

[Skip to content](#)

Scene Operators

`bpy.ops.scene.delete()`

Delete active scene

`bpy.ops.scene.freestyle_add_edge_marks_to_keying_set()`

Add the data paths to the Freestyle Edge Mark property of selected edges to the active keying set

FILE:

[startup/bl_operators/freestyle.py:136](#)

`bpy.ops.scene.freestyle_add_face_marks_to_keying_set()`

Add the data paths to the Freestyle Face Mark property of selected polygons to the active keying set

FILE:

[startup/bl_operators/freestyle.py:167](#)

`bpy.ops.scene.freestyle_alpha_modifier_add(*, type='ALONG_STROKE')`

Add an alpha transparency modifier to the line style associated with the active lineset

PARAMETERS:

type (enum in [Linestyle Alpha Modifier Type Items](#), (optional)) – Type

`bpy.ops.scene.freestyle_color_modifier_add(*, type='ALONG_STROKE')`

Add a line color modifier to the line style associated with the active lineset

PARAMETERS:

type (enum in [Linestyle Color Modifier Type Items](#), (optional)) – Type

`bpy.ops.scene.freestyle_fill_range_by_selection(*, type='COLOR', name='')`

Fill the Range Min/Max entries by the min/max distance between selected mesh objects and the source object (either a user-specified object or the active camera)

PARAMETERS:

- **type** (enum in `['COLOR', 'ALPHA', 'THICKNESS']`, (optional)) – Type, Type of the modifier to work on
 - `COLOR` Color – Color modifier type.
 - `ALPHA` Alpha – Alpha modifier type.
 - `THICKNESS` Thickness – Thickness modifier type.
- **name** (string, (optional, never None)) – Name, Name of the modifier to work on

FILE:

[startup/bl_operators/freestyle.py:42](#)

`bpy.ops.scene.freestyle_geometry_modifier_add(*, type='2D_OFFSET')`

Add a stroke geometry modifier to the line style associated with the active lineset

PARAMETERS:

type (enum in [Linestyle Geometry Modifier Type Items](#), (optional)) – Type

`bpy.ops.scene.freestyle_lineset_add()`

Add a line set into the list of line sets

`bpy.ops.scene.freestyle_lineset_copy()`

Copy the active line set to the internal clipboard

`bpy.ops.scene.freestyle_lineset_move(*, direction='UP')`

Change the position of the active line set within the list of line sets

PARAMETERS:

direction (*enum in ['UP', 'DOWN'], (optional)*) – Direction, Direction to move the active line set towards

`bpy.ops.scene.freestyle_lineset_paste()`

Paste the internal clipboard content to the active line set

`bpy.ops.scene.freestyle_lineset_remove()`

Remove the active line set from the list of line sets

`bpy.ops.scene.freestyle_linestyle_new()`

Create a new line style, reusable by multiple line sets

`bpy.ops.scene.freestyle_modifier_copy()`

Duplicate the modifier within the list of modifiers

`bpy.ops.scene.freestyle_modifier_move(*, direction='UP')`

Move the modifier within the list of modifiers

PARAMETERS:

direction (*enum in ['UP', 'DOWN'], (optional)*) – Direction, Direction to move the chosen modifier towards

`bpy.ops.scene.freestyle_modifier_remove()`

Remove the modifier from the list of modifiers

`bpy.ops.scene.freestyle_module_add()`

Add a style module into the list of modules

`bpy.ops.scene.freestyle_module_move(*, direction='UP')`

Change the position of the style module within in the list of style modules

PARAMETERS:

direction (*enum in ['UP', 'DOWN'], (optional)*) – Direction, Direction to move the chosen style module towards

`bpy.ops.scene.freestyle_module_open(*, filepath="", make_internal=True)`

Open a style module file

PARAMETERS:

- **filepath** (*string, (optional, never None)*) – filepath
- **make_internal** (*boolean, (optional)*) – Make internal, Make module file internal after loading

FILE:

[startup/bl_operators/freestyle.py:212](#)

`bpy.ops.scene.freestyle_module_remove()`

Remove the style module from the stack

`bpy.ops.scene.freestyle_stroke_material_create()`

Create Freestyle stroke material for testing

`bpy.ops.scene.freestyle_thickness_modifier_add(*, type='ALONG_STROKE')`

Add a line thickness modifier to the line style associated with the active lineset

PARAMETERS:

type (enum in [Linestyle Thickness Modifier Type Items](#), (optional)) – Type

bpy.ops.scene.gltf2_action_filter_refresh()

Refresh list of actions

FILE:

[addons_core/io_scene_gltf2/blender/com/gltf2_blender_ui.py:615](#)

bpy.ops.scene.gpencil_brush_preset_add(*, name="", remove_name=False, remove_active=False)

Add or remove grease pencil brush preset

PARAMETERS:

- **name** (*string, (optional, never None)*) – Name, Name of the preset, used to make the path name
- **remove_name** (*boolean, (optional)*) – remove_name
- **remove_active** (*boolean, (optional)*) – remove_active

FILE:

[startup/bl_operators/presets.py:119](#)

bpy.ops.scene.gpencil_material_preset_add(*, name="", remove_name=False, remove_active=False)

Add or remove Grease Pencil material preset

PARAMETERS:

- **name** (*string, (optional, never None)*) – Name, Name of the preset, used to make the path name
- **remove_name** (*boolean, (optional)*) – remove_name
- **remove_active** (*boolean, (optional)*) – remove_active

FILE:

[startup/bl_operators/presets.py:119](#)

bpy.ops.scene.new(*, type='NEW')

Add new scene by type

PARAMETERS:

type (enum in ['NEW', 'EMPTY', 'LINK_COPY', 'FULL_COPY'], (optional)) –

Type

- **NEW** New – Add a new, empty scene with default settings.
- **EMPTY** Copy Settings – Add a new, empty scene, and copy settings from the current scene.
- **LINK_COPY** Linked Copy – Link in the collections from the current scene (shallow copy).
- **FULL_COPY** Full Copy – Make a full copy of the current scene.

bpy.ops.scene.new_sequencer(*, type='NEW')

Add new scene by type in the sequence editor and assign to active strip

PARAMETERS:

type (enum in ['NEW', 'EMPTY', 'LINK_COPY', 'FULL_COPY'], (optional)) –

Type

- **NEW** New – Add a new, empty scene with default settings.
- **EMPTY** Copy Settings – Add a new, empty scene, and copy settings from the current scene.
- **LINK_COPY** Linked Copy – Link in the collections from the current scene (shallow copy).
- **FULL_COPY** Full Copy – Make a full copy of the current scene.

bpy.ops.scene.render_view_add()

Add a render view

`bpy.ops.scene.render_view_remove()`

Remove the selected render view

`bpy.ops.scene.view_layer_add(*, type='NEW')`

Add a view layer

PARAMETERS:

type (*enum in ['NEW', 'COPY', 'EMPTY'], (optional)*) –

Type

- **NEW** New – Add a new view layer.
- **COPY** Copy Settings – Copy settings of current view layer.
- **EMPTY** Blank – Add a new view layer with all collections disabled.

`bpy.ops.scene.view_layer_add_aov()`

Add a Shader AOV

`bpy.ops.scene.view_layer_add_lightgroup(*, name='')`

Add a Light Group

PARAMETERS:

name (*string, (optional, never None)*) – Name, Name of newly created lightgroup

`bpy.ops.scene.view_layer_add_used_lightgroups()`

Add all used Light Groups

`bpy.ops.scene.view_layer_remove()`

Remove the selected view layer

`bpy.ops.scene.view_layer_remove_aov()`

Remove Active AOV

`bpy.ops.scene.view_layer_remove_lightgroup()`

Remove Active Lightgroup

`bpy.ops.scene.view_layer_remove_unused_lightgroups()`

Remove all unused Light Groups