Skip to content

# SpaceNLA(Space)

base classes — `bpy_struct`, `Space`

**class** bpy.types.**SpaceNLA(Space)**

NLA editor space data

**dopesheet**

Settings for filtering animation data

**TYPE:**

`DopeSheet`, (readonly)

**show_local_markers**

Show action-local markers on the strips, useful when synchronizing timing across strips

**TYPE:**

boolean, default False

**show_markers**

If any exists, show markers in a separate row at the bottom of the editor

**TYPE:**

boolean, default False

**show_region_channels**

**TYPE:**

boolean, default False

**show_region_hud**

**TYPE:**

boolean, default False

**show_region_ui**

**TYPE:**

boolean, default False

**show_seconds**

Show timing as a timecode instead of frames

**TYPE:**

boolean, default False

**show_strip_curves**

Show influence F-Curves on strips

**TYPE:**

boolean, default False

**use_realtime_update**

When transforming strips, changes to the animation data are flushed to other views

**TYPE:**

boolean, default False

**classmethod bl_rna_get_subclass(id, default=None)**

**PARAMETERS:**

**id** (*str*) – The RNA type identifier.

**RETURNS:**

The RNA type or default when not found.

**RETURN TYPE:**

`bpy.types.Struct` subclass

**classmethod bl_rna_get_subclass_py(id, default=None)**

**PARAMETERS:**

**id** (*str*) – The RNA type identifier.

**RETURNS:**

The class or default when not found.

**RETURN TYPE:**

type

**classmethod draw_handler_add(callback, args, region_type, draw_type)**

Add a new draw handler to this space type. It will be called every time the specified region in the space type will be drawn. Note: All argumer are positional only for now.

**PARAMETERS:**

- **callback** (*Callable[[Any, ...], Any]*) – A function that will be called when the region is drawn. It gets the specified arguments as input, it's return value is ignored.
- **args** (*tuple[Any, ...]*) – Arguments that will be passed to the callback.
- **region_type** (*str*) – The region type the callback draws in; usually `WINDOW`. (`bpy.types.Region.type`)
- **draw_type** (*str*) – Usually `POST_PIXEL` for 2D drawing and `POST_VIEW` for 3D drawing. In some cases `PRE_VIEW` can be used. `BACKDROP` can be used for backdrops in the node editor.

**RETURNS:**

Handler that can be removed later on.

**RETURN TYPE:**

object

**classmethod draw_handler_remove(handler, region_type)**

Remove a draw handler that was added previously.

**PARAMETERS:**

- **handler** (*object*) – The draw handler that should be removed.
- **region_type** (*str*) – Region type the callback was added to.

## Inherited Properties

- `bpy_struct.id_data`
- `Space.type`
- `Space.show_locked_time`
- `Space.show_region_header`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`

- `bpy_struct.get`
- `bpy_struct.id_properties_clear`
- `bpy_struct.id_properties_ensure`
- `bpy_struct.id_properties_ui`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_overridable_library`
- `bpy_struct.is_property_readonly`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.path_resolve`
- `bpy_struct.pop`
- `bpy_struct.property_overridable_library_set`
- `bpy_struct.property_unset`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Space.bl_rna_get_subclass`
- `Space.bl_rna_get_subclass_py`
- `Space.draw_handler_add`
- `Space.draw_handler_remove`