Skip to content

Mask Operators

bpy.ops.mask.**add_feather_vertex(*, location=(0.0, 0.0))**

Add vertex to feather

PARAMETERS:

location (`mathutils.Vector` of 2 items in [-inf, inf], (optional)) – Location, Location of vertex in normalized space

bpy.ops.mask.**add_feather_vertex_slide(*, MASK_OT_add_feather_vertex=None, MASK_OT_slide_point=None)**

Add new vertex to feather and slide it

PARAMETERS:

- **MASK_OT_add_feather_vertex** (`MASK_OT_add_feather_vertex`, (optional)) – Add Feather Vertex, Add vertex to feather
- **MASK_OT_slide_point** (`MASK_OT_slide_point`, (optional)) – Slide Point, Slide control points

bpy.ops.mask.**add_vertex(*, location=(0.0, 0.0))**

Add vertex to active spline

PARAMETERS:

location (`mathutils.Vector` of 2 items in [-inf, inf], (optional)) – Location, Location of vertex in normalized space

bpy.ops.mask.**add_vertex_slide(*, MASK_OT_add_vertex=None, MASK_OT_slide_point=None)**

Add new vertex and slide it

PARAMETERS:

- **MASK_OT_add_vertex** (`MASK_OT_add_vertex`, (optional)) – Add Vertex, Add vertex to active spline
- **MASK_OT_slide_point** (`MASK_OT_slide_point`, (optional)) – Slide Point, Slide control points

bpy.ops.mask.**copy_splines()**

Copy the selected splines to the internal clipboard

bpy.ops.mask.**cyclic_toggle()**

Toggle cyclic for selected splines

bpy.ops.mask.**delete(*, confirm=True)**

Delete selected control points or splines

PARAMETERS:

confirm (*boolean, (optional)*) – Confirm, Prompt for confirmation

bpy.ops.mask.**duplicate()**

Duplicate selected control points and segments between them

bpy.ops.mask.**duplicate_move(*, MASK_OT_duplicate=None, TRANSFORM_OT_translate=None)**

Duplicate mask and move

PARAMETERS:

- **MASK_OT_duplicate** (`MASK_OT_duplicate`, (optional)) – Duplicate Mask, Duplicate selected control points and segments betwee them
- **TRANSFORM_OT_translate** (`TRANSFORM_OT_translate`, (optional)) – Move, Move selected items

bpy.ops.mask.**feather_weight_clear()**

Reset the feather weight to zero

bpy.ops.mask.**handle_type_set(*, type='AUTO')**

bpy.ops.mask.[illegible]

Set type of handles for selected control points

PARAMETERS:

type (*enum in ['AUTO', 'VECTOR', 'ALIGNED', 'ALIGNED_DOUBLESIDE', 'FREE'], (optional)*) – Type, Spline type

bpy.ops.mask.**hide_view_clear(*, select=True)**

Reveal temporarily hidden mask layers

PARAMETERS:

select (*boolean, (optional)*) – Select

bpy.ops.mask.**hide_view_set(*, unselected=False)**

Temporarily hide mask layers

PARAMETERS:

unselected (*boolean, (optional)*) – Unselected, Hide unselected rather than selected layers

bpy.ops.mask.**layer_move(*, direction='UP')**

Move the active layer up/down in the list

PARAMETERS:

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

bpy.ops.mask.**layer_new(*, name='')**

Add new mask layer for masking

PARAMETERS:

name (*string, (optional, never None)*) – Name, Name of new mask layer

bpy.ops.mask.**layer_remove()**

Remove mask layer

bpy.ops.mask.**new(*, name='')**

Create new mask

PARAMETERS:

name (*string, (optional, never None)*) – Name, Name of new mask

bpy.ops.mask.**normals_make_consistent()**

Recalculate the direction of selected handles

bpy.ops.mask.**parent_clear()**

Clear the mask's parenting

bpy.ops.mask.**parent_set()**

Set the mask's parenting

bpy.ops.mask.**paste_splines()**

Paste splines from the internal clipboard

bpy.ops.mask.**primitive_circle_add(*, size=100.0, location=(0.0, 0.0))**

Add new circle-shaped spline

PARAMETERS:

- **size** (*float in [-inf, inf], (optional)*) – Size, Size of new circle
- **location** (`mathutils.Vector` of 2 items in [-inf, inf], (optional)) – Location, Location of new circle

bpy.ops.mask.**primitive_square_add(*, size=100.0, location=(0.0, 0.0))**

Add new square-shaped spline

PARAMETERS:

- **size** (*float in [-inf, inf], (optional)*) – Size, Size of new circle
- **location** (`mathutils.Vector` of 2 items in [-inf, inf], (optional)) – Location, Location of new circle

bpy.ops.mask.**select(*, extend=False, deselect=False, toggle=False, deselect_all=False, select_passthrough=False, location=(0.0, 0.0))**

Select spline points

PARAMETERS:

- **extend** (*boolean, (optional)*) – Extend, Extend selection instead of deselecting everything first
- **deselect** (*boolean, (optional)*) – Deselect, Remove from selection
- **toggle** (*boolean, (optional)*) – Toggle Selection, Toggle the selection
- **deselect_all** (*boolean, (optional)*) – Deselect On Nothing, Deselect all when nothing under the cursor
- **select_passthrough** (*boolean, (optional)*) – Only Select Unselected, Ignore the select action when the element is already selected
- **location** (`mathutils.Vector` of 2 items in [-inf, inf], (optional)) – Location, Location of vertex in normalized space

bpy.ops.mask.**select_all(*, action='TOGGLE')**

Change selection of all curve points

PARAMETERS:

action (*enum in ['TOGGLE', 'SELECT', 'DESELECT', 'INVERT'], (optional)*) –

Action, Selection action to execute

- `TOGGLE` Toggle – Toggle selection for all elements.
- `SELECT` Select – Select all elements.
- `DESELECT` Deselect – Deselect all elements.
- `INVERT` Invert – Invert selection of all elements.

bpy.ops.mask.**select_box(*, xmin=0, xmax=0, ymin=0, ymax=0, wait_for_input=True, mode='SET')**

Select curve points using box selection

PARAMETERS:

- **xmin** (*int in [-inf, inf], (optional)*) – X Min
- **xmax** (*int in [-inf, inf], (optional)*) – X Max
- **ymin** (*int in [-inf, inf], (optional)*) – Y Min
- **ymax** (*int in [-inf, inf], (optional)*) – Y Max
- **wait_for_input** (*boolean, (optional)*) – Wait for Input
- **mode** (*enum in ['SET', 'ADD', 'SUB'], (optional)*) –

  Mode

  - `SET` Set – Set a new selection.
  - `ADD` Extend – Extend existing selection.
  - `SUB` Subtract – Subtract existing selection.

bpy.ops.mask.**select_circle(*, x=0, y=0, radius=25, wait_for_input=True, mode='SET')**

Select curve points using circle selection

PARAMETERS:

- **x** (*int in [-inf, inf], (optional)*) – X
- **y** (*int in [-inf, inf], (optional)*) – Y
- **radius** (*int in [1, inf], (optional)*) – Radius
- **wait_for_input** (*boolean, (optional)*) – Wait for Input

- **wait_for_input** (*boolean, (optional)*) – Wait for Input
- **mode** (*enum in ['SET', 'ADD', 'SUB'], (optional)*) –
  Mode
  - `SET` Set – Set a new selection.
  - `ADD` Extend – Extend existing selection.
  - `SUB` Subtract – Subtract existing selection.

bpy.ops.mask.**select_lasso(*, path=None, use_smooth_stroke=False, smooth_stroke_factor=0.75, smooth_stroke_radius=35, mode='SET')**

Select curve points using lasso selection

PARAMETERS:

- **path** (`bpy_prop_collection` of `OperatorMousePath`, (optional)) – Path
- **use_smooth_stroke** (*boolean, (optional)*) – Stabilize Stroke, Selection lags behind mouse and follows a smoother path
- **smooth_stroke_factor** (*float in [0.5, 0.99], (optional)*) – Smooth Stroke Factor, Higher values gives a smoother stroke
- **smooth_stroke_radius** (*int in [10, 200], (optional)*) – Smooth Stroke Radius, Minimum distance from last point before selection continues
- **mode** (*enum in ['SET', 'ADD', 'SUB'], (optional)*) –
  Mode
  - `SET` Set – Set a new selection.
  - `ADD` Extend – Extend existing selection.
  - `SUB` Subtract – Subtract existing selection.

bpy.ops.mask.**select_less()**

Deselect spline points at the boundary of each selection region

bpy.ops.mask.**select_linked()**

Select all curve points linked to already selected ones

bpy.ops.mask.**select_linked_pick(*, deselect=False)**

(De)select all points linked to the curve under the mouse cursor

PARAMETERS:

deselect (*boolean, (optional)*) – Deselect

bpy.ops.mask.**select_more()**

Select more spline points connected to initial selection

bpy.ops.mask.**shape_key_clear()**

Remove mask shape keyframe for active mask layer at the current frame

bpy.ops.mask.**shape_key_feather_reset()**

Reset feather weights on all selected points animation values

bpy.ops.mask.**shape_key_insert()**

Insert mask shape keyframe for active mask layer at the current frame

bpy.ops.mask.**shape_key_rekey(*, location=True, feather=True)**

Recalculate animation data on selected points for frames selected in the dopesheet

PARAMETERS:

- **location** (*boolean, (optional)*) – Location
- **feather** (*boolean, (optional)*) – Feather

bpy.ops.mask.**slide_point(*, slide_feather=False, is_new_point=False)**

Slide control points

PARAMETERS:

- **slide_feather** (*boolean, (optional)*) – Slide Feather, First try to slide feather instead of vertex
- **is_new_point** (*boolean, (optional)*) – Slide New Point, Newly created vertex is being slid

bpy.ops.mask.**slide_spline_curvature()**

Slide a point on the spline to define its curvature

bpy.ops.mask.**switch_direction()**

Switch direction of selected splines