I-SIMPA Scripting Guide

1.1.4

Generated by Doxygen 1.6.1

Tue Jan 19 09:41:48 2010

Contents

Chapter 1

I-SIMPA scripting guide

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

Chapter 2

I-SIMPA adding functionnality

Adding popup menu functionnality 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 rigth 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 suffisant.

You must take control of theses fields to implement constraint and to adding more functionnality.

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 initia
      lisation
            #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 existance 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 overide the gettreelabel 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, whe 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.ELEME
  NT_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:

Chapter 5

Class Index

5.1 Class Hierarchy

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

Chapter 6

Class Index

6.1 Class List

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

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_RECP, 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_RECP_-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_RECP_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

Avaible 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 prefrence tree

ELEMENT_TYPE_USER_PREFERENCE_ITEM User preference item, in the user prefrence tree

ELEMENT_TYPE_USER_PREFERENCE_ITEM_ISOTEMPLATE User preference item, in the user prefrence 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

Avaible 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_RECP_COMPUTE_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 savelog (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 functionnality

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 functionnality

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:

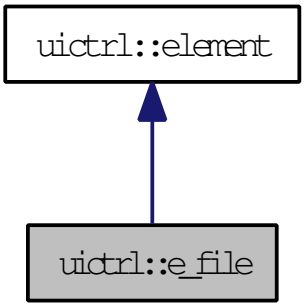
pyel Corresponding tree item node.

idevent Built-in uictrl::idevent or python defined event by application::register_event

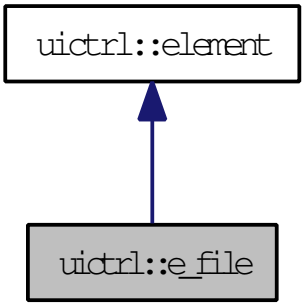
parameters See uictrl::idevent member for more specific details.

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, reprentative 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

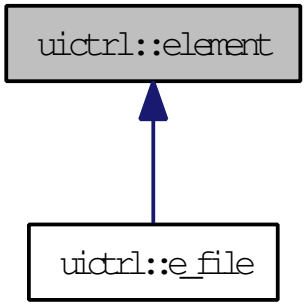
- typedef element_type **ELEMENT_TYPE**
- typedef graph **GRAPH**
- typedef idevent **IDEVENT**

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 recusively 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