PySI

Generated by Doxygen 1.8.15

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Namespace Documentation

5.1 SIEffect Namespace Reference

Documentation for this module / class.

Classes

- class SIEffect

  Super Class from which all subsequent plugins are derived.

5.1.1 Detailed Description

Documentation for this module / class.

Used as central entry point for all SIGRun plugins

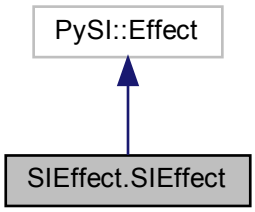
Chapter 6

Class Documentation

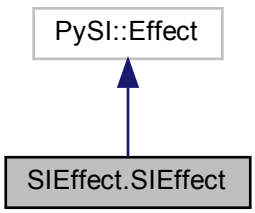
6.1 SIEffect.SIEffect Class Reference

Super Class from which all subsequent plugins are derived.

Inheritance diagram for SIEffect.SIEffect:



Collaboration diagram for SIEffect.SIEffect:



Public Member Functions

- def __init__ (self, shape, uuid, texture_path, regiontype, regionname, kwargs, __source__="custom")

  constructor
- def get_region_width (self)

  member function for retrieving the maximum width of a region
- def get_region_height (self)

  member function for retrieving the maximum height of a region
- def set_position_from_position (self, rel_x, rel_y, abs_x, abs_y)

  member function for setting the position of a region based on the positional data of another region.
- def relative_x_pos (self)

  member function for getting the relative x coordinate of the parent region's top left corner
- def relative_y_pos (self)

  member function for getting the relative y coordinate of the parent region's top left corner
- def absolute_x_pos (self)

  member function for getting the absolute x coordinate of the parent region's top left corner
- def absolute_y_pos (self)

  member function for getting the absolute y coordinate of the parent region's top left corner
- def on_move_enter_recv (self, cursor_id, link_attrib)

  member function for receiving data from the PySI.MOVE capability for the PySI.ON_ENTER collision event
- def on_move_continuous_recv (self)

  member function for the PySI.MOVE capability for the PySI.ON_CONTINUOUS collision event
- def on_move_leave_recv (self, cursor_id, link_attrib)

  member function for receiving data from the PySI.MOVE capability for the PySI.ON_LEAVE collision event
- def enable_effect (self, capability, is_emit, on_enter, on_continuous, on_leave)

  member function for enabling the emission or reception of an effect
- def is_effect_enabled (self, capability, is_emit)
- def override_effect (self, capability, is_emit, on_enter, on_continuous, on_leave)

  member function for overriding the emission or reception of an effect
- def disable_effect (self, capability, is_emit)

  member function for disabling the emission or reception of an effect
- def enable_link_emission (self, emission_capability, emission_function)

  member function for enabling the emission of data in the context of a link event
- def enable_link_reception (self, emission_capability, reception_capability, reception_function)

  member function for enabling the emission of data in the context of a link event
- def disable_link_emission (self, emission_capability)

  member function for disabling the emission of data in the context of a link event
- def disable_link_reception (self, emission_capability, reception_capability="")

  member function for disabling the reception of data in the context of a link event
- def create_link (self, sender_uuid, sender_attribute, receiver_uuid, receiver_attribute)

  member function for establishing a specified link between two regions according to given attributes
- def remove_link (self, sender_uuid, sender_attribute, receiver_uuid, receiver_attribute)

  member function for removing a specified link between two regions according to given attributes
- def set_QML_data (self, key, value, datatype, data_kwargs={})

  member function for setting data in the associated qml file of a region effect
- def set_QML_path (self, filename)

  member function for setting the path to an plugin's associated qml file
- def add_point_to_region_drawing (self, x, y, cursor_id)

  member function for adding a point to a region drawing based on a cursor id.
- def register_region_from_drawing (self, cursor_id)

  member function for registering a region drawing according to a cursor id

- def start_standard_application (self, file_uuid, file_path)

  member function for starting the standard application of a file given its uuid as a region and its path in the filesystem
- def close_standard_application (self, file_uuid)

  member function for closing the standard application of a file given its uuid as a region and its path in the filesystem
- def display_folder_contents_page (self, page, source_uuid, with_buttons=True)

  member function for displaying the contents of a folder in the filesystem as pages of other filesystem entries
- def delete (self)

  member function for deleting a region
- def create_region_via_name (self, shape, effect_name, as_selector=False, kwargs={})

  member function for creating a new region
- def create_region_via_id (self, shape, effect_type, kwargs={})

  member function for creating a new region
- def available_plugins (self)

  member function for retrieving the plugins which are available for sketching as a dict of names.
- def snap_to_mouse (self)

  member function for snapping a region's center to the mouse cursor
- def context_dimensions (self)

  member function for retrieving the dimensions of the active SI-Context (width in px, and height in px)
- def assign_effect (self, effect_name_to_assign, effect_display_name, kwargs)

  member function for assigning a new effect to a region if the region is a cursor, the effect that cursor can draw is changed instead!
- def move (self, x, y)

  member function for moving the effect's associated region to the point (x, y)

Public Attributes

- shape

  member attribute variable containing the shape (contour) of a drawn region as a PySI.PointVector
- width

  member attribute variable containing the axis-aligned bounding-box (aabb) of a drawn region as a PySI.PointVector
- height

  member variable containing the maximum height of the region
- name

  member attribute variable containing the name of a drawn region as a str
- region_type

  member attribute variable containing the type of effect of a drawn region as a PySI.EffectType
- source

  member attribute variable containing the source of effect of a drawn region as a str
- qml_path

  member attribute variable containing the path to a QML file for styling of a drawn region as a str
- delta_x

  member attribute variable containing the last relative movement of the region according to the x axis as a float
- delta_y

  member attribute variable containing the last relative movement of the region according to the y axis as a float
- last_x

  member attribute variable containing the last absolute x coordinate as a float
- last_y

  member attribute variable containing the last absolute y coordinate as a float
- color

  member attribute variable containing the fill color of a region in RGBA as a PySI.Color

- is_under_user_control

  member attribute variable which is true when an user directly controls the region (e.g.
- texture_path

  member attribute variable storing the path to the image file used as texture for a region
- texture_width

  member attribute variable storing the width of a texture of a region drawing as a float
- texture_height

  member attribute variable storing the height of a texture of a region drawing as a float
- cap_emit

  member attribute variable storing keys to functions which are called when collision events occur for emitting data to receiving regions
- cap_recv

  member attribute variable storing keys to functions which are called when collision events occur for receiving data from emitting regions
- cap_link_emit

  member attribute variable storing keys to functions which are called when linking events occur for emitting data to receiving regions
- cap_link_recv

  member attribute variable storing keys to functions which are called when linking events occur for emitting data to receiving regions
- mouse_x

  member attribute variable storing the x position of the mouse cursor
- mouse_y

  member attribute variable storing the y position of the mouse cursor
- x
- y

Static Public Attributes

- bool EMISSION = True

  member constant to mark an effect or link emittable
- bool RECEPTION = False

  static member attribute to mark an effect or link receivable
- string TEXTURE_PATH_NONE = ""

  static member attribute to signal that it's associated effect does not display an icon (texture) when drawn as a region
- bool RESAMPLING = True

  static member attribute to notify SIGRun to resample a region's shape when changed from PySI
- bool NO_RESAMPLING = False

  static member attribute to notify SIGRun to not resample a region's shape when changed from PySI Use with caution! May lead to unexpected / barely debugable behaviour!

6.1.1 Detailed Description

Super Class from which all subsequent plugins are derived.

This Class itself is derived from PySI written in C++ which is documented separately within SIGRun

Definition at line 15 of file SIEffect.py.

6.1.2 Constructor & Destructor Documentation

6.1.2.1 __init__()

```
def SIEffect.SIEffect.__init__ (
              self,
              shape,
              uuid,
              texture_path,
              regiontype,
              regionname,
              kwargs,
              __source__ = "custom" )
```

constructor

Constructs a new SIEffect object based on the given arguments.

Parameters

self	the object pointer
shape	the contour of the drawn region (PySI.PointVector)
aabb	the axis-aligned bounding-box of the drawn region (PySI.PointVector)
uuid	the universally unique identifier of the drawn region (str)
texture_path	the path to an image intended to be used as an icon for the drawn region (str)
kwargs	keyworded arguments which may necessary for more specific implementations of region effects (dict)
source	the source of the plugin e.g. standard environment library (str)

Definition at line 45 of file SIEffect.py.

6.1.3 Member Function Documentation

6.1.3.1 absolute_x_pos()

```
def SIEffect.SIEffect.absolute_x_pos (
              self )
```

member function for getting the absolute x coordinate of the parent region's top left corner

Parameters

self	the object pointer

Definition at line 242 of file SIEffect.py.

6.1.3.2 absolute_y_pos()

```
def SIEffect.SIEffect.absolute_y_pos (
              self )
```

member function for getting the absolute y coordinate of the parent region's top left corner

Parameters

self	the object pointer

Definition at line 248 of file SIEffect.py.

6.1.3.3 add_point_to_region_drawing()

```
def SIEffect.SIEffect.add_point_to_region_drawing (
              self,
              x,
              y,
              cursor_id )
```

member function for adding a point to a region drawing based on a cursor id.

Parameters

self	the object pointer
x	the x coordinate of the cursor (float)
y	the y coordinate of the cursor (float)
cursor_id	the id of cursor currently drawing (str)

This function is specific to effects of PySI.EffectType.SI_CANVAS. Therefore, this function does nothing when called with other effect types.

This function uses self.__partial_regions__ (c++-bindings)

Definition at line 429 of file SIEffect.py.

6.1.3.4 assign_effect()

```
def SIEffect.SIEffect.assign_effect (
              self,
```

```
                effect_name_to_assign,
                effect_display_name,
                kwargs )
```

member function for assigning a new effect to a region if the region is a cursor, the effect that cursor can draw is changed instead!

Parameters

self	the object pointer
effect_name_to_assign	the name of the effect which is intended to be written to a region
effect_display_name	the name of the effect which is intended to be visible to a user
kwargs	key-worded arguments containing specifics of certain regions

Definition at line 527 of file SIEffect.py.

6.1.3.5 available_plugins()

```
def SIEffect.SIEffect.available_plugins (
                self )
```

member function for retrieving the plugins which are available for sketching as a dict of names.

This dict of names uses region_name attributes as keys and region_display_name attributes as values

Parameters

self	the object pointer

Definition at line 504 of file SIEffect.py.

6.1.3.6 close_standard_application()

```
def SIEffect.SIEffect.close_standard_application (
                self,
                file_uuid )
```

member function for closing the standard application of a file given its uuid as a region and its path in the filesystem

Parameters

self	the object pointer
file_uuid	the uuid of the region associated to a file icon representing a file of the filesystem (str)

This function calls self.__destroy_embedded_window__ (c++-bindings)

Definition at line 465 of file SIEffect.py.

6.1.3.7 context_dimensions()

```
def SIEffect.SIEffect.context_dimensions (
              self )
```

member function for retrieving the dimensions of the active SI-Context (width in px, and height in px)

Parameters

self	the object pointer

Definition at line 517 of file SIEffect.py.

6.1.3.8 create_link()

```
def SIEffect.SIEffect.create_link (
              self,
              sender_uuid,
              sender_attribute,
              receiver_uuid,
              receiver_attribute )
```

member function for establishing a specified link between two regions according to given attributes

Parameters

self	the object pointer
sender_uuid	the uuid of the emitting region (str)
sender_attribute	the attribute to be linked by the emitting region (str)
receiver_uuid	the uuid of the receiving region (str)
receiver_attribute	the attribute to be linked by the receiving region (str)

Definition at line 380 of file SIEffect.py.

6.1.3.9 create_region_via_id()

```
def SIEffect.SIEffect.create_region_via_id (
              self,
              shape,
              effect_type,
              kwargs = {} )
```

member function for creating a new region

Parameters

self	the object pointer
shape	the shape / contour of the region as a PySI.PointVector or list [[x1, x1], [x2, y2], ... [xn, yn]]
effect_name	the name (region_name) of the effect which shall be assigned to the region (region_display_name does not work)

Definition at line 497 of file SIEffect.py.

6.1.3.10 create_region_via_name()

```
def SIEffect.SIEffect.create_region_via_name (
              self,
              shape,
              effect_name,
              as_selector = False,
              kwargs = {} )
```

member function for creating a new region

Parameters

self	the object pointer
shape	the shape / contour of the region as a PySI.PointVector or list [[x1, x1], [x2, y2], ... [xn, yn]]
effect_name	the name (region_name) of the effect which shall be assigned to the region (region_display_name does not work)

Definition at line 489 of file SIEffect.py.

6.1.3.11 delete()

```
def SIEffect.SIEffect.delete (
              self )
```

member function for deleting a region

Parameters

self	the object pointer

Definition at line 481 of file SIEffect.py.

6.1.3.12 disable_effect()

```
def SIEffect.SIEffect.disable_effect (
              self,
              capability,
              is_emit )
```

member function for disabling the emission or reception of an effect

Parameters

self	the object pointer
capability	the capability of the collision event (str)
is_emit	the variable depicting if a region emits (True) or receives (False) an effect (bool)

Definition at line 323 of file SIEffect.py.

6.1.3.13 disable_link_emission()

```
def SIEffect.SIEffect.disable_link_emission (
              self,
              emission_capability )
```

member function for disabling the emission of data in the context of a link event

Parameters

self	the object pointer
emission_capability	the capability of the linking event used by the emitting region (str)

Definition at line 352 of file SIEffect.py.

6.1.3.14 disable_link_reception()

```
def SIEffect.SIEffect.disable_link_reception (
              self,
              emission_capability,
              reception_capability = "" )
```

member function for disabling the reception of data in the context of a link event

Parameters

self	the object pointer
emission_capability	the capability of the linking event used by the emitting region (str)
reception_capability	the capability of the linking event of a receiving region with default value "" (str)

If no reception_capability is specified, the emission_capability is deleted from self.cap_link_recv. If reception_↩ capability is specified and present in self.cap_link_recv, the specified relation is deleted from emission_capability.

See also

self.cap_link_recv

Definition at line 364 of file SIEffect.py.

6.1.3.15 display_folder_contents_page()

```
def SIEffect.SIEffect.display_folder_contents_page (
              self,
              page,
              source_uuid,
              with_buttons = True )
```

member function for displaying the contents of a folder in the filesystem as pages of other filesystem entries

Parameters

self	the object pointer
source_uuid	the uuid of the region associated to a folder icon representing a folder of the filesystem (str)
with_buttons	a flag depicting whether buttons for browsing pages is wanted (True) or not (False) (bool)

This function calls self.__show_folder_contents_page__ (c++-bindings)

Definition at line 475 of file SIEffect.py.

6.1.3.16 enable_effect()

```
def SIEffect.SIEffect.enable_effect (
              self,
              capability,
              is_emit,
              on_enter,
              on_continuous,
              on_leave )
```

member function for enabling the emission or reception of an effect

Parameters

self	the object pointer
capability	the capability of the collision event (str)
is_emit	the variable depicting if a region emits (True) or receives (False) an effect (bool)
on_enter	the function to be called for the collision event PySI.ON_ENTER
on_continuous	the function to be called for the collision event PySI.ON_CONTINUOUS
on_leave	the function to be called for the collision event PySI.ON_LEAVE

Definition at line 292 of file SIEffect.py.

6.1.3.17 enable_link_emission()

```
def SIEffect.SIEffect.enable_link_emission (
              self,
              emission_capability,
              emission_function )
```

member function for enabling the emission of data in the context of a link event

Parameters

self	the object pointer
emission_capability	the capability of the linking event (str)
emission_function	the function to be called for emitting data

Definition at line 336 of file SIEffect.py.

6.1.3.18 enable_link_reception()

```
def SIEffect.SIEffect.enable_link_reception (
              self,
              emission_capability,
              reception_capability,
              reception_function )
```

member function for enabling the emission of data in the context of a link event

Parameters

self	the object pointer
emission_capability	the capability of the linking event used by the emitting region (str)
reception_capability	the capability of the linking event of a receiving region (str)
reception_function	the function to be called for receiving data

Definition at line 345 of file SIEffect.py.

6.1.3.19 get_region_height()

```
def SIEffect.SIEffect.get_region_height (
              self )
```

member function for retrieving the maximum height of a region

Definition at line 203 of file SIEffect.py.

6.1.3.20 get_region_width()

```
def SIEffect.SIEffect.get_region_width (
              self )
```

member function for retrieving the maximum width of a region

Definition at line 199 of file SIEffect.py.

6.1.3.21 is_effect_enabled()

```
def SIEffect.SIEffect.is_effect_enabled (
              self,
              capability,
              is_emit )
```

Definition at line 298 of file SIEffect.py.

6.1.3.22 move()

```
def SIEffect.SIEffect.move (
              self,
              x,
              y )
```

member function for moving the effect's associated region to the point (x, y)

Parameters

self	the object pointer
x	the absolute x coordinate of the point
y	the absolute y coordinate of the point

Definition at line 535 of file SIEffect.py.

6.1.3.23 on_move_continuous_recv()

```
def SIEffect.SIEffect.on_move_continuous_recv (
              self )
```

member function for the PySI.MOVE capability for the PySI.ON_CONTINUOUS collision event

Parameters

self	the object pointer

Definition at line 264 of file SIEffect.py.

6.1.3.24 on_move_enter_recv()

```
def SIEffect.SIEffect.on_move_enter_recv (
              self,
              cursor_id,
              link_attrib )
```

member function for receiving data from the PySI.MOVE capability for the PySI.ON_ENTER collision event

Parameters

self	the object pointer
cursor_id	the cursor which is intended to move the region (str)
link_attribute	the linking attribute defining how the cursor and the region are intended to be linked (str)

Definition at line 256 of file SIEffect.py.

6.1.3.25 on_move_leave_recv()

```
def SIEffect.SIEffect.on_move_leave_recv (
              self,
              cursor_id,
              link_attrib )
```

member function for receiving data from the PySI.MOVE capability for the PySI.ON_LEAVE collision event

Parameters

self	the object pointer
cursor_id	the cursor which is intended to move the region (str)
link_attribute	the linking attribute defining how the cursor and the region are intended to be linked (str)

Definition at line 272 of file SIEffect.py.

6.1.3.26 override_effect()

```
def SIEffect.SIEffect.override_effect (
              self,
              capability,
              is_emit,
              on_enter,
              on_continuous,
              on_leave )
```

member function for overriding the emission or reception of an effect

Parameters

self	the object pointer
capability	the capability of the collision event (str)
is_emit	the variable depicting if a region emits (True) or receives (False) an effect (bool)
on_enter	the function to be called for the collision event PySI.ON_ENTER
on_continuous	the function to be called for the collision event PySI.ON_CONTINUOUS
on_leave	the function to be called for the collision event PySI.ON_LEAVE

This function then calls self.enable_effect(capability, is_emit, on_enter, on_continuous, on_leave)

See also

self.enable_effect(capability, is_emit, on_enter, on_continuous, on_leave)

Definition at line 315 of file SIEffect.py.

6.1.3.27 register_region_from_drawing()

```
def SIEffect.SIEffect.register_region_from_drawing (
              self,
              cursor_id )
```

member function for registering a region drawing according to a cursor id

Parameters

self	the object pointer
cursor_id	the id of the cursor which is currently drawing (str)

This function is specific to effects of PySI.EffectType.SI_CANVAS. Therefore, this function does nothing when called with other effect types.

This function uses self.__registered_regions__ (c++-bindings)

Definition at line 445 of file SIEffect.py.

6.1.3.28 relative_x_pos()

```
def SIEffect.SIEffect.relative_x_pos (
              self )
```

member function for getting the relative x coordinate of the parent region's top left corner

Parameters

self	the object pointer

Definition at line 230 of file SIEffect.py.

6.1.3.29 relative_y_pos()

```
def SIEffect.SIEffect.relative_y_pos (
              self )
```

member function for getting the relative y coordinate of the parent region's top left corner

Parameters

self	the object pointer

Definition at line 236 of file SIEffect.py.

6.1.3.30 remove_link()

```
def SIEffect.SIEffect.remove_link (
              self,
              sender_uuid,
              sender_attribute,
              receiver_uuid,
              receiver_attribute )
```

member function for removing a specified link between two regions according to given attributes

Parameters

self	the object pointer
sender_uuid	the uuid of the emitting region (str)
sender_attribute	the attribute to be linked by the emitting region (str)
receiver_uuid	the uuid of the receiving region (str)
receiver_attribute	the attribute to be linked by the receiving region (str)

Definition at line 391 of file SIEffect.py.

6.1.3.31 set_position_from_position()

```
def SIEffect.SIEffect.set_position_from_position (
              self,
              rel_x,
              rel_y,
              abs_x,
              abs_y )
```

member function for setting the position of a region based on the positional data of another region.

This function is used as a reception function for linking events where positional data is emitted by another region and applied to the position of a region.

Parameters

self	the object pointer
rel_x	the relative positional change on the x axis (float)
rel_y	the relative positional change on the y axis (float)
abs_x	the absolute position on the x axis (float)
abs_y	the absolute position on the y axis (float)

Definition at line 215 of file SIEffect.py.

6.1.3.32 set_QML_data()

```
def SIEffect.SIEffect.set_QML_data (
              self,
              key,
              value,
              datatype,
              data_kwargs = {} )
```

member function for setting data in the associated qml file of a region effect

Parameters

self	the object pointer
key	the variable specified in the qml file (str)
value	the value to set in the variable in the qml file (variant)
datatype	the data type of the value (PySI.INT, PySI.FLOAT, ...) (int)

Calls the function **set_data** (c++-bindings)

Definition at line 406 of file SIEffect.py.

6.1.3.33 set_QML_path()

```
def SIEffect.SIEffect.set_QML_path (
              self,
              filename )
```

member function for setting the path to an plugin's associated qml file

Parameters

self	the object pointer
filename	the file name of the target qml file

Returns

the absolute path to the qml file

Definition at line 415 of file SIEffect.py.

6.1.3.34 snap_to_mouse()

```
def SIEffect.SIEffect.snap_to_mouse (
              self )
```

member function for snapping a region's center to the mouse cursor

Parameters

self	the object pointer

Definition at line 510 of file SIEffect.py.

6.1.3.35 start_standard_application()

```
def SIEffect.SIEffect.start_standard_application (
              self,
              file_uuid,
              file_path )
```

member function for starting the standard application of a file given its uuid as a region and its path in the filesystem

Parameters

self	the object pointer
file_uuid	the uuid of the region associated to a file icon representing a file of the filesystem (str)
file_path	the path of the file in the filesystem (str)

This function calls self.__embed_file_standard_appliation_into_context__ (c++-bindings)

Definition at line 456 of file SIEffect.py.

6.1.4 Member Data Documentation

6.1.4.1 cap_emit

`SIEffect.SIEffect.cap_emit`

member attribute variable storing keys to functions which are called when collision events occur for emitting data to receiving regions

This variable is a PySI.String2_String2FunctionMap_Map (c++-bindings) and uses capabilities (str) as keys to the inner String2FunctionMap. The inner String2FunctionMap uses collision event names (PySI.ON_ENTER ("on_enter"), PySI:ON_CONTINUOUS ("on_continuous"), PySI.ON_LEAVE ("on_leave")) as keys to their corresponding functions as values

Example:

self.cap_emit["CAPABILITY"] = {PySI.ON_ENTER: self.<function_enter>, PySI:ON_CONTINUOUS: self.<function_continuous>, PySI.ON_LEAVE: self.<function_leave>

Therefore, this example allows a region to emit an effect of CAPABILITY once a collision event occurred

Definition at line 142 of file SIEffect.py.

6.1.4.2 cap_link_emit

`SIEffect.SIEffect.cap_link_emit`

member attribute variable storing keys to functions which are called when linking events occur for emitting data to receiving regions

This variable is a String2FunctionMap (c++-bindings) containing capabilities (str) as keys and functions as values

Example with SI-integrated linking of positions for emission case: self.cap_link_emit[PySI.POSITION] = self.<function_position_emit> Therefore, this example emits the positional data of the region to a linked region.

Definition at line 167 of file SIEffect.py.

6.1.4.3 cap_link_recv

SIEffect.SIEffect.cap_link_recv

member attribute variable storing keys to functions which are called when linking events occur for emitting data to receiving regions

This variable is a PySI.String2_String2FunctionMap_Map (c++-bindings) and uses linking event capability names (str) as keys to the inner String2FunctionMap. The inner String2FunctionMap uses linking event capability names (PySI.POSITION, <own name="" as="" str>="">) as keys to their corresponding functions as values. The outer key corresponds to the emission capability. The inner key corresponds to the reception capability of the targeted region and points towards the function which is to be called during the linking event Therefore, it is possible to map e.g. incomimg positional data to the color of the receiving region.

Example with SI-integrated linking of positions for reception case: self.cap_link_recv[PySI.POSITION][PySI.POSI↩ TION] = self.<function_position_emit> self.cap_link_recv[PySI.POSITION][PySI.COLOR] = self.<function_color↩ _emit> Therefore, this example receives the positional data of a linked region and can apply this data to other categories of data according to the linking relationship.

Definition at line 185 of file SIEffect.py.

6.1.4.4 cap_recv

SIEffect.SIEffect.cap_recv

member attribute variable storing keys to functions which are called when collision events occur for receiving data from emitting regions

This variable is a PySI.String2_String2FunctionMap_Map (c++-bindings) and uses capabilities (str) as keys to the inner String2FunctionMap. The inner String2FunctionMap uses collision event names (PySI.ON_ENTER ("on_↩ enter"), PySI:ON_CONTINUOUS ("on_continuous"), PySI.ON_LEAVE ("on_leave")) as keys to their corresponding functions as values

Example:

self.cap_recv["CAPABILITY"] = {PySI.ON_ENTER: self.<function_enter>, PySI:ON_CONTINUOUS: self.<function↩ _continuous>, PySI.ON_LEAVE: self.<function_leave>

Therefore, this example allows a region to receive an effect of CAPABILITY once a collision event occurred

Definition at line 154 of file SIEffect.py.

6.1.4.5 color

SIEffect.SIEffect.color

member attribute variable containing the fill color of a region in RGBA as a PySI.Color

Definition at line 103 of file SIEffect.py.

6.1.4.6 delta_x

`SIEffect.SIEffect.delta_x`

member attribute variable containing the last relative movement of the region according to the x axis as a float

Definition at line 91 of file SIEffect.py.

6.1.4.7 delta_y

`SIEffect.SIEffect.delta_y`

member attribute variable containing the last relative movement of the region according to the y axis as a float

Definition at line 94 of file SIEffect.py.

6.1.4.8 EMISSION

`bool SIEffect.SIEffect.EMISSION = True  [static]`

member constant to mark an effect or link emittable

Definition at line 18 of file SIEffect.py.

6.1.4.9 height

`SIEffect.SIEffect.height`

member variable containing the maximum height of the region

computed via aabb

Definition at line 65 of file SIEffect.py.

6.1.4.10 is_under_user_control

`SIEffect.SIEffect.is_under_user_control`

member attribute variable which is true when an user directly controls the region (e.g.

moving it around) as a bool

Definition at line 106 of file SIEffect.py.

6.1.4.11 last_x

`SIEffect.SIEffect.last_x`

member attribute variable containing the last absolute x coordinate as a float

Definition at line 97 of file SIEffect.py.

6.1.4.12 last_y

`SIEffect.SIEffect.last_y`

member attribute variable containing the last absolute y coordinate as a float

Definition at line 100 of file SIEffect.py.

6.1.4.13 mouse_x

`SIEffect.SIEffect.mouse_x`

member attribute variable storing the x position of the mouse cursor

Definition at line 188 of file SIEffect.py.

6.1.4.14 mouse_y

`SIEffect.SIEffect.mouse_y`

member attribute variable storing the y position of the mouse cursor

Definition at line 191 of file SIEffect.py.

6.1.4.15 name

`SIEffect.SIEffect.name`

member attribute variable containing the name of a drawn region as a str

Definition at line 71 of file SIEffect.py.

6.1.4.16 NO_RESAMPLING

```
bool SIEffect.SIEffect.NO_RESAMPLING = False  [static]
```

static member attribute to notify SIGRun to not resample a region's shape when changed from PySI Use with caution! May lead to unexpected / barely debugable behaviour!

Definition at line 32 of file SIEffect.py.

6.1.4.17 qml_path

```
SIEffect.SIEffect.qml_path
```

member attribute variable containing the path to a QML file for styling of a drawn region as a str

This value can be left empty if no visualization of the region is intended (e.g. Container-Regions for External Applications or MouseCursor)

See also

Container
MouseCursor

Definition at line 88 of file SIEffect.py.

6.1.4.18 RECEPTION

```
bool SIEffect.SIEffect.RECEPTION = False  [static]
```

static member attribute to mark an effect or link receivable

Definition at line 21 of file SIEffect.py.

6.1.4.19 region_type

```
SIEffect.SIEffect.region_type
```

member attribute variable containing the type of effect of a drawn region as a PySI.EffectType

Effect implementation which are currently not part of the Standard Environment Library of SIGRun are required to be of type SI_CUSTOM

Definition at line 76 of file SIEffect.py.

6.1.4.20 RESAMPLING

```
bool SIEffect.SIEffect.RESAMPLING = True  [static]
```

static member attribute to notify SIGRun to resample a region's shape when changed from PySI

Definition at line 27 of file SIEffect.py.

6.1.4.21 shape

```
SIEffect.SIEffect.shape
```

member attribute variable containing the shape (contour) of a drawn region as a PySI.PointVector

Definition at line 49 of file SIEffect.py.

6.1.4.22 source

```
SIEffect.SIEffect.source
```

member attribute variable containing the source of effect of a drawn region as a str

Effect implementation which are currently not part of the Standard Environment Library of SIGRun are encouraged to not start with "libStdSI"

Definition at line 81 of file SIEffect.py.

6.1.4.23 texture_height

```
SIEffect.SIEffect.texture_height
```

member attribute variable storing the height of a texture of a region drawing as a float

This value is only set if texture_path is a valid path

Definition at line 123 of file SIEffect.py.

6.1.4.24 texture_path

```
SIEffect.SIEffect.texture_path
```

member attribute variable storing the path to the image file used as texture for a region

Definition at line 112 of file SIEffect.py.

6.1.4.25 TEXTURE_PATH_NONE

```
string SIEffect.SIEffect.TEXTURE_PATH_NONE = ""  [static]
```

static member attribute to signal that it's associated effect does not display an icon (texture) when drawn as a region

Definition at line 24 of file SIEffect.py.

6.1.4.26 texture_width

```
SIEffect.SIEffect.texture_width
```

member attribute variable storing the width of a texture of a region drawing as a float

This value is only set if texture_path is a valid path

Definition at line 118 of file SIEffect.py.

6.1.4.27 width

```
SIEffect.SIEffect.width
```

member attribute variable containing the axis-aligned bounding-box (aabb) of a drawn region as a PySI.PointVector

This variable is automatically computed when shape is changed. It is recommended to use this variable read-only.
member variable containing the maximum width of the region

computed via aabb

Definition at line 60 of file SIEffect.py.

6.1.4.28 x

```
SIEffect.SIEffect.x
```

Definition at line 511 of file SIEffect.py.

6.1.4.29 y

```
SIEffect.SIEffect.y
```

Definition at line 512 of file SIEffect.py.

The documentation for this class was generated from the following file:

- SIEffect.py

Chapter 7

File Documentation

7.1 SIEffect.py File Reference

Classes

- class SIEffect.SIEffect

  Super Class from which all subsequent plugins are derived.

Namespaces

- SIEffect

  Documentation for this module / class.

Index