BevelModifier(Modifier)

base classes — bpy_struct, Modifier

class bpy.types.**BevelModifier(Modifier)**

Bevel modifier to make edges and vertices more rounded

affect

Affect edges or vertices

- VERTICES Vertices – Affect only vertices.
- EDGES Edges – Affect only edges.

TYPE:

enum in ['VERTICES', 'EDGES'], default 'EDGES'

angle_limit

Angle above which to bevel edges

TYPE:

float in [0, 3.14159], default 0.523599

custom_profile

The path for the custom profile

TYPE:

CurveProfile, (readonly)

edge_weight

Attribute name for edge weight

TYPE:

string, default '', (never None)

face_strength_mode

Whether to set face strength, and which faces to set it on

- FSTR_NONE None – Do not set face strength.
- FSTR_NEW New – Set face strength on new faces only.
- FSTR_AFFECTED Affected – Set face strength on new and affected faces only.
- FSTR_ALL All – Set face strength on all faces.

TYPE:

enum in ['FSTR_NONE', 'FSTR_NEW', 'FSTR_AFFECTED', 'FSTR_ALL'], default 'FSTR_NONE'

harden_normals

Match normals of new faces to adjacent faces

TYPE:

boolean, default False

invert_vertex_group

Invert vertex group influence

TYPE:

boolean, default False

limit_method

- `NONE` None – Bevel the entire mesh by a constant amount.
- `ANGLE` Angle – Only bevel edges with sharp enough angles between faces.
- `WEIGHT` Weight – Use bevel weights to determine how much bevel is applied in edge mode.
- `VGROUP` Vertex Group – Use vertex group weights to select whether vertex or edge is beveled.

TYPE:

enum in [‘NONE’, ‘ANGLE’, ‘WEIGHT’, ‘VGROUP’], default ‘ANGLE’

loop_slide

Prefer sliding along edges to having even widths

TYPE:

boolean, default True

mark_seam

Mark Seams along beveled edges

TYPE:

boolean, default False

mark_sharp

Mark beveled edges as sharp

TYPE:

boolean, default False

material

Material index of generated faces, -1 for automatic

TYPE:

int in [-1, 32767], default -1

miter_inner

Pattern to use for inside of miters

- `MITER_SHARP` Sharp – Inside of miter is sharp.
- `MITER_ARC` Arc – Inside of miter is arc.

TYPE:

enum in [‘MITER_SHARP’, ‘MITER_ARC’], default ‘MITER_SHARP’

miter_outer

Pattern to use for outside of miters

- `MITER_SHARP` Sharp – Outside of miter is sharp.
- `MITER_PATCH` Patch – Outside of miter is squared-off patch.
- `MITER_ARC` Arc – Outside of miter is arc.

TYPE:

enum in [‘MITER_SHARP’, ‘MITER_PATCH’, ‘MITER_ARC’], default ‘MITER_SHARP’

offset_type

What distance Width measures

- `OFFSET` Offset – Amount is offset of new edges from original.

- `WIDTH` Width – Amount is width of new face.
- `DEPTH` Depth – Amount is perpendicular distance from original edge to bevel face.
- `PERCENT` Percent – Amount is percent of adjacent edge length.
- `ABSOLUTE` Absolute – Amount is absolute distance along adjacent edge.

TYPE:

enum in ['OFFSET', 'WIDTH', 'DEPTH', 'PERCENT', 'ABSOLUTE'], default 'OFFSET'

profile

The profile shape (0.5 = round)

TYPE:

float in [0, 1], default 0.5

profile_type

The type of shape used to rebuild a beveled section

- `SUPERELLIPSE` Superellipse – The profile can be a concave or convex curve.
- `CUSTOM` Custom – The profile can be any arbitrary path between its endpoints.

TYPE:

enum in ['SUPERELLIPSE', 'CUSTOM'], default 'SUPERELLIPSE'

segments

Number of segments for round edges/verts

TYPE:

int in [1, 1000], default 1

spread

Spread distance for inner miter arcs

TYPE:

float in [0, inf], default 0.1

use_clamp_overlap

Clamp the width to avoid overlap

TYPE:

boolean, default True

vertex_group

Vertex group name

TYPE:

string, default "", (never None)

vertex_weight

Attribute name for vertex weight

TYPE:

string, default "", (never None)

vmesh_method

The method to use to create the mesh at intersections

- `ADJ` Grid Fill – Default patterned fill.
- `CUTOFF` Cutoff – A cut-off at the end of each profile before the intersection.

TYPE:
enum in ['ADJ', 'CUTOFF'], default 'ADJ'

width
Bevel amount

TYPE:
float in [0, inf], default 0.1

width_pct
Bevel amount for percentage method

TYPE:
float in [0, inf], default 0.1

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

Inherited Properties

- `bpy_struct.id_data`
- `Modifier.name`
- `Modifier.type`
- `Modifier.show_viewport`
- `Modifier.show_render`
- `Modifier.show_in_editmode`
- `Modifier.show_on_cage`
- `Modifier.show_expanded`
- `Modifier.is_active`
- `Modifier.use_pin_to_last`
- `Modifier.is_override_data`
- `Modifier.use_apply_on_spline`
- `Modifier.execution_time`
- `Modifier.persistent_uid`

Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.id_properties_clear`
- `bpy_struct.id_properties_ensure`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.pop`

- 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.property_overridable_library_set
- bpy_struct.property_unset
- bpy_struct.type_recast
- bpy_struct.values
- Modifier.bl_rna_get_subclass
- Modifier.bl_rna_get_subclass_py

Previous
BakeSettings(bpy_struct)
Report issue on this page

Ne
BezierSplinePoint(bpy_stru