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

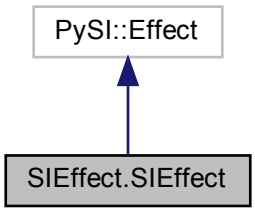
Chapter 6

Class Documentation

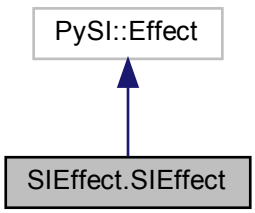
6.1 SIEffect.SIEffect Class Reference

Super Class from which all subsequent plugins are derived.

Inheritance diagram for SIEffect.SIEffect:



Collaboration diagram for SIEffect.SIEffect:



Public Member Functions

- def print_calling_info (func)

  decorator function for printing which object called the the given function func
- None __init__ (self, PySI.PointVector shape, str uuid, str texture_path, int regiontype, str regionname, dict kwargs, str __source__="custom")

  constructor
- list current_regions (self)

  member function for retrieving all effects currently represented as regions
- bool was_moved (self)

  member function for retrieving whether a region was moved the frame before
- list conditional_variables (self)

  member function for retrieving all variables which were annotated with SIEffect.SI_CONDITION
- int get_region_width (self)

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

  member function for retrieving the maximum height of a region
- int relative_x_pos (self)

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

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

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

  member function for getting the absolute y coordinate of the parent region's top left corner
- None enable_effect (self, str capability, bool is_emit, object on_enter, object on_continuous, object on_leave)

  member function for enabling the emission or reception of an effect
- bool is_effect_enabled (self, str capability, bool is_emit)

  member function for determining whether a collision event exists
- None override_effect (self, str capability, bool is_emit, object on_enter, object on_continuous, object on_leave)

  member function for overriding the emission or reception of an effect
- None disable_effect (self, str capability, bool is_emit)

  member function for disabling the emission or reception of an effect
- None enable_link_emission (self, str emission_capability, object emission_function)

  member function for enabling the emission of data in the context of a link event
- None enable_link_reception (self, str emission_capability, str reception_capability, object reception_function)

  member function for enabling the emission of data in the context of a link event
- None disable_link_emission (self, str emission_capability)

  member function for disabling the emission of data in the context of a link event
- None disable_link_reception (self, str emission_capability, str reception_capability="")

  member function for disabling the reception of data in the context of a link event
- None create_link (self, str sender_uuid, str sender_attribute, str receiver_uuid, str receiver_attribute)

  member function for establishing a specified link between two regions according to given attributes
- None remove_link (self, str sender_uuid, str sender_attribute, str receiver_uuid, str receiver_attribute)

  member function for removing a specified link between two regions according to given attributes
- bool is_linked (self, send_uuid, send_attr, recv_uuid, recv_attr)
- None emit_linking_action (self, object sender, str capability, tuple args)

  member function for emitting a linking action
- None set_QML_data (self, str key, object value, int datatype, data_kwargs={})

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

- object get_QML_data (self, str key, int datatype)

  member function for getting data set from an associated qml file of a region effect
- str set_QML_path (self, str filename)

  member function for setting the path to an plugin's associated qml file
- None add_point_to_region_drawing (self, float x, float y, str cursor_id)

  member function for adding a point to a region drawing based on a cursor id.
- None register_region_from_drawing (self, str cursor_id, dict kwargs={})

  member function for registering a region drawing according to a cursor id
- None cancel_region_drawing (self, str cursor_id)

  member function for canceling the current drawing of a region
- None start_standard_application (self, str file_uuid, str file_path)

  member function for starting the standard application of a file given its uuid as a region and its path in the filesystem
- None close_standard_application (self, str file_uuid)

  member function for closing the standard application of a file given its uuid as a region and its path in the filesystem
- None display_folder_contents_page (self, int page, str source_uuid, with_buttons=True)

  member function for displaying the contents of a folder in the filesystem as pages of other filesystem entries
- None delete (self, str uuid=None)

  member function for deleting a region
- bool is_flagged_for_deletion (self)

  member function which provides whether the region self is flagged for deletion, i.e.
- bool enveloped_by (self, object other)

  member function which provides whether a given region is fully enveloped by this region
- None create_region_via_name (self, PySI.PointVector shape, str effect_name, as_selector=False, kwargs={})

  member function for creating a new region
- None create_region_via_id (self, PySI.PointVector shape, str effect_type, kwargs={})

  member function for creating a new region
- None create_region_via_class (self, list shape, object clazz, kwargs={})

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

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

  member function for retrieving the plugins which are exluded from use This list of names contains regionname attributes
- None snap_to_mouse (self)

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

  member function for retrieving the dimensions of the active SI-Context (width in px, and height in px)
- None assign_effect (self, str effect_name_to_assign, str effect_display_name, str effect_texture, dict 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!
- None move (self, x, y)

  member function for moving the effect's associated region to the point (x, y)
- None set_drawing_additions (self, list drawing_additions)

  member function for registering additional drawings to a region without having them to add as a region
- list get_drawing_additions (self)

  member function for retrieving the additional drawings of a region
- None run_in_thread (self, object function, tuple args)

  member function for offloading a function call to a thread
- list present_collisions (self)

  member function which provides a list of region uuids which are currently overlapping with the region self
- list present_collisions_names (self)

member function which provides a list of region names which are currently overlapping with the region self

- list present_collisions_uuids (self)

  member function which provides a list of region names which are currently overlapping with the region self
- dict selected_effects_by_cursor_id (self)

  member function which provides the regionnames of current cursors
- None si_print (self, *tuple args)

  member function which is used by SIGRun during the python interpreter embedding step, in order to substitute the builtin print function with this one
- None set_cursor_stroke_width_by_cursorid (self, str cursor_id, int stroke_width)

  member function which is used to control the stroke width with which a cursor can draw
- None set_cursor_stroke_color_by_cursorid (self, str cursor_id, PySI.Color color)

  member function which is used to control the color with which a cursor can draw
- list round_edge (self, list pts)

  member function which is used when a rectangular regions is created in order to rounds the corners of the rectangular region
- None __handle_exception__ (self, Exception ex, str file)

  member function for generally handling exceptions which may occur in constructors of plugins

Static Public Member Functions

- def on_enter (capability, transmission_type)

  Decorator for registering on_enter collision events.
- def on_continuous (capability, transmission_type)

  Decorator for registering on_continuous collision events.
- def on_leave (capability, transmission_type)

  Decorator for registering on_leave collision events.
- def on_link (transmission_type, emission_capability, reception_capability=None)

  Decorator for registering linking actions.

Public Attributes

- is_resampling_enabled
- with_border

  member attribute variable serving as a rendering hint for showing a regions border
- border_color
- context_height
- border_width
- width

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

  member variable containing the maximum height of the region
- visualization_width
- visualization_height
- 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.
- was_under_user_control
- 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 debuggable behaviour!

- SI_CONDITION = None

  static member attribute used as an type annotations for variables which are marked as variables used for SI drawable conditionals

- bool resampling_enabled = True

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

6.1.2 Constructor & Destructor Documentation

6.1.2.1 __init__()

```
None SIEffect.SIEffect.__init__ (
              self,
             PySI.PointVector shape,
             str uuid,
             str texture_path,
             int regiontype,
             str regionname,
             dict kwargs,
             str __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)

Returns

None

6.1.3 Member Function Documentation

6.1.3.1 __handle_exception__()

```
None SIEffect.SIEffect.__handle_exception__ (
              self,
             Exception ex,
             str file )
```

member function for generally handling exceptions which may occur in constructors of plugins

Author

Robert Fent (as part of his Bachelor's Thesis)

Parameters

ex	the thrown exception as an Exception object
file	the absolute path to the plugin file in which the exception occurred

Returns

None

6.1.3.2 absolute_x_pos()

```
int 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

Returns

the absolute x coordinate of the associated region's top left corner

6.1.3.3 absolute_y_pos()

```
int 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

Returns

the absolute y coordinate of the associated region's top left corner

6.1.3.4 add_point_to_region_drawing()

```
None SIEffect.SIEffect.add_point_to_region_drawing (
              self,
             float x,
             float y,
             str 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.

Returns

None

6.1.3.5 assign_effect()

```
None SIEffect.SIEffect.assign_effect (
              self,
             str effect_name_to_assign,
             str effect_display_name,
             str effect_texture,
             dict 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

Returns

None

6.1.3.6 available_plugins()

```
list SIEffect.SIEffect.available_plugins (
              self )
```

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

This list of names contains regionname attributes

Parameters

self	the object pointer

Returns

a list containing all names of available plugins as str values

6.1.3.7 cancel_region_drawing()

```
None SIEffect.SIEffect.cancel_region_drawing (
              self,
             str cursor_id )
```

member function for canceling the current drawing of a region

Parameters

cursor_id	the uuid of the mouse cursor which is currently used for drawing for which the drawing is to be cancelled

Returns

None

6.1.3.8 close_standard_application()

```
None SIEffect.SIEffect.close_standard_application (
              self,
             str 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)

Returns

None

6.1.3.9 conditional_variables()

```
list SIEffect.SIEffect.conditional_variables (
              self )
```

member function for retrieving all variables which were annotated with SIEffect.SI_CONDITION

Returns

the list of condition variables as list

6.1.3.10 context_dimensions()

```
tuple 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

Returns

the dimensions of the active SI context as a tuple

6.1.3.11 create_link()

```
None SIEffect.SIEffect.create_link (
              self,
             str sender_uuid,
             str sender_attribute,
             str receiver_uuid,
             str 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)

Returns

None

6.1.3.12 create_region_via_class()

```
None SIEffect.SIEffect.create_region_via_class (
              self,
             list shape,
             object clazz,
              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]]
clazz	the object which can be used to call the constructor from

Returns

None

6.1.3.13 create_region_via_id()

```
None SIEffect.SIEffect.create_region_via_id (
             self,
            PySI.PointVector shape,
            str 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)

Returns

None

6.1.3.14 create_region_via_name()

```
None SIEffect.SIEffect.create_region_via_name (
             self,
            PySI.PointVector shape,
            str 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)

Returns

None

6.1.3.15 current_regions()

```
list SIEffect.SIEffect.current_regions (
              self )
```

member function for retrieving all effects currently represented as regions

Returns

the list of effects as a list

6.1.3.16 delete()

```
None SIEffect.SIEffect.delete (
              self,
             str uuid = None )
```

member function for deleting a region

Parameters

self	the object pointer

Returns

None

6.1.3.17 disable_effect()

```
None SIEffect.SIEffect.disable_effect (
              self,
             str capability,
             bool 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)

Returns

None

6.1.3.18 disable_link_emission()

```
None SIEffect.SIEffect.disable_link_emission (
              self,
             str 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)

Returns

None

6.1.3.19 disable_link_reception()

```
None SIEffect.SIEffect.disable_link_reception (
              self,
             str emission_capability,
             str 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

Returns

None

6.1.3.20 display_folder_contents_page()

```
None SIEffect.SIEffect.display_folder_contents_page (
              self,
             int page,
             str 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
page	the number of the current page which browsed in a folder region
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)

Returns

None

6.1.3.21 emit_linking_action()

```
None SIEffect.SIEffect.emit_linking_action (
              self,
             object sender,
             str capability,
             tuple args )
```

member function for emitting a linking action

Parameters

sender	the source of the the linking action
capability	the capability with which the linking action shall be emitted
args	the data which is to be received by receivers

Returns

None

6.1.3.22 enable_effect()

```
None SIEffect.SIEffect.enable_effect (
              self,
```

```
            str capability,
            bool is_emit,
            object on_enter,
            object on_continuous,
            object on_leave )
```

member function for enabling the emission or reception of an effect

This function is used in order to register collision events. During loading of plugins, the SIGRun plugin transpiler adds this function to the constructor of transpiled plugins based on the information provided in the associated Decorator

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

See also

on_enter(capability, transmission_type):
on_continuous(capability, transmission_type):
on_leave(capability, transmission_type):

Returns

None

6.1.3.23 enable_link_emission()

```
None SIEffect.SIEffect.enable_link_emission (
             self,
            str emission_capability,
            object emission_function )
```

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

This function is used in order to register linking actions for emission. During loading of plugins, the SIGRun plugin transpiler adds this function to the constructor of transpiled plugins based on the information provided in the associated decorator.

Parameters

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

See also

on_link(transmission_type, emission_capability, reception_capability=None)

Returns

None

6.1.3.24 enable_link_reception()

```
None SIEffect.SIEffect.enable_link_reception (
              self,
             str emission_capability,
             str reception_capability,
             object reception_function )
```

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

This function is used in order to register linking actions for reception. During loading of plugins, the SIGRun plugin transpiler adds this function to the constructor of transpiled plugins based on the information provided in the associated decorator.

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

See also

on_link(transmission_type, emission_capability, reception_capability=None)

Returns

None

6.1.3.25 enveloped_by()

```
bool SIEffect.SIEffect.enveloped_by (
              self,
             object other )
```

member function which provides whether a given region is fully enveloped by this region

Parameters

other	a colliding effect which is checked if it is fully enveloped by self

Returns

a bool representing if the region self is flagged for deletion or not

6.1.3.26 excluded_plugins()

```
list SIEffect.SIEffect.excluded_plugins (
              self )
```

member function for retrieving the plugins which are exluded from use This list of names contains regionname attributes

Returns

a list containing all names of excluded plugins as str values

6.1.3.27 get_drawing_additions()

```
list SIEffect.SIEffect.get_drawing_additions (
              self )
```

member function for retrieving the additional drawings of a region

Returns

the list containing the additional drawings

6.1.3.28 get_QML_data()

```
object SIEffect.SIEffect.get_QML_data (
              self,
             str key,
             int datatype )
```

member function for getting data set from an associated qml file of a region effect

Parameters

self	the object pointer
key	the key specified in QML to address the required data
datatype	the data type of the requested value (PySI.DataType.INT, PySI.DataType.FLOAT, ...) (int)

Returns

the value queried by the key as the given datatype

6.1.3.29 get_region_height()

```
int SIEffect.SIEffect.get_region_height (
              self )
```

member function for retrieving the maximum height of a region

Parameters

self	the pointer to the object

Returns

the width of the associated region as int

6.1.3.30 get_region_width()

```
int SIEffect.SIEffect.get_region_width (
              self )
```

member function for retrieving the maximum width of a region

Parameters

self	the pointer to the object

Returns

the width of the associated region as int

6.1.3.31 is_effect_enabled()

```
bool SIEffect.SIEffect.is_effect_enabled (
              self,
             str capability,
             bool is_emit )
```

member function for determining whether a collision event exists

Parameters

self	the object pointer
capability	the capability of the collision event (str)
is_emit	the transmission type (bool)

Returns

True if a collision event exists with the given capability and transmission type, False else

6.1.3.32 is_flagged_for_deletion()

```
bool SIEffect.SIEffect.is_flagged_for_deletion (
              self )
```

member function which provides whether the region self is flagged for deletion, i.e.

will be deleted next frame

Returns

a bool representing if the region self is flagged for deletion or not

6.1.3.33 is_linked()

```
bool SIEffect.SIEffect.is_linked (
              self,
              send_uuid,
              send_attr,
              recv_uuid,
              recv_attr )
```

6.1.3.34 move()

```
None 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

Returns

None

6.1.3.35 on_continuous()

```
def SIEffect.SIEffect.on_continuous (
              capability,
              transmission_type )  [static]
```

Decorator for registering on_continuous collision events.

Decorates a specific function in other plugin files to be used as an on_continuous collision event. Recommended use: @SIEffect.on_continuous(<capability>, <transmission_type>)

This decorator adds no functionality and only provides easier syntax for defining on_continuous collision events. The decorator is detected by the SIGRun plugin transpiler during the plugin loading step. In this step, the transpiler removes the decorator and appends an equivalent function call to the plugin's constructor, in order to register the on_continuous collision event.

Parameters

capability	the str value serving as the identifier for the on_continuous collision event
transmission_type	the bool value serving to determine whether the event shall be emitted (SIEffect.EMISSION) or received (SIEffect.RECEPTION)

Returns

the decorated function

6.1.3.36 on_enter()

```
def SIEffect.SIEffect.on_enter (
              capability,
              transmission_type )  [static]
```

Decorator for registering on_enter collision events.

Decorates a specific function in other plugin files to be used as an on_enter collision event. Recommended use: @SIEffect.on_enter(<capability>, <transmission_type>)

This decorator adds no functionality and only provides easier syntax for defining on_enter collision events. The decorator is detected by the SIGRun plugin transpiler during the plugin loading step. In this step, the transpiler removes the decorator and appends an equivalent function call to the plugin's constructor, in order to register the on_enter collision event.

Parameters

capability	the str value serving as the identifier for the on_enter collision event
transmission_type	the bool value serving to determine whether the event shall be emitted (SIEffect.EMISSION) or received (SIEffect.RECEPTION)

Returns

the decorated function

6.1.3.37 on_leave()

```
def SIEffect.SIEffect.on_leave (
              capability,
              transmission_type )  [static]
```

Decorator for registering on_leave collision events.

Decorates a specific function in other plugin files to be used as an on_leave collision event. Recommended use: @SIEffect.on_leave(<capability>, <transmission_type>)

This decorator adds no functionality and only provides easier syntax for defining on_leave collision events. The decorator is detected by the SIGRun plugin transpiler during the plugin loading step. In this step, the transpiler removes the decorator and appends an equivalent function call to the plugin's constructor, in order to register the on_leave collision event.

Parameters

capability	the str value serving as the identifier for the on_leave collision event
transmission_type	the bool value serving to determine whether the event shall be emitted (SIEffect.EMISSION) or received (SIEffect.RECEPTION)

Returns

the decorated function

6.1.3.38 on_link()

```
def SIEffect.SIEffect.on_link (
              transmission_type,
              emission_capability,
              reception_capability = None )  [static]
```

Decorator for registering linking actions.

Decorates a specific function in other plugin files to be used as an linking action. Recommended use: @SI↩ Effect.on_link(<transmission_type>, <emission_capability>, <reception_capability>)

This decorator adds no functionality and only provides easier syntax for defining linking actions. The decorator is detected by the SIGRun plugin transpiler during the plugin loading step. In this step, the transpiler removes the decorator and appends an equivalent function call to the plugin's constructor, in order to register the linking action. Here, the transpiler differentiates the emission of a linking action: @SIEffect.on_link(SIEffect.EMISSION, <capability>) and the reception of a linking action: @SIEffect.on_link(SIEffect.Reception, <emission_capability>, <reception_capability>)

Parameters

transmission_type	the bool value serving to determine whether the event shall be emitted (SIEffect.EMISSION) or received (SIEffect.RECEPTION)
emission_capability	the str value serving as the identifier of with which the linking action was emitted from its source
reception_capability	the str value serving as the identifier of with which the linking action shall be received

Returns

the decorated function

6.1.3.39 override_effect()

```
None SIEffect.SIEffect.override_effect (
              self,
             str capability,
             bool is_emit,
             object on_enter,
             object on_continuous,
             object 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)

Returns

None

6.1.3.40 present_collisions()

```
list SIEffect.SIEffect.present_collisions (
              self )
```

member function which provides a list of region uuids which are currently overlapping with the region self

Returns

a list which contains the uuids and names of colliding regions

6.1.3.41 present_collisions_names()

```
list SIEffect.SIEffect.present_collisions_names (
              self )
```

member function which provides a list of region names which are currently overlapping with the region self

Returns

a list which contains the names of the colliding regions

6.1.3.42 present_collisions_uuids()

```
list SIEffect.SIEffect.present_collisions_uuids (
              self )
```

member function which provides a list of region names which are currently overlapping with the region self

Returns

a list which contains the uuids of the colliding regions

6.1.3.43 print_calling_info()

```
def SIEffect.SIEffect.print_calling_info (
              func )
```

decorator function for printing which object called the the given function func

Parameters

func	the function that will be executed and it will be printed which object did call it

Returns

the wrapper function of the decorator

6.1.3.44 register_region_from_drawing()

```
None SIEffect.SIEffect.register_region_from_drawing (
              self,
             str cursor_id,
             dict kwargs = {} )
```

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.

Returns

None

6.1.3.45 relative_x_pos()

```
int 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

Returns

the relative x coordinate of the associated region's top left corner

6.1.3.46 relative_y_pos()

```
int 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

Returns

the relative y coordinate of the associated region's top left corner

6.1.3.47 remove_link()

```
None SIEffect.SIEffect.remove_link (
              self,
             str sender_uuid,
             str sender_attribute,
             str receiver_uuid,
             str 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)

Returns

None

6.1.3.48 round_edge()

```
list SIEffect.SIEffect.round_edge (
              self,
             list pts )
```

member function which is used when a rectangular regions is created in order to rounds the corners of the rectangular region

Parameters

pts	a list of lists containing the coordinates of the points

Returns

the list of list containing the new coordinates of the points

6.1.3.49 run_in_thread()

```
None SIEffect.SIEffect.run_in_thread (
              self,
             object function,
             tuple args )
```

member function for offloading a function call to a thread

This function launches a given function in another thread. The threaded function's return value cannot be retrieved. This function should be used when a long operation (procedure) has to be computed which at the start of its computation is completely independent of any other function or variables.

Parameters

function	the function to be offloaded
args	the arguments with which the function is intended to be called

Returns

None

6.1.3.50 selected_effects_by_cursor_id()

```
dict SIEffect.SIEffect.selected_effects_by_cursor_id (
              self )
```

member function which provides the regionnames of current cursors

Returns

dictionary in which cursor ids are keys and the selected effects regionname as value

6.1.3.51 set_cursor_stroke_color_by_cursorid()

```
None SIEffect.SIEffect.set_cursor_stroke_color_by_cursorid (
              self,
             str cursor_id,
             PySI.Color color )
```

member function which is used to control the color with which a cursor can draw

Parameters

cursor_id	the uuid of the cursor
color	the PySI.Color with which the color of the stroke of the drawing on the canvas is defined

Returns

None

6.1.3.52 set_cursor_stroke_width_by_cursorid()

```
None SIEffect.SIEffect.set_cursor_stroke_width_by_cursorid (
              self,
             str cursor_id,
             int stroke_width )
```

member function which is used to control the stroke width with which a cursor can draw

Parameters

cursor_id	the uuid of the cursor
stroke_with	the integer with which the width of the stroke of the drawing on the canvas is defined

Returns

None

6.1.3.53 set_drawing_additions()

```
None SIEffect.SIEffect.set_drawing_additions (
              self,
             list drawing_additions )
```

member function for registering additional drawings to a region without having them to add as a region

Parameters

drawing_additions	the list containing further lists which represents lines or shapes. Such lines or shapes consist of points (x, y) also represented as a list

Usage: self.set_drawing_additions([[[px, py], ...], [[qx, qy,], ...], ...]

Returns

None

6.1.3.54 set_QML_data()

```
None SIEffect.SIEffect.set_QML_data (
              self,
             str key,
             object value,
             int 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)

Returns

None

6.1.3.55 set_QML_path()

```
str SIEffect.SIEffect.set_QML_path (
              self,
             str 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 (str)

6.1.3.56 si_print()

```
None SIEffect.SIEffect.si_print (
              self,
             *tuple args )
```

member function which is used by SIGRun during the python interpreter embedding step, in order to substitute the builtin print function with this one

@detail the corresponding call in SIGRun (C++): bp::exec((std::string("import builtins\nimport os\n\n") + "os.↩
remove(\".TEST.TXT")
" + "open(".TEST.TXT", 'x').close()
" + "def si_print(filename):
" " def wrap(func):
" + " def wrapped_func(∗args, ∗∗kwargs):
" + " with open(filename, \'a\') as outputfile:
" + " out = str(args).replace(chr(0), '')
" + " outputfile.write(out)
" + " outputfile.write('\n')
" + " return func("PySI:", ∗args, ∗∗kwargs)
" + " return wrapped_func
" + " return wrap

" + "builtins.print = si_print(".TEST.TXT")(builtins.print)
").c_str(), d_globals);

Parameters

args	an arbitrary amount if non-keyword parameters passed as a tuple which is forwarded to builtin print

Returns

None

6.1.3.57 snap_to_mouse()

```
None SIEffect.SIEffect.snap_to_mouse (
              self )
```

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

Parameters

self	the object pointer

Returns

None

6.1.3.58 start_standard_application()

```
None SIEffect.SIEffect.start_standard_application (
              self,
             str file_uuid,
             str 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)

Returns

None

6.1.3.59 was_moved()

```
bool SIEffect.SIEffect.was_moved (
              self )
```

member function for retrieving whether a region was moved the frame before

Returns

the bool if the region was moved or not

6.1.4 Member Data Documentation

6.1.4.1 border_color

SIEffect.SIEffect.border_color

6.1.4.2 border_width

SIEffect.SIEffect.border_width

6.1.4.3 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

6.1.4.4 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.

6.1.4.5 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.

6.1.4.6 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

6.1.4.7 color

SIEffect.SIEffect.color

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

6.1.4.8 context_height

SIEffect.SIEffect.context_height

6.1.4.9 default_border_color

```
SIEffect.SIEffect.default_border_color
```

6.1.4.10 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

6.1.4.11 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

6.1.4.12 EMISSION

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

member constant to mark an effect or link emittable

6.1.4.13 height

```
SIEffect.SIEffect.height
```

member variable containing the maximum height of the region

computed via aabb

6.1.4.14 is_resampling_enabled

```
SIEffect.SIEffect.is_resampling_enabled
```

6.1.4.15 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

6.1.4.16 last_x

`SIEffect.SIEffect.last_x`

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

6.1.4.17 last_y

`SIEffect.SIEffect.last_y`

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

6.1.4.18 mouse_x

`SIEffect.SIEffect.mouse_x`

member attribute variable storing the x position of the mouse cursor

6.1.4.19 mouse_y

`SIEffect.SIEffect.mouse_y`

member attribute variable storing the y position of the mouse cursor

6.1.4.20 name

`SIEffect.SIEffect.name`

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

6.1.4.21 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 debuggable behaviour!

6.1.4.22 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

6.1.4.23 RECEPTION

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

static member attribute to mark an effect or link receivable

6.1.4.24 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

6.1.4.25 RESAMPLING

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

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

6.1.4.26 resampling_enabled

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

6.1.4.27 SI_CONDITION

```
SIEffect.SIEffect.SI_CONDITION = None  [static]
```

static member attribute used as an type annotations for variables which are marked as variables used for SI drawable conditionals

Usage self.<identifier>: SIEffect.SI_CONDITION = <bool value>="">

6.1.4.28 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"

6.1.4.29 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

6.1.4.30 texture_path

```
SIEffect.SIEffect.texture_path
```

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

6.1.4.31 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

6.1.4.32 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

6.1.4.33 visualization_height

`SIEffect.SIEffect.visualization_height`

6.1.4.34 visualization_width

`SIEffect.SIEffect.visualization_width`

6.1.4.35 was_under_user_control

`SIEffect.SIEffect.was_under_user_control`

6.1.4.36 width

`SIEffect.SIEffect.width`

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

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

6.1.4.37 with_border

`SIEffect.SIEffect.with_border`

member attribute variable serving as a rendering hint for showing a regions border

6.1.4.38 x

`SIEffect.SIEffect.x`

6.1.4.39 y

`SIEffect.SIEffect.y`

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