RECEIVER_CLOSE, GRAPH_SOUND_SOURCE_OPEN, GRAPH_
SOUND_SOURCE_CLOSE, GRAPH_VOLUME_OPEN,
GRAPH_VOLUME_CLOSE, GRAPH_PREF_ANIMATION, GRAPH_PREF_GENERAL,
GRAPH_PREF_LEGEND,
GRAPH_PREF_NOISE_MAP, GRAPH_PREF_PARTICLES, GRAPH_USER_PREF_
ROOT_CLOSE, GRAPH_USER_PREF_ROOT_OPEN,
GRAPH_LAST_STATIC_GRAPH }

```

```

• enum idevent {
    IDEVENT_DELETE_ELEMENT,    IDEVENT_RENAME_ELEMENT,    IDEVENT_COPIER,
    IDEVENT_COLLER,

    IDEVENT_NEW_SURFACE_GROUP, IDEVENT_GETPROPERTIES, IDEVENT_SELECT_-
    TREE_ITEM, IDEVENT_NEW_RECEPTEUR_P,

    IDEVENT_SELECT_POSITION,    IDEVENT_NEW_SOURCE,    IDEVENT_NEW_-
    RECEPTEUR_S, IDEVENT_NEW_USERFREQ,

    IDEVENT_NEW_ENCOMBREMENT,    IDEVENT_NEW_ENCOMBREMENT_CUBOIDE,
    IDEVENT_NEW_USERMAT, IDEVENT_NEW_MATERIAL_GROUP,

    IDEVENT_RUN_CALCULATION,    IDEVENT_IMPORT_MATERIAL,    IDEVENT_LOAD_-
    PARTICLE_SIMULATION, IDEVENT_LOAD_PARTICLE_SIMULATION_PATH,

    IDEVENT_RELOAD_FOLDER,    IDEVENT_DELETE_FOLDER,    IDEVENT_LOAD_-
    RECEPTEURSS_SIMULATION_BY_TIMESTEP,    IDEVENT_LOAD_RECEPTEURSS_-
    SIMULATION_BY_TIMESTEP_SUM,

    IDEVENT_LOAD_RECEPTEURSS_SIMULATION_SUM,    IDEVENT_INVERT_FACE_-
    ORIENTATION,    IDEVENT_EMPTY_POINTER_VERTEX_GROUP,    IDEVENT_REC_-
    COMPUTE_ACOUSTIC_PARAMETERS,

    IDEVENT_RECEPTEURS_COMPUTE_TR,    IDEVENT_RECEPTEURS_COMPUTE_EDT,
    IDEVENT_REPORT_PARTICULES_MAKE_GABE,    IDEVENT_BFREQ_PRESELECTION_-
    NONE,

    IDEVENT_BFREQ_PRESELECTION_THIRD_BAND, IDEVENT_BFREQ_PRESELECTION_-
    BAND,    IDEVENT_BFREQ_PRESELECTION_BUILDING_THIRD_BAND,    IDEVENT_-
    BFREQ_PRESELECTION_BUILDING_BAND,

    IDEVENT_REC_-
    COMPUTE_ADVANCED_ACOUSTIC_PARAMETERS, IDEVENT_NEW_-
    RECEPTEURP_GROUP, IDEVENT_NEW_SOURCE_GROUP, IDEVENT_NEW_SURFACE_-
    GROUP_FROM_SELECTION,

    IDEVENT_LOAD_RECEPTEURSP_SIMULATION,    IDEVENT_BUILD_VOLUMES_FROM_-
    TRIMESH, IDEVENT_NEW_VOLUME, IDEVENT_OPEN_FOLDER,

    IDEVENT_CONVERT_VOL_TO_FITTING,    IDEVENT_NEW_RECEPTEUR_S_COUPE,
    IDEVENT_LAST_FIXED }

```

### 7.1.1 Detailed Description

Python embedding of c++ class.

### 7.1.2 Enumeration Type Documentation

#### 7.1.2.1 enum uictrl::element\_type

Available built-in [element](#) types

#### Enumerator:

*ELEMENT\_TYPE\_CORE\_ROOT* Built-in [element](#) type  
*ELEMENT\_TYPE\_RESULT\_ROOT* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_ROOT* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_PROJET\_CONFIGURATION* Built-in [element](#) type

*ELEMENT\_TYPE\_SCENE\_GROUPESURFACES* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_GROUPESURFACES\_GROUPE* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_GROUPESURFACES\_GROUPE\_VERTEX* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_RECEPTEURSP* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_RECEPTEURSP\_RECEPTEUR* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_RECEPTEURSP\_RECEPTEUR\_PROPRIETES* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_RECEPTEURSP\_RECEPTEUR\_RENDU* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_RECEPTEURSS* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_RECEPTEURSS\_RECEPTEUR* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_RECEPTEURSS\_RECEPTEUR\_PROPRIETES* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_RECEPTEURSS\_RECEPTEUR\_RENDU* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_SOURCES* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_SOURCES\_SOURCE* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_SOURCES\_SOURCE\_PROPRIETES* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_SOURCES\_SOURCE\_PUISSANCE* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_SOURCES\_SOURCE\_RENDU* Built-in [element](#) type  
*ELEMENT\_TYPE\_TEXT* Built-in [element](#) type  
*ELEMENT\_TYPE\_COLOR* Built-in [element](#) type  
*ELEMENT\_TYPE\_POSITION* Built-in [element](#) type  
*ELEMENT\_TYPE\_LIST* Built-in [element](#) type  
*ELEMENT\_TYPE\_INTEGER* Built-in [element](#) type  
*ELEMENT\_TYPE\_FLOAT* Built-in [element](#) type  
*ELEMENT\_TYPE\_BOOL* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_PROJET\_RENDU* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_PROJET\_RENDU\_ORIGINE* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_PROJET\_RENDU\_MODEL* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_BDD* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_BDD\_SPECTRUMS* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_BDD\_SPECTRUMS\_USER* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_BDD\_SPECTRUMS\_APP* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_BDD\_MATERIAUX* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_BDD\_MATERIAUX\_APP* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_BDD\_MATERIAUX\_APP\_GROUP* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_BDD\_MATERIAUX\_APP\_MATERIAU* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_BDD\_MATERIAUX\_USER* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_BDD\_MATERIAUX\_USER\_GROUP* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_BDD\_MATERIAUX\_USER\_MATERIAU* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_BDD\_MATERIAUX\_PROPMATERIAU* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_BDD\_CATMATERIAL* Built-in [element](#) type  
*ELEMENT\_TYPE\_MATERIAU\_APP* Built-in [element](#) type

*ELEMENT\_TYPE\_MATERIAU\_USER* Built-in [element](#) type  
*ELEMENT\_TYPE\_GAMMEFREQ\_APP* Built-in [element](#) type  
*ELEMENT\_TYPE\_GAMMEFREQ\_USER* Built-in [element](#) type  
*ELEMENT\_TYPE\_PROPERTY\_FREQ* Built-in [element](#) type  
*ELEMENT\_TYPE\_ROW* Built-in [element](#) type  
*ELEMENT\_TYPE\_ROW\_BFREQ* Built-in [element](#) type  
*ELEMENT\_TYPE\_ROW\_MATERIAU* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_ENCOMBREMENTS* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_ENCOMBREMENTS\_ENCOMBREMENT* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_ENCOMBREMENTS\_ENCOMBREMENT\_PROPRIETES* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_ENCOMBREMENTS\_ENCOMBREMENT\_CUBOIDE* Built-in [element](#) type  
*ELEMENT\_TYPE\_GAMMEABSORPTION* Built-in [element](#) type  
*ELEMENT\_TYPE\_CORE\_SPPS* Built-in [element](#) type  
*ELEMENT\_TYPE\_CORE\_CORE\_CONFIG* Built-in [element](#) type  
*ELEMENT\_TYPE\_CORE\_CORE\_CONFMAILLAGE* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_PROJET* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_PROJET\_USERCONFIGURATION* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_PROJET\_RENDU\_PARTICULES* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_DONNEES* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_ENCOMBREMENTS\_ENCOMBREMENT\_RENDU* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_PROJET\_ENVIRONNEMENTCONF* Built-in [element](#) type  
*ELEMENT\_TYPE\_DRAWABLE* 3D object [element](#)  
*ELEMENT\_TYPE\_CORE\_CORE\_BFREQSELECTION* Built-in [element](#) type  
*ELEMENT\_TYPE\_BOOL\_BFREQ* Built-in [element](#) type  
*ELEMENT\_TYPE\_REPORT\_FOLDER* Built-in [element](#) type  
*ELEMENT\_TYPE\_REPORT\_PARTVISUALISATION* Built-in [element](#) type  
*ELEMENT\_TYPE\_REPORT\_RECEPTEURSSVISUALISATION* Built-in [element](#) type  
*ELEMENT\_TYPE\_REPORT\_GABE* Built-in [element](#) type  
*ELEMENT\_TYPE\_REPORT\_GABE\_RECP* Built-in [element](#) type  
*ELEMENT\_TYPE\_TREE\_LIST* Built-in [element](#) type  
*ELEMENT\_TYPE\_CORE\_TC* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_PROJET\_INFORMATION* Built-in [element](#) type  
*ELEMENT\_TYPE\_SCENE\_BDD\_MATERIAUX\_MATERIAU\_RENDER* Built-in [element](#) type  
*ELEMENT\_TYPE\_FONT* Built-in [element](#) type  
*ELEMENT\_TYPE\_CORE\_TLM* Built-in [element](#) type  
*ELEMENT\_TYPE\_REPORT\_GABE\_GAP* Built-in [element](#) type  
*ELEMENT\_TYPE\_REPORT\_UNKNOWN* Fichier inconnu par PSPS mais connu par le système d'exploitation  
*ELEMENT\_TYPE\_CORE\_SPPS\_OCTREE* Built-in [element](#) type

*ELEMENT\_TYPE\_REPORT\_RPI* Built-in [element](#) type

*ELEMENT\_TYPE\_SCENE\_VOLUMES* Built-in [element](#) type

*ELEMENT\_TYPE\_SCENE\_VOLUMES\_VOLUME* Built-in [element](#) type

*ELEMENT\_TYPE\_SCENE\_VOLUMES\_VOLUME\_RENDU* Built-in [element](#) type

*ELEMENT\_TYPE\_SCENE\_VOLUMES\_VOLUME\_PROPRIETES* Built-in [element](#) type

*ELEMENT\_TYPE\_PYTHON\_EXTENSION* Built-in [element](#) type

*ELEMENT\_TYPE\_ELEMENT* Lors de la déclaration d'un élément utilisateur, ce type permet d'exprimer le fait que l'élément hérite directement de l'élément de base

*ELEMENT\_TYPE\_CORE\_CORE* Built-in [element](#) type

*ELEMENT\_TYPE\_REPORT\_FILE* Built-in [element](#) type

*ELEMENT\_TYPE\_REPORT\_RECEPTEURSSVISUALISATION\_STANDART* Built-in [element](#) type

*ELEMENT\_TYPE\_REPORT\_RECEPTEURSSVISUALISATION\_GAIN* Built-in [element](#) type

*ELEMENT\_TYPE\_REPORT\_RECEPTEURSSVISUALISATION\_TR* Built-in [element](#) type

*ELEMENT\_TYPE\_REPORT\_RECEPTEURSSVISUALISATION\_EDT* Built-in [element](#) type

*ELEMENT\_TYPE\_SCENE\_RECEPTEURSS\_RECEPTEURCOUPE* Built-in [element](#) type

*ELEMENT\_TYPE\_SCENE\_RECEPTEURSS\_RECEPTEURCOUPE\_PROPRIETES* Built-in [element](#) type

*ELEMENT\_TYPE\_SCENE\_RECEPTEURSS\_RECEPTEURCOUPE\_RENDU* Built-in [element](#) type

*ELEMENT\_TYPE\_USER\_PREFERENCE\_NODE* User preference node, in the user preference tree

*ELEMENT\_TYPE\_USER\_PREFERENCE\_ITEM* User preference item, in the user preference tree

*ELEMENT\_TYPE\_USER\_PREFERENCE\_ITEM\_ISOTEMPLATE* User preference item, in the user preference tree where user can choose iso palette.

### 7.1.2.2 enum uictrl::graph

Application tree icons

#### Enumerator:

*GRAPH\_FOLDER* Built-in picture declaration

*GRAPH\_ITEM* Built-in picture declaration

*GRAPH\_FOLDER\_OPEN* Built-in picture declaration

*GRAPH\_FITTINGS\_OPEN* Built-in picture declaration

*GRAPH\_FITTINGS\_CLOSE* Built-in picture declaration

*GRAPH\_FITTING\_OPEN* Built-in picture declaration

*GRAPH\_FITTING\_CLOSE* Built-in picture declaration

*GRAPH\_PUNCTUAL\_RECEIVERS\_OPEN* Built-in picture declaration

*GRAPH\_PUNCTUAL\_RECEIVERS\_CLOSE* Built-in picture declaration

*GRAPH\_SURFACE\_RECEIVERS\_OPEN* Built-in picture declaration

*GRAPH\_SURFACE\_RECEIVERS\_CLOSE* Built-in picture declaration

*GRAPH\_SOUND\_SOURCES\_OPEN* Built-in picture declaration  
*GRAPH\_SOUND\_SOURCES\_CLOSE* Built-in picture declaration  
*GRAPH\_SURFACES\_OPEN* Built-in picture declaration  
*GRAPH\_SURFACES\_CLOSE* Built-in picture declaration  
*GRAPH\_LAST\_STATIC\_GRAPH* Last graph id

### 7.1.2.3 enum uictrl:idevent

Available built-in events

Enumerator:

*IDEVENT\_DELETE\_ELEMENT* [Element](#) delete  
**Parameters:**

*IDEVENT\_RENAME\_ELEMENT* {"name",: "new label"} [Element](#) rename  
*IDEVENT\_COPIER* built-in event  
*IDEVENT\_COLLER* built-in event  
*IDEVENT\_NEW\_SURFACE\_GROUP* built-in event  
*IDEVENT\_GETPROPERTIES* built-in event  
*IDEVENT\_SELECT\_TREE\_ITEM* built-in event  
*IDEVENT\_NEW\_RECEPTEUR\_P* built-in event  
*IDEVENT\_SELECT\_POSITION* built-in event  
*IDEVENT\_NEW\_SOURCE* built-in event  
*IDEVENT\_NEW\_RECEPTEUR\_S* built-in event  
*IDEVENT\_NEW\_USERFREQ* built-in event  
*IDEVENT\_NEW\_ENCOMBREMENT* built-in event  
*IDEVENT\_NEW\_ENCOMBREMENT\_CUBOIDE* built-in event  
*IDEVENT\_NEW\_USERMAT* built-in event  
*IDEVENT\_NEW\_MATERIAL\_GROUP* built-in event  
*IDEVENT\_RUN\_CALCULATION* built-in event  
**Parameters:**

*IDEVENT\_IMPORT\_MATERIAL* {"path",: "material file path"} Import material file from odeon or Catt  
*IDEVENT\_LOAD\_PARTICLE\_SIMULATION* built-in event  
*IDEVENT\_LOAD\_PARTICLE\_SIMULATION\_PATH* built-in event  
*IDEVENT\_RELOAD\_FOLDER* built-in event  
*IDEVENT\_DELETE\_FOLDER* built-in event  
*IDEVENT\_LOAD\_RECEPTEURSS\_SIMULATION\_BY\_TIMESTEP* built-in event  
*IDEVENT\_LOAD\_RECEPTEURSS\_SIMULATION\_BY\_TIMESTEP\_SUM* built-in event  
*IDEVENT\_LOAD\_RECEPTEURSS\_SIMULATION\_SUM* built-in event  
*IDEVENT\_INVERT\_FACE\_ORIENTATION* built-in event  
*IDEVENT\_EMPTY\_POINTER\_VERTEX\_GROUP* built-in event  
*IDEVENT\_RECPCOMPUTE\_ACOUSTIC\_PARAMETERS* built-in event

*IDEVENT\_RECEPTEURS\_COMPUTE\_TR* built-in event  
*IDEVENT\_RECEPTEURS\_COMPUTE\_EDT* built-in event  
*IDEVENT\_REPORT\_PARTICULES\_MAKE\_GABE* built-in event  
*IDEVENT\_BFREQ\_PRESELECTION\_NONE* built-in event  
*IDEVENT\_BFREQ\_PRESELECTION\_THIRD\_BAND* built-in event  
*IDEVENT\_BFREQ\_PRESELECTION\_BAND* built-in event  
*IDEVENT\_BFREQ\_PRESELECTION\_BUILDING\_THIRD\_BAND* built-in event  
*IDEVENT\_BFREQ\_PRESELECTION\_BUILDING\_BAND* built-in event  
*IDEVENT\_RECP\_COMPUTE\_ADVANCED\_ACOUSTIC\_PARAMETERS* built-in event  
*IDEVENT\_NEW\_RECEPTEURP\_GROUP* built-in event  
*IDEVENT\_NEW\_SOURCE\_GROUP* built-in event  
*IDEVENT\_NEW\_SURFACE\_GROUP\_FROM\_SELECTION* built-in event  
*IDEVENT\_LOAD\_RECEPTEURSP\_SIMULATION* built-in event  
*IDEVENT\_BUILD\_VOLUMES\_FROM\_TRIMESH* built-in event  
*IDEVENT\_NEW\_VOLUME* built-in event  
*IDEVENT\_OPEN\_FOLDER* built-in event  
*IDEVENT\_CONVERT\_VOL\_TO\_FITTING* built-in event  
*IDEVENT\_NEW\_RECEPTEUR\_S\_COUPE* Add a cutting plan receiver event





# Chapter 8

## Class Documentation

### 8.1 uictrl::application Class Reference

Python [application](#) control class.

#### Static Public Member Functions

- static void [clearlogdata](#) ()
- static void [clearshellhisto](#) ()
- static std::string [getcachedir](#) ()
- static boost::python::list [getdataarray](#) (const [element](#) &pyel)
- static std::string [getlastcalculationpath](#) ()
- static std::string [getlocale](#) ()
- static int [getrootcore](#) ()
- static int [getrootpreference](#) ()
- static int [getrootreport](#) ()
- static int [getrootscene](#) ()
- static boost::python::tuple [getuserinput](#) (const std::string &title, const std::string &msg, boost::python::dict rows)
- static bool [importscene](#) (const std::string &path, bool keepexistingfacegroup=true, bool docorrection=true, bool domeshsurface=false, const std::string &paramTetgen="")
- static void [loadproject](#) (const std::string &path)
- static void [newproject](#) ()
- static int [register\\_event](#) (boost::python::object &func)
- static void [register\\_menu\\_manager](#) (const int &element\_typeid, boost::python::object &manager)
- static void [reloadgroupsfrommodel](#) ()
- static void [saveelog](#) (const std::string &path)
- static void [saveproject](#) (const std::string &path="")
- static void [saveshell](#) (const std::string &path)
- static void [sendevent](#) (const [element](#) &pyel, const int &idevent, boost::python::dict parameters=boost::python::dict())

#### 8.1.1 Detailed Description

Python [application](#) control class.

## 8.1.2 Member Function Documentation

### 8.1.2.1 `static void uictrl::application::clearlogdata () [static]`

Clear the log window history

### 8.1.2.2 `static void uictrl::application::clearshellhisto () [static]`

Clear the python log window history

### 8.1.2.3 `static std::string uictrl::application::getcachedir () [static]`

Return the projet cache directory

### 8.1.2.4 `static boost::python::list uictrl::application::getdataarray (const element & pyel) [static]`

Return the associated data array with the [element](#). For scene and core elements, the returned array is the property tab. For report [element](#) that herits from gabe [element](#), this method return the post-processed array.

#### Parameters:

*pyel* The data array will be extracted from this parameter.

### 8.1.2.5 `static std::string uictrl::application::getlastcalculationpath () [static]`

Return the last computation result folder.

### 8.1.2.6 `static std::string uictrl::application::getlocale () [static]`

Return the user selected language corresponding to the canonical form of current locale name. Canonical form is the one that is used on UNIX systems: it is a two- or five-letter string in `xx` or `xx_YY` format, where `xx` is ISO 639 code of language and `YY` is ISO 3166 code of the country. Examples are "en", "en\_GB", "en\_US" or "fr\_FR".

### 8.1.2.7 `static int uictrl::application::getrootcore () [static]`

Give the access to the root node of a projet tree

#### Returns:

The [element](#) id of the root core node

### 8.1.2.8 `static int uictrl::application::getrootpreference () [static]`

Give the access to the user preference root node of the [application](#) tree

#### Returns:

The [element](#) id of the root user preference

**8.1.2.9 static int uictrl::application::getrootreport () [static]**

Give the access to the root node of a projet tree

**Returns:**

The [element](#) id of the root report node

**8.1.2.10 static int uictrl::application::getrootscene () [static]**

Give the access to the root node of a projet tree

**Returns:**

The [element](#) id of the root scene node

**8.1.2.11 static boost::python::tuple uictrl::application::getuserinput (const std::string & title, const std::string & msg, boost::python::dict rows) [static]**

Show a window form where the user can write text in each field.

**Parameters:**

*title* Title of the window

*msg* Message text information.

*rows* Message fields.dict sample {"field one" : "default value" ,"field two" : "default value", "field three" : ["Value 1", "Value 2"] }

**Returns:**

Tuple (Bool, list) The first cell is the button user choice ok:True cancel:False. The second cell contain a list that had the same size of the rows parameter but contains the new fields values.

**8.1.2.12 static bool uictrl::application::importscene (const std::string & path, bool keepexistingfacegroup = true, bool docorrection = true, bool domeshsurface = false, const std::string & paramTetgen = "") [static]**

Import an outside defined model. Supported file format is \*.3ds;\*.ply;\*.bin;\*.poly

**Parameters:**

*path* Model file path

*keepexistingfacegroup* If True, it will try to fill face groups with the new faces thanks to old-new faces position.

*docorrection* Execute preprocess.exe to split or destroy triangle faces if needed.

*domeshsurface* Remesh the surface of the scene. This operation may increase the number of facets and destroy material color and textures data.

*paramTetgen* User defined parameter for the mesh software. Useless parameter if domeshsurface at False.

**Returns:**

True on success, false if import fails, see message log for details

**8.1.2.13 static void uictrl::application::loadproject (const std::string & *path*) [static]**

Load a file project

**Parameters:**

*path* File load path

**8.1.2.14 static void uictrl::application::newproject () [static]**

Close current project and make a new one

**8.1.2.15 static int uictrl::application::register\_event (boost::python::object & *func*) [static]**

Append a new event to I-SIMPA. See [I-SIMPA adding fonctionnality](#)

**Parameters:**

*func* Reference to the python function.

**Returns:**

Integer id of the new function

**8.1.2.16 static void uictrl::application::register\_menu\_manager (const int & *element\_typeid*, boost::python::object & *manager*) [static]**

Append a new menu manager for this [element](#) type. See [I-SIMPA adding fonctionnality](#)

**Parameters:**

*element\_typeid* [uictrl::element\\_type](#) Manager's linked [element](#) type.

*manager* Manager object instance.

**8.1.2.17 static void uictrl::application::reloadgroupsfrommodel () [static]**

Recharge les faces à partir du modèle

**8.1.2.18 static void uictrl::application::savelog (const std::string & *path*) [static]**

Save the content of the message log window

**Parameters:**

*path* Save file path

**8.1.2.19 static void uictrl::application::saveproject (const std::string & path = "") [static]**

Save the loaded project

**Parameters:**

*path* File save path

**8.1.2.20 static void uictrl::application::saveshell (const std::string & path) [static]**

Save the content of the python log window

**Parameters:**

*path* Save file path

**8.1.2.21 static void uictrl::application::sendevent (const element & pyel, const int & idevent, boost::python::dict parameters = boost::python::dict()) [static]**

Send an event to the interface.

**Parameters:**

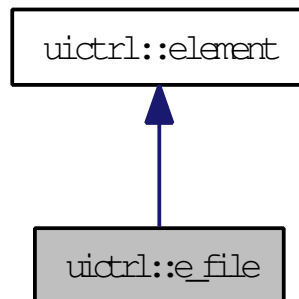
*pyel* Corresponding tree item node.

*idevent* Built-in [uictrl::idevent](#) or python defined event by [application::register\\_event](#)

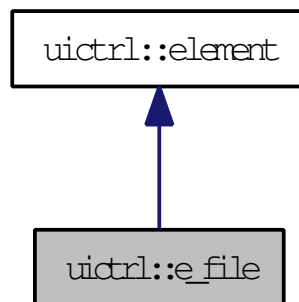
*parameters* See [uictrl::idevent](#) member for more specific details.

## 8.2 uictrl::e\_file Class Reference

Inheritance diagram for uictrl::e\_file:



Collaboration diagram for uictrl::e\_file:



### Public Member Functions

- `std::string buildfullpath ()`
- `e_file (const element &cpyFrom)`
- `e_file (const wxInt32 &_xmlId)`

### 8.2.1 Detailed Description

Specification of `element`, representative of a file or a folder

### 8.2.2 Member Function Documentation

#### 8.2.2.1 `std::string uictrl::e_file::buildfullpath ()`

Return the relative path of this file `element`.

## 8.3 uictrl::Element Struct Reference

Alias.

### Public Types

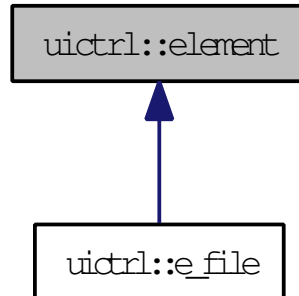
- typedef [element\\_type](#) **ELEMENT\_TYPE**
- typedef [graph](#) **GRAPH**
- typedef [idevent](#) **IDEVENT**

### 8.3.1 Detailed Description

Alias.

## 8.4 uictrl::element Class Reference

Give control on a built-in(c++) or python implemented [element](#). Inheritance diagram for uictrl::element:



### Public Member Functions

- `int appendfilsbytype (Element::ELEMENT_TYPE etypefils, const std::string &libelle="")`
- `int appendpropertybool (const std::string &propertyName, const std::string &propertyLabel, bool propertyDefaultValue, bool exportToCore=false)`
- `int appendpropertycolor (const std::string &propertyName, const std::string &propertyLabel, long defaultRed=0, long defaultGreen=0, long defaultBlue=0)`
- `int appendpropertydecimal (const std::string &propertyName, const std::string &propertyLabel, float propertyDefaultValue=0.f, bool readOnly=false, int precision=4, bool isMaxValue=false, bool isMinValue=false, float maxValue=0, float minValue=0, bool exportToCore=false)`
- `int appendpropertyentier (const std::string &propertyName, const std::string &propertyLabel, int propertyDefaultValue=0, bool exportToCore=false, bool isMaxValue=false, bool isMinValue=false, int maxValue=0, int minValue=0)`
- `int appendpropertyfont (const std::string &propertyName, const std::string &propertyLabel, const std::string &propertyDefaultValue="")`
- `int appendpropertylist (const std::string &propertyName, const std::string &propertyLabel, const boost::python::list &values, long defaultValue, bool asTitle=false, int hSize=1, bool exportToCore=false)`
- `int appendpropertyposition (const std::string &propertyName, const std::string &propertyLabel, const boost::python::list &propertyDefaultValue, bool exportToCore=false)`
- `int appendpropertytext (const std::string &propertyName, const std::string &propertyLabel, const std::string &propertyDefaultValue, bool readOnly=false, bool exportToCore=false)`
- `boost::python::object appenduserelement (const Element::ELEMENT_TYPE &baseType, const std::string &moduleName, const std::string &className)`
- `boost::python::list childs ()`
- `void deleteallelementbytype (Element::ELEMENT_TYPE typeElement)`
- `void deleteallelementbytyper (Element::ELEMENT_TYPE typeElementToDelete)`
- `bool deleteelementbyxmlid (int xmlIdElement, bool setModification=true)`
- `element (const element &cpyFrom)`
- `element (const wxInt32 &_xmlId)`
- `boost::python::list getallelementbytype (Element::ELEMENT_TYPE typeElement)`
- `bool getboolconfig (const std::string &name)`
- `boost::python::list getcolorconfig (const std::string &name)`
- `float getdecimalconfig (const std::string &name)`
- `int getelementbylibelle (std::string libelle)`



- int [getelementbytype](#) (Element::ELEMENT\_TYPE typeElement)
- int [getentierconfig](#) (const std::string &name)
- wxInt32 [getid](#) ()
- wxInt32 [getindice](#) () const
- boost::python::dict [getinfos](#) ()
- int [getlistconfig](#) (const std::string &name)
- boost::python::list [getmenu](#) ()
- boost::python::list [getpositionconfig](#) (const std::string &name)
- std::string [getstringconfig](#) (const std::string &name)
- bool [hasproperty](#) (const std::string &name)
- void [hide](#) (bool visible=false)
- void [modified](#) (int elementUpdated)
- void [register\\_update\\_manager](#) (boost::python::object &pymethod)
- void [setreadonlyallconfig](#) (bool readOnly=true, int col=0)
- void [setreadonlyconfig](#) (const std::string &name, bool readOnly=true, int col=0)
- bool [updateboolconfig](#) (const std::string &name, bool newValue)
- bool [updatedecimalconfig](#) (const std::string &name, float newValue)
- bool [updateentierconfig](#) (const std::string &name, int newValue)
- bool [updatelistconfig](#) (const std::string &name, int newIndex)
- bool [updatepositionconfig](#) (const std::string &name, const boost::python::list &newValue)
- bool [updatestringconfig](#) (const std::string &name, const std::string &newValue)

## Protected Attributes

- wxInt32 [xmlId](#)

### 8.4.1 Detailed Description

Give control on a built-in(c++) or python implemented [element](#). An [element](#) is the base class of all project data. From the basic string,bool,float and integer to a project tree node.

### 8.4.2 Member Function Documentation

#### 8.4.2.1 int uictrl::element::appendfilsbytype (Element::ELEMENT\_TYPE *etypefils*, const std::string & *libelle* = "")

Add a new child of etypefils [element](#) type and return this xml id. Return -1 in case of failure.

#### Parameters:

*etypefils* [Element](#) type of the children

#### 8.4.2.2 int uictrl::element::appendpropertybool (const std::string & *propertyName*, const std::string & *propertyLabel*, bool *propertyDefaultValue*, bool *exportToCore* = false)

Append a new property to an [element](#).

#### Parameters:

*propertyName* Name of the property. Each property name must be unique.

*propertyLabel* Untranslated property label. Use the language catalog (.po) to append the translation correspondance.

*propertyDefaultValue* Default value of the property

*exportToCore* Set it to True to make this property visible in calculation core xml configuration document.

**Returns:**

The [element](#) index of the new property or of the property of the same name. -1 in case of failure.

**8.4.2.3** `int uictrl::element::appendpropertycolor (const std::string & propertyName, const std::string & propertyLabel, long defaultRed = 0, long defaultGreen = 0, long defaultBlue = 0)`

Append a new property to an [element](#).

**Parameters:**

*propertyName* Name of the property. Each property name must be unique.

*propertyLabel* Untranslated property label. Use the language catalog (.po) to append the translation correspondance.

*defaultRed* Default red color. [0-255]

*defaultGreen* Default green color. [0-255]

*defaultBlue* Default blue color. [0-255]

**Returns:**

The [element](#) index of the new property or of the property of the same name. -1 in case of failure.

**8.4.2.4** `int uictrl::element::appendpropertydecimal (const std::string & propertyName, const std::string & propertyLabel, float propertyDefaultValue = 0.f, bool readOnly = false, int precision = 4, bool isMaxValue = false, bool isMinValue = false, float maxValue = 0, float minValue = 0, bool exportToCore = false)`

Append a new property to an [element](#).

**Parameters:**

*propertyName* Name of the property. Each property name must be unique.

*propertyLabel* Untranslated property label. Use the language catalog (.po) to append the translation correspondance.

*propertyDefaultValue* Default value of the property

*readOnly* Set it to True to forbid write user access to this property.

*precision* Precision showed after dot decimal separator.

*isMaxValue* True to activate max value constraint.

*isMinValue* True to activate min value constraint.

*maxValue* Maximum value of the field.

*minValue* Minimum value of the field.

*exportToCore* Set it to True to make this property visible in calculation core xml configuration document.

**Returns:**

The [element](#) index of the new property or of the property of the same name. -1 in case of failure.

**8.4.2.5** `int uictrl::element::appendpropertyentier (const std::string & propertyName, const std::string & propertyLabel, int propertyDefaultValue = 0, bool exportToCore = false, bool isMaxValue = false, bool isMinValue = false, int maxValue = 0, int minValue = 0)`

Append a new property to an [element](#).

**Parameters:**

*propertyName* Name of the property. Each property name must be unique.

*propertyLabel* Untranslated property label. Use the language catalog (.po) to append the translation correspondance.

*propertyDefaultValue* Default value of the property

*exportToCore* Set it to True to make this property visible in calculation core xml configuration document.

*isMaxValue* True to activate max value constraint.

*isMinValue* True to activate min value constraint.

*maxValue* Maximum value of the field.

*minValue* Minimum value of the field.

**Returns:**

The [element](#) index of the new property or of the property of the same name. -1 in case of failure.

**8.4.2.6** `int uictrl::element::appendpropertyfont (const std::string & propertyName, const std::string & propertyLabel, const std::string & propertyDefaultValue = "")`

Not implemented

**8.4.2.7** `int uictrl::element::appendpropertylist (const std::string & propertyName, const std::string & propertyLabel, const boost::python::list & values, long defaultValue, bool asTitle = false, int hSize = 1, bool exportToCore = false)`

Append a new property to an [element](#).

**Parameters:**

*propertyName* Name of the property. Each property name must be unique.

*propertyLabel* Untranslated property label. Use the language catalog (.po) to append the translation correspondance.

*values* List option with untranslated labels and index. [{"first list item"}, "second list item"], [0, 1]

*defaultValue* Default value of the property

*asTitle* At true this list will be the first property shown.

*hSize* Number of cols occupied by this property.

*exportToCore* Set it to True to make this property visible in calculation core xml configuration document.

**Returns:**

The [element](#) index of the new property or of the property of the same name. -1 in case of failure.

**8.4.2.8 int uictrl::element::appendpropertyposition (const std::string & *propertyName*, const std::string & *propertyLabel*, const boost::python::list & *propertyDefaultValue*, bool *exportToCore* = false)**

Append a new property to an [element](#).

**Parameters:**

*propertyName* Name of the property. Each property name must be unique.

*propertyLabel* Untranslated property label. Use the language catalog (.po) to append the translation correspondance.

*propertyDefaultValue* Default value of the property [x,y,z]

*exportToCore* Set it to True to make this property visible in calculation core xml configuration document.

**Returns:**

The [element](#) index of the new property or of the property of the same name. -1 in case of failure.

**8.4.2.9 int uictrl::element::appendpropertytext (const std::string & *propertyName*, const std::string & *propertyLabel*, const std::string & *propertyDefaultValue*, bool *readOnly* = false, bool *exportToCore* = false)**

Append a new property to an [element](#).

**Parameters:**

*propertyName* Name of the property. Each property name must be unique.

*propertyLabel* Untranslated property label. Use the language catalog (.po) to append the translation correspondance.

*propertyDefaultValue* Default value of the property

*readOnly* Set it to True to forbid write user access to this property.

*exportToCore* Set it to True to make this property visible in calculation core xml configuration document.

**Returns:**

The [element](#) index of the new property or of the property of the same name. -1 in case of failure.

#### 8.4.2.10 boost::python::object uictrl::element::appenduserelement (const Element::ELEMENT\_TYPE & *baseType*, const std::string & *moduleName*, const std::string & *className*)

Set user defined python object as a children of this [element](#)

##### Parameters:

*baseType* Base type of the new [element](#)

*moduleName* Module name of the new [element](#)

*className* Class name of the new [element](#)

#### 8.4.2.11 boost::python::list uictrl::element::childs ()

For each [element](#) child this function return The index of [element](#),the type, the name

#### 8.4.2.12 void uictrl::element::deleteallelementbytype (Element::ELEMENT\_TYPE *typeElement*)

Delete immediate childs elements corresponding to this [element](#) type

##### Parameters:

*typeElement* [Element](#) type

#### 8.4.2.13 void uictrl::element::deleteallelementbytyper (Element::ELEMENT\_TYPE *typeElementToDelete*)

Delete recursively childs elements corresponding to this [element](#) type

##### Parameters:

*typeElement* [Element](#) type

#### 8.4.2.14 bool uictrl::element::deleteelementbyxmlid (int *xmlIdElement*, bool *setModification* = true)

Delete immediate child [element](#) corresponding to this id.

##### Parameters:

*xmlIdElement* [Element](#) id

*setModification* If true, call [element::modified](#) automatically

#### 8.4.2.15 boost::python::list uictrl::element::getallelementbytype (Element::ELEMENT\_TYPE *typeElement*)

Navigate recursively through childrens and return an index list of all corresponding elements.

#### 8.4.2.16 `bool uictrl::element::getboolconfig (const std::string & name)`

Return the value of a property

**Parameters:**

*name* Name of the property. Please check with [element::hasproperty](#) if there are any doubt.

**Returns:**

Value of the property

#### 8.4.2.17 `boost::python::list uictrl::element::getcolorconfig (const std::string & name)`

Return the value of a property

**Parameters:**

*name* Name of the property. Please check with [element::hasproperty](#) if there are any doubt.

**Returns:**

Value of the property [red,green,blue] [0-255]

#### 8.4.2.18 `float uictrl::element::getdecimalconfig (const std::string & name)`

Return the value of a property

**Parameters:**

*name* Name of the property. Please check with [element::hasproperty](#) if there are any doubt.

**Returns:**

Value of the property

#### 8.4.2.19 `int uictrl::element::getelementbylibelle (std::string libelle)`

Return the [element](#) id of the first children with the same [element](#) name

#### 8.4.2.20 `int uictrl::element::getelementbytype (Element::ELEMENT_TYPE typeElement)`

Return the [element](#) id of the first children with the `element_type`

#### 8.4.2.21 `int uictrl::element::getentierconfig (const std::string & name)`

Return the value of a property

**Parameters:**

*name* Name of the property. Please check with [element::hasproperty](#) if there are any doubt.

**Returns:**

Value of the property

#### 8.4.2.22 wxInt32 uictrl::element::getid () [inline]

**Returns:**

[Element](#) index

#### 8.4.2.23 boost::python::dict uictrl::element::getinfos ()

Return a dict with the following keys :

- typeElement : [Element](#) Type ([uictrl::element\\_type](#))
- xmlIdElement : [Element](#) index
- expanded : True if [element](#) tree is expanded
- userDestroyable : True if the user is able to destroy this [element](#)
- label : Untranslated version of the [element](#) label
- label\_located : Translated [element](#) label using the locale language of [application](#).
- name : [Element](#) name.
- parentid : Index of the parent [element](#). -1 if none.

#### 8.4.2.24 int uictrl::element::getlistconfig (const std::string & name)

Return the value of a property

**Parameters:**

*name* Name of the property. Please check with [element::hasproperty](#) if there are any doubt.

**Returns:**

Value of the property

#### 8.4.2.25 boost::python::list uictrl::element::getmenu ()

Return the final (built-in+python) menu with nested list containing tuple (translated name, event\_id)

#### 8.4.2.26 boost::python::list uictrl::element::getpositionconfig (const std::string & name)

Return the value of a property

**Parameters:**

*name* Name of the property. Please check with [element::hasproperty](#) if there are any doubt.

**Returns:**

Value of the property [x,y,z] (m)

#### 8.4.2.27 `std::string uictrl::element::getstringconfig (const std::string & name)`

Return the value of a property

**Parameters:**

*name* Name of the property. Please check with [element::hasproperty](#) if there are any doubt.

**Returns:**

Value of the property

#### 8.4.2.28 `bool uictrl::element::hasproperty (const std::string & name)`

Search inside the element's properties, and return True if a property had this name.

**Parameters:**

*hasproperty* Property name

#### 8.4.2.29 `void uictrl::element::hide (bool visible = false)`

Hide this [element](#). It can't be seen by the user.

**Parameters:**

*visible* New state of visibility

#### 8.4.2.30 `void uictrl::element::modified (int elementUpdated)`

Tag this [element](#) and its parents as modified and will be saved later.

**Parameters:**

*elementUpdated* Updated [element](#) xml id

#### 8.4.2.31 `void uictrl::element::register_update_manager (boost::python::object & pymethod)`

Link a python function to the event of [element](#) update. You can add multiple links. Links are alive until project close.

**Parameters:**

*func* Python class with a function called OnUpdate with one parameter, the updated [element](#) index, may be this [element](#) or a child.

#### 8.4.2.32 `void uictrl::element::setreadonlyallconfig (bool readOnly = true, int col = 0)`

Enable or disable the write access to all property of an [element](#).

**Parameters:**

*readOnly* New state of access

*col* For row property, you can set read only on a specific col only.



**8.4.2.33 void uictrl::element::setreadonlyconfig (const std::string & *name*, bool *readOnly* = true, int *col* = 0)**

Enable or disable the write access to a property

**Parameters:**

*name* Name of the property

*readOnly* New state of access

*col* For row property, you can set read only on a specific col only.

**8.4.2.34 bool uictrl::element::updateboolconfig (const std::string & *name*, bool *newValue*)**

Update the value of a property.

**Parameters:**

*name* Name of the property

*newValue* New value of the property

**Returns:**

True if the property has been found and updated

**8.4.2.35 bool uictrl::element::updatedecimalconfig (const std::string & *name*, float *newValue*)**

Update the value of a property.

**Parameters:**

*name* Name of the property

*newValue* New value of the property

**Returns:**

True if the property has been found and updated

**8.4.2.36 bool uictrl::element::updateentierconfig (const std::string & *name*, int *newValue*)**

Update the value of a property.

**Parameters:**

*name* Name of the property

*newValue* New value of the property

**Returns:**

True if the property has been found and updated

**8.4.2.37** `bool uictrl::element::updatelistconfig (const std::string & name, int newIndex)`

Update the value of a property.

**Parameters:**

*name* Name of the property

*newIndex* New value of the property

**Returns:**

True if the property has been found and updated

**8.4.2.38** `bool uictrl::element::updatepositionconfig (const std::string & name, const boost::python::list & newValue)`

Update the value of a property.

**Parameters:**

*name* Name of the property

*newValue* New value of the property [x,y,z]

**Returns:**

True if the property has been found and updated

**8.4.2.39** `bool uictrl::element::updatestringconfig (const std::string & name, const std::string & newValue)`

Update the value of a property.

**Parameters:**

*name* Name of the property

*newValue* New value of the property

**Returns:**

True if the property has been found and updated

# Index

- appendfilesbytype
  - uictrl::element, 35
- appendpropertybool
  - uictrl::element, 35
- appendpropertycolor
  - uictrl::element, 36
- appendpropertydecimal
  - uictrl::element, 36
- appendpropertyentier
  - uictrl::element, 37
- appendpropertyfont
  - uictrl::element, 37
- appendpropertylist
  - uictrl::element, 37
- appendpropertyposition
  - uictrl::element, 38
- appendpropertytext
  - uictrl::element, 38
- appenduserelement
  - uictrl::element, 38
  
- buildfullpath
  - uictrl::e\_file, 32
  
- childs
  - uictrl::element, 39
- clearlogdata
  - uictrl::application, 28
- clearshellhisto
  - uictrl::application, 28
  
- deleteallelementbytype
  - uictrl::element, 39
- deleteallelementbytyper
  - uictrl::element, 39
- deleteelementbyxmlid
  - uictrl::element, 39
  
- ELEMENT\_TYPE\_BOOL
  - uictrl, 21
- ELEMENT\_TYPE\_BOOL\_BFREQ
  - uictrl, 22
- ELEMENT\_TYPE\_COLOR
  - uictrl, 21
- ELEMENT\_TYPE\_CORE\_CORE\_
  - uictrl, 23
- ELEMENT\_TYPE\_CORE\_CORE\_
  - BFREQSELECTION
  - uictrl, 22
- ELEMENT\_TYPE\_CORE\_CORE\_CONFIG
  - uictrl, 22
- ELEMENT\_TYPE\_CORE\_CORE\_
  - CONFMAILLAGE
  - uictrl, 22
- ELEMENT\_TYPE\_CORE\_ROOT
  - uictrl, 20
- ELEMENT\_TYPE\_CORE\_SPPS
  - uictrl, 22
- ELEMENT\_TYPE\_CORE\_SPPS\_OCTREE
  - uictrl, 22
- ELEMENT\_TYPE\_CORE\_TC
  - uictrl, 22
- ELEMENT\_TYPE\_CORE\_TLM
  - uictrl, 22
- ELEMENT\_TYPE\_DRAWABLE
  - uictrl, 22
- ELEMENT\_TYPE\_ELEMENT
  - uictrl, 23
- ELEMENT\_TYPE\_FLOAT
  - uictrl, 21
- ELEMENT\_TYPE\_FONT
  - uictrl, 22
- ELEMENT\_TYPE\_GAMMEABSORPTION
  - uictrl, 22
- ELEMENT\_TYPE\_GAMMEFREQ\_APP
  - uictrl, 22
- ELEMENT\_TYPE\_GAMMEFREQ\_USER
  - uictrl, 22
- ELEMENT\_TYPE\_INTEGER
  - uictrl, 21
- ELEMENT\_TYPE\_LIST
  - uictrl, 21
- ELEMENT\_TYPE\_MATERIAU\_APP
  - uictrl, 21
- ELEMENT\_TYPE\_MATERIAU\_USER
  - uictrl, 21
- ELEMENT\_TYPE\_POSITION
  - uictrl, 21
- ELEMENT\_TYPE\_PROPERTY\_FREQ
  - uictrl, 22
- ELEMENT\_TYPE\_PYTHON\_EXTENSION

- uictrl, 23
- ELEMENT\_TYPE\_REPORT\_FILE
  - uictrl, 23
- ELEMENT\_TYPE\_REPORT\_FOLDER
  - uictrl, 22
- ELEMENT\_TYPE\_REPORT\_GABE
  - uictrl, 22
- ELEMENT\_TYPE\_REPORT\_GABE\_GAP
  - uictrl, 22
- ELEMENT\_TYPE\_REPORT\_GABE\_RECPC
  - uictrl, 22
- ELEMENT\_TYPE\_REPORT\_-
  - PARTVISUALISATION
    - uictrl, 22
- ELEMENT\_TYPE\_REPORT\_-
  - RECEPTEURSSVISUALISATION
    - uictrl, 22
- ELEMENT\_TYPE\_REPORT\_-
  - RECEPTEURSSVISUALISATION\_-
    - EDT
      - uictrl, 23
- ELEMENT\_TYPE\_REPORT\_-
  - RECEPTEURSSVISUALISATION\_-
    - GAIN
      - uictrl, 23
- ELEMENT\_TYPE\_REPORT\_-
  - RECEPTEURSSVISUALISATION\_-
    - STANDART
      - uictrl, 23
- ELEMENT\_TYPE\_REPORT\_-
  - RECEPTEURSSVISUALISATION\_TR
    - uictrl, 23
- ELEMENT\_TYPE\_REPORT\_RPI
  - uictrl, 22
- ELEMENT\_TYPE\_REPORT\_UNKNOWN
  - uictrl, 22
- ELEMENT\_TYPE\_RESULT\_ROOT
  - uictrl, 20
- ELEMENT\_TYPE\_ROW
  - uictrl, 22
- ELEMENT\_TYPE\_ROW\_BFREQ
  - uictrl, 22
- ELEMENT\_TYPE\_ROW\_MATERIAU
  - uictrl, 22
- ELEMENT\_TYPE\_SCENE\_BDD
  - uictrl, 21
- ELEMENT\_TYPE\_SCENE\_BDD\_-
  - CATMATERIAL
    - uictrl, 21
- ELEMENT\_TYPE\_SCENE\_BDD\_MATERIAUX
  - uictrl, 21
- ELEMENT\_TYPE\_SCENE\_BDD\_-
  - MATERIAUX\_APP
    - uictrl, 21
- ELEMENT\_TYPE\_SCENE\_BDD\_-
  - MATERIAUX\_APP\_GROUP
    - uictrl, 21
- ELEMENT\_TYPE\_SCENE\_BDD\_-
  - MATERIAUX\_APP\_MATERIAU
    - uictrl, 21
- ELEMENT\_TYPE\_SCENE\_BDD\_-
  - MATERIAUX\_MATERIAU\_RENDER
    - uictrl, 22
- ELEMENT\_TYPE\_SCENE\_BDD\_-
  - MATERIAUX\_PROPMATERIAU
    - uictrl, 21
- ELEMENT\_TYPE\_SCENE\_BDD\_-
  - MATERIAUX\_USER
    - uictrl, 21
- ELEMENT\_TYPE\_SCENE\_BDD\_-
  - MATERIAUX\_USER\_GROUP
    - uictrl, 21
- ELEMENT\_TYPE\_SCENE\_BDD\_-
  - MATERIAUX\_USER\_MATERIAU
    - uictrl, 21
- ELEMENT\_TYPE\_SCENE\_BDD\_SPECTRUMS
  - uictrl, 21
- ELEMENT\_TYPE\_SCENE\_BDD\_-
  - SPECTRUMS\_APP
    - uictrl, 21
- ELEMENT\_TYPE\_SCENE\_BDD\_-
  - SPECTRUMS\_USER
    - uictrl, 21
- ELEMENT\_TYPE\_SCENE\_DONNEES
  - uictrl, 22
- ELEMENT\_TYPE\_SCENE\_-
  - ENCOMBREMENTS
    - uictrl, 22
- ELEMENT\_TYPE\_SCENE\_-
  - ENCOMBREMENTS\_-
    - ENCOMBREMENT
      - uictrl, 22
- ELEMENT\_TYPE\_SCENE\_-
  - ENCOMBREMENTS\_-
    - ENCOMBREMENT\_CUBOIDE
      - uictrl, 22
- ELEMENT\_TYPE\_SCENE\_-
  - ENCOMBREMENTS\_-
    - ENCOMBREMENT\_PROPRIETES
      - uictrl, 22
- ELEMENT\_TYPE\_SCENE\_-
  - ENCOMBREMENTS\_-
    - ENCOMBREMENT\_RENDU
      - uictrl, 22
- ELEMENT\_TYPE\_SCENE\_-
  - GROUPESURFACES
    - uictrl, 20

- ELEMENT\_TYPE\_SCENE\_-  
  GROUPESURFACES\_GROUPE  
  uictrl, 21
- ELEMENT\_TYPE\_SCENE\_-  
  GROUPESURFACES\_GROUPE\_-  
  VERTEX  
  uictrl, 21
- ELEMENT\_TYPE\_SCENE\_PROJET  
  uictrl, 22
- ELEMENT\_TYPE\_SCENE\_PROJET\_-  
  CONFIGURATION  
  uictrl, 20
- ELEMENT\_TYPE\_SCENE\_PROJET\_-  
  ENVIRONNEMENTCONF  
  uictrl, 22
- ELEMENT\_TYPE\_SCENE\_PROJET\_-  
  INFORMATION  
  uictrl, 22
- ELEMENT\_TYPE\_SCENE\_PROJET\_RENDU  
  uictrl, 21
- ELEMENT\_TYPE\_SCENE\_PROJET\_RENDU\_-  
  MODEL  
  uictrl, 21
- ELEMENT\_TYPE\_SCENE\_PROJET\_RENDU\_-  
  ORIGINE  
  uictrl, 21
- ELEMENT\_TYPE\_SCENE\_PROJET\_RENDU\_-  
  PARTICULES  
  uictrl, 22
- ELEMENT\_TYPE\_SCENE\_PROJET\_-  
  USERCONFIGURATION  
  uictrl, 22
- ELEMENT\_TYPE\_SCENE\_RECEPTEURSP  
  uictrl, 21
- ELEMENT\_TYPE\_SCENE\_RECEPTEURSP\_-  
  RECEPTEUR  
  uictrl, 21
- ELEMENT\_TYPE\_SCENE\_RECEPTEURSP\_-  
  RECEPTEUR\_PROPRIETES  
  uictrl, 21
- ELEMENT\_TYPE\_SCENE\_RECEPTEURSP\_-  
  RECEPTEUR\_RENDU  
  uictrl, 21
- ELEMENT\_TYPE\_SCENE\_RECEPTEURSS  
  uictrl, 21
- ELEMENT\_TYPE\_SCENE\_RECEPTEURSS\_-  
  RECEPTEUR  
  uictrl, 21
- ELEMENT\_TYPE\_SCENE\_RECEPTEURSS\_-  
  RECEPTEUR\_PROPRIETES  
  uictrl, 21
- ELEMENT\_TYPE\_SCENE\_RECEPTEURSS\_-  
  RECEPTEUR\_RENDU  
  uictrl, 21
- ELEMENT\_TYPE\_SCENE\_RECEPTEURSS\_-  
  RECEPTEURCOUPE  
  uictrl, 23
- ELEMENT\_TYPE\_SCENE\_RECEPTEURSS\_-  
  RECEPTEURCOUPE\_PROPRIETES  
  uictrl, 23
- ELEMENT\_TYPE\_SCENE\_RECEPTEURSS\_-  
  RECEPTEURCOUPE\_RENDU  
  uictrl, 23
- ELEMENT\_TYPE\_SCENE\_ROOT  
  uictrl, 20
- ELEMENT\_TYPE\_SCENE\_SOURCES  
  uictrl, 21
- ELEMENT\_TYPE\_SCENE\_SOURCES\_-  
  SOURCE  
  uictrl, 21
- ELEMENT\_TYPE\_SCENE\_SOURCES\_-  
  SOURCE\_PROPRIETES  
  uictrl, 21
- ELEMENT\_TYPE\_SCENE\_SOURCES\_-  
  SOURCE\_PUISSANCE  
  uictrl, 21
- ELEMENT\_TYPE\_SCENE\_SOURCES\_-  
  SOURCE\_RENDU  
  uictrl, 21
- ELEMENT\_TYPE\_SCENE\_VOLUMES  
  uictrl, 23
- ELEMENT\_TYPE\_SCENE\_VOLUMES\_-  
  VOLUME  
  uictrl, 23
- ELEMENT\_TYPE\_SCENE\_VOLUMES\_-  
  VOLUME\_PROPRIETES  
  uictrl, 23
- ELEMENT\_TYPE\_SCENE\_VOLUMES\_-  
  VOLUME\_RENDU  
  uictrl, 23
- ELEMENT\_TYPE\_TEXT  
  uictrl, 21
- ELEMENT\_TYPE\_TREE\_LIST  
  uictrl, 22
- ELEMENT\_TYPE\_USER\_PREFERENCE\_ITEM  
  uictrl, 23
- ELEMENT\_TYPE\_USER\_PREFERENCE\_-  
  ITEM\_ISOTEMPLATE  
  uictrl, 23
- ELEMENT\_TYPE\_USER\_PREFERENCE\_-  
  NODE  
  uictrl, 23
- element\_type  
  uictrl, 20
- getallementbytype  
  uictrl::element, 39
- getboolconfig

- uictrl::element, 39
- getcachedir
  - uictrl::application, 28
- getcolorconfig
  - uictrl::element, 40
- getdataarray
  - uictrl::application, 28
- getdecimalconfig
  - uictrl::element, 40
- getelementbylibelle
  - uictrl::element, 40
- getelementbytype
  - uictrl::element, 40
- getentierconfig
  - uictrl::element, 40
- getid
  - uictrl::element, 40
- getinfos
  - uictrl::element, 41
- getlastcalculationpath
  - uictrl::application, 28
- getlistconfig
  - uictrl::element, 41
- getlocale
  - uictrl::application, 28
- getmenu
  - uictrl::element, 41
- getpositionconfig
  - uictrl::element, 41
- getrootcore
  - uictrl::application, 28
- getrootpreference
  - uictrl::application, 28
- getrootreport
  - uictrl::application, 28
- getrootscene
  - uictrl::application, 29
- getstringconfig
  - uictrl::element, 41
- getuserinput
  - uictrl::application, 29
- graph
  - uictrl, 23
- GRAPH\_FITTING\_CLOSE
  - uictrl, 23
- GRAPH\_FITTING\_OPEN
  - uictrl, 23
- GRAPH\_FITTINGS\_CLOSE
  - uictrl, 23
- GRAPH\_FITTINGS\_OPEN
  - uictrl, 23
- GRAPH\_FOLDER
  - uictrl, 23
- GRAPH\_FOLDER\_OPEN
  - uictrl, 23
- GRAPH\_ITEM
  - uictrl, 23
- GRAPH\_LAST\_STATIC\_GRAPH
  - uictrl, 24
- GRAPH\_PUNCTUAL\_RECEIVERS\_CLOSE
  - uictrl, 23
- GRAPH\_PUNCTUAL\_RECEIVERS\_OPEN
  - uictrl, 23
- GRAPH\_SOUND\_SOURCES\_CLOSE
  - uictrl, 24
- GRAPH\_SOUND\_SOURCES\_OPEN
  - uictrl, 23
- GRAPH\_SURFACE\_RECEIVERS\_CLOSE
  - uictrl, 23
- GRAPH\_SURFACE\_RECEIVERS\_OPEN
  - uictrl, 23
- GRAPH\_SURFACES\_CLOSE
  - uictrl, 24
- GRAPH\_SURFACES\_OPEN
  - uictrl, 24
- hasproperty
  - uictrl::element, 42
- hide
  - uictrl::element, 42
- idevent
  - uictrl, 24
- IDEVENT\_BFREQ\_PRESELECTION\_BAND
  - uictrl, 25
- IDEVENT\_BFREQ\_PRESELECTION\_-  
BUILDING\_BAND
  - uictrl, 25
- IDEVENT\_BFREQ\_PRESELECTION\_-  
BUILDING\_THIRD\_BAND
  - uictrl, 25
- IDEVENT\_BFREQ\_PRESELECTION\_NONE
  - uictrl, 25
- IDEVENT\_BFREQ\_PRESELECTION\_THIRD\_-  
BAND
  - uictrl, 25
- IDEVENT\_BUILD\_VOLUMES\_FROM\_-  
TRIMESH
  - uictrl, 25
- IDEVENT\_COLLER
  - uictrl, 24
- IDEVENT\_CONVERT\_VOL\_TO\_FITTING
  - uictrl, 25
- IDEVENT\_COPIER
  - uictrl, 24
- IDEVENT\_DELETE\_ELEMENT
  - uictrl, 24
- IDEVENT\_DELETE\_FOLDER
  - uictrl, 24

- uictrl, 24
- IDEVENT\_EMPTY\_POINTER\_VERTEX\_-  
GROUP
  - uictrl, 24
- IDEVENT\_GETPROPERTIES
  - uictrl, 24
- IDEVENT\_IMPORT\_MATERIAL
  - uictrl, 24
- IDEVENT\_INVERT\_FACE\_ORIENTATION
  - uictrl, 24
- IDEVENT\_LOAD\_PARTICLE\_SIMULATION
  - uictrl, 24
- IDEVENT\_LOAD\_PARTICLE\_SIMULATION\_-  
PATH
  - uictrl, 24
- IDEVENT\_LOAD\_RECEPTEURSP\_-  
SIMULATION
  - uictrl, 25
- IDEVENT\_LOAD\_RECEPTEURSS\_-  
SIMULATION\_BY\_TIMESTEP
  - uictrl, 24
- IDEVENT\_LOAD\_RECEPTEURSS\_-  
SIMULATION\_BY\_TIMESTEP\_SUM
  - uictrl, 24
- IDEVENT\_LOAD\_RECEPTEURSS\_-  
SIMULATION\_SUM
  - uictrl, 24
- IDEVENT\_NEW\_ENCOMBREMENT
  - uictrl, 24
- IDEVENT\_NEW\_ENCOMBREMENT\_-  
CUBOIDE
  - uictrl, 24
- IDEVENT\_NEW\_MATERIAL\_GROUP
  - uictrl, 24
- IDEVENT\_NEW\_RECEPTEUR\_P
  - uictrl, 24
- IDEVENT\_NEW\_RECEPTEUR\_S
  - uictrl, 24
- IDEVENT\_NEW\_RECEPTEUR\_S\_COUPE
  - uictrl, 25
- IDEVENT\_NEW\_RECEPTEURP\_GROUP
  - uictrl, 25
- IDEVENT\_NEW\_SOURCE
  - uictrl, 24
- IDEVENT\_NEW\_SOURCE\_GROUP
  - uictrl, 25
- IDEVENT\_NEW\_SURFACE\_GROUP
  - uictrl, 24
- IDEVENT\_NEW\_SURFACE\_GROUP\_FROM\_-  
SELECTION
  - uictrl, 25
- IDEVENT\_NEW\_USERFREQ
  - uictrl, 24
- IDEVENT\_NEW\_USERMAT
  - uictrl, 24
- IDEVENT\_NEW\_VOLUME
  - uictrl, 25
- IDEVENT\_OPEN\_FOLDER
  - uictrl, 25
- IDEVENT\_RECEPTEURS\_COMPUTE\_EDT
  - uictrl, 25
- IDEVENT\_RECEPTEURS\_COMPUTE\_TR
  - uictrl, 24
- IDEVENT\_RECP\_COMPUTE\_ACOUSTIC\_-  
PARAMETERS
  - uictrl, 24
- IDEVENT\_RECP\_COMPUTE\_ADVANCED\_-  
ACOUSTIC\_PARAMETERS
  - uictrl, 25
- IDEVENT\_RELOAD\_FOLDER
  - uictrl, 24
- IDEVENT\_RENAME\_ELEMENT
  - uictrl, 24
- IDEVENT\_REPORT\_PARTICULES\_MAKE\_-  
GABE
  - uictrl, 25
- IDEVENT\_RUN\_CALCULATION
  - uictrl, 24
- IDEVENT\_SELECT\_POSITION
  - uictrl, 24
- IDEVENT\_SELECT\_TREE\_ITEM
  - uictrl, 24
- importscene
  - uictrl::application, 29
- loadproject
  - uictrl::application, 29
- modified
  - uictrl::element, 42
- newproject
  - uictrl::application, 30
- register\_event
  - uictrl::application, 30
- register\_menu\_manager
  - uictrl::application, 30
- register\_update\_manager
  - uictrl::element, 42
- reloadgroupsfrommodel
  - uictrl::application, 30
- savelog
  - uictrl::application, 30
- saveproject
  - uictrl::application, 30
- saveshell
  - uictrl::application, 31

- sendevent
  - uictrl::application, 31
- setreadonlyallconfig
  - uictrl::element, 42
- setreadonlyconfig
  - uictrl::element, 42
- uictrl, 17
  - ELEMENT\_TYPE\_BOOL, 21
  - ELEMENT\_TYPE\_BOOL\_BFREQ, 22
  - ELEMENT\_TYPE\_COLOR, 21
  - ELEMENT\_TYPE\_CORE\_CORE, 23
  - ELEMENT\_TYPE\_CORE\_CORE\_-  
BFREQSELECTION, 22
  - ELEMENT\_TYPE\_CORE\_CORE\_CONFIG,  
22
  - ELEMENT\_TYPE\_CORE\_CORE\_-  
CONFMAILLAGE, 22
  - ELEMENT\_TYPE\_CORE\_ROOT, 20
  - ELEMENT\_TYPE\_CORE\_SPPS, 22
  - ELEMENT\_TYPE\_CORE\_SPPS\_OCTREE,  
22
  - ELEMENT\_TYPE\_CORE\_TC, 22
  - ELEMENT\_TYPE\_CORE\_TLM, 22
  - ELEMENT\_TYPE\_DRAWABLE, 22
  - ELEMENT\_TYPE\_ELEMENT, 23
  - ELEMENT\_TYPE\_FLOAT, 21
  - ELEMENT\_TYPE\_FONT, 22
  - ELEMENT\_TYPE\_GAMMEABSORPTION,  
22
  - ELEMENT\_TYPE\_GAMMEFREQ\_APP, 22
  - ELEMENT\_TYPE\_GAMMEFREQ\_USER,  
22
  - ELEMENT\_TYPE\_INTEGER, 21
  - ELEMENT\_TYPE\_LIST, 21
  - ELEMENT\_TYPE\_MATERIAU\_APP, 21
  - ELEMENT\_TYPE\_MATERIAU\_USER, 21
  - ELEMENT\_TYPE\_POSITION, 21
  - ELEMENT\_TYPE\_PROPERTY\_FREQ, 22
  - ELEMENT\_TYPE\_PYTHON\_EXTENSION,  
23
  - ELEMENT\_TYPE\_REPORT\_FILE, 23
  - ELEMENT\_TYPE\_REPORT\_FOLDER, 22
  - ELEMENT\_TYPE\_REPORT\_GABE, 22
  - ELEMENT\_TYPE\_REPORT\_GABE\_GAP,  
22
  - ELEMENT\_TYPE\_REPORT\_GABE\_RECIP,  
22
  - ELEMENT\_TYPE\_REPORT\_-  
PARTVISUALISATION, 22
  - ELEMENT\_TYPE\_REPORT\_-  
RECEPTEURSSVISUALISATION,  
22
  - ELEMENT\_TYPE\_REPORT\_-  
RECEPTEURSSVISUALISATION\_-  
EDT, 23
  - ELEMENT\_TYPE\_REPORT\_-  
RECEPTEURSSVISUALISATION\_-  
GAIN, 23
  - ELEMENT\_TYPE\_REPORT\_-  
RECEPTEURSSVISUALISATION\_-  
STANDART, 23
  - ELEMENT\_TYPE\_REPORT\_-  
RECEPTEURSSVISUALISATION\_TR,  
23
  - ELEMENT\_TYPE\_REPORT\_RPI, 22
  - ELEMENT\_TYPE\_REPORT\_UNKNOWN,  
22
  - ELEMENT\_TYPE\_RESULT\_ROOT, 20
  - ELEMENT\_TYPE\_ROW, 22
  - ELEMENT\_TYPE\_ROW\_BFREQ, 22
  - ELEMENT\_TYPE\_ROW\_MATERIAU, 22
  - ELEMENT\_TYPE\_SCENE\_BDD, 21
  - ELEMENT\_TYPE\_SCENE\_BDD\_-  
CATMATERIAL, 21
  - ELEMENT\_TYPE\_SCENE\_BDD\_-  
MATERIAUX, 21
  - ELEMENT\_TYPE\_SCENE\_BDD\_-  
MATERIAUX\_APP, 21
  - ELEMENT\_TYPE\_SCENE\_BDD\_-  
MATERIAUX\_APP\_GROUP, 21
  - ELEMENT\_TYPE\_SCENE\_BDD\_-  
MATERIAUX\_APP\_MATERIAU,  
21
  - ELEMENT\_TYPE\_SCENE\_BDD\_-  
MATERIAUX\_MATERIAU\_RENDER,  
22
  - ELEMENT\_TYPE\_SCENE\_BDD\_-  
MATERIAUX\_PROPMATERIAU,  
21
  - ELEMENT\_TYPE\_SCENE\_BDD\_-  
MATERIAUX\_USER, 21
  - ELEMENT\_TYPE\_SCENE\_BDD\_-  
MATERIAUX\_USER\_GROUP, 21
  - ELEMENT\_TYPE\_SCENE\_BDD\_-  
MATERIAUX\_USER\_MATERIAU,  
21
  - ELEMENT\_TYPE\_SCENE\_BDD\_-  
SPECTRUMS, 21
  - ELEMENT\_TYPE\_SCENE\_BDD\_-  
SPECTRUMS\_APP, 21
  - ELEMENT\_TYPE\_SCENE\_BDD\_-  
SPECTRUMS\_USER, 21
  - ELEMENT\_TYPE\_SCENE\_DONNEES, 22
  - ELEMENT\_TYPE\_SCENE\_-  
ENCOMBREMENTS, 22



- ELEMENT\_TYPE\_SCENE\_-
  - ENCOMBREMENTS\_-
  - ENCOMBREMENT, [22](#)
- ELEMENT\_TYPE\_SCENE\_-
  - ENCOMBREMENTS\_-
  - ENCOMBREMENT\_CUBOIDE, [22](#)
- ELEMENT\_TYPE\_SCENE\_-
  - ENCOMBREMENTS\_-
  - ENCOMBREMENT\_PROPRIETES, [22](#)
- ELEMENT\_TYPE\_SCENE\_-
  - ENCOMBREMENTS\_-
  - ENCOMBREMENT\_RENDU, [22](#)
- ELEMENT\_TYPE\_SCENE\_-
  - GROUPESURFACES, [20](#)
- ELEMENT\_TYPE\_SCENE\_-
  - GROUPESURFACES\_GROUPE, [21](#)
- ELEMENT\_TYPE\_SCENE\_-
  - GROUPESURFACES\_GROUPE\_-
  - VERTEX, [21](#)
- ELEMENT\_TYPE\_SCENE\_PROJET, [22](#)
- ELEMENT\_TYPE\_SCENE\_PROJET\_-
  - CONFIGURATION, [20](#)
- ELEMENT\_TYPE\_SCENE\_PROJET\_-
  - ENVIRONNEMENTCONF, [22](#)
- ELEMENT\_TYPE\_SCENE\_PROJET\_-
  - INFORMATION, [22](#)
- ELEMENT\_TYPE\_SCENE\_PROJET\_-
  - RENDU, [21](#)
- ELEMENT\_TYPE\_SCENE\_PROJET\_-
  - RENDU\_MODEL, [21](#)
- ELEMENT\_TYPE\_SCENE\_PROJET\_-
  - RENDU\_ORIGINE, [21](#)
- ELEMENT\_TYPE\_SCENE\_PROJET\_-
  - RENDU\_PARTICULES, [22](#)
- ELEMENT\_TYPE\_SCENE\_PROJET\_-
  - USERCONFIGURATION, [22](#)
- ELEMENT\_TYPE\_SCENE\_-
  - RECEPTEURSP, [21](#)
- ELEMENT\_TYPE\_SCENE\_-
  - RECEPTEURSP\_RECEPTEUR, [21](#)
- ELEMENT\_TYPE\_SCENE\_-
  - RECEPTEURSP\_RECEPTEUR\_-
  - PROPRIETES, [21](#)
- ELEMENT\_TYPE\_SCENE\_-
  - RECEPTEURSP\_RECEPTEUR\_-
  - RENDU, [21](#)
- ELEMENT\_TYPE\_SCENE\_-
  - RECEPTEURSS, [21](#)
- ELEMENT\_TYPE\_SCENE\_-
  - RECEPTEURSS\_RECEPTEUR, [21](#)
- ELEMENT\_TYPE\_SCENE\_-
  - RECEPTEURSS\_RECEPTEUR\_-
  - PROPRIETES, [21](#)
- ELEMENT\_TYPE\_SCENE\_-
  - RECEPTEURSS\_RECEPTEUR\_-
  - RENDU, [21](#)
- ELEMENT\_TYPE\_SCENE\_-
  - RECEPTEURSS\_-
  - RECEPTEURCOUPE\_PROPRIETES, [23](#)
- ELEMENT\_TYPE\_SCENE\_-
  - RECEPTEURSS\_-
  - RECEPTEURCOUPE\_RENDU, [23](#)
- ELEMENT\_TYPE\_SCENE\_ROOT, [20](#)
- ELEMENT\_TYPE\_SCENE\_SOURCES, [21](#)
- ELEMENT\_TYPE\_SCENE\_SOURCES\_-
  - SOURCE, [21](#)
- ELEMENT\_TYPE\_SCENE\_SOURCES\_-
  - SOURCE\_PROPRIETES, [21](#)
- ELEMENT\_TYPE\_SCENE\_SOURCES\_-
  - SOURCE\_PUISSANCE, [21](#)
- ELEMENT\_TYPE\_SCENE\_SOURCES\_-
  - SOURCE\_RENDU, [21](#)
- ELEMENT\_TYPE\_SCENE\_VOLUMES, [23](#)
- ELEMENT\_TYPE\_SCENE\_VOLUMES\_-
  - VOLUME, [23](#)
- ELEMENT\_TYPE\_SCENE\_VOLUMES\_-
  - VOLUME\_PROPRIETES, [23](#)
- ELEMENT\_TYPE\_SCENE\_VOLUMES\_-
  - VOLUME\_RENDU, [23](#)
- ELEMENT\_TYPE\_TEXT, [21](#)
- ELEMENT\_TYPE\_TREE\_LIST, [22](#)
- ELEMENT\_TYPE\_USER\_PREFERENCE\_-
  - ITEM, [23](#)
- ELEMENT\_TYPE\_USER\_PREFERENCE\_-
  - ITEM\_ISOTEMPLATE, [23](#)
- ELEMENT\_TYPE\_USER\_PREFERENCE\_-
  - NODE, [23](#)
- element\_type, [20](#)
- graph, [23](#)
- GRAPH\_FITTING\_CLOSE, [23](#)
- GRAPH\_FITTING\_OPEN, [23](#)
- GRAPH\_FITTINGS\_CLOSE, [23](#)
- GRAPH\_FITTINGS\_OPEN, [23](#)
- GRAPH\_FOLDER, [23](#)
- GRAPH\_FOLDER\_OPEN, [23](#)
- GRAPH\_ITEM, [23](#)
- GRAPH\_LAST\_STATIC\_GRAPH, [24](#)
- GRAPH\_PUNCTUAL\_RECEIVERS\_-
  - CLOSE, [23](#)
- GRAPH\_PUNCTUAL\_RECEIVERS\_OPEN, [23](#)
- GRAPH\_SOUND\_SOURCES\_CLOSE, [24](#)
- GRAPH\_SOUND\_SOURCES\_OPEN, [23](#)

- GRAPH\_SURFACE\_RECEIVERS\_CLOSE, 23
- GRAPH\_SURFACE\_RECEIVERS\_OPEN, 23
- GRAPH\_SURFACES\_CLOSE, 24
- GRAPH\_SURFACES\_OPEN, 24
- idevent, 24
- IDEVENT\_BFREQ\_PRESELECTION\_BAND, 25
- IDEVENT\_BFREQ\_PRESELECTION\_BUILDING\_BAND, 25
- IDEVENT\_BFREQ\_PRESELECTION\_BUILDING\_THIRD\_BAND, 25
- IDEVENT\_BFREQ\_PRESELECTION\_NONE, 25
- IDEVENT\_BFREQ\_PRESELECTION\_THIRD\_BAND, 25
- IDEVENT\_BUILD\_VOLUMES\_FROM\_TRIMESH, 25
- IDEVENT\_COLLER, 24
- IDEVENT\_CONVERT\_VOL\_TO\_FITTING, 25
- IDEVENT\_COPIER, 24
- IDEVENT\_DELETE\_ELEMENT, 24
- IDEVENT\_DELETE\_FOLDER, 24
- IDEVENT\_EMPTY\_POINTER\_VERTEX\_GROUP, 24
- IDEVENT\_GETPROPERTIES, 24
- IDEVENT\_IMPORT\_MATERIAL, 24
- IDEVENT\_INVERT\_FACE\_ORIENTATION, 24
- IDEVENT\_LOAD\_PARTICLE\_SIMULATION, 24
- IDEVENT\_LOAD\_PARTICLE\_SIMULATION\_PATH, 24
- IDEVENT\_LOAD\_RECEPTEURSP\_SIMULATION, 25
- IDEVENT\_LOAD\_RECEPTEURSS\_SIMULATION\_BY\_TIMESTEP, 24
- IDEVENT\_LOAD\_RECEPTEURSS\_SIMULATION\_BY\_TIMESTEP\_SUM, 24
- IDEVENT\_LOAD\_RECEPTEURSS\_SIMULATION\_SUM, 24
- IDEVENT\_NEW\_ENCOMBREMENT, 24
- IDEVENT\_NEW\_ENCOMBREMENT\_CUBOIDE, 24
- IDEVENT\_NEW\_MATERIAL\_GROUP, 24
- IDEVENT\_NEW\_RECEPTEUR\_P, 24
- IDEVENT\_NEW\_RECEPTEUR\_S, 24
- IDEVENT\_NEW\_RECEPTEUR\_S\_COUPE, 25
- IDEVENT\_NEW\_RECEPTEURP\_GROUP, 25
- IDEVENT\_NEW\_SOURCE, 24
- IDEVENT\_NEW\_SOURCE\_GROUP, 25
- IDEVENT\_NEW\_SURFACE\_GROUP, 24
- IDEVENT\_NEW\_SURFACE\_GROUP\_FROM\_SELECTION, 25
- IDEVENT\_NEW\_USERFREQ, 24
- IDEVENT\_NEW\_USERMAT, 24
- IDEVENT\_NEW\_VOLUME, 25
- IDEVENT\_OPEN\_FOLDER, 25
- IDEVENT\_RECEPTEURS\_COMPUTE\_EDT, 25
- IDEVENT\_RECEPTEURS\_COMPUTE\_TR, 24
- IDEVENT\_RECPCOMPUTE\_ACOUSTIC\_PARAMETERS, 24
- IDEVENT\_RECPCOMPUTE\_ADVANCED\_ACOUSTIC\_PARAMETERS, 25
- IDEVENT\_RELOAD\_FOLDER, 24
- IDEVENT\_RENAME\_ELEMENT, 24
- IDEVENT\_REPORT\_PARTICULES\_MAKE\_GABE, 25
- IDEVENT\_RUN\_CALCULATION, 24
- IDEVENT\_SELECT\_POSITION, 24
- IDEVENT\_SELECT\_TREE\_ITEM, 24
- uictrl::application, 27
- clearlogdata, 28
- clearshellhisto, 28
- getcachdir, 28
- getdataarray, 28
- getlastcalculationpath, 28
- getlocale, 28
- getrootcore, 28
- getrootpreference, 28
- getrootreport, 28
- getrootscene, 29
- getuserinput, 29
- importscene, 29
- loadproject, 29
- newproject, 30
- register\_event, 30
- register\_menu\_manager, 30
- reloadgroupsfrommodel, 30
- savelog, 30
- saveproject, 30
- saveshell, 31
- sendevent, 31
- uictrl::e\_file, 32
- buildfullpath, 32
- uictrl::Element, 33
- uictrl::element, 34
- appendfilsbytype, 35
- appendpropertybool, 35
- appendpropertycolor, 36

appendpropertydecimal, 36  
appendpropertyentier, 37  
appendpropertyfont, 37  
appendpropertylist, 37  
appendpropertyposition, 38  
appendpropertytext, 38  
appenduserelement, 38  
childs, 39  
deleteallementbytype, 39  
deleteallementbytyper, 39  
deleteelementbyxmlid, 39  
getallementbytype, 39  
getboolconfig, 39  
getcolorconfig, 40  
getdecimalconfig, 40  
getelementbylibelle, 40  
getelementbytype, 40  
getentierconfig, 40  
getid, 40  
getinfos, 41  
getlistconfig, 41  
getmenu, 41  
getpositionconfig, 41  
getstringconfig, 41  
hasproperty, 42  
hide, 42  
modified, 42  
register\_update\_manager, 42  
setreadonlyallconfig, 42  
setreadonlyconfig, 42  
updateboolconfig, 43  
updatedecimalconfig, 43  
updateentierconfig, 43  
updatelistconfig, 43  
updatepositionconfig, 44  
updatestringconfig, 44  
updateboolconfig  
  uictrl::element, 43  
updatedecimalconfig  
  uictrl::element, 43  
updateentierconfig  
  uictrl::element, 43  
updatelistconfig  
  uictrl::element, 43  
updatepositionconfig  
  uictrl::element, 44  
updatestringconfig  
  uictrl::element, 44