

PySI

Generated by Doxygen 1.8.15



<b>1 Namespace Index</b>	<b>1</b>
1.1 Packages	1
<b>2 Hierarchical Index</b>	<b>3</b>
2.1 Class Hierarchy	3
<b>3 Class Index</b>	<b>5</b>
3.1 Class List	5
<b>4 File Index</b>	<b>7</b>
4.1 File List	7
<b>5 Namespace Documentation</b>	<b>9</b>
5.1 SIEffect Namespace Reference	9
5.1.1 Detailed Description	9
<b>6 Class Documentation</b>	<b>11</b>
6.1 SIEffect.SIEffect Class Reference	11
6.1.1 Detailed Description	14
6.1.2 Constructor & Destructor Documentation	14
6.1.2.1 __init__()	15
6.1.3 Member Function Documentation	15
6.1.3.1 absolute_x_pos()	15
6.1.3.2 absolute_y_pos()	16
6.1.3.3 add_point_to_region_drawing()	16
6.1.3.4 add_QML_data()	16
6.1.3.5 assign_effect()	17
6.1.3.6 available_plugins()	17
6.1.3.7 close_standard_application()	18
6.1.3.8 context_dimensions()	18
6.1.3.9 create_link()	18
6.1.3.10 create_region_via_id()	19
6.1.3.11 create_region_via_name()	19
6.1.3.12 delete()	20
6.1.3.13 disable_effect()	20
6.1.3.14 disable_link_emission()	20
6.1.3.15 disable_link_reception()	21
6.1.3.16 display_folder_contents_page()	21
6.1.3.17 enable_effect()	22
6.1.3.18 enable_link_emission()	22
6.1.3.19 enable_link_reception()	23
6.1.3.20 get_region_height()	23
6.1.3.21 get_region_width()	23
6.1.3.22 is_effect_enabled()	23

6.1.3.23 move()	24
6.1.3.24 on_move_continuous_recv()	24
6.1.3.25 on_move_enter_recv()	24
6.1.3.26 on_move_leave_recv()	25
6.1.3.27 override_effect()	25
6.1.3.28 register_region_from_drawing()	26
6.1.3.29 relative_x_pos()	26
6.1.3.30 relative_y_pos()	26
6.1.3.31 remove_link()	27
6.1.3.32 set_position_from_position()	27
6.1.3.33 snap_to_mouse()	28
6.1.3.34 start_standard_application()	28
6.1.4 Member Data Documentation	28
6.1.4.1 cap_emit	28
6.1.4.2 cap_link_emit	29
6.1.4.3 cap_link_recv	29
6.1.4.4 cap_recv	30
6.1.4.5 color	30
6.1.4.6 delta_x	30
6.1.4.7 delta_y	30
6.1.4.8 EMISSION	31
6.1.4.9 height	31
6.1.4.10 is_under_user_control	31
6.1.4.11 last_x	31
6.1.4.12 last_y	31
6.1.4.13 mouse_x	32
6.1.4.14 mouse_y	32
6.1.4.15 name	32
6.1.4.16 NO_RESAMPLING	32
6.1.4.17 qml_path	32
6.1.4.18 RECEPTION	33
6.1.4.19 region_type	33
6.1.4.20 RESAMPLING	33
6.1.4.21 shape	33
6.1.4.22 source	33
6.1.4.23 texture_height	34
6.1.4.24 texture_path	34
6.1.4.25 TEXTURE_PATH_NONE	34
6.1.4.26 texture_width	34
6.1.4.27 width	34
6.1.4.28 x	35
6.1.4.29 y	35

---

<b>7 File Documentation</b>	<b>37</b>
7.1 SIEffect.py File Reference . . . . .	37
<b>Index</b>	<b>39</b>



# Chapter 1

## Namespace Index

### 1.1 Packages

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

<a href="#">SIEffect</a>	Documentation for this module / class . . . . .	9
--------------------------	---	---





## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

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

Effect	
SIEffect.SIEffect . . . . .	<a href="#">11</a>



## Chapter 3

# Class Index

### 3.1 Class List

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

<a href="#">SIEffect.SIEffect</a>	
Super Class from which all subsequent plugins are derived . . . . .	11



## Chapter 4

# File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

<a href="#">SIEffect.py</a>	.....	37
-----------------------------	-------	----



## 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





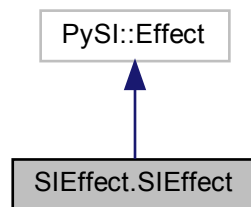
## Chapter 6

# Class Documentation

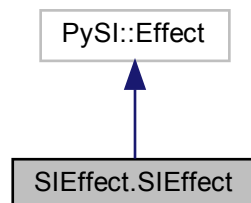
### 6.1 SIEffect.SIEffect Class Reference

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 `add_QML_data` (self, `key`, `value`, `datatype`)  
*member function for setting data in the associated qml file of a region effect*
- 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\\_rcv](#)  
*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\\_rcv](#)  
*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 12 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

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

Definition at line 42 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

<i>self</i>	the object pointer
-------------	--------------------

Definition at line 239 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

<i>self</i>	the object pointer
-------------	--------------------

Definition at line 245 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

<i>self</i>	the object pointer
<i>x</i>	the x coordinate of the cursor (float)
<i>y</i>	the y coordinate of the cursor (float)
<i>cursor↔ _id</i>	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 417 of file SIEffect.py.

### 6.1.3.4 add\_QML\_data()

```
def SIEffect.SIEffect.add_QML_data (
    self,
    key,
    value,
    datatype )
```

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

## Parameters

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

Calls the function **add\_data** (c++-bindings)

Definition at line 403 of file SIEffect.py.

## 6.1.3.5 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

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

Definition at line 515 of file SIEffect.py.

## 6.1.3.6 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

<i>self</i>	the object pointer
-------------	--------------------

Definition at line 492 of file SIEffect.py.

#### 6.1.3.7 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

<i>self</i>	the object pointer
<i>file_uuid</i>	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 453 of file SIEffect.py.

#### 6.1.3.8 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

<i>self</i>	the object pointer
-------------	--------------------

Definition at line 505 of file SIEffect.py.

#### 6.1.3.9 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

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

Definition at line 377 of file SIEffect.py.

## 6.1.3.10 create\_region\_via\_id()

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

member function for creating a new region

## Parameters

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

Definition at line 485 of file SIEffect.py.

## 6.1.3.11 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

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

Definition at line 477 of file SIEffect.py.

#### 6.1.3.12 delete()

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

member function for deleting a region

##### Parameters

<i>self</i>	the object pointer
-------------	--------------------

Definition at line 469 of file SIEffect.py.

#### 6.1.3.13 disable\_effect()

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

member function for disabling the emission or reception of an effect

##### Parameters

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

Definition at line 320 of file SIEffect.py.

#### 6.1.3.14 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

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

Definition at line 349 of file SIEffect.py.

## 6.1.3.15 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

<i>self</i>	the object pointer
<i>emission_capability</i>	the capability of the linking event used by the emitting region (str)
<i>reception_capability</i>	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\_rcv*. If *reception\_capability* is specified and present in *self.cap\_link\_rcv*, the specified relation is deleted from *emission\_capability*.

## See also

*self.cap\_link\_rcv*

Definition at line 361 of file SIEffect.py.

## 6.1.3.16 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

<i>self</i>	the object pointer
<i>source_uuid</i>	the uuid of the region associated to a folder icon representing a folder of the filesystem (str)
<i>with_buttons</i>	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 463 of file `SIEffect.py`.

#### 6.1.3.17 `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

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

Definition at line 289 of file `SIEffect.py`.

#### 6.1.3.18 `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

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

Definition at line 333 of file `SIEffect.py`.

#### 6.1.3.19 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

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

Definition at line 342 of file SIEffect.py.

#### 6.1.3.20 get\_region\_height()

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

member function for retrieving the maximum height of a region

Definition at line 200 of file SIEffect.py.

#### 6.1.3.21 get\_region\_width()

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

member function for retrieving the maximum width of a region

Definition at line 196 of file SIEffect.py.

#### 6.1.3.22 is\_effect\_enabled()

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

Definition at line 295 of file SIEffect.py.

#### 6.1.3.23 move()

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

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

##### Parameters

<i>self</i>	the object pointer
<i>x</i>	the absolute x coordinate of the point
<i>y</i>	the absolute y coordinate of the point

Definition at line 523 of file SIEffect.py.

#### 6.1.3.24 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

<i>self</i>	the object pointer
-------------	--------------------

Definition at line 261 of file SIEffect.py.

#### 6.1.3.25 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

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

Definition at line 253 of file SIEffect.py.

#### 6.1.3.26 on\_move\_leave\_rcv()

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

member function for receiving data from the PySI.MOVE capability for the PySI.ON\_LEAVE collision event

##### Parameters

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

Definition at line 269 of file SIEffect.py.

#### 6.1.3.27 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

<i>self</i>	the object pointer
<i>capability</i>	the capability of the collision event (str)
<i>is_emit</i>	the variable depicting if a region emits (True) or receives (False) an effect (bool)
<i>on_enter</i>	the function to be called for the collision event PySI.ON_ENTER
<i>on_continuous</i>	the function to be called for the collision event PySI.ON_CONTINUOUS
<i>on_leave</i>	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 312 of file SIEffect.py.

#### 6.1.3.28 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

<i>self</i>	the object pointer
<i>cursor↔ _id</i>	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 433 of file SIEffect.py.

#### 6.1.3.29 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

<i>self</i>	the object pointer
-------------	--------------------

Definition at line 227 of file SIEffect.py.

#### 6.1.3.30 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

<i>self</i>	the object pointer
-------------	--------------------

Definition at line 233 of file SIEffect.py.

## 6.1.3.31 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

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

Definition at line 388 of file SIEffect.py.

## 6.1.3.32 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

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

Definition at line 212 of file SIEffect.py.

#### 6.1.3.33 snap\_to\_mouse()

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

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

##### Parameters

<i>self</i>	the object pointer
-------------	--------------------

Definition at line 498 of file SIEffect.py.

#### 6.1.3.34 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

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

This function calls self.\_\_embed\_file\_standard\_appliation\_into\_context\_\_ (c++-bindings)

Definition at line 444 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 139 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 164 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. incoming 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 182 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 151 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 100 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 88 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 91 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 15 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 62 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 103 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 94 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 97 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 185 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 188 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 68 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 29 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 85 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 18 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 73 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 24 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 46 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 78 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 120 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 109 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 21 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 115 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 57 of file SIEffect.py.



#### 6.1.4.28 x

`SIEffect.SIEffect.x`

Definition at line 499 of file SIEffect.py.

#### 6.1.4.29 y

`SIEffect.SIEffect.y`

Definition at line 500 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

`__init__`  
    SIEffect.SIEffect, [14](#)

`absolute_x_pos`  
    SIEffect.SIEffect, [15](#)

`absolute_y_pos`  
    SIEffect.SIEffect, [15](#)

`add_point_to_region_drawing`  
    SIEffect.SIEffect, [16](#)

`add_QML_data`  
    SIEffect.SIEffect, [16](#)

`assign_effect`  
    SIEffect.SIEffect, [17](#)

`available_plugins`  
    SIEffect.SIEffect, [17](#)

`cap_emit`  
    SIEffect.SIEffect, [28](#)

`cap_link_emit`  
    SIEffect.SIEffect, [29](#)

`cap_link_recv`  
    SIEffect.SIEffect, [29](#)

`cap_recv`  
    SIEffect.SIEffect, [29](#)

`close_standard_application`  
    SIEffect.SIEffect, [18](#)

`color`  
    SIEffect.SIEffect, [30](#)

`context_dimensions`  
    SIEffect.SIEffect, [18](#)

`create_link`  
    SIEffect.SIEffect, [18](#)

`create_region_via_id`  
    SIEffect.SIEffect, [19](#)

`create_region_via_name`  
    SIEffect.SIEffect, [19](#)

`delete`  
    SIEffect.SIEffect, [20](#)

`delta_x`  
    SIEffect.SIEffect, [30](#)

`delta_y`  
    SIEffect.SIEffect, [30](#)

`disable_effect`  
    SIEffect.SIEffect, [20](#)

`disable_link_emission`  
    SIEffect.SIEffect, [20](#)

`disable_link_reception`  
    SIEffect.SIEffect, [21](#)

`display_folder_contents_page`  
    SIEffect.SIEffect, [21](#)

`EMISSION`  
    SIEffect.SIEffect, [30](#)

`enable_effect`  
    SIEffect.SIEffect, [22](#)

`enable_link_emission`  
    SIEffect.SIEffect, [22](#)

`enable_link_reception`  
    SIEffect.SIEffect, [22](#)

`get_region_height`  
    SIEffect.SIEffect, [23](#)

`get_region_width`  
    SIEffect.SIEffect, [23](#)

`height`  
    SIEffect.SIEffect, [31](#)

`is_effect_enabled`  
    SIEffect.SIEffect, [23](#)

`is_under_user_control`  
    SIEffect.SIEffect, [31](#)

`last_x`  
    SIEffect.SIEffect, [31](#)

`last_y`  
    SIEffect.SIEffect, [31](#)

`mouse_x`  
    SIEffect.SIEffect, [31](#)

`mouse_y`  
    SIEffect.SIEffect, [32](#)

`move`  
    SIEffect.SIEffect, [23](#)

`name`  
    SIEffect.SIEffect, [32](#)

`NO_RESAMPLING`  
    SIEffect.SIEffect, [32](#)

`on_move_continuous_recv`  
    SIEffect.SIEffect, [24](#)

`on_move_enter_recv`  
    SIEffect.SIEffect, [24](#)

`on_move_leave_recv`  
    SIEffect.SIEffect, [25](#)

`override_effect`  
    SIEffect.SIEffect, [25](#)

`qml_path`

- SIEffect.SIEffect, 32
- RECEPTION
  - SIEffect.SIEffect, 32
- region\_type
  - SIEffect.SIEffect, 33
- register\_region\_from\_drawing
  - SIEffect.SIEffect, 26
- relative\_x\_pos
  - SIEffect.SIEffect, 26
- relative\_y\_pos
  - SIEffect.SIEffect, 26
- remove\_link
  - SIEffect.SIEffect, 27
- RESAMPLING
  - SIEffect.SIEffect, 33
- set\_position\_from\_position
  - SIEffect.SIEffect, 27
- shape
  - SIEffect.SIEffect, 33
- SIEffect, 9
- SIEffect.py, 37
- SIEffect.SIEffect, 11
  - \_\_init\_\_, 14
  - absolute\_x\_pos, 15
  - absolute\_y\_pos, 15
  - add\_point\_to\_region\_drawing, 16
  - add\_QML\_data, 16
  - assign\_effect, 17
  - available\_plugins, 17
  - cap\_emit, 28
  - cap\_link\_emit, 29
  - cap\_link\_rcv, 29
  - cap\_rcv, 29
  - close\_standard\_application, 18
  - color, 30
  - context\_dimensions, 18
  - create\_link, 18
  - create\_region\_via\_id, 19
  - create\_region\_via\_name, 19
  - delete, 20
  - delta\_x, 30
  - delta\_y, 30
  - disable\_effect, 20
  - disable\_link\_emission, 20
  - disable\_link\_reception, 21
  - display\_folder\_contents\_page, 21
  - EMISSION, 30
  - enable\_effect, 22
  - enable\_link\_emission, 22
  - enable\_link\_reception, 22
  - get\_region\_height, 23
  - get\_region\_width, 23
  - height, 31
  - is\_effect\_enabled, 23
  - is\_under\_user\_control, 31
  - last\_x, 31
  - last\_y, 31
  - mouse\_x, 31
  - mouse\_y, 32
  - move, 23
  - name, 32
  - NO\_RESAMPLING, 32
  - on\_move\_continuous\_rcv, 24
  - on\_move\_enter\_rcv, 24
  - on\_move\_leave\_rcv, 25
  - override\_effect, 25
  - qml\_path, 32
  - RECEPTION, 32
  - region\_type, 33
  - register\_region\_from\_drawing, 26
  - relative\_x\_pos, 26
  - relative\_y\_pos, 26
  - remove\_link, 27
  - RESAMPLING, 33
  - set\_position\_from\_position, 27
  - shape, 33
  - snap\_to\_mouse, 28
  - source, 33
  - start\_standard\_application, 28
  - texture\_height, 33
  - texture\_path, 34
  - TEXTURE\_PATH\_NONE, 34
  - texture\_width, 34
  - width, 34
  - x, 34
  - y, 35
- snap\_to\_mouse
  - SIEffect.SIEffect, 28
- source
  - SIEffect.SIEffect, 33
- start\_standard\_application
  - SIEffect.SIEffect, 28
- texture\_height
  - SIEffect.SIEffect, 33
- texture\_path
  - SIEffect.SIEffect, 34
- TEXTURE\_PATH\_NONE
  - SIEffect.SIEffect, 34
- texture\_width
  - SIEffect.SIEffect, 34
- width
  - SIEffect.SIEffect, 34
- x
  - SIEffect.SIEffect, 34
- y
  - SIEffect.SIEffect, 35