

---

# **Blender Index**

*Release 2.60.0 - API*

**Blender Foundation**

November 03, 2011



# CONTENTS



Welcome, this document is an API reference for Blender 2.60.0. built Unknown.

A PDF version of this document is also available



# BLENDER/PYTHON DOCUMENTATION

## 1.1 Quickstart Introduction

### 1.1.1 Intro

This API is generally stable but some areas are still being added and improved.

The Blender/Python API can do the following:

- Edit any data the user interface can (Scenes, Meshes, Particles etc.)
- Modify user preferences, keymaps and themes
- Run tools with own settings
- Create user interface elements such as menus, headers and panels
- Create new tools
- Create interactive tools
- Create new rendering engines that integrate with Blender
- Define new settings in existing Blender data
- Draw in the 3D view using OpenGL commands from Python

The Blender/Python API **can't** (yet)...

- Create new space types.
- Assign custom properties to every type.
- Define callbacks or listeners to be notified when data is changed.

### 1.1.2 Before Starting

This document isn't intended to fully cover each topic. Rather, its purpose is to familiarize you with Blender 2.5's new Python API.

A quick list of helpful things to know before starting:

- Blender uses Python 3.x; some 3rd party extensions are not available yet.
- The interactive console in Blender 2.5 has been improved; testing one-liners in the console is a good way to learn.
- Button tool tips show Python attributes and operator names.

- Right clicking on buttons and menu items directly links to API documentation.
- For more examples, the text menu has a templates section where some example operators can be found.
- To examine further scripts distributed with Blender, see `~/ .blender/scripts/startup/bl_ui` for the user interface and `~/ .blender/scripts/startup/bl_op` for operators.

### 1.1.3 Key Concepts

#### Data Access

##### Accessing datablocks

Python accesses Blender's data in the same way as the animation system and user interface; this implies that any setting that can be changed via a button can also be changed from Python.

Accessing data from the currently loaded blend file is done with the module `bpy.data`. This gives access to library data. For example:

```
>>> bpy.data.objects
<bpy_collection[3], BlendDataObjects>

>>> bpy.data.scenes
<bpy_collection[1], BlendDataScenes>

>>> bpy.data.materials
<bpy_collection[1], BlendDataMaterials>
```

##### About Collections

You'll notice that an index as well as a string can be used to access members of the collection.

Unlike Python's dictionaries, both methods are acceptable; however, the index of a member may change while running Blender.

```
>>> list(bpy.data.objects)
[bpy.data.objects["Cube"], bpy.data.objects["Plane"]]

>>> bpy.data.objects['Cube']
bpy.data.objects["Cube"]

>>> bpy.data.objects[0]
bpy.data.objects["Cube"]
```

##### Accessing attributes

Once you have a data block, such as a material, object, groups etc., its attributes can be accessed much like you would change a setting using the graphical interface. In fact, the tooltip for each button also displays the Python attribute which can help in finding what settings to change in a script.

```
>>> bpy.data.objects[0].name
'Camera'
```



```
>>> bpy.data.scenes["Scene"]
bpy.data.scenes['Scene']

>>> bpy.data.materials.new("MyMaterial")
bpy.data.materials['MyMaterial']
```

For testing what data to access it's useful to use the “Console”, which is its own space type in Blender 2.5. This supports auto-complete, giving you a fast way to dig into different data in your file.

Example of a data path that can be quickly found via the console:

```
>>> bpy.data.scenes[0].render.resolution_percentage
100
>>> bpy.data.scenes[0].objects["Torus"].data.vertices[0].co.x
1.0
```

## Custom Properties

Python can access properties on any datablock that has an ID (data that can be linked in and accessed from `bpy.data`). When assigning a property, you can make up your own names, these will be created when needed or overwritten if they exist.

This data is saved with the blend file and copied with objects.

Example:

```
bpy.context.object["MyOwnProperty"] = 42

if "SomeProp" in bpy.context.object:
    print("Property found")

# Use the get function like a python dictionary
# which can have a fallback value.
value = bpy.data.scenes["Scene"].get("test_prop", "fallback value")

# dictionaries can be assigned as long as they only use basic types.
group = bpy.data.groups.new("MyTestGroup")
group["GameSettings"] = {"foo": 10, "bar": "spam", "baz": {}}

del group["GameSettings"]
```

Note that these properties can only be assigned basic Python types.

- int, float, string
- array of ints/floats
- dictionary (only string keys are supported, values must be basic types too)

These properties are valid outside of Python. They can be animated by curves or used in driver paths.

## Context

While it's useful to be able to access data directly by name or as a list, it's more common to operate on the user's selection. The context is always available from “`bpy.context`” and can be used to get the active object, scene, tool settings along with many other attributes.

Common-use cases:

```
>>> bpy.context.object
>>> bpy.context.selected_objects
>>> bpy.context.visible_bones
```

Note that the context is read-only. These values cannot be modified directly, though they may be changed by running API functions or by using the data API.

So `bpy.context.object = obj` will raise an error.

But `bpy.context.scene.objects.active = obj` will work as expected.

The context attributes change depending on where they are accessed. The 3D view has different context members than the console, so take care when accessing context attributes that the user state is known.

See [bpy.context](#) API reference

## Operators (Tools)

Operators are tools generally accessed by the user from buttons, menu items or key shortcuts. From the user perspective they are a tool but Python can run these with its own settings through the `bpy.ops` module.

Examples:

```
>>> bpy.ops.mesh.flip_normals()
{'FINISHED'}
>>> bpy.ops.mesh.hide(unselected=False)
{'FINISHED'}
>>> bpy.ops.object.scale_apply()
{'FINISHED'}
```

---

**Note:** The menu item: Help -> Operator Cheat Sheet” gives a list of all operators and their default values in Python syntax, along with the generated docs. This is a good way to get an overview of all blender’s operators.

---

### Operator Poll()

Many operators have a “poll” function which may check that the mouse is a valid area or that the object is in the correct mode (Edit Mode, Weight Paint etc). When an operator’s poll function fails within python, an exception is raised.

For example, calling `bpy.ops.view3d.render_border()` from the console raises the following error:

```
RuntimeError: Operator bpy.ops.view3d.render_border.poll() failed, context is incorrect
```

In this case the context must be the 3d view with an active camera.

To avoid using try/except clauses wherever operators are called you can call the operators own `.poll()` function to check if it can run in the current context.

```
if bpy.ops.view3d.render_border.poll():
    bpy.ops.view3d.render_border()
```

### 1.1.4 Integration

Python scripts can integrate with Blender in the following ways:

- By defining a rendering engine.

- By defining operators.
- By defining menus, headers and panels.
- By inserting new buttons into existing menus, headers and panels

In Python, this is done by defining a class, which is a subclass of an existing type.

## Example Operator

```
import bpy

def main(context):
    for ob in context.scene.objects:
        print(ob)

class SimpleOperator(bpy.types.Operator):
    '''Tooltip'''
    bl_idname = "object.simple_operator"
    bl_label = "Simple Object Operator"

    @classmethod
    def poll(cls, context):
        return context.active_object is not None

    def execute(self, context):
        main(context)
        return {'FINISHED'}

def register():
    bpy.utils.register_class(SimpleOperator)

def unregister():
    bpy.utils.unregister_class(SimpleOperator)

if __name__ == "__main__":
    register()

    # test call
    bpy.ops.object.simple_operator()
```

Once this script runs, `SimpleOperator` is registered with Blender and can be called from the operator search popup or added to the toolbar.

To run the script:

1. Highlight the above code then press Ctrl+C to copy it.
2. Start Blender
3. Press Ctrl+Right twice to change to the Scripting layout.
4. Click the button labeled `New` and the confirmation pop up in order to create a new text block.
5. Press Ctrl+V to paste the code into the text panel (the upper left frame).

6. Click on the button **Run Script**.
7. Move your mouse into the 3D view, press spacebar for the operator search menu, and type “Simple”.
8. Click on the “Simple Operator” item found in search.

### See Also:

The class members with the **bl\_** prefix are documented in the API reference `bpy.types.Operator`

---

**Note:** The output from the `main` function is sent to the terminal; in order to see this, be sure to *use the terminal*.

---

## Example Panel

Panels register themselves as a class, like an operator. Notice the extra **bl\_** variables used to set the context they display in.

```
import bpy

class HelloWorldPanel(bpy.types.Panel):
    bl_label = "Hello World Panel"
    bl_idname = "OBJECT_PT_hello"
    bl_space_type = "PROPERTIES"
    bl_region_type = "WINDOW"
    bl_context = "object"

    def draw(self, context):
        layout = self.layout

        obj = context.object

        row = layout.row()
        row.label(text="Hello world!", icon='WORLD_DATA')

        row = layout.row()
        row.label(text="Active object is: " + obj.name)
        row = layout.row()
        row.prop(obj, "name")

def register():
    bpy.utils.register_class(HelloWorldPanel)

def unregister():
    bpy.utils.unregister_class(HelloWorldPanel)

if __name__ == "__main__":
    register()
```

To run the script:

1. Highlight the above code then press Ctrl+C to copy it
2. Start Blender
3. Press Ctrl+Right twice to change to the Scripting layout

4. Click the button labeled `New` and the confirmation pop up in order to create a new text block.
5. Press `Ctrl+V` to paste the code into the text panel (the upper left frame)
6. Click on the button **Run Script**.

To view the results:

1. Select the the default cube.
2. Click on the Object properties icon in the buttons panel (far right; appears as a tiny cube).
3. Scroll down to see a panel named **Hello World Panel**.
4. Changing the object name also updates **Hello World Panel's** Name: field.

Note the row distribution and the label and properties that are available through the code.

#### See Also:

`bpy.types.Panel`

## 1.1.5 Types

Blender defines a number of Python types but also uses Python native types.

Blender's Python API can be split up into 3 categories.

### Native Types

In simple cases returning a number or a string as a custom type would be cumbersome, so these are accessed as normal python types.

- blender float/int/boolean -> float/int/boolean
- blender enumerator -> string

```
>>> C.object.rotation_mode = 'AXIS_ANGLE'
```

- blender enumerator (multiple) -> set of strings

```
# setting multiple camera overlay guides
bpy.context.scene.camera.data.show_guide = {'GOLDEN', 'CENTER'}
```

```
# passing as an operator argument for report types
self.report({'WARNING', 'INFO'}, "Some message!")
```

### Internal Types

Used for Blender datablocks and collections: `bpy.types.bpy_struct`

For data that contains its own attributes groups/meshes/bones/scenes... etc.

There are 2 main types that wrap Blenders data, one for datablocks (known internally as `bpy_struct`), another for properties.

```
>>> bpy.context.object
bpy.data.objects['Cube']
```

```
>>> C.scene.objects
bpy.data.scenes['Scene'].objects
```

Note that these types reference Blender's data so modifying them is immediately visible.

### Mathutils Types

Used for vectors, quaternion, eulers, matrix and color types, accessible from `mathutils`

Some attributes such as `bpy.types.Object.location`, `bpy.types.PoseBone.rotation_euler` and `bpy.types.Scene.cursor_location` can be accessed as special math types which can be used together and manipulated in various useful ways.

Example of a matrix, vector multiplication:

```
bpy.context.object.matrix_world * bpy.context.object.data.verts[0].co
```

---

**Note:** mathutils types keep a reference to Blender's internal data so changes can be applied back.

Example:

```
# modifies the Z axis in place.
bpy.context.object.location.z += 2.0

# location variable holds a reference to the object too.
location = bpy.context.object.location
location *= 2.0

# Copying the value drops the reference so the value can be passed to
# functions and modified without unwanted side effects.
location = bpy.context.object.location.copy()
```

---

### 1.1.6 Animation

There are 2 ways to add keyframes through Python.

The first is through key properties directly, which is similar to inserting a keyframe from the button as a user. You can also manually create the curves and keyframe data, then set the path to the property. Here are examples of both methods.

Both examples insert a keyframe on the active object's Z axis.

Simple example:

```
obj = bpy.context.object
obj.location[2] = 0.0
obj.keyframe_insert(data_path="location", frame=10.0, index=2)
obj.location[2] = 1.0
obj.keyframe_insert(data_path="location", frame=20.0, index=2)
```

Using Low-Level Functions:

```
obj = bpy.context.object
obj.animation_data_create()
obj.animation_data.action = bpy.data.actions.new(name="MyAction")
fcu_z = obj.animation_data.action.fcurves.new(data_path="location", index=2)
fcu_z.keyframe_points.add(2)
fcu_z.keyframe_points[0].co = 10.0, 0.0
fcu_z.keyframe_points[1].co = 20.0, 1.0
```

## 1.2 Python API Overview

This document is to give an understanding of how python and blender fit together, covering some of the functionality that isn't obvious from reading the API reference and example scripts.

### 1.2.1 Python in Blender

Blender embeds a python interpreter which is started with blender and stays active. This interpreter runs scripts to draw the user interface and is used for some of Blender's internal tools too.

This is a typical python environment so tutorials on how to write python scripts will work running the scripts in blender too. Blender provides the `bpy` module to the python interpreter. This module can be imported in a script and gives access to blender data, classes, and functions. Scripts that deal with blender data will need to import this module.

Here is a simple example of moving a vertex of the object named **Cube**:

```
import bpy
bpy.data.objects["Cube"].data.vertices[0].co.x += 1.0
```

This modifies Blender's internal data directly. When you run this in the interactive console you will see the 3D viewport update.

### 1.2.2 The Default Environment

When developing your own scripts it may help to understand how blender sets up its python environment. Many python scripts come bundled with blender and can be used as a reference because they use the same API that script authors write tools in. Typical usage for scripts include: user interface, import/export, scene manipulation, automation, defining your own toolset and customization.

On startup blender scans the `scripts/startup/` directory for python modules and imports them. The exact location of this directory depends on your installation. [See the directory layout docs](#)

### 1.2.3 Script Loading

This may seem obvious but it's important to note the difference between executing a script directly or importing it as a module.

Scripts that extend blender - define classes that exist beyond the scripts execution, this makes future access to these classes (to unregister for example) more difficult than importing as a module where class instance is kept in the module and can be accessed by importing that module later on.

For this reason it's preferable to only use directly execute scripts that don't extend blender by registering classes.

Here are some ways to run scripts directly in blender.

- Loaded in the text editor and press **Run Script**.
- Typed or pasted into the interactive console.
- Execute a python file from the command line with blender, eg:

```
blender --python /home/me/my_script.py
```

To run as modules:

- The obvious way, `import some_module` command from the text window or interactive console.
- Open as a text block and tick "Register" option, this will load with the blend file.

- copy into one of the directories `scripts/startup`, where they will be automatically imported on startup.
- define as an addon, enabling the addon will load it as a python module.

### Addons

Some of blenders functionality is best kept optional, alongside scripts loaded at startup we have addons which are kept in their own directory `scripts/addons`, and only load on startup if selected from the user preferences.

The only difference between addons and built-in python modules is that addons must contain a `bl_info` variable which blender uses to read metadata such as name, author, category and URL.

The user preferences addon listing uses `bl_info` to display information about each addon.

See [Addons](#) for details on the `bl_info` dictionary.

## 1.2.4 Integration through Classes

Running python scripts in the text editor is useful for testing but you'll want to extend blender to make tools accessible like other built-in functionality.

The blender python api allows integration for:

- `bpy.types.Panel`
- `bpy.types.Menu`
- `bpy.types.Operator`
- `bpy.types.PropertyGroup`
- `bpy.types.KeyingSet`
- `bpy.types.RenderEngine`

This is intentionally limited. Currently, for more advanced features such as mesh modifiers, object types, or shader nodes, C/C++ must be used.

For python intergration Blender defines methods which are common to all types. This works by creating a python subclass of a Blender class which contains variables and functions specified by the parent class which are pre-defined to interface with Blender.

For example:

```
import bpy
class SimpleOperator(bpy.types.Operator):
    bl_idname = "object.simple_operator"
    bl_label = "Tool Name"

    def execute(self, context):
        print("Hello World")
        return {'FINISHED' }
```

```
bpy.utils.register_class(SimpleOperator)
```

First note that we subclass a member of `bpy.types`, this is common for all classes which can be integrated with blender and used so we know if this is an Operator and not a Panel when registering.

Both class properties start with a `bl_` prefix. This is a convention used to distinguish blender properties from those you add yourself.



Next see the execute function, which takes an instance of the operator and the current context. A common prefix is not used for functions.

Lastly the register function is called, this takes the class and loads it into blender. See [Class Registration](#).

Regarding inheritance, blender doesn't impose restrictions on the kinds of class inheritance used, the registration checks will use attributes and functions defined in parent classes.

class mix-in example:

```
import bpy
class BaseOperator:
    def execute(self, context):
        print("Hello World BaseClass")
        return {'FINISHED'}

class SimpleOperator(bpy.types.Operator, BaseOperator):
    bl_idname = "object.simple_operator"
    bl_label = "Tool Name"

bpy.utils.register_class(SimpleOperator)
```

Notice these classes don't define an `__init__(self)` function. While `__init__()` and `__del__()` will be called if defined, the class instances lifetime only spans the execution. So a panel for example will have a new instance for every redraw, for this reason there is rarely a cause to store variables in the panel instance. Instead, persistent variables should be stored in Blenders data so that the state can be restored when blender is restarted.

---

**Note:** Modal operators are an exception, keeping their instance variable as blender runs, see modal operator template.

---

So once the class is registered with blender, instancing the class and calling the functions is left up to blender. In fact you cannot instance these classes from the script as you would expect with most python API's.

To run operators you can call them through the operator api, eg:

```
import bpy
bpy.ops.object.simple_operator()
```

User interface classes are given a context in which to draw, buttons window, file header, toolbar etc, then they are drawn when that area is displayed so they are never called by python scripts directly.

## 1.2.5 Registration

### Module Registration

Blender modules loaded at startup require `register()` and `unregister()` functions. These are the *only* functions that blender calls from your code, which is otherwise a regular python module.

A simple blender/python module can look like this:

```
import bpy

class SimpleOperator(bpy.types.Operator):
    """ See example above """

def register():
    bpy.utils.register_class(SimpleOperator)

def unregister():
```

```
bpy.utils.unregister_class(SimpleOperator)

if __name__ == "__main__":
    register()
```

These functions usually appear at the bottom of the script containing class registration sometimes adding menu items. You can also use them for internal purposes setting up data for your own tools but take care since register won't re-run when a new blend file is loaded.

The register/unregister calls are used so it's possible to toggle addons and reload scripts while blender runs. If the register calls were placed in the body of the script, registration would be called on import, meaning there would be no distinction between importing a module or loading its classes into blender.

This becomes problematic when a script imports classes from another module making it difficult to manage which classes are being loaded and when.

The last 2 lines are only for testing:

```
if __name__ == "__main__":
    register()
```

This allows the script to be run directly in the text editor to test changes. This `register()` call won't run when the script is imported as a module since `__main__` is reserved for direct execution.

### Class Registration

Registering a class with blender results in the class definition being loaded into blender, where it becomes available alongside existing functionality.

Once this class is loaded you can access it from `bpy.types`, using the `bl_idname` rather than the classes original name.

When loading a class, blender performs sanity checks making sure all required properties and functions are found, that properties have the correct type, and that functions have the right number of arguments.

Mostly you will not need concern yourself with this but if there is a problem with the class definition it will be raised on registering:

Using the function arguments `def execute(self, context, spam)`, will raise an exception:

```
ValueError: expected Operator, SimpleOperator class "execute" function to
have 2 args, found 3
```

Using `bl_idname = 1` will raise.

```
TypeError: validating class error: Operator.bl_idname expected a string
type, not int
```

### Multiple-Classes

Loading classes into blender is described above, for simple cases calling `bpy.utils.register_class` (Some-Class) is sufficient, but when there are many classes or a packages submodule has its own classes it can be tedious to list them all for registration.

For more convenient loading/unloading `bpy.utils.register_module` (module) and `bpy.utils.unregister_module` (module) functions exist.

A script which defines many of its own operators, panels menus etc. you only need to write:

```
def register():
    bpy.utils.register_module(__name__)

def unregister():
    bpy.utils.unregister_module(__name__)
```

Internally blender collects subclasses on registrable types, storing them by the module in which they are defined. By passing the module name to `bpy.utils.register_module` blender can register all classes created by this module and its submodules.

## Inter Classes Dependencies

When customizing blender you may want to group your own settings together, after all, they will likely have to co-exist with other scripts. To group these properties classes need to be defined, for groups within groups or collections within groups you can find yourself having to deal with order of registration/unregistration.

Custom properties groups are themselves classes which need to be registered.

Say you want to store material settings for a custom engine.

```
# Create new property
# bpy.data.materials[0].my_custom_props.my_float
import bpy

class MyMaterialProps(bpy.types.PropertyGroup):
    my_float = bpy.props.FloatProperty()

def register():
    bpy.utils.register_class(MyMaterialProps)
    bpy.types.Material.my_custom_props = bpy.props.PointerProperty(type=MyMaterialProps)

def unregister():
    del bpy.types.Material.my_custom_props
    bpy.utils.unregister_class(MyMaterialProps)

if __name__ == "__main__":
    register()
```

---

**Note:** *The class must be registered before being used in a property, failing to do so will raise an error:*

```
ValueError: bpy_struct "Material" registration error: my_custom_props could not register
```

---

```
# Create new property group with a sub property
# bpy.data.materials[0].my_custom_props.sub_group.my_float
import bpy

class MyMaterialSubProps(bpy.types.PropertyGroup):
    my_float = bpy.props.FloatProperty()

class MyMaterialGroupProps(bpy.types.PropertyGroup):
    sub_group = bpy.props.PointerProperty(type=MyMaterialSubProps)

def register():
    bpy.utils.register_class(MyMaterialSubProps)
    bpy.utils.register_class(MyMaterialGroupProps)
```

```
bpy.types.Material.my_custom_props = bpy.props.PointerProperty(type=MyMaterialGroupProps)

def unregister():
    del bpy.types.Material.my_custom_props
    bpy.utils.unregister_class(MyMaterialGroupProps)
    bpy.utils.unregister_class(MyMaterialSubProps)

if __name__ == "__main__":
    register()
```

---

**Note:** *The lower most class needs to be registered first and that unregister() is a mirror of register()*

---

### Manipulating Classes

Properties can be added and removed as blender runs, normally happens on register or unregister but for some special cases it may be useful to modify types as the script runs.

For example:

```
# add a new property to an existing type
bpy.types.Object.my_float = bpy.props.FloatProperty()
# remove
del bpy.types.Object.my_float
```

This works just as well for PropertyGroup subclasses you define yourself.

```
class MyPropGroup(bpy.types.PropertyGroup):
    pass
MyPropGroup.my_float = bpy.props.FloatProperty()
```

...this is equivalent to:

```
class MyPropGroup(bpy.types.PropertyGroup):
    my_float = bpy.props.FloatProperty()
```

### Dynamic Defined-Classes (Advanced)

In some cases the specifier for data may not be in blender, renderman shader definitions for example and it may be useful to define types and remove them on the fly.

```
for i in range(10):
    idname = "object.operator_%d" % i

    def func(self, context):
        print("Hello World", self.bl_idname)
        return {'FINISHED'}

    opclass = type("DynOp%d" % i,
                  (bpy.types.Operator, ),
                  {"bl_idname": idname, "bl_label": "Test", "execute": func},
                  )
    bpy.utils.register_class(opclass)
```

**Note:** Notice `type()` is called to define the class. This is an alternative syntax for class creation in python, better suited to constructing classes dynamically.

Calling these operators:

```
>>> bpy.ops.object.operator_1()
Hello World OBJECT_OT_operator_1
{'FINISHED'}
```

```
>>> bpy.ops.object.operator_2()
Hello World OBJECT_OT_operator_2
{'FINISHED'}
```

## 1.3 Best Practice

When writing you're own scripts python is great for new developers to pick up and become productive, but you can also pick up odd habits or at least write scripts that are not easy for others to understand.

For you're own work this is of course fine, but if you want to collaborate with others or have you're work included with blender there are practices we encourage.

### 1.3.1 Style Conventions

For Blender 2.5 we have chosen to follow python suggested style guide to avoid mixing styles amongst our own scripts and make it easier to use python scripts from other projects.

Using our style guide for your own scripts makes it easier if you eventually want to contribute them to blender.

This style guide is known as pep8 and can be found [here](#)

A brief listing of pep8 criteria.

- camel caps for class names: `MyClass`
- all lower case underscore separated module names: `my_module`
- indentation of 4 spaces (no tabs)
- spaces around operators. `1 + 1`, not `1+1`
- only use explicit imports, (no importing `*`)
- don't use single line: `if val: body`, separate onto 2 lines instead.

As well as pep8 we have other conventions used for blender python scripts.

- Use single quotes for enums, and double quotes for strings.

Both are of course strings but in our internal API enums are unique items from a limited set. eg.

```
bpy.context.scene.render.file_format = 'PNG'
bpy.context.scene.render.filepath = "//render_out"
```

- pep8 also defines that lines should not exceed 79 characters, we felt this is too restrictive so this is optional per script.

Periodically we run checks for pep8 compliance on blender scripts, for scripts to be included in this check add this line as a comment at the top of the script.

```
# <pep8 compliant>
```

To enable line length checks use this instead.

```
# <pep8-80 compliant>
```

## 1.3.2 User Interface Layout

TODO: Thomas

## 1.3.3 Script Efficiency

### List Manipulation (General Python Tips)

#### Searching for list items

In Python there are some handy list functions that save you having to search through the list.

Even though you're not looping on the list data **python is**, so you need to be aware of functions that will slow down your script by searching the whole list.

```
my_list.count(list_item)
my_list.index(list_item)
my_list.remove(list_item)
if list_item in my_list: ...
```

#### Modifying Lists

In python we can add and remove from a list, This is slower when the list length is modifier, especially at the start of the list, since all the data after the index of modification needs to be moved up or down 1 place.

The most simple way to add onto the end of the list is to use `my_list.append(list_item)` or `my_list.extend(some_list)` and the fastest way to remove an item is `my_list.pop()` or `del my_list[-1]`.

To use an index you can use `my_list.insert(index, list_item)` or `list.pop(index)` for list removal, but these are slower.

Sometimes its faster (but more memory hungry) to just rebuild the list.

Say you want to remove all triangle faces in a list.

Rather than...

```
faces = mesh.faces[:] # make a list copy of the meshes faces
f_idx = len(faces)    # Loop backwards
while f_idx:         # while the value is not 0
    f_idx -= 1

    if len(faces[f_idx].vertices) == 3:
        faces.pop(f_idx) # remove the triangle
```

It's faster to build a new list with list comprehension.

```
faces = [f for f in mesh.faces if len(f.vertices) != 3]
```

## Adding List Items

If you have a list that you want to add onto another list, rather than...

```
for l in some_list:
    my_list.append(l)
```

Use...

```
my_list.extend([a, b, c...])
```

Note that insert can be used when needed, but it is slower than append especially when inserting at the start of a long list.

This example shows a very sub-optimal way of making a reversed list.

```
reverse_list = []
for list_item in some_list:
    reverse_list.insert(0, list_item)
```

## Removing List Items

Use `my_list.pop(index)` rather than `my_list.remove(list_item)`

This requires you to have the index of the list item but is faster since `remove()` will search the list.

Here is an example of how to remove items in 1 loop, removing the last items first, which is faster (as explained above).

```
list_index = len(my_list)

while list_index:
    list_index -= 1
    if my_list[list_index].some_test_attribute == 1:
        my_list.pop(list_index)
```

This example shows a fast way of removing items, for use in cases where you can alter the list order without breaking the scripts functionality. This works by swapping 2 list items, so the item you remove is always last.

```
pop_index = 5

# swap so the pop_index is last.
my_list[-1], my_list[pop_index] = my_list[pop_index], my_list[-1]

# remove last item (pop_index)
my_list.pop()
```

When removing many items in a large list this can provide a good speedup.

## Avoid Copying Lists

When passing a list/dictionary to a function, it is faster to have the function modify the list rather than returning a new list so python doesn't have to duplicate the list in memory.

Functions that modify a list in-place are more efficient than functions that create new lists.

This is generally slower so only use for functions when it makes sense not to modify the list in place.

```
>>> my_list = some_list_func(my_list)
```

This is generally faster since there is no re-assignment and no list duplication.

```
>>> some_list_func(vec)
```

Also note that passing a sliced list makes a copy of the list in python memory

```
>>> foobar(my_list[:])
```

If my\_list was a large array containing 10000's of items, a copy could use a lot of extra memory.

### Writing Strings to a File (Python General)

Here are 3 ways of joining multiple strings into 1 string for writing

This really applies to any area of your code that involves a lot of string joining.

Python's string addition, *don't use if you can help it, especially when writing data in a loop.*

```
>>> file.write(str1 + " " + str2 + " " + str3 + "\n")
```

String formatting. Use this when you're writing string data from floats and int's

```
>>> file.write("%s %s %s\n" % (str1, str2, str3))
```

Python's string joining function. To join a list of strings

```
>>> file.write(" ".join([str1, str2, str3, "\n"]))
```

join is fastest on many strings, string formatting is quite fast too (better for converting data types). String arithmetic is slowest.

### Parsing Strings (Import/Exporting)

Since many file formats are ASCII, the way you parse/export strings can make a large difference in how fast your script runs.

When importing strings to make into blender there are a few ways to parse the string.

#### Parsing Numbers

Use `float(string)` rather than `eval(string)`, if you know the value will be an int then `int(string)`, `float()` will work for an int too but its faster to read ints with `int()`.

#### Checking String Start/End

If your checking the start of a string for a keyword, rather than...

```
>>> if line[0:5] == "vert ": ...
```

Use...

```
>>> if line.startswith("vert "):
```



Using `startswith()` is slightly faster (approx 5%) and also avoids a possible error with the slice length not matching the string length.

`my_string.endswith("foo_bar")` can be used for line endings too.

if your unsure whether the text is upper or lower case use lower or upper string function.

```
>>> if line.lower().startswith("vert ")
```

## Use try/except Sparingly

The `try` statement useful to save time writing error checking code.

However `try` is significantly slower then an `if` since an exception has to be set each time, so avoid using `try` in areas of your code that execute in a loop and runs many times.

There are cases where using `try` is faster than checking weather the condition will raise an error, so it is worth experimenting.

## Value Comparison

Python has two ways to compare values `a == b` and `a is b`, The difference is that `==` may run the objects comparison function `__cmp__()` where as `is` compares identity, that both variables reference the same item in memory.

In cases where you know you are checking for the same value which is referenced from multiple places, `is` is faster.

## Time You're Code

While developing a script its good to time it to be aware of any changes in performance, this can be done simply.

```
import time
time_start = time.time()

# do something...

print("My Script Finished: %.4f sec" % time.time() - time_start)
```

# 1.4 Tips and Tricks

Here are various suggestions that you might find useful when writing scripts.

Some of these are just python features that scripters may not have thought to use with blender, others are blender specific.

## 1.4.1 Use The Terminal

When writing python scripts, it's useful to have a terminal open, this is not the built-in python console but a terminal application which is used to start blender.

There are 3 main uses for the terminal, these are:

- You can see the output of `print()` as you're script runs, which is useful to view debug info.
- The error trace-back is printed in full to the terminal which won't always generate an error popup in blender's user interface (depending on how the script is executed).

- If the script runs for too long or you accidentally enter an infinite loop, Ctrl+C in the terminal (Ctrl+Break on Windows) will quit the script early.

---

**Note:** For Linux and OSX users this means starting the terminal first, then running blender from within it. On Windows the terminal can be enabled from the help menu.

---

### 1.4.2 Show All Operators

While blender logs operators in the Info space, this only reports operators with the REGISTER option enabled so as not to flood the Info view with calls to `bpy.ops.view3d.smoothview` and `bpy.ops.view3d.zoom`.

However, for testing it can be useful to see **every** operator called in a terminal, do this by enabling the debug option either by passing the `--debug` argument when starting blender or by setting `bpy.app.debug` to True while blender is running.

### 1.4.3 Use an External Editor

Blenders text editor is fine for small changes and writing tests but its not full featured, for larger projects you'll probably want to use a standalone editor or python IDE.

Editing a text file externally and having the same text open in blender does work but isn't that optimal so here are 2 ways you can easily use an external file from blender.

Using the following examples you'll still need textblock in blender to execute, but reference an external file rather than including it directly.

#### Executing External Scripts

This is the equivalent to running the script directly, referencing a scripts path from a 2 line textblock.

```
filename = "/full/path/to/myscript.py"
exec(compile(open(filename).read(), filename, 'exec'))
```

You might want to reference a script relative to the blend file.

```
import bpy
import os
```

```
filename = os.path.join(os.path.basename(bpy.data.filepath), "myscript.py")
exec(compile(open(filename).read(), filename, 'exec'))
```

#### Executing Modules

This example shows loading a script in as a module and executing a module function.

```
import myscript
import imp

imp.reload(myscript)
myscript.main()
```

Notice that the script is reloaded every time, this forces use of the modified version, otherwise the cached one in `sys.modules` would be used until blender was restarted.

The important difference between this and executing the script directly is it has to call a function in the module, in this case `main()` but it can be any function, an advantage with this is you can pass arguments to the function from this small script which is often useful for testing different settings quickly.

The other issue with this is the script has to be in python's module search path. While this is not best practice - for testing you can extend the search path, this example adds the current blend files directory to the search path, then loads the script as a module.

```
import sys
import os
import bpy

blend_dir = os.path.basename(bpy.data.filepath)
if blend_dir not in sys.path:
    sys.path.append(blend_dir)

import myscript
import imp
imp.reload(myscript)
myscript.main()
```

### 1.4.4 Don't Use Blender!

While developing your own scripts blenders interface can get in the way, manually reloading, running the scripts, opening file import etc. adds overhead.

For scripts that are not interactive it can end up being more efficient not to use blenders interface at all and instead execute the script on the command line.

```
blender --background --python myscript.py
```

You might want to run this with a blend file so the script has some data to operate on.

```
blender myscene.blend --background --python myscript.py
```

---

**Note:** Depending on your setup you might have to enter the full path to the blender executable.

---

Once the script is running properly in background mode, you'll want to check the output of the script, this depends completely on the task at hand however here are some suggestions.

- render the output to an image, use an image viewer and keep writing over the same image each time.
- save a new blend file, or export the file using one of blenders exporters.
- if the results can be displayed as text - print them or write them to a file.

This can take a little time to setup, but it can be well worth the effort to reduce the time it takes to test changes - you can even have blender running the script ever few seconds with a viewer updating the results, so no need to leave you're text editor to see changes.

### 1.4.5 Use External Tools

When there are no readily available python modules to perform specific tasks it's worth keeping in mind you may be able to have python execute an external command on you're data and read the result back in.

Using external programs adds an extra dependency and may limit who can use the script but to quickly setup you're own custom pipeline or writing one-off scripts this can be handy.

Examples include:

- Run The Gimp in batch mode to execute custom scripts for advanced image processing.
- Write out 3D models to use external mesh manipulation tools and read back in the results.
- Convert files into recognizable formats before reading.

### 1.4.6 Bundled Python & Extensions

The Blender releases distributed from blender.org include a complete python installation on all platforms, this has the disadvantage that any extensions you have installed in you're systems python wont be found by blender.

There are 2 ways around this:

- remove blender python sub-directory, blender will then fallback on the systems python and use that instead **python version must match the one that blender comes with.**
- copy the extensions into blender's python sub-directory so blender can access them, you could also copy the entire python installation into blenders sub-directory, replacing the one blender comes with. This works as long as the python versions match and the paths are created in the same relative locations. Doing this has the advantage that you can redistribute this bundle to others with blender and/or the game player, including any extensions you rely on.

### 1.4.7 Drop Into a Python Interpreter in You're Script

In the middle of a script you may want to inspect some variables, run some function and generally dig about to see whats going on.

```
import code
code.interact(local=locals())
```

If you want to access both global and local variables do this...

```
import code
namespace = globals().copy()
namespace.update(locals())
code.interact(local=namespace)
```

The next example is an equivalent single line version of the script above which is easier to paste into you're code:

```
__import__('code').interact(local={k: v for ns in (globals(), locals()) for k, v in ns.items()})
```

`code.interact` can be added at any line in the script and will pause the script an launch an interactive interpreter in the terminal, when you're done you can quit the interpreter and the script will continue execution.

Admittedly this highlights the lack of any python debugging support built into blender, but its still handy to know.

---

**Note:** This works in the game engine as well, it can be handy to inspect the state of a running game.

---

## 1.4.8 Advanced

### Blender as a module

From a python perspective it's nicer to have everything as an extension which lets the python script combine many components.

Advantages include:

- you can use external editors/IDE's with blenders python API and execute scripts within the IDE (step over code, inspect variables as the script runs).
- editors/IDE's can auto complete blender modules & variables.
- existing scripts can import blender API's without having to run inside blender.

This is marked advanced because to run blender as a python module requires a special build option.

For instructions on building see [Building blender as a python module](#)

### Python Safety (Build Option)

Since it's possible to access data which has been removed (see Gotcha's), this can be hard to track down the cause of crashes.

To raise python exceptions on accessing freed data (rather than crashing), enable the CMake build option `WITH_PYTHON_SAFETY`.

This enables data tracking which makes data access about 2x slower which is why the option is not enabled in release builds.

## 1.5 Gotchas

This document attempts to help you work with the Blender API in areas that can be troublesome and avoid practices that are known to give instability.

### 1.5.1 Using Operators

Blender's operators are tools for users to access, that python can access them too is very useful nevertheless operators have limitations that can make them cumbersome to script.

Main limits are...

- Can't pass data such as objects, meshes or materials to operate on (operators use the context instead)
- The return value from calling an operator gives the success (if it finished or was canceled), in some cases it would be more logical from an API perspective to return the result of the operation.
- Operators poll function can fail where an API function would raise an exception giving details on exactly why.

#### Why does an operator's poll fail?

When calling an operator gives an error like this:

```
>>> bpy.ops.action.clean(threshold=0.001)
RuntimeError: Operator bpy.ops.action.clean.poll() failed, context is incorrect
```

Which raises the question as to what the correct context might be?

Typically operators check for the active area type, a selection or active object they can operate on, but some operators are more picky about when they run.

In most cases you can figure out what context an operator needs simply by seeing how it's used in Blender and thinking about what it does.

Unfortunately if you're still stuck - the only way to **really** know what's going on is to read the source code for the poll function and see what it's checking.

For python operators it's not so hard to find the source since it's included with Blender and the source file/line is included in the operator reference docs.

Downloading and searching the C code isn't so simple, especially if you're not familiar with the C language but by searching the operator name or description you should be able to find the poll function with no knowledge of C.

---

**Note:** Blender does have the functionality for poll functions to describe why they fail, but it's currently not used much, if you're interested to help improve our API feel free to add calls to `CTX_wm_operator_poll_msg_set` where it's not obvious why poll fails.

```
>>> bpy.ops.gpencil.draw()
RuntimeError: Operator bpy.ops.gpencil.draw.poll() Failed to find Grease Pencil data to draw into
```

---

### The operator still doesn't work!

Certain operators in Blender are only intended for use in a specific context, some operators for example are only called from the properties window where they check the current material, modifier or constraint.

Examples of this are:

- `bpy.ops.texture.slot_move`
- `bpy.ops.constraint.limitdistance_reset`
- `bpy.ops.object.modifier_copy`
- `bpy.ops.buttons.file_browse`

Another possibility is that you are the first person to attempt to use this operator in a script and some modifications need to be made to the operator to run in a different context, if the operator should logically be able to run but fails when accessed from a script it should be reported to the bug tracker.

## 1.5.2 Stale Data

### No updates after setting values

Sometimes you want to modify values from python and immediately access the updated values, eg:

Once changing the objects `bpy.types.Object.location` you may want to access its transformation right after from `bpy.types.Object.matrix_world`, but this doesn't work as you might expect.

Consider the calculations that might go into working out the object's final transformation, this includes:

- animation function curves.
- drivers and their python expressions.
- constraints

- parent objects and all of their f-curves, constraints etc.

To avoid expensive recalculations every time a property is modified, Blender defers making the actual calculations until they are needed.

However, while the script runs you may want to access the updated values.

This can be done by calling `bpy.types.Scene.update` after modifying values which recalculates all data that is tagged to be updated.

### Can I redraw during the script?

The official answer to this is no, or... *“You don’t want to do that”*.

To give some background on the topic...

While a script executes Blender waits for it to finish and is effectively locked until its done, while in this state Blender won’t redraw or respond to user input. Normally this is not such a problem because scripts distributed with Blender tend not to run for an extended period of time, nevertheless scripts *can* take ages to execute and its nice to see whats going on in the view port.

Tools that lock Blender in a loop and redraw are highly discouraged since they conflict with Blenders ability to run multiple operators at once and update different parts of the interface as the tool runs.

So the solution here is to write a **modal** operator, that is - an operator which defines a modal() function, See the modal operator template in the text editor.

Modal operators execute on user input or setup their own timers to run frequently, they can handle the events or pass through to be handled by the keymap or other modal operators.

Transform, Painting, Fly-Mode and File-Select are example of a modal operators.

Writing modal operators takes more effort than a simple `for` loop that happens to redraw but is more flexible and integrates better with Blenders design.

#### Ok, Ok! I still want to draw from python

If you insist - yes its possible, but scripts that use this hack wont be considered for inclusion in Blender and any issues with using it wont be considered bugs, this is also not guaranteed to work in future releases.

```
bpy.ops.wm.redraw_timer(type='DRAW_WIN_SWAP', iterations=1)
```

### 1.5.3 Matrix multiplication is wrong

Every so often users complain that Blenders matrix math is wrong, the confusion comes from mathutils matrices being column-major to match OpenGL and the rest of Blenders matrix operations and stored matrix data.

This is different to **numpy** which is row-major which matches what you would expect when using conventional matrix math notation.

### 1.5.4 I can’t edit the mesh in edit-mode!

Blender’s EditMesh is an internal data structure (not saved and not exposed to python), this gives the main annoyance that you need to exit edit-mode to edit the mesh from python.

The reason we have not made much attempt to fix this yet is because we will likely move to BMesh mesh API eventually, so any work on the API now will be wasted effort.

With the BMesh API we may expose mesh data to python so we can write useful tools in python which are also fast to execute while in edit-mode.

For the time being this limitation just has to be worked around but we're aware its frustrating needs to be addressed.

### 1.5.5 EditBones, PoseBones, Bone... Bones

Armature Bones in Blender have three distinct data structures that contain them. If you are accessing the bones through one of them, you may not have access to the properties you really need.

---

**Note:** In the following examples `bpy.context.object` is assumed to be an armature object.

---

#### Edit Bones

`bpy.context.object.data.edit_bones` contains a editbones; to access them you must set the armature mode to edit mode first (editbones do not exist in object or pose mode). Use these to create new bones, set their head/tail or roll, change their parenting relationships to other bones, etc.

Example using `bpy.types.EditBone` in armature editmode:

This is only possible in edit mode.

```
>>> bpy.context.object.data.edit_bones["Bone"].head = Vector((1.0, 2.0, 3.0))
```

This will be empty outside of editmode.

```
>>> mybones = bpy.context.selected_editable_bones
```

Returns an editbone only in edit mode.

```
>>> bpy.context.active_bone
```

#### Bones (Object Mode)

`bpy.context.object.data.bones` contains bones. These *live* in object mode, and have various properties you can change, note that the head and tail properties are read-only.

Example using `bpy.types.Bone` in object or pose mode:

Returns a bone (not an editbone) outside of edit mode

```
>>> bpy.context.active_bone
```

This works, as with blender the setting can be edited in any mode

```
>>> bpy.context.object.data.bones["Bone"].use_deform = True
```

Accessible but read-only

```
>>> tail = myobj.data.bones["Bone"].tail
```



## Pose Bones

`bpy.context.object.pose.bones` contains pose bones. This is where animation data resides, i.e. animatable transformations are applied to pose bones, as are constraints and ik-settings.

Examples using `bpy.types.PoseBone` in object or pose mode:

```
# Gets the name of the first constraint (if it exists)
bpy.context.object.pose.bones["Bone"].constraints[0].name

# Gets the last selected pose bone (pose mode only)
bpy.context.active_pose_bone
```

---

**Note:** Notice the pose is accessed from the object rather than the object data, this is why blender can have 2 or more objects sharing the same armature in different poses.

---

**Note:** Strictly speaking `PoseBone`'s are not bones, they are just the state of the armature, stored in the `bpy.types.Object` rather than the `bpy.types.Armature`, the real bones are however accessible from the pose bones - `bpy.types.PoseBone.bone`

---

## Armature Mode Switching

While writing scripts that deal with armatures you may find you have to switch between modes, when doing so take care when switching out of editmode not to keep references to the edit-bones or their head/tail vectors. Further access to these will crash blender so its important the script clearly separates sections of the code which operate in different modes.

This is mainly an issue with editmode since pose data can be manipulated without having to be in pose mode, however for operator access you may still need to enter pose mode.

### 1.5.6 Relative File Paths

Blenders relative file paths are not compatible with standard python modules such as `sys` and `os`.

Built in python functions don't understand blenders `//` prefix which denotes the blend file path.

A common case where you would run into this problem is when exporting a material with associated image paths.

```
>>> bpy.path.abspath(image.filepath)
```

When using blender data from linked libraries there is an unfortunate complication since the path will be relative to the library rather than the open blend file. When the data block may be from an external blend file pass the library argument from the `bpy.types.ID`.

```
>>> bpy.path.abspath(image.filepath, library=image.library)
```

These returns the absolute path which can be used with native python modules.

### 1.5.7 Unicode Problems

Python supports many different encodings so there is nothing stopping you from writing a script in latin1 or iso-8859-15.

See [pep-0263](#)

However this complicates things for the python api because blend files themselves don't have an encoding.

To simplify the problem for python integration and script authors we have decided all strings in blend files **must** be UTF-8 or ASCII compatible.

This means assigning strings with different encodings to an object names for instance will raise an error.

Paths are an exception to this rule since we cannot ignore the existance of non-utf-8 paths on peoples filesystems.

This means seemingly harmless expressions can raise errors, eg.

```
>>> print(bpy.data.filepath)
UnicodeEncodeError: 'ascii' codec can't encode characters in position 10-21: ordinal not in range(128)

>>> bpy.context.object.name = bpy.data.filepath
Traceback (most recent call last):
  File "<blender_console>", line 1, in <module>
TypeError: bpy_struct: item.attr= val: Object.name expected a string type, not str
```

Here are 2 ways around filesystem encoding issues:

```
>>> print(repr(bpy.data.filepath))

>>> import os
>>> filepath_bytes = os.fsencode(bpy.data.filepath)
>>> filepath_utf8 = filepath_bytes.decode('utf-8', "replace")
>>> bpy.context.object.name = filepath_utf8
```

Unicode encoding/decoding is a big topic with comprehensive python documentation, to avoid getting stuck too deep in encoding problems - here are some suggestions:

- Always use utf-8 encoding or convert to utf-8 where the input is unknown.
- Avoid manipulating filepaths as strings directly, use `os.path` functions instead.
- Use `os.fsencode()` / `os.fsdecode()` rather than the built in string decoding functions when operating on paths.
- To print paths or to include them in the user interface use `repr(path)` first or `"%r" % path` with string formatting.
- **Possibly** - use bytes instead of python strings, when reading some input its less trouble to read it as binary data though you will still need to decide how to treat any strings you want to use with Blender, some importers do this.

### 1.5.8 Strange errors using 'threading' module

Python threading with Blender only works properly when the threads finish up before the script does. By using `threading.join()` for example.

Heres an example of threading supported by Blender:

```
import threading
import time

def prod():
    print(threading.current_thread().name, "Starting")

    # do something vaguely useful
```

```

import bpy
from mathutils import Vector
from random import random

prod_vec = Vector((random() - 0.5, random() - 0.5, random() - 0.5))
print("Prodding", prod_vec)
bpy.data.objects["Cube"].location += prod_vec
time.sleep(random() + 1.0)
# finish

print(threading.current_thread().name, "Exiting")

threads = [threading.Thread(name="Prod %d" % i, target=prod) for i in range(10)]

print("Starting threads...")

for t in threads:
    t.start()

print("Waiting for threads to finish...")

for t in threads:
    t.join()

```

This an example of a timer which runs many times a second and moves the default cube continuously while Blender runs (Unsupported).

```

def func():
    print("Running...")
    import bpy
    bpy.data.objects['Cube'].location.x += 0.05

def my_timer():
    from threading import Timer
    t = Timer(0.1, my_timer)
    t.start()
    func()

my_timer()

```

Use cases like the one above which leave the thread running once the script finishes may seem to work for a while but end up causing random crashes or errors in Blender's own drawing code.

So far, no work has gone into making Blender's python integration thread safe, so until its properly supported, best not make use of this.

---

**Note:** Python's threads only allow co-currency and won't speed up your scripts on multi-processor systems, the `subprocess` and `multiprocess` modules can be used with blender and make use of multiple CPU's too.

---

## 1.5.9 Help! My script crashes Blender

Ideally it would be impossible to crash Blender from python however there are some problems with the API where it can be made to crash.

Strictly speaking this is a bug in the API but fixing it would mean adding memory verification on every access since

most crashes are caused by the python objects referencing Blenders memory directly, whenever the memory is freed, further python access to it can crash the script. But fixing this would make the scripts run very slow, or writing a very different kind of API which doesn't reference the memory directly.

Here are some general hints to avoid running into these problems.

- Be aware of memory limits, especially when working with large lists since Blender can crash simply by running out of memory.
- Many hard to fix crashes end up being because of referencing freed data, when removing data be sure not to hold any references to it.
- Modules or classes that remain active while Blender is used, should not hold references to data the user may remove, instead, fetch data from the context each time the script is activated.
- Crashes may not happen every time, they may happen more on some configurations/operating-systems.

### Undo/Redo

Undo invalidates all `bpy.types.ID` instances (Object, Scene, Mesh etc).

This example shows how you can tell undo changes the memory locations.

```
>>> hash(bpy.context.object)
-9223372036849950810
>>> hash(bpy.context.object)
-9223372036849950810

# ... move the active object, then undo

>>> hash(bpy.context.object)
-9223372036849951740
```

As suggested above, simply not holding references to data when Blender is used interactively by the user is the only way to ensure the script doesn't become unstable.

### Edit Mode / Memory Access

Switching edit-mode `bpy.ops.object.mode_set(mode='EDIT')` / `bpy.ops.object.mode_set(mode='OBJECT')` will re-allocate objects data, any references to a meshes vertices/faces/uvs, armatures bones, curves points etc cannot be accessed after switching edit-mode.

Only the reference to the data its self can be re-accessed, the following example will crash.

```
mesh = bpy.context.active_object.data
faces = mesh.faces
bpy.ops.object.mode_set(mode='EDIT')
bpy.ops.object.mode_set(mode='OBJECT')

# this will crash
print(faces)
```

So after switching edit-mode you need to re-access any object data variables, the following example shows how to avoid the crash above.

```
mesh = bpy.context.active_object.data
faces = mesh.faces
bpy.ops.object.mode_set(mode='EDIT')
bpy.ops.object.mode_set(mode='OBJECT')
```

```
# faces have been re-allocated
faces = mesh.faces
print(faces)
```

These kinds of problems can happen for any functions which re-allocate the object data but are most common when switching edit-mode.

## Array Re-Allocation

When adding new points to a curve or vertices's/edges/faces to a mesh, internally the array which stores this data is re-allocated.

```
bpy.ops.curve.primitive_bezier_curve_add()
point = bpy.context.object.data.splines[0].bezier_points[0]
bpy.context.object.data.splines[0].bezier_points.add()

# this will crash!
point.co = 1.0, 2.0, 3.0
```

This can be avoided by re-assigning the point variables after adding the new one or by storing indices's to the points rather than the points themselves.

The best way is to sidestep the problem altogether add all the points to the curve at once. This means you don't have to worry about array re-allocation and its faster too since reallocating the entire array for every point added is inefficient.

## Removing Data

**Any** data that you remove shouldn't be modified or accessed afterwards, this includes f-curves, drivers, render layers, timeline markers, modifiers, constraints along with objects, scenes, groups, bones.. etc.

This is a problem in the API at the moment that we should eventually solve.



# APPLICATION MODULES

## 2.1 Context Access (bpy.context)

The context members available depend on the area of blender which is currently being accessed.

Note that all context values are readonly, but may be modified through the data api or by running operators

### 2.1.1 Screen Context

`bpy.context.scene`

**Type** `bpy.types.Scene`

`bpy.context.visible_objects`

**Type** sequence of `bpy.types.Object`

`bpy.context.visible_bases`

**Type** sequence of `bpy.types.ObjectBase`

`bpy.context.selectable_objects`

**Type** sequence of `bpy.types.Object`

`bpy.context.selectable_bases`

**Type** sequence of `bpy.types.ObjectBase`

`bpy.context.selected_objects`

**Type** sequence of `bpy.types.Object`

`bpy.context.selected_bases`

**Type** sequence of `bpy.types.ObjectBase`

`bpy.context.selected_editable_objects`

**Type** sequence of `bpy.types.Object`

`bpy.context.selected_editable_bases`

**Type** sequence of `bpy.types.ObjectBase`

`bpy.context.visible_bones`

**Type** sequence of `bpy.types.Object`

`bpy.context.editable_bones`

**Type** sequence of `bpy.types.EditBone`  
`bpy.context.selected_bones`

**Type** sequence of `bpy.types.Bone`  
`bpy.context.selected_editable_bones`

**Type** sequence of `bpy.types.Bone`  
`bpy.context.visible_pose_bones`

**Type** sequence of `bpy.types.PoseBone`  
`bpy.context.selected_pose_bones`

**Type** sequence of `bpy.types.PoseBone`  
`bpy.context.active_bone`

**Type** `bpy.types.Bone`  
`bpy.context.active_pose_bone`

**Type** `bpy.types.PoseBone`  
`bpy.context.active_base`

**Type** `bpy.types.ObjectBase`  
`bpy.context.active_object`

**Type** `bpy.types.Object`  
`bpy.context.object`

**Type** `bpy.types.Object`  
`bpy.context.edit_object`

**Type** `bpy.types.Object`  
`bpy.context.sculpt_object`

**Type** `bpy.types.Object`  
`bpy.context.vertex_paint_object`

**Type** `bpy.types.Object`  
`bpy.context.weight_paint_object`

**Type** `bpy.types.Object`  
`bpy.context.image_paint_object`

**Type** `bpy.types.Object`  
`bpy.context.particle_edit_object`

**Type** `bpy.types.Object`  
`bpy.context.sequences`

**Type** sequence of `bpy.types.Sequence`  
`bpy.context.selected_sequences`

**Type** sequence of `bpy.types.Sequence`  
`bpy.context.selected_editable_sequences`



**Type** sequence of `bpy.types.Sequence`

## 2.1.2 View3D Context

`bpy.context.selected_objects`

**Type** sequence of `bpy.types.Object`

`bpy.context.selected_bases`

**Type** sequence of `bpy.types.ObjectBase`

`bpy.context.selected_editable_objects`

**Type** sequence of `bpy.types.Object`

`bpy.context.selected_editable_bases`

**Type** sequence of `bpy.types.ObjectBase`

`bpy.context.visible_objects`

**Type** sequence of `bpy.types.Object`

`bpy.context.visible_bases`

**Type** sequence of `bpy.types.ObjectBase`

`bpy.context.selectable_objects`

**Type** sequence of `bpy.types.Object`

`bpy.context.selectable_bases`

**Type** sequence of `bpy.types.ObjectBase`

`bpy.context.active_base`

**Type** `bpy.types.ObjectBase`

`bpy.context.active_object`

**Type** `bpy.types.Object`

## 2.1.3 Buttons Context

`bpy.context.world`

**Type** `bpy.types.World`

`bpy.context.object`

**Type** `bpy.types.Object`

`bpy.context.mesh`

**Type** `bpy.types.Mesh`

`bpy.context.armature`

**Type** `bpy.types.Armature`

`bpy.context.lattice`

**Type** `bpy.types.Lattice`

`bpy.context.curve`

**Type** `bpy.types.Curve`  
`bpy.context.meta_ball`  
**Type** `bpy.types.MetaBall`  
`bpy.context.lamp`  
**Type** `bpy.types.Lamp`  
`bpy.context.speaker`  
**Type** `bpy.types.Speaker`  
`bpy.context.camera`  
**Type** `bpy.types.Camera`  
`bpy.context.material`  
**Type** `bpy.types.Material`  
`bpy.context.material_slot`  
**Type** `bpy.types.MaterialSlot`  
`bpy.context.texture`  
**Type** `bpy.types.Texture`  
`bpy.context.texture_slot`  
**Type** `bpy.types.MaterialTextureSlot`  
`bpy.context.bone`  
**Type** `bpy.types.Bone`  
`bpy.context.edit_bone`  
**Type** `bpy.types.EditBone`  
`bpy.context.pose_bone`  
**Type** `bpy.types.PoseBone`  
`bpy.context.particle_system`  
**Type** `bpy.types.ParticleSystem`  
`bpy.context.particle_system_editable`  
**Type** `bpy.types.ParticleSystem`  
`bpy.context.cloth`  
**Type** `bpy.types.ClothModifier`  
`bpy.context.soft_body`  
**Type** `bpy.types.SoftBodyModifier`  
`bpy.context.fluid`  
**Type** `bpy.types.FluidSimulationModifier`  
`bpy.context.smoke`  
**Type** `bpy.types.SmokeModifier`  
`bpy.context.collision`

**Type** `bpy.types.CollisionModifier`

`bpy.context.brush`

**Type** `bpy.types.Brush`

## 2.1.4 Image Context

`bpy.context.edit_image`

**Type** `bpy.types.Image`

## 2.1.5 Node Context

`bpy.context.selected_nodes`

**Type** sequence of `bpy.types.Node`

## 2.1.6 Text Context

`bpy.context.edit_text`

**Type** `bpy.types.Text`

## 2.2 Data Access (`bpy.data`)

This module is used for all blender/python access.

`bpy.data`

Access to blenders internal data

**Type** `bpy.types.BlendData`

```
import bpy
```

```
# print all objects
```

```
for obj in bpy.data.objects:
    print(obj.name)
```

```
# print all scene names in a list
```

```
print(bpy.data.scenes.keys())
```

```
# remove mesh Cube
```

```
if "Cube" in bpy.data.meshes:
    mesh = bpy.data.meshes["Cube"]
    print("removing mesh", mesh)
    bpy.data.meshes.remove(mesh)
```

```
# write images into a file next to the blend
```

```
import os
file = open(os.path.splitext(bpy.data.filepath)[0] + ".txt", 'w')
```

```
for image in bpy.data.images:
    file.write("%s %d x %d\n" % (image.filepath, image.size[0], image.size[1]))

file.close()
```

## 2.3 Operators (bpy.ops)

### 2.3.1 Calling Operators

Provides python access to calling operators, this includes operators written in C, Python or Macros.

Only keyword arguments can be used to pass operator properties.

Operators don't have return values as you might expect, instead they return a set() which is made up of: {'RUNNING\_MODAL', 'CANCELLED', 'FINISHED', 'PASS\_THROUGH'}. Common return values are {'FINISHED'} and {'CANCELLED'}.

Calling an operator in the wrong context will raise a RuntimeError, there is a poll() method to avoid this problem.

Note that the operator ID (bl\_idname) in this example is 'mesh.subdivide', 'bpy.ops' is just the access path for python.

```
import bpy

# calling an operator
bpy.ops.mesh.subdivide(number_cuts=3, smoothness=0.5)

# check poll() to avoid exception.
if bpy.ops.object.mode_set.poll():
    bpy.ops.object.mode_set(mode='EDIT')
```

### 2.3.2 Execution Context

When calling an operator you may want to pass the execution context.

This determines the context that's given to the operator to run in, and whether invoke() is called or execute().

'EXEC\_DEFAULT' is used by default but you may want the operator to take user interaction with 'INVOKE\_DEFAULT'.

The execution context is as a non keyword, string argument in: ('INVOKE\_DEFAULT', 'INVOKE\_REGION\_WIN', 'INVOKE\_REGION\_CHANNELS', 'INVOKE\_REGION\_PREVIEW', 'INVOKE\_AREA', 'INVOKE\_SCREEN', 'EXEC\_DEFAULT', 'EXEC\_REGION\_WIN', 'EXEC\_REGION\_CHANNELS', 'EXEC\_REGION\_PREVIEW', 'EXEC\_AREA', 'EXEC\_SCREEN')

```
# group add popup
import bpy
bpy.ops.object.group_instance_add('INVOKE_DEFAULT')
```

### Action Operators

```
bpy.ops.action.clean(threshold=0.001)
Simplify F-Curves by removing closely spaced keyframes
```

**Parameters** `threshold` (float in [0, inf], (optional)) – Threshold

`bpy.ops.action.clickselect` (*extend=False, column=False*)  
Select keyframes by clicking on them

**Parameters**

- **extend** (*boolean, (optional)*) – Extend Select
- **column** (*boolean, (optional)*) – Column Select

`bpy.ops.action.copy` ()  
Copy selected keyframes to the copy/paste buffer

`bpy.ops.action.delete` ()  
Remove all selected keyframes

`bpy.ops.action.duplicate` ()  
Make a copy of all selected keyframes

`bpy.ops.action.duplicate_move` (*ACTION\_OT\_duplicate=None, TRANSFORM\_OT\_transform=None*)  
Undocumented ([contribute](#))

**Parameters**

- **ACTION\_OT\_duplicate** (*ACTION\_OT\_duplicate, (optional)*) – Duplicate Keyframes, Make a copy of all selected keyframes
- **TRANSFORM\_OT\_transform** (*TRANSFORM\_OT\_transform, (optional)*) – Transform, Transform selected items by mode type

`bpy.ops.action.extrapolation_type` (*type='CONSTANT'*)  
Set extrapolation mode for selected F-Curves

**Parameters** **type** (*enum in ['CONSTANT', 'LINEAR', 'MAKE\_CYCLIC', 'CLEAR\_CYCLIC'], (optional)*) – Type

- **CONSTANT** Constant Extrapolation.
- **LINEAR** Linear Extrapolation.
- **MAKE\_CYCLIC** Make Cyclic (F-Modifier), Add Cycles F-Modifier if one doesn't exist already.
- **CLEAR\_CYCLIC** Clear Cyclic (F-Modifier), Remove Cycles F-Modifier if not needed anymore.

`bpy.ops.action.frame_jump` ()  
Set the current frame to the average frame of the selected keyframes

`bpy.ops.action.handle_type` (*type='FREE'*)  
Set type of handle for selected keyframes

**Parameters** **type** (*enum in ['FREE', 'VECTOR', 'ALIGNED', 'AUTO', 'AUTO\_CLAMPED'], (optional)*) – Type

- **FREE** Free.
- **VECTOR** Vector.
- **ALIGNED** Aligned.
- **AUTO** Automatic.
- **AUTO\_CLAMPED** Auto Clamped, Auto handles clamped to not overshoot.

`bpy.ops.action.interpolation_type` (*type='CONSTANT'*)  
Set interpolation mode for the F-Curve segments starting from the selected keyframes

**Parameters** **type** (*enum in ['CONSTANT', 'LINEAR', 'BEZIER'], (optional)*) – Type

`bpy.ops.action.keyframe_insert` (*type='ALL'*)  
Insert keyframes for the specified channels

**Parameters** **type** (*enum in ['ALL', 'SEL', 'GROUP'], (optional)*) – Type

`bpy.ops.action.keyframe_type` (*type='KEYFRAME'*)  
Set type of keyframe for the selected keyframes

**Parameters** **type** (*enum in ['KEYFRAME', 'BREAKDOWN', 'EXTREME', 'JITTER'], (optional)*)  
– Type

`bpy.ops.action.markers_make_local` ()  
Move selected scene markers to the active Action as local 'pose' markers

`bpy.ops.action.mirror` (*type='CFRA'*)  
Flip selected keyframes over the selected mirror line

**Parameters** **type** (*enum in ['CFRA', 'XAXIS', 'MARKER'], (optional)*) – Type

`bpy.ops.action.new` ()  
Create new action

`bpy.ops.action.paste` (*offset='START', merge='MIX'*)  
Paste keyframes from copy/paste buffer for the selected channels, starting on the current frame

**Parameters**

- **offset** (*enum in ['START', 'END', 'RELATIVE', 'NONE'], (optional)*) – Offset, Paste time offset of keys
  - `START` Frame Start, Paste keys starting at current frame.
  - `END` Frame End, Paste keys ending at current frame.
  - `RELATIVE` Frame Relative, Paste keys relative to the current frame when copying.
  - `NONE` No Offset, Paste keys from original time.
- **merge** (*enum in ['MIX', 'OVER\_ALL', 'OVER\_RANGE', 'OVER\_RANGE\_ALL'], (optional)*) – Type, Method of merging pasted keys and existing
  - `MIX` Mix, Overlay existing with new keys.
  - `OVER_ALL` Overwrite All, Replace all keys.
  - `OVER_RANGE` Overwrite Range, Overwrite keys in pasted range.
  - `OVER_RANGE_ALL` Overwrite Entire Range, Overwrite keys in pasted range, using the range of all copied keys.

`bpy.ops.action.previewrange_set` ()  
Set Preview Range based on extents of selected Keyframes

`bpy.ops.action.sample` ()  
Add keyframes on every frame between the selected keyframes

`bpy.ops.action.select_all_toggle` (*invert=False*)  
Toggle selection of all keyframes

**Parameters** **invert** (*boolean, (optional)*) – Invert

`bpy.ops.action.select_border` (*gesture\_mode=0, xmin=0, xmax=0, ymin=0, ymax=0, axis\_range=False*)  
Select all keyframes within the specified region

**Parameters**

- **gesture\_mode** (*int in [-inf, inf], (optional)*) – Gesture Mode
- **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
- **axis\_range** (*boolean, (optional)*) – Axis Range

`bpy.ops.action.select_column` (*mode='KEYS'*)  
Select all keyframes on the specified frame(s)

**Parameters mode** (*enum in ['KEYS', 'CFRA', 'MARKERS\_COLUMN', 'MARKERS\_BETWEEN'], (optional)*) – Mode

`bpy.ops.action.select_leftright` (*mode='CHECK', extend=False*)  
Select keyframes to the left or the right of the current frame

**Parameters**

- **mode** (*enum in ['CHECK', 'LEFT', 'RIGHT'], (optional)*) – Mode
- **extend** (*boolean, (optional)*) – Extend Select

`bpy.ops.action.select_less` ()  
Deselect keyframes on ends of selection islands

`bpy.ops.action.select_linked` ()  
Select keyframes occurring in the same F-Curves as selected ones

`bpy.ops.action.select_more` ()  
Select keyframes beside already selected ones

`bpy.ops.action.snap` (*type='CFRA'*)  
Snap selected keyframes to the times specified

**Parameters type** (*enum in ['CFRA', 'NEAREST\_FRAME', 'NEAREST\_SECOND', 'NEAREST\_MARKER'], (optional)*) – Type

`bpy.ops.action.view_all` ()  
Reset viewable area to show full keyframe range

`bpy.ops.action.view_selected` ()  
Reset viewable area to show selected keyframes range

**Anim Operators**

`bpy.ops.anim.change_frame` (*frame=0*)  
Interactively change the current frame number

**Parameters frame** (*int in [-300000, 300000], (optional)*) – Frame

`bpy.ops.anim.channels_click` (*extend=False, children\_only=False*)  
Handle mouse-clicks over animation channels

**Parameters**

- **extend** (*boolean, (optional)*) – Extend Select
- **children\_only** (*boolean, (optional)*) – Select Children Only

`bpy.ops.anim.channels_collapse` (*all=True*)

Collapse (i.e. close) all selected expandable animation channels

**Parameters** *all* (*boolean, (optional)*) – All, Collapse all channels (not just selected ones)

`bpy.ops.anim.channels_delete` ()

Delete all selected animation channels

`bpy.ops.anim.channels_editable_toggle` (*mode='TOGGLE', type='PROTECT'*)

Toggle editability of selected channels

**Parameters**

- **mode** (*enum in ['TOGGLE', 'DISABLE', 'ENABLE', 'INVERT'], (optional)*) – Mode
- **type** (*enum in ['PROTECT', 'MUTE'], (optional)*) – Type

`bpy.ops.anim.channels_expand` (*all=True*)

Expand (i.e. open) all selected expandable animation channels

**Parameters** *all* (*boolean, (optional)*) – All, Expand all channels (not just selected ones)

`bpy.ops.anim.channels_fcurves_enable` ()

Clears 'disabled' tag from all F-Curves to get broken F-Curves working again

`bpy.ops.anim.channels_move` (*direction='DOWN'*)

Rearrange selected animation channels

**Parameters** *direction* (*enum in ['TOP', 'UP', 'DOWN', 'BOTTOM'], (optional)*) – Direction

`bpy.ops.anim.channels_rename` ()

Rename animation channel under mouse

`bpy.ops.anim.channels_select_all_toggle` (*invert=False*)

Toggle selection of all animation channels

**Parameters** *invert* (*boolean, (optional)*) – Invert

`bpy.ops.anim.channels_select_border` (*gesture\_mode=0, xmin=0, xmax=0, ymin=0, ymax=0*)

Select all animation channels within the specified region

**Parameters**

- **gesture\_mode** (*int in [-inf, inf], (optional)*) – Gesture Mode
- **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

`bpy.ops.anim.channels_setting_disable` (*mode='DISABLE', type='PROTECT'*)

Disable specified setting on all selected animation channels

**Parameters**

- **mode** (*enum in ['TOGGLE', 'DISABLE', 'ENABLE', 'INVERT'], (optional)*) – Mode
- **type** (*enum in ['PROTECT', 'MUTE'], (optional)*) – Type

`bpy.ops.anim.channels_setting_enable` (*mode='ENABLE', type='PROTECT'*)

Enable specified setting on all selected animation channels

**Parameters**

- **mode** (*enum in ['TOGGLE', 'DISABLE', 'ENABLE', 'INVERT'], (optional)*) – Mode



- **type** (*enum in ['PROTECT', 'MUTE'], (optional)*) – Type

`bpy.ops.anim.channels_setting_toggle` (*mode='TOGGLE', type='PROTECT'*)  
Toggle specified setting on all selected animation channels

#### Parameters

- **mode** (*enum in ['TOGGLE', 'DISABLE', 'ENABLE', 'INVERT'], (optional)*) – Mode
- **type** (*enum in ['PROTECT', 'MUTE'], (optional)*) – Type

`bpy.ops.anim.channels_visibility_set` ()  
Make only the selected animation channels visible in the Graph Editor

`bpy.ops.anim.channels_visibility_toggle` ()  
Toggle visibility in Graph Editor of all selected animation channels

`bpy.ops.anim.clear_useless_actions` (*only\_unused=True*)  
Mark actions with no F-Curves for deletion after save+reload of file preserving “action libraries”

**Parameters** **only\_unused** (*boolean, (optional)*) – Only Unused, Only unused (Fake User only) actions get considered

**File** `startup/bl_operators/anim.py:244`

`bpy.ops.anim.copy_driver_button` ()  
Copy the driver for the highlighted button

`bpy.ops.anim.driver_button_add` (*all=True*)  
Add driver(s) for the property(s) connected represented by the highlighted button

**Parameters** **all** (*boolean, (optional)*) – All, Create drivers for all elements of the array

`bpy.ops.anim.driver_button_remove` (*all=True*)  
Remove the driver(s) for the property(s) connected represented by the highlighted button

**Parameters** **all** (*boolean, (optional)*) – All, Delete drivers for all elements of the array

`bpy.ops.anim.keyframe_delete` (*type='DEFAULT', confirm\_success=True*)  
Delete keyframes on the current frame for all properties in the specified Keying Set

#### Parameters

- **type** (*enum in ['DEFAULT'], (optional)*) – Keying Set, The Keying Set to use
- **confirm\_success** (*boolean, (optional)*) – Confirm Successful Insert, Show a popup when the keyframes get successfully added

`bpy.ops.anim.keyframe_delete_button` (*all=True*)  
Undocumented (`contribute`)

**Parameters** **all** (*boolean, (optional)*) – All, Delete keyframes from all elements of the array

`bpy.ops.anim.keyframe_delete_v3d` ()  
Remove keyframes on current frame for selected object

`bpy.ops.anim.keyframe_insert` (*type='DEFAULT', confirm\_success=True*)  
Insert keyframes on the current frame for all properties in the specified Keying Set

#### Parameters

- **type** (*enum in ['DEFAULT'], (optional)*) – Keying Set, The Keying Set to use
- **confirm\_success** (*boolean, (optional)*) – Confirm Successful Insert, Show a popup when the keyframes get successfully added

`bpy.ops.anim.keyframe_insert_button` (*all=True*)  
Undocumented ([contribute](#))

**Parameters** *all* (*boolean, (optional)*) – All, Insert a keyframe for all element of the array

`bpy.ops.anim.keyframe_insert_menu` (*type='DEFAULT', confirm\_success=False, always\_prompt=False*)

Insert Keyframes for specified Keying Set, with menu of available Keying Sets if undefined

**Parameters**

- **type** (*enum in ['DEFAULT'], (optional)*) – Keying Set, The Keying Set to use
- **confirm\_success** (*boolean, (optional)*) – Confirm Successful Insert, Show a popup when the keyframes get successfully added
- **always\_prompt** (*boolean, (optional)*) – Always Show Menu

`bpy.ops.anim.keying_set_active_set` (*type=0*)  
Undocumented ([contribute](#))

**Parameters** *type* (*int in [-inf, inf], (optional)*) – Keying Set Number, Index (determined internally) of the Keying Set to use

`bpy.ops.anim.keying_set_add` ()  
Add a new (empty) Keying Set to the active Scene

`bpy.ops.anim.keying_set_export` (*filepath=""*, *filter\_folder=True*, *filter\_text=True*, *filter\_python=True*)

Export Keying Set to a python script

**Parameters**

- **filepath** (*string, (optional)*) – File Path
- **filter\_folder** (*boolean, (optional)*) – Filter folders
- **filter\_text** (*boolean, (optional)*) – Filter text
- **filter\_python** (*boolean, (optional)*) – Filter python

**File** `startup/bl_operators/anim.py:59`

`bpy.ops.anim.keying_set_path_add` ()  
Add empty path to active Keying Set

`bpy.ops.anim.keying_set_path_remove` ()  
Remove active Path from active Keying Set

`bpy.ops.anim.keying_set_remove` ()  
Remove the active Keying Set

`bpy.ops.anim.keyingset_button_add` (*all=True*)  
Undocumented ([contribute](#))

**Parameters** *all* (*boolean, (optional)*) – All, Add all elements of the array to a Keying Set

`bpy.ops.anim.keyingset_button_remove` ()  
Undocumented ([contribute](#))

`bpy.ops.anim.paste_driver_button` ()  
Paste the driver in the copy/paste buffer for the highlighted button

`bpy.ops.anim.previewrange_clear` ()  
Clear Preview Range

`bpy.ops.anim.previewrange_set` (*xmin=0, xmax=0, ymin=0, ymax=0*)  
Interactively define frame range used for playback

#### 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

`bpy.ops.anim.time_toggle` ()  
Toggle whether timing is displayed in frames or seconds for active timeline view

`bpy.ops.anim.update_data_paths` ()  
Update data paths from 2.56 and previous versions, modifying data paths of drivers and fcurves

**File** `startup/bl_operators/anim.py:271`

## Armature Operators

`bpy.ops.armature.align` ()  
Align selected bones to the active bone (or to their parent)

`bpy.ops.armature.armature_layers` (*layers=(False, False)*)

Change the visible armature layers

**Parameters** **layers** (*boolean array of 32 items, (optional)*) – Layer, Armature layers to make visible

`bpy.ops.armature.autoside_names` (*type='XAXIS'*)  
Automatically renames the selected bones according to which side of the target axis they fall on

**Parameters** **type** (*enum in ['XAXIS', 'YAXIS', 'ZAXIS'], (optional)*) – Axis, Axis tag names with

- XAXIS X-Axis, Left/Right.
- YAXIS Y-Axis, Front/Back.
- ZAXIS Z-Axis, Top/Bottom.

`bpy.ops.armature.bone_layers` (*layers=(False, False)*)

Change the layers that the selected bones belong to

**Parameters** **layers** (*boolean array of 32 items, (optional)*) – Layer, Armature layers that bone belongs to

`bpy.ops.armature.bone_primitive_add` (*name="Bone"*)  
Add a new bone located at the 3D-Cursor

**Parameters** **name** (*string, (optional)*) – Name, Name of the newly created bone

`bpy.ops.armature.calculate_roll` (*type='X', axis\_flip=False, axis\_only=False*)  
Automatically fix alignment of select bones' axes

#### Parameters

- **type** (*enum in ['X', 'Y', 'Z', 'ACTIVE', 'VIEW', 'CURSOR'], (optional)*) – Type

- **axis\_flip** (*boolean, (optional)*) – Flip Axis, Negate the alignment axis
- **axis\_only** (*boolean, (optional)*) – Shortest Rotation, Ignore the axis direction, use the shortest rotation to align

`bpy.ops.armature.click_extrude()`  
Create a new bone going from the last selected joint to the mouse position

`bpy.ops.armature.delete()`  
Remove selected bones from the armature

`bpy.ops.armature.duplicate()`  
Make copies of the selected bones within the same armature

`bpy.ops.armature.duplicate_move` (*ARMATURE\_OT\_duplicate=None, TRANS-*  
*FORM\_OT\_translate=None*)  
Undocumented ([contribute](#))

#### Parameters

- **ARMATURE\_OT\_duplicate** (*ARMATURE\_OT\_duplicate, (optional)*) – Duplicate Selected Bone(s), Make copies of the selected bones within the same armature
- **TRANSFORM\_OT\_translate** (*TRANSFORM\_OT\_translate, (optional)*) – Translate, Translate selected items

`bpy.ops.armature.extrude` (*forked=False*)  
Create new bones from the selected joints

#### Parameters **forked** (*boolean, (optional)*) – Forked

`bpy.ops.armature.extrude_forked` (*ARMATURE\_OT\_extrude=None, TRANS-*  
*FORM\_OT\_translate=None*)  
Undocumented ([contribute](#))

#### Parameters

- **ARMATURE\_OT\_extrude** (*ARMATURE\_OT\_extrude, (optional)*) – Extrude, Create new bones from the selected joints
- **TRANSFORM\_OT\_translate** (*TRANSFORM\_OT\_translate, (optional)*) – Translate, Translate selected items

`bpy.ops.armature.extrude_move` (*ARMATURE\_OT\_extrude=None, TRANS-*  
*FORM\_OT\_translate=None*)  
Undocumented ([contribute](#))

#### Parameters

- **ARMATURE\_OT\_extrude** (*ARMATURE\_OT\_extrude, (optional)*) – Extrude, Create new bones from the selected joints
- **TRANSFORM\_OT\_translate** (*TRANSFORM\_OT\_translate, (optional)*) – Translate, Translate selected items

`bpy.ops.armature.fill()`  
Add bone between selected joint(s) and/or 3D-Cursor

`bpy.ops.armature.flip_names()`  
Flips (and corrects) the axis suffixes of the names of selected bones

`bpy.ops.armature.hide` (*unselected=False*)  
Tag selected bones to not be visible in Edit Mode

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

- `bpy.ops.armature.layers_show_all` (*all=True*)  
Make all armature layers visible
- Parameters** `all` (*boolean, (optional)*) – All Layers, Enable all layers or just the first 16 (top row)
- `bpy.ops.armature.merge` (*type='WITHIN\_CHAIN'*)  
Merge continuous chains of selected bones
- Parameters** `type` (*enum in ['WITHIN\_CHAIN'], (optional)*) – Type
- `bpy.ops.armature.parent_clear` (*type='CLEAR'*)  
Remove the parent-child relationship between selected bones and their parents
- Parameters** `type` (*enum in ['CLEAR', 'DISCONNECT'], (optional)*) – ClearType, What way to clear parenting
- `bpy.ops.armature.parent_set` (*type='CONNECTED'*)  
Set the active bone as the parent of the selected bones
- Parameters** `type` (*enum in ['CONNECTED', 'OFFSET'], (optional)*) – ParentType, Type of parenting
- `bpy.ops.armature.reveal` ()  
Unhide all bones that have been tagged to be hidden in Edit Mode
- `bpy.ops.armature.select_all` (*action='TOGGLE'*)  
Toggle selection status of all bones
- 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.armature.select_hierarchy` (*direction='PARENT', extend=False*)  
Select immediate parent/children of selected bones
- Parameters**
- `direction` (*enum in ['PARENT', 'CHILD'], (optional)*) – Direction
  - `extend` (*boolean, (optional)*) – Add to Selection
- `bpy.ops.armature.select_inverse` ()  
Flip the selection status of bones (selected -> unselected, unselected -> selected)
- `bpy.ops.armature.select_linked` (*extend=False*)  
Select bones related to selected ones by parent/child relationships
- Parameters** `extend` (*boolean, (optional)*) – Extend, Extend selection instead of deselecting everything first
- `bpy.ops.armature.separate` ()  
Isolate selected bones into a separate armature
- `bpy.ops.armature.subdivide` (*number\_cuts=1*)  
Break selected bones into chains of smaller bones
- Parameters** `number_cuts` (*int in [1, inf], (optional)*) – Number of Cuts

`bpy.ops.armature.switch_direction()`  
Change the direction that a chain of bones points in (head <-> tail swap)

## Boid Operators

`bpy.ops.boid.rule_add(type='GOAL')`  
Add a boid rule to the current boid state

**Parameters** `type` (*enum in ['GOAL', 'AVOID', 'AVOID\_COLLISION', 'SEPARATE', 'FLOCK', 'FOLLOW\_LEADER', 'AVERAGE\_SPEED', 'FIGHT'], (optional)*) – Type

- **GOAL** Goal, Go to assigned object or loudest assigned signal source.
- **AVOID** Avoid, Get away from assigned object or loudest assigned signal source.
- **AVOID\_COLLISION** Avoid Collision, Manoeuvre to avoid collisions with other boids and deflector objects in near future.
- **SEPARATE** Separate, Keep from going through other boids.
- **FLOCK** Flock, Move to center of neighbors and match their velocity.
- **FOLLOW\_LEADER** Follow Leader, Follow a boid or assigned object.
- **AVERAGE\_SPEED** Average Speed, Maintain speed, flight level or wander.
- **FIGHT** Fight, Go to closest enemy and attack when in range.

`bpy.ops.boid.rule_del()`  
Undocumented ([contribute](#))

`bpy.ops.boid.rule_move_down()`  
Move boid rule down in the list

`bpy.ops.boid.rule_move_up()`  
Move boid rule up in the list

`bpy.ops.boid.state_add()`  
Add a boid state to the particle system

`bpy.ops.boid.state_del()`  
Undocumented ([contribute](#))

`bpy.ops.boid.state_move_down()`  
Move boid state down in the list

`bpy.ops.boid.state_move_up()`  
Move boid state up in the list

## Brush Operators

`bpy.ops.brush.active_index_set(mode="", index=0)`  
Set active sculpt/paint brush from it's number

### Parameters

- **mode** (*string, (optional)*) – mode, Paint mode to set brush for
- **index** (*int in [-inf, inf], (optional)*) – number, Brush number

**File** `startup/bl_operators/wm.py:161`

- `bpy.ops.brush.add()`  
Add brush by mode type
- `bpy.ops.brush.curve_preset` (*shape='SMOOTH'*)  
Set brush shape
- Parameters** `shape` (*enum in ['SHARP', 'SMOOTH', 'MAX', 'LINE', 'ROUND', 'ROOT'], (optional)*) – Mode
- `bpy.ops.brush.image_tool_set` (*tool='DRAW'*)  
Set the image tool
- Parameters** `tool` (*enum in ['DRAW', 'SOFTEN', 'SMEAR', 'CLONE'], (optional)*) – Tool
- `bpy.ops.brush.reset()`  
Return brush to defaults based on current tool
- `bpy.ops.brush.scale_size` (*scalar=1.0*)  
Change brush size by a scalar
- Parameters** `scalar` (*float in [0, 2], (optional)*) – Scalar, Factor to scale brush size by
- `bpy.ops.brush.sculpt_tool_set` (*tool='BLOB'*)  
Set the sculpt tool
- Parameters** `tool` (*enum in ['BLOB', 'CLAY', 'CREASE', 'DRAW', 'FILL', 'FLATTEN', 'GRAB', 'INFLATE', 'LAYER', 'NUDGE', 'PINCH', 'ROTATE', 'SCRAPE', 'SMOOTH', 'SNAKE\_HOOK', 'THUMB'], (optional)*) – Tool
- `bpy.ops.brush.vertex_tool_set` (*tool='MIX'*)  
Set the vertex paint tool
- Parameters** `tool` (*enum in ['MIX', 'ADD', 'SUB', 'MUL', 'BLUR', 'LIGHTEN', 'DARKEN'], (optional)*) – Tool
- MIX Mix, Use mix blending mode while painting.
  - ADD Add, Use add blending mode while painting.
  - SUB Subtract, Use subtract blending mode while painting.
  - MUL Multiply, Use multiply blending mode while painting.
  - BLUR Blur, Blur the color with surrounding values.
  - LIGHTEN Lighten, Use lighten blending mode while painting.
  - DARKEN Darken, Use darken blending mode while painting.
- `bpy.ops.brush.weight_tool_set` (*tool='MIX'*)  
Set the weight paint tool
- Parameters** `tool` (*enum in ['MIX', 'ADD', 'SUB', 'MUL', 'BLUR', 'LIGHTEN', 'DARKEN'], (optional)*) – Tool
- MIX Mix, Use mix blending mode while painting.
  - ADD Add, Use add blending mode while painting.
  - SUB Subtract, Use subtract blending mode while painting.
  - MUL Multiply, Use multiply blending mode while painting.
  - BLUR Blur, Blur the color with surrounding values.
  - LIGHTEN Lighten, Use lighten blending mode while painting.

- **DARKEN** Darken, Use darken blending mode while painting.

## Buttons Operators

`bpy.ops.buttons.directory_browse` (*directory=""*, *filter\_blender=False*, *filter\_image=False*, *filter\_movie=False*, *filter\_python=False*, *filter\_font=False*, *filter\_sound=False*, *filter\_text=False*, *filter\_btx=False*, *filter\_collada=False*, *filter\_folder=False*, *filemode=9*, *relative\_path=True*)

Open a directory browser, Hold Shift to open the file, Alt to browse containing directory

### Parameters

- **directory** (*string, (optional)*) – Directory, Directory of the file
- **filter\_blender** (*boolean, (optional)*) – Filter .blend files
- **filter\_image** (*boolean, (optional)*) – Filter image files
- **filter\_movie** (*boolean, (optional)*) – Filter movie files
- **filter\_python** (*boolean, (optional)*) – Filter python files
- **filter\_font** (*boolean, (optional)*) – Filter font files
- **filter\_sound** (*boolean, (optional)*) – Filter sound files
- **filter\_text** (*boolean, (optional)*) – Filter text files
- **filter\_btx** (*boolean, (optional)*) – Filter btx files
- **filter\_collada** (*boolean, (optional)*) – Filter COLLADA files
- **filter\_folder** (*boolean, (optional)*) – Filter folders
- **filemode** (*int in [1, 9], (optional)*) – File Browser Mode, The setting for the file browser mode to load a .blend file, a library or a special file
- **relative\_path** (*boolean, (optional)*) – Relative Path, Select the file relative to the blend file

`bpy.ops.buttons.file_browse` (*filepath=""*, *filter\_blender=False*, *filter\_image=False*, *filter\_movie=False*, *filter\_python=False*, *filter\_font=False*, *filter\_sound=False*, *filter\_text=False*, *filter\_btx=False*, *filter\_collada=False*, *filter\_folder=False*, *filemode=9*, *relative\_path=True*)

Open a file browser, Hold Shift to open the file, Alt to browse containing directory

### Parameters

- **filepath** (*string, (optional)*) – File Path, Path to file
- **filter\_blender** (*boolean, (optional)*) – Filter .blend files
- **filter\_image** (*boolean, (optional)*) – Filter image files
- **filter\_movie** (*boolean, (optional)*) – Filter movie files
- **filter\_python** (*boolean, (optional)*) – Filter python files
- **filter\_font** (*boolean, (optional)*) – Filter font files
- **filter\_sound** (*boolean, (optional)*) – Filter sound files
- **filter\_text** (*boolean, (optional)*) – Filter text files
- **filter\_btx** (*boolean, (optional)*) – Filter btx files



- **filter\_collada** (*boolean, (optional)*) – Filter COLLADA files
- **filter\_folder** (*boolean, (optional)*) – Filter folders
- **filemode** (*int in [1, 9], (optional)*) – File Browser Mode, The setting for the file browser mode to load a .blend file, a library or a special file
- **relative\_path** (*boolean, (optional)*) – Relative Path, Select the file relative to the blend file

`bpy.ops.buttons.toolbox()`  
Display button panel toolbox

## Cloth Operators

`bpy.ops.cloth.preset_add(name="", remove_active=False)`  
Add a Cloth Preset

**Parameters** **name** (*string, (optional)*) – Name, Name of the preset, used to make the path name

**File** `startup/bl_operators/presets.py:50`

## Console Operators

`bpy.ops.console.autocomplete()`  
Evaluate the namespace up until the cursor and give a list of options or complete the name if there is only one

**File** `startup/bl_operators/console.py:51`

`bpy.ops.console.banner()`  
Print a message when the terminal initializes

**File** `startup/bl_operators/console.py:69`

`bpy.ops.console.clear(scrollback=True, history=False)`  
Clear text by type

### Parameters

- **scrollback** (*boolean, (optional)*) – Scrollback, Clear the scrollbar history
- **history** (*boolean, (optional)*) – History, Clear the command history

`bpy.ops.console.copy()`  
Copy selected text to clipboard

`bpy.ops.console.delete(type='NEXT_CHARACTER')`  
Delete text by cursor position

**Parameters** **type** (*enum in ['NEXT\_CHARACTER', 'PREVIOUS\_CHARACTER'], (optional)*) – Type, Which part of the text to delete

`bpy.ops.console.execute()`  
Execute the current console line as a python expression

**File** `startup/bl_operators/console.py:31`

`bpy.ops.console.history_append(text="", current_character=0, remove_duplicates=False)`  
Append history at cursor position

### Parameters

- **text** (*string, (optional)*) – Text, Text to insert at the cursor position
- **current\_character** (*int in [0, inf], (optional)*) – Cursor, The index of the cursor

- **remove\_duplicates** (*boolean, (optional)*) – Remove Duplicates, Remove duplicate items in the history

`bpy.ops.console.history_cycle` (*reverse=False*)

Cycle through history

**Parameters** **reverse** (*boolean, (optional)*) – Reverse, Reverse cycle history

`bpy.ops.console.insert` (*text=""*)

Insert text at cursor position

**Parameters** **text** (*string, (optional)*) – Text, Text to insert at the cursor position

`bpy.ops.console.language` (*language=""*)

Set the current language for this console

**Parameters** **language** (*string, (optional)*) – Language

**File** `startup/bl_operators/console.py:97`

`bpy.ops.console.move` (*type='LINE\_BEGIN'*)

Move cursor position

**Parameters** **type** (*enum in ['LINE\_BEGIN', 'LINE\_END', 'PREVIOUS\_CHARACTER', 'NEXT\_CHARACTER', 'PREVIOUS\_WORD', 'NEXT\_WORD'], (optional)*) – Type, Where to move cursor to

`bpy.ops.console.paste` ()

Paste text from clipboard

`bpy.ops.console.scrollback_append` (*text="", type='OUTPUT'*)

Append scrollbar text by type

**Parameters**

- **text** (*string, (optional)*) – Text, Text to insert at the cursor position
- **type** (*enum in ['OUTPUT', 'INPUT', 'INFO', 'ERROR'], (optional)*) – Type, Console output type

`bpy.ops.console.select_set` ()

Set the console selection

## Constraint Operators

`bpy.ops.constraint.childof_clear_inverse` (*constraint="", owner='OBJECT'*)

Clear inverse correction for ChildOf constraint

**Parameters**

- **constraint** (*string, (optional)*) – Constraint, Name of the constraint to edit
- **owner** (*enum in ['OBJECT', 'BONE'], (optional)*) – Owner, The owner of this constraint
  - `OBJECT` Object, Edit a constraint on the active object.
  - `BONE` Bone, Edit a constraint on the active bone.

`bpy.ops.constraint.childof_set_inverse` (*constraint="", owner='OBJECT'*)

Set inverse correction for ChildOf constraint

**Parameters**

- **constraint** (*string, (optional)*) – Constraint, Name of the constraint to edit

- **owner** (*enum in ['OBJECT', 'BONE'], (optional)*) – Owner, The owner of this constraint
  - OBJECT Object, Edit a constraint on the active object.
  - BONE Bone, Edit a constraint on the active bone.

`bpy.ops.constraint.delete()`  
Remove constraint from constraint stack

`bpy.ops.constraint.limitdistance_reset (constraint="", owner='OBJECT')`  
Reset limiting distance for Limit Distance Constraint

#### Parameters

- **constraint** (*string, (optional)*) – Constraint, Name of the constraint to edit
- **owner** (*enum in ['OBJECT', 'BONE'], (optional)*) – Owner, The owner of this constraint
  - OBJECT Object, Edit a constraint on the active object.
  - BONE Bone, Edit a constraint on the active bone.

`bpy.ops.constraint.move_down (constraint="", owner='OBJECT')`  
Move constraint down in constraint stack

#### Parameters

- **constraint** (*string, (optional)*) – Constraint, Name of the constraint to edit
- **owner** (*enum in ['OBJECT', 'BONE'], (optional)*) – Owner, The owner of this constraint
  - OBJECT Object, Edit a constraint on the active object.
  - BONE Bone, Edit a constraint on the active bone.

`bpy.ops.constraint.move_up (constraint="", owner='OBJECT')`  
Move constraint up in constraint stack

#### Parameters

- **constraint** (*string, (optional)*) – Constraint, Name of the constraint to edit
- **owner** (*enum in ['OBJECT', 'BONE'], (optional)*) – Owner, The owner of this constraint
  - OBJECT Object, Edit a constraint on the active object.
  - BONE Bone, Edit a constraint on the active bone.

`bpy.ops.constraint.stretchto_reset (constraint="", owner='OBJECT')`  
Reset original length of bone for Stretch To Constraint

#### Parameters

- **constraint** (*string, (optional)*) – Constraint, Name of the constraint to edit
- **owner** (*enum in ['OBJECT', 'BONE'], (optional)*) – Owner, The owner of this constraint
  - OBJECT Object, Edit a constraint on the active object.
  - BONE Bone, Edit a constraint on the active bone.

## Curve Operators

`bpy.ops.curve.cyclic_toggle (direction='CYCLIC_U')`  
Make active spline closed/opened loop

**Parameters** `direction` (*enum in ['CYCLIC\_U', 'CYCLIC\_V'], (optional)*) – Direction, Direction to make surface cyclic in

`bpy.ops.curve.de_select_first()`  
Undocumented ([contribute](#))

`bpy.ops.curve.de_select_last()`  
Undocumented ([contribute](#))

`bpy.ops.curve.delete` (*type='SELECTED'*)  
Delete selected control points or segments

**Parameters** `type` (*enum in ['SELECTED', 'SEGMENT', 'ALL'], (optional)*) – Type, Which elements to delete

`bpy.ops.curve.duplicate` (*mode='TRANSLATION'*)  
Duplicate selected control points and segments between them

**Parameters** `mode` (*enum in ['INIT', 'DUMMY', 'TRANSLATION', 'ROTATION', 'RESIZE', 'TOSPHERE', 'SHEAR', 'WARP', 'SHRINKFATTEN', 'TILT', 'TRACKBALL', 'PUSHPULL', 'CREASE', 'MIRROR', 'BONE\_SIZE', 'BONE\_ENVELOPE', 'CURVE\_SHRINKFATTEN', 'BONE\_ROLL', 'TIME\_TRANSLATE', 'TIME\_SLIDE', 'TIME\_SCALE', 'TIME\_EXTEND', 'BAKE\_TIME', 'BEVEL', 'BWEIGHT', 'ALIGN', 'EDGESLIDE', 'SEQSLIDE'], (optional)*) – Mode

`bpy.ops.curve.extrude` (*mode='TRANSLATION'*)  
Extrude selected control point(s) and move

**Parameters** `mode` (*enum in ['INIT', 'DUMMY', 'TRANSLATION', 'ROTATION', 'RESIZE', 'TOSPHERE', 'SHEAR', 'WARP', 'SHRINKFATTEN', 'TILT', 'TRACKBALL', 'PUSHPULL', 'CREASE', 'MIRROR', 'BONE\_SIZE', 'BONE\_ENVELOPE', 'CURVE\_SHRINKFATTEN', 'BONE\_ROLL', 'TIME\_TRANSLATE', 'TIME\_SLIDE', 'TIME\_SCALE', 'TIME\_EXTEND', 'BAKE\_TIME', 'BEVEL', 'BWEIGHT', 'ALIGN', 'EDGESLIDE', 'SEQSLIDE'], (optional)*) – Mode

`bpy.ops.curve.handle_type_set` (*type='AUTOMATIC'*)  
Set type of handles for selected control points

**Parameters** `type` (*enum in ['AUTOMATIC', 'VECTOR', 'ALIGNED', 'FREE\_ALIGN', 'TOGGLE\_FREE\_ALIGN'], (optional)*) – Type, Spline type

`bpy.ops.curve.hide` (*unselected=False*)  
Undocumented ([contribute](#))

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

`bpy.ops.curve.make_segment()`  
Undocumented ([contribute](#))

`bpy.ops.curve.primitive_bezier_circle_add` (*view\_align=False, enter\_editmode=False, location=(0.0, 0.0, 0.0), rotation=(0.0, 0.0, 0.0), layers=(False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False)*)

Construct a Bezier Circle

#### Parameters

- `view_align` (*boolean, (optional)*) – Align to View, Align the new object to the view
- `enter_editmode` (*boolean, (optional)*) – Enter Editmode, Enter editmode when adding this object

- **location** (*float array of 3 items in [-inf, inf], (optional)*) – Location, Location for the newly added object
- **rotation** (*float array of 3 items in [-inf, inf], (optional)*) – Rotation, Rotation for the newly added object
- **layers** (*boolean array of 20 items, (optional)*) – Layer

```
bpy.ops.curve.primitive_bezier_curve_add (view_align=False, enter_editmode=False, location=(0.0, 0.0, 0.0), rotation=(0.0, 0.0, 0.0), layers=(False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False))
```

Construct a Bezier Curve

#### Parameters

- **view\_align** (*boolean, (optional)*) – Align to View, Align the new object to the view
- **enter\_editmode** (*boolean, (optional)*) – Enter Editmode, Enter editmode when adding this object
- **location** (*float array of 3 items in [-inf, inf], (optional)*) – Location, Location for the newly added object
- **rotation** (*float array of 3 items in [-inf, inf], (optional)*) – Rotation, Rotation for the newly added object
- **layers** (*boolean array of 20 items, (optional)*) – Layer

```
bpy.ops.curve.primitive_nurbs_circle_add (view_align=False, enter_editmode=False, location=(0.0, 0.0, 0.0), rotation=(0.0, 0.0, 0.0), layers=(False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False))
```

Construct a Nurbs Circle

#### Parameters

- **view\_align** (*boolean, (optional)*) – Align to View, Align the new object to the view
- **enter\_editmode** (*boolean, (optional)*) – Enter Editmode, Enter editmode when adding this object
- **location** (*float array of 3 items in [-inf, inf], (optional)*) – Location, Location for the newly added object
- **rotation** (*float array of 3 items in [-inf, inf], (optional)*) – Rotation, Rotation for the newly added object
- **layers** (*boolean array of 20 items, (optional)*) – Layer

```
bpy.ops.curve.primitive_nurbs_curve_add (view_align=False, enter_editmode=False, location=(0.0, 0.0, 0.0), rotation=(0.0, 0.0, 0.0), layers=(False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False))
```

Construct a Nurbs Curve

#### Parameters

- **view\_align** (*boolean, (optional)*) – Align to View, Align the new object to the view

- **enter\_editmode** (*boolean, (optional)*) – Enter Editmode, Enter editmode when adding this object
- **location** (*float array of 3 items in [-inf, inf], (optional)*) – Location, Location for the newly added object
- **rotation** (*float array of 3 items in [-inf, inf], (optional)*) – Rotation, Rotation for the newly added object
- **layers** (*boolean array of 20 items, (optional)*) – Layer

`bpy.ops.curve.primitive_nurbs_path_add` (*view\_align=False, enter\_editmode=False, location=(0.0, 0.0, 0.0), rotation=(0.0, 0.0, 0.0), layers=(False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False)*)

Construct a Path

#### Parameters

- **view\_align** (*boolean, (optional)*) – Align to View, Align the new object to the view
- **enter\_editmode** (*boolean, (optional)*) – Enter Editmode, Enter editmode when adding this object
- **location** (*float array of 3 items in [-inf, inf], (optional)*) – Location, Location for the newly added object
- **rotation** (*float array of 3 items in [-inf, inf], (optional)*) – Rotation, Rotation for the newly added object
- **layers** (*boolean array of 20 items, (optional)*) – Layer

`bpy.ops.curve.radius_set` (*radius=1.0*)  
Set per-point radius which is used for bevel tapering

**Parameters** **radius** (*float in [0, inf], (optional)*) – Radius

`bpy.ops.curve.reveal` ()  
Undocumented ([contribute](#))

`bpy.ops.curve.select_all` (*action='TOGGLE'*)  
Undocumented ([contribute](#))

**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.curve.select_inverse` ()  
Undocumented ([contribute](#))

`bpy.ops.curve.select_less` ()  
Undocumented ([contribute](#))

`bpy.ops.curve.select_linked` ()  
Undocumented ([contribute](#))

`bpy.ops.curve.select_linked_pick` (*deselect=False*)  
Undocumented ([contribute](#))

**Parameters** `deselect` (*boolean, (optional)*) – Deselect, Deselect linked control points rather than selecting them

`bpy.ops.curve.select_more` ()  
Undocumented ([contribute](#))

`bpy.ops.curve.select_next` ()  
Undocumented ([contribute](#))

`bpy.ops.curve.select_nth` (*nth=2*)  
Undocumented ([contribute](#))

**Parameters** `nth` (*int in [2, 100], (optional)*) – Nth Selection

`bpy.ops.curve.select_previous` ()  
Undocumented ([contribute](#))

`bpy.ops.curve.select_random` (*percent=50.0, extend=False*)  
Undocumented ([contribute](#))

**Parameters**

- **percent** (*float in [0, 100], (optional)*) – Percent, Percentage of elements to select randomly
- **extend** (*boolean, (optional)*) – Extend Selection, Extend selection instead of deselecting everything first

`bpy.ops.curve.select_row` ()  
Undocumented ([contribute](#))

`bpy.ops.curve.separate` ()  
Undocumented ([contribute](#))

`bpy.ops.curve.shade_flat` ()  
Undocumented ([contribute](#))

`bpy.ops.curve.shade_smooth` ()  
Undocumented ([contribute](#))

`bpy.ops.curve.smooth` ()  
Flatten angles of selected points

`bpy.ops.curve.smooth_radius` ()  
Flatten radiuses of selected points

`bpy.ops.curve.spin` (*center=(0.0, 0.0, 0.0), axis=(0.0, 0.0, 0.0)*)  
Undocumented ([contribute](#))

**Parameters**

- **center** (*float array of 3 items in [-inf, inf], (optional)*) – Center, Center in global view space
- **axis** (*float array of 3 items in [-1, 1], (optional)*) – Axis, Axis in global view space

`bpy.ops.curve.spline_type_set` (*type='POLY'*)  
Set type of active spline

**Parameters** `type` (*enum in ['POLY', 'BEZIER', 'NURBS'], (optional)*) – Type, Spline type

`bpy.ops.curve.spline_weight_set` (*weight=1.0*)  
Set softbody goal weight for selected points

**Parameters** `weight` (*float in [0, 1], (optional)*) – Weight

`bpy.ops.curve.subdivide` (*number\_cuts=1*)  
Subdivide selected segments

**Parameters** `number_cuts` (*int in [1, inf], (optional)*) – Number of cuts

`bpy.ops.curve.switch_direction()`  
Switch direction of selected splines

`bpy.ops.curve.tilt_clear()`  
Undocumented ([contribute](#))

`bpy.ops.curve.vertex_add(location=(0.0, 0.0, 0.0))`  
Undocumented ([contribute](#))

**Parameters** `location` (*float array of 3 items in [-inf, inf], (optional)*) – Location, Location to add new vertex at

## Ed Operators

`bpy.ops.ed.redo()`  
Redo previous action

`bpy.ops.ed.undo()`  
Undo previous action

`bpy.ops.ed.undo_history(item=0)`  
Redo specific action in history

**Parameters** `item` (*int in [0, inf], (optional)*) – Item

`bpy.ops.ed.undo_push(message="Add an undo step *function may be moved*")`  
Add an undo state (internal use only)

**Parameters** `message` (*string, (optional)*) – Undo Message

## Export Anim Operators

`bpy.ops.export_anim.bvh(filepath="", check_existing=True, filter_glob="*.bvh", global_scale=1.0, frame_start=0, frame_end=0, rotate_mode='NATIVE', root_transform_only=False)`  
Save a BVH motion capture file from an armature

### Parameters

- **filepath** (*string, (optional)*) – File Path, Filepath used for exporting the file
- **check\_existing** (*boolean, (optional)*) – Check Existing, Check and warn on overwriting existing files
- **global\_scale** (*float in [0.0001, 1e+06], (optional)*) – Scale, Scale the BVH by this value
- **frame\_start** (*int in [-inf, inf], (optional)*) – Start Frame, Starting frame to export
- **frame\_end** (*int in [-inf, inf], (optional)*) – End Frame, End frame to export
- **rotate\_mode** (*enum in ['NATIVE', 'XYZ', 'XZY', 'YXZ', 'YZX', 'ZXY', 'ZYX'], (optional)*) – Rotation, Rotation conversion
  - NATIVE Euler (Native), Use the rotation order defined in the BVH file.
  - XYZ Euler (XYZ), Convert rotations to euler XYZ.
  - XZY Euler (XZY), Convert rotations to euler XZY.
  - YXZ Euler (YXZ), Convert rotations to euler YXZ.



- YZX Euler (YZX), Convert rotations to euler YZX.
- ZXY Euler (ZXY), Convert rotations to euler ZXY.
- ZYX Euler (ZYX), Convert rotations to euler ZYX.
- **root\_transform\_only** (*boolean, (optional)*) – Root Transform Only, Only write out transform channels for the root bone

**File** [addons/io\\_anim\\_bvh/\\_\\_init\\_\\_.py:205](#)

## Export Mesh Operators

```
bpy.ops.export_mesh.ply (filepath="", check_existing=True, filter_glob="*.ply",
                        use_modifiers=True, use_normals=True, use_uv_coords=True,
                        use_colors=True)
```

Export a single object as a stanford PLY with normals, colours and texture coordinates

### Parameters

- **filepath** (*string, (optional)*) – File Path, Filepath used for exporting the file
- **check\_existing** (*boolean, (optional)*) – Check Existing, Check and warn on overwriting existing files
- **use\_modifiers** (*boolean, (optional)*) – Apply Modifiers, Apply Modifiers to the exported mesh
- **use\_normals** (*boolean, (optional)*) – Normals, Export Normals for smooth and hard shaded faces
- **use\_uv\_coords** (*boolean, (optional)*) – UVs, Export the active UV layer
- **use\_colors** (*boolean, (optional)*) – Vertex Colors, Exort the active vertex color layer

**File** [addons/io\\_mesh\\_ply/\\_\\_init\\_\\_.py:114](#)

```
bpy.ops.export_mesh.stl (filepath="", check_existing=True, ascii=False, apply_modifiers=True)
```

Save STL triangle mesh data from the active object

### Parameters

- **filepath** (*string, (optional)*) – File Path, Filepath used for exporting the file
- **check\_existing** (*boolean, (optional)*) – Check Existing, Check and warn on overwriting existing files
- **ascii** (*boolean, (optional)*) – Ascii, Save the file in ASCII file format
- **apply\_modifiers** (*boolean, (optional)*) – Apply Modifiers, Apply the modifiers before saving

**File** [addons/io\\_mesh\\_stl/\\_\\_init\\_\\_.py:127](#)

## Export Scene Operators

```
bpy.ops.export_scene.autodesk_3ds (filepath="", check_existing=True, filter_glob="*.3ds",
                                   use_selection=False, axis_forward='Y', axis_up='Z')
```

Export to 3DS file format (.3ds)

### Parameters

- **filepath** (*string, (optional)*) – File Path, Filepath used for exporting the file

- **check\_existing** (*boolean, (optional)*) – Check Existing, Check and warn on overwriting existing files
- **use\_selection** (*boolean, (optional)*) – Selection Only, Export selected objects only
- **axis\_forward** (*enum in ['X', 'Y', 'Z', '-X', '-Y', '-Z'], (optional)*) – Forward
- **axis\_up** (*enum in ['X', 'Y', 'Z', '-X', '-Y', '-Z'], (optional)*) – Up

**File** `addons/io_scene_3ds/__init__.py:164`

```
bpy.ops.export_scene.fbx(filepath="", check_existing=True, filter_glob="*.fbx",
                        use_selection=False, global_scale=1.0, axis_forward='-
                        Z', axis_up='Y', object_types={'ARMATURE', 'LAMP',
                        'CAMERA', 'MESH', 'EMPTY'}, use_mesh_modifiers=True,
                        mesh_smooth_type='FACE', use_mesh_edges=False,
                        use_anim=True, use_anim_action_all=True, use_default_take=True,
                        use_anim_optimize=True, anim_optimize_precision=6.0,
                        path_mode='AUTO', use_rotate_workaround=False,
                        xna_validate=False, batch_mode='OFF', use_batch_own_dir=True,
                        use_metadata=True)
```

Selection to an ASCII Autodesk FBX

### Parameters

- **filepath** (*string, (optional)*) – File Path, Filepath used for exporting the file
- **check\_existing** (*boolean, (optional)*) – Check Existing, Check and warn on overwriting existing files
- **use\_selection** (*boolean, (optional)*) – Selected Objects, Export selected objects on visible layers
- **global\_scale** (*float in [0.01, 1000], (optional)*) – Scale, Scale all data (Some importers do not support scaled armatures!)
- **axis\_forward** (*enum in ['X', 'Y', 'Z', '-X', '-Y', '-Z'], (optional)*) – Forward
- **axis\_up** (*enum in ['X', 'Y', 'Z', '-X', '-Y', '-Z'], (optional)*) – Up
- **object\_types** (*enum set in {'EMPTY', 'CAMERA', 'LAMP', 'ARMATURE', 'MESH'}, (optional)*) – Object Types
- **use\_mesh\_modifiers** (*boolean, (optional)*) – Apply Modifiers, Apply modifiers to mesh objects
- **mesh\_smooth\_type** (*enum in ['OFF', 'FACE', 'EDGE'], (optional)*) – Smoothing
  - OFF Off, Don't write smoothing.
  - FACE Face, Write face smoothing.
  - EDGE Edge, Write edge smoothing.
- **use\_mesh\_edges** (*boolean, (optional)*) – Include Edges, Edges may not be necessary, can cause import pipeline errors with XNA
- **use\_anim** (*boolean, (optional)*) – Include Animation, Export keyframe animation
- **use\_anim\_action\_all** (*boolean, (optional)*) – All Actions, Export all actions for armatures or just the currently selected action
- **use\_default\_take** (*boolean, (optional)*) – Include Default Take, Export currently assigned object and armature animations into a default take from the scene start/end frames

- **use\_anim\_optimize** (*boolean, (optional)*) – Optimize Keyframes, Remove double keyframes
- **anim\_optimize\_precision** (*float in [1, 16], (optional)*) – Precision, Tolerance for comparing double keyframes (higher for greater accuracy)
- **path\_mode** (*enum in ['AUTO', 'ABSOLUTE', 'RELATIVE', 'MATCH', 'STRIP', 'COPY'], (optional)*) – Path Mode, Method used to reference paths
  - AUTO Auto, Use Relative paths with subdirectories only.
  - ABSOLUTE Absolute, Always write absolute paths.
  - RELATIVE Relative, Always write relative path (where possible).
  - MATCH Match, Match Absolute/Relative setting with input path.
  - STRIP Strip Path, Filename only.
  - COPY Copy, copy the file to the destination path (or subdirectory).
- **use\_rotate\_workaround** (*boolean, (optional)*) – XNA Rotate Animation Hack, Disable global rotation, for XNA compatibility
- **xna\_validate** (*boolean, (optional)*) – XNA Strict Options, Make sure options are compatible with Microsoft XNA
- **batch\_mode** (*enum in ['OFF', 'SCENE', 'GROUP'], (optional)*) – Batch Mode
  - OFF Off, Active scene to file.
  - SCENE Scene, Each scene as a file.
  - GROUP Group, Each group as a file.
- **use\_batch\_own\_dir** (*boolean, (optional)*) – Batch Own Dir, Create a dir for each exported file
- **use\_metadata** (*boolean, (optional)*) – Use Metadata

**File** `addons/io_scene_fbx/__init__.py:236`

```
bpy.ops.export_scene.obj (filepath="", check_existing=True, filter_glob="*.obj;*.mtl",
                        use_selection=False, use_animation=False, use_apply_modifiers=True,
                        use_edges=True, use_normals=False, use_uvns=True,
                        use_materials=True, use_triangles=False, use_nurbs=False,
                        use_vertex_groups=False, use_blen_objects=True,
                        group_by_object=False, group_by_material=False,
                        keep_vertex_order=False, global_scale=1.0, axis_forward='-Z',
                        axis_up='Y', path_mode='AUTO')
```

Save a Wavefront OBJ File

#### Parameters

- **filepath** (*string, (optional)*) – File Path, Filepath used for exporting the file
- **check\_existing** (*boolean, (optional)*) – Check Existing, Check and warn on overwriting existing files
- **use\_selection** (*boolean, (optional)*) – Selection Only, Export selected objects only
- **use\_animation** (*boolean, (optional)*) – Animation, Write out an OBJ for each frame
- **use\_apply\_modifiers** (*boolean, (optional)*) – Apply Modifiers, Apply modifiers (preview resolution)
- **use\_edges** (*boolean, (optional)*) – Include Edges

- **use\_normals** (*boolean, (optional)*) – Include Normals
- **use\_uv**s (*boolean, (optional)*) – Include UVs, Write out the active UV coordinates
- **use\_materials** (*boolean, (optional)*) – Write Materials, Write out the MTL file
- **use\_triangles** (*boolean, (optional)*) – Triangulate Faces, Convert all faces to triangles
- **use\_nurbs** (*boolean, (optional)*) – Write Nurbs, Write nurbs curves as OBJ nurbs rather than converting to geometry
- **use\_vertex\_groups** (*boolean, (optional)*) – Polygroups
- **use\_blen\_objects** (*boolean, (optional)*) – Objects as OBJ Objects
- **group\_by\_object** (*boolean, (optional)*) – Objects as OBJ Groups
- **group\_by\_material** (*boolean, (optional)*) – Material Groups
- **keep\_vertex\_order** (*boolean, (optional)*) – Keep Vertex Order
- **global\_scale** (*float in [0.01, 1000], (optional)*) – Scale, Scale all data
- **axis\_forward** (*enum in ['X', 'Y', 'Z', '-X', '-Y', '-Z'], (optional)*) – Forward
- **axis\_up** (*enum in ['X', 'Y', 'Z', '-X', '-Y', '-Z'], (optional)*) – Up
- **path\_mode** (*enum in ['AUTO', 'ABSOLUTE', 'RELATIVE', 'MATCH', 'STRIP', 'COPY'], (optional)*) – Path Mode, Method used to reference paths
  - AUTO Auto, Use Relative paths with subdirectories only.
  - ABSOLUTE Absolute, Always write absolute paths.
  - RELATIVE Relative, Always write relative path (where possible).
  - MATCH Match, Match Absolute/Relative setting with input path.
  - STRIP Strip Path, Filename only.
  - COPY Copy, copy the file to the destination path (or subdirectory).

**File** `addons/io_scene_obj/_init__.py:328`

```
bpy.ops.export_scene.x3d(filepath="", check_existing=True, filter_glob="*.x3d",  
                        use_selection=False, use_apply_modifiers=True, use_triangulate=False,  
                        use_normals=False, use_compress=False, use_hierarchy=True,  
                        use_h3d=False, axis_forward='Z', axis_up='Y', path_mode='AUTO')
```

Export selection to Extensible 3D file (.x3d)

### Parameters

- **filepath** (*string, (optional)*) – File Path, Filepath used for exporting the file
- **check\_existing** (*boolean, (optional)*) – Check Existing, Check and warn on overwriting existing files
- **use\_selection** (*boolean, (optional)*) – Selection Only, Export selected objects only
- **use\_apply\_modifiers** (*boolean, (optional)*) – Apply Modifiers, Use transformed mesh data from each object
- **use\_triangulate** (*boolean, (optional)*) – Triangulate, Write quads into 'IndexedTriangleSet'
- **use\_normals** (*boolean, (optional)*) – Normals, Write normals with geometry
- **use\_compress** (*boolean, (optional)*) – Compress, Compress the exported file
- **use\_hierarchy** (*boolean, (optional)*) – Hierarchy, Export parent child relationships

- **use\_h3d** (*boolean, (optional)*) – H3D Extensions, Export shaders for H3D
- **axis\_forward** (*enum in ['X', 'Y', 'Z', '-X', '-Y', '-Z'], (optional)*) – Forward
- **axis\_up** (*enum in ['X', 'Y', 'Z', '-X', '-Y', '-Z'], (optional)*) – Up
- **path\_mode** (*enum in ['AUTO', 'ABSOLUTE', 'RELATIVE', 'MATCH', 'STRIP', 'COPY'], (optional)*) – Path Mode, Method used to reference paths
  - AUTO Auto, Use Relative paths with subdirectories only.
  - ABSOLUTE Absolute, Always write absolute paths.
  - RELATIVE Relative, Always write relative patsh (where possible).
  - MATCH Match, Match Absolute/Relative setting with input path.
  - STRIP Strip Path, Filename only.
  - COPY Copy, copy the file to the destination path (or subdirectory).

**File** `addons/io_scene_x3d/__init__.py:169`

## File Operators

`bpy.ops.file.bookmark_add()`  
Add a bookmark for the selected/active directory

`bpy.ops.file.bookmark_toggle()`  
Toggle bookmarks display

`bpy.ops.file.cancel()`  
Cancel loading of selected file

`bpy.ops.file.delete()`  
Delete selected file

`bpy.ops.file.delete_bookmark(index=-1)`  
Delete selected bookmark

**Parameters** `index` (*int in [-1, 20000], (optional)*) – Index

`bpy.ops.file.directory()`  
Enter a directory name

`bpy.ops.file.directory_new(directory=""“)`  
Create a new directory

**Parameters** `directory` (*string, (optional)*) – Directory, Name of new directory

`bpy.ops.file.execute(need_active=False)`  
Execute selected file

**Parameters** `need_active` (*boolean, (optional)*) – Need Active, Only execute if there’s an active selected file in the file list

`bpy.ops.file.fileenum(increment=1)`  
Increment number in filename

**Parameters** `increment` (*int in [-100, 100], (optional)*) – Increment

`bpy.ops.file.find_missing_files` (*filepath=""*, *filter\_blender=False*, *filter\_image=False*, *filter\_movie=False*, *filter\_python=False*, *filter\_font=False*, *filter\_sound=False*, *filter\_text=False*, *filter\_btx=False*, *filter\_collada=False*, *filter\_folder=False*, *filemode=9*)

Undocumented ([contribute](#))

#### Parameters

- **filepath** (*string, (optional)*) – File Path, Path to file
- **filter\_blender** (*boolean, (optional)*) – Filter .blend files
- **filter\_image** (*boolean, (optional)*) – Filter image files
- **filter\_movie** (*boolean, (optional)*) – Filter movie files
- **filter\_python** (*boolean, (optional)*) – Filter python files
- **filter\_font** (*boolean, (optional)*) – Filter font files
- **filter\_sound** (*boolean, (optional)*) – Filter sound files
- **filter\_text** (*boolean, (optional)*) – Filter text files
- **filter\_btx** (*boolean, (optional)*) – Filter btx files
- **filter\_collada** (*boolean, (optional)*) – Filter COLLADA files
- **filter\_folder** (*boolean, (optional)*) – Filter folders
- **filemode** (*int in [1, 9], (optional)*) – File Browser Mode, The setting for the file browser mode to load a .blend file, a library or a special file

`bpy.ops.file.hidedot` ()  
Toggle hide hidden dot files

`bpy.ops.file.highlight` ()  
Highlight selected file(s)

`bpy.ops.file.make_paths_absolute` ()  
Undocumented ([contribute](#))

`bpy.ops.file.make_paths_relative` ()  
Undocumented ([contribute](#))

`bpy.ops.file.next` ()  
Move to next folder

`bpy.ops.file.pack_all` ()  
Undocumented ([contribute](#))

`bpy.ops.file.parent` ()  
Move to parent directory

`bpy.ops.file.previous` ()  
Move to previous folder

`bpy.ops.file.refresh` ()  
Refresh the file list

`bpy.ops.file.rename` ()  
Rename file or file directory

`bpy.ops.file.report_missing_files` ()  
Undocumented ([contribute](#))

`bpy.ops.file.select` (*extend=False, fill=False*)  
Activate/select file

#### Parameters

- **extend** (*boolean, (optional)*) – Extend, Extend selection instead of deselecting everything first
- **fill** (*boolean, (optional)*) – Fill, Select everything beginning with the last selection

`bpy.ops.file.select_all_toggle` ()  
Select/deselect all files

`bpy.ops.file.select_bookmark` (*dir=""*)  
Select a bookmarked directory

**Parameters** *dir* (*string, (optional)*) – Dir

`bpy.ops.file.select_border` (*gesture\_mode=0, xmin=0, xmax=0, ymin=0, ymax=0*)  
Activate/select the file(s) contained in the border

#### Parameters

- **gesture\_mode** (*int in [-inf, inf], (optional)*) – Gesture Mode
- **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

`bpy.ops.file.smoothscroll` ()  
Smooth scroll to make editable file visible

`bpy.ops.file.unpack_all` (*method='USE\_LOCAL'*)  
Undocumented ([contribute](#))

**Parameters** *method* (*enum in ['USE\_LOCAL', 'WRITE\_LOCAL', 'USE\_ORIGINAL', 'WRITE\_ORIGINAL', 'KEEP', 'ASK'], (optional)*) – Method, How to unpack

## Fluid Operators

`bpy.ops.fluid.bake` ()  
Bake fluid simulation

## Font Operators

`bpy.ops.font.buffer_paste` ()  
Paste text from OS buffer

`bpy.ops.font.case_set` (*case='LOWER'*)  
Set font case

**Parameters** *case* (*enum in ['LOWER', 'UPPER'], (optional)*) – Case, Lower or upper case

`bpy.ops.font.case_toggle` ()  
Toggle font case

`bpy.ops.font.change_character` (*delta=1*)  
Change font character code

**Parameters** **delta** (*int in [-255, 255], (optional)*) – Delta, Number to increase or decrease character code with

`bpy.ops.font.change_spacing(delta=1)`

Change font spacing

**Parameters** **delta** (*int in [-20, 20], (optional)*) – Delta, Amount to decrease or increase character spacing with

`bpy.ops.font.delete(type='ALL')`

Delete text by cursor position

**Parameters** **type** (*enum in ['ALL', 'NEXT\_CHARACTER', 'PREVIOUS\_CHARACTER', 'SELECTION', 'NEXT\_OR\_SELECTION', 'PREVIOUS\_OR\_SELECTION'], (optional)*) – Type, Which part of the text to delete

`bpy.ops.font.file_paste(filepath="", filter_blender=False, filter_image=False, filter_movie=False, filter_python=False, filter_font=False, filter_sound=False, filter_text=True, filter_btx=False, filter_collada=False, filter_folder=True, filemode=9)`

Paste contents from file

**Parameters**

- **filepath** (*string, (optional)*) – File Path, Path to file
- **filter\_blender** (*boolean, (optional)*) – Filter .blend files
- **filter\_image** (*boolean, (optional)*) – Filter image files
- **filter\_movie** (*boolean, (optional)*) – Filter movie files
- **filter\_python** (*boolean, (optional)*) – Filter python files
- **filter\_font** (*boolean, (optional)*) – Filter font files
- **filter\_sound** (*boolean, (optional)*) – Filter sound files
- **filter\_text** (*boolean, (optional)*) – Filter text files
- **filter\_btx** (*boolean, (optional)*) – Filter btx files
- **filter\_collada** (*boolean, (optional)*) – Filter COLLADA files
- **filter\_folder** (*boolean, (optional)*) – Filter folders
- **filemode** (*int in [1, 9], (optional)*) – File Browser Mode, The setting for the file browser mode to load a .blend file, a library or a special file

`bpy.ops.font.insert_lorem()`

Insert placeholder text

`bpy.ops.font.line_break(ctrl=False)`

Insert line break at cursor position

**Parameters** **ctrl** (*boolean, (optional)*) – Ctrl

`bpy.ops.font.move(type='LINE_BEGIN')`

Move cursor to position type

**Parameters** **type** (*enum in ['LINE\_BEGIN', 'LINE\_END', 'PREVIOUS\_CHARACTER', 'NEXT\_CHARACTER', 'PREVIOUS\_WORD', 'NEXT\_WORD', 'PREVIOUS\_LINE', 'NEXT\_LINE', 'PREVIOUS\_PAGE', 'NEXT\_PAGE'], (optional)*) – Type, Where to move cursor to



`bpy.ops.font.move_select` (*type*='LINE\_BEGIN')

Make selection from current cursor position to new cursor position type

**Parameters** *type* (enum in ['LINE\_BEGIN', 'LINE\_END', 'PREVIOUS\_CHARACTER', 'NEXT\_CHARACTER', 'PREVIOUS\_WORD', 'NEXT\_WORD', 'PREVIOUS\_LINE', 'NEXT\_LINE', 'PREVIOUS\_PAGE', 'NEXT\_PAGE'], (optional)) – Type, Where to move cursor to, to make a selection

`bpy.ops.font.open` (*filepath*="", *filter\_blender*=False, *filter\_image*=False, *filter\_movie*=False, *filter\_python*=False, *filter\_font*=True, *filter\_sound*=False, *filter\_text*=False, *filter\_btx*=False, *filter\_collada*=False, *filter\_folder*=True, *filemode*=9, *relative\_path*=True)

Undocumented ([contribute](#))

#### Parameters

- **filepath** (*string*, (optional)) – File Path, Path to file
- **filter\_blender** (*boolean*, (optional)) – Filter .blend files
- **filter\_image** (*boolean*, (optional)) – Filter image files
- **filter\_movie** (*boolean*, (optional)) – Filter movie files
- **filter\_python** (*boolean*, (optional)) – Filter python files
- **filter\_font** (*boolean*, (optional)) – Filter font files
- **filter\_sound** (*boolean*, (optional)) – Filter sound files
- **filter\_text** (*boolean*, (optional)) – Filter text files
- **filter\_btx** (*boolean*, (optional)) – Filter btx files
- **filter\_collada** (*boolean*, (optional)) – Filter COLLADA files
- **filter\_folder** (*boolean*, (optional)) – Filter folders
- **filemode** (*int* in [1, 9], (optional)) – File Browser Mode, The setting for the file browser mode to load a .blend file, a library or a special file
- **relative\_path** (*boolean*, (optional)) – Relative Path, Select the file relative to the blend file

`bpy.ops.font.style_set` (*style*='BOLD', *clear*=False)

Set font style

#### Parameters

- **style** (enum in ['BOLD', 'ITALIC', 'UNDERLINE', 'SMALL\_CAPS'], (optional)) – Style, Style to set selection to
- **clear** (*boolean*, (optional)) – Clear, Clear style rather than setting it

`bpy.ops.font.style_toggle` (*style*='BOLD')

Toggle font style

**Parameters** *style* (enum in ['BOLD', 'ITALIC', 'UNDERLINE', 'SMALL\_CAPS'], (optional)) – Style, Style to set selection to

`bpy.ops.font.text_copy` ()

Copy selected text to clipboard

`bpy.ops.font.text_cut` ()

Cut selected text to clipboard

`bpy.ops.font.text_insert` (*text*="", *accent*=False)

Insert text at cursor position

### Parameters

- **text** (*string, (optional)*) – Text, Text to insert at the cursor position
- **accent** (*boolean, (optional)*) – Accent mode, Next typed character will strike through previous, for special character input

`bpy.ops.font.text_paste()`  
Paste text from clipboard

`bpy.ops.font.textbox_add()`  
Add a new text box

`bpy.ops.font.textbox_remove(index=0)`  
Remove the textbox

**Parameters** `index` (*int in [0, inf], (optional)*) – Index, The current text box

`bpy.ops.font.unlink()`  
Unlink active font data block

### Gpencil Operators

`bpy.ops.gpencil.active_frame_delete()`  
Delete the active frame for the active Grease Pencil datablock

`bpy.ops.gpencil.convert(type='PATH')`  
Convert the active Grease Pencil layer to a new Object

**Parameters** `type` (*enum in ['PATH', 'CURVE'], (optional)*) – Type

`bpy.ops.gpencil.data_add()`  
Add new Grease Pencil datablock

`bpy.ops.gpencil.data_unlink()`  
Unlink active Grease Pencil datablock

`bpy.ops.gpencil.draw(mode='DRAW', stroke=None)`  
Make annotations on the active data

### Parameters

- **mode** (*enum in ['DRAW', 'DRAW\_STRAIGHT', 'DRAW\_POLY', 'ERASER'], (optional)*) – Mode, Way to interpret mouse movements
- **stroke** (*bpy\_prop\_collection of OperatorStrokeElement, (optional)*) – Stroke

`bpy.ops.gpencil.layer_add()`  
Add new Grease Pencil layer for the active Grease Pencil datablock

### Graph Operators

`bpy.ops.graph.bake()`  
Bake selected F-Curves to a set of sampled points defining a similar curve

`bpy.ops.graph.clean(threshold=0.001)`  
Simplify F-Curves by removing closely spaced keyframes

**Parameters** `threshold` (*float in [0, inf], (optional)*) – Threshold

`bpy.ops.graph.click_insert` (*frame=1.0, value=1.0*)  
 Insert new keyframe at the cursor position for the active F-Curve

#### Parameters

- **frame** (*float in [-inf, inf], (optional)*) – Frame Number, Frame to insert keyframe on
- **value** (*float in [-inf, inf], (optional)*) – Value, Value for keyframe on

`bpy.ops.graph.clickselect` (*extend=False, column=False, curves=False*)  
 Select keyframes by clicking on them

#### Parameters

- **extend** (*boolean, (optional)*) – Extend Select
- **column** (*boolean, (optional)*) – Column Select, Select all keyframes that occur on the same frame as the one under the mouse
- **curves** (*boolean, (optional)*) – Only Curves, Select all the keyframes in the curve

`bpy.ops.graph.copy` ()  
 Copy selected keyframes to the copy/paste buffer

`bpy.ops.graph.cursor_set` (*frame=0, value=0.0*)  
 Interactively set the current frame number and value cursor

#### Parameters

- **frame** (*int in [-300000, 300000], (optional)*) – Frame
- **value** (*float in [1.17549e-38, inf], (optional)*) – Value

`bpy.ops.graph.delete` ()  
 Remove all selected keyframes

`bpy.ops.graph.duplicate` (*mode='TRANSLATION'*)  
 Make a copy of all selected keyframes

**Parameters mode** (*enum in ['INIT', 'DUMMY', 'TRANSLATION', 'ROTATION', 'RESIZE', 'TO-SPHERE', 'SHEAR', 'WARP', 'SHRINKFATTEN', 'TILT', 'TRACKBALL', 'PUSHPULL', 'CREASE', 'MIRROR', 'BONE\_SIZE', 'BONE\_ENVELOPE', 'CURVE\_SHRINKFATTEN', 'BONE\_ROLL', 'TIME\_TRANSLATE', 'TIME\_SLIDE', 'TIME\_SCALE', 'TIME\_EXTEND', 'BAKE\_TIME', 'BEVEL', 'BWEIGHT', 'ALIGN', 'EDGESLIDE', 'SEQSLIDE'], (optional)*) – Mode

`bpy.ops.graph.duplicate_move` (*GRAPH\_OT\_duplicate=None, TRANSFORM\_OT\_transform=None*)  
 Undocumented ([contribute](#))

#### Parameters

- **GRAPH\_OT\_duplicate** (*GRAPH\_OT\_duplicate, (optional)*) – Duplicate Keyframes, Make a copy of all selected keyframes
- **TRANSFORM\_OT\_transform** (*TRANSFORM\_OT\_transform, (optional)*) – Transform, Transform selected items by mode type

`bpy.ops.graph.euler_filter` ()  
 Fixes the most common causes of gimbal lock in the selected Euler Rotation F-Curves

`bpy.ops.graph.extrapolation_type` (*type='CONSTANT'*)  
 Set extrapolation mode for selected F-Curves

**Parameters type** (*enum in ['CONSTANT', 'LINEAR', 'MAKE\_CYCLIC', 'CLEAR\_CYCLIC'], (optional)*) – Type

- **CONSTANT** Constant Extrapolation.
- **LINEAR** Linear Extrapolation.
- **MAKE\_CYCLIC** Make Cyclic (F-Modifier), Add Cycles F-Modifier if one doesn't exist already.
- **CLEAR\_CYCLIC** Clear Cyclic (F-Modifier), Remove Cycles F-Modifier if not needed anymore.

`bpy.ops.graph.fmodifier_add` (*type='NULL', only\_active=True*)  
Add F-Modifiers to the selected F-Curves

**Parameters**

- **type** (*enum in ['NULL', 'GENERATOR', 'FNGENERATOR', 'ENVELOPE', 'CYCLES', 'NOISE', 'FILTER', 'LIMITS', 'STEPPED'], (optional)*) – Type
- **only\_active** (*boolean, (optional)*) – Only Active, Only add F-Modifier to active F-Curve

`bpy.ops.graph.fmodifier_copy` ()  
Copy the F-Modifier(s) of the active F-Curve

`bpy.ops.graph.fmodifier_paste` ()  
Add copied F-Modifiers to the selected F-Curves

`bpy.ops.graph.frame_jump` ()  
Set the current frame to the average frame of the selected keyframes

`bpy.ops.graph.ghost_curves_clear` ()  
Clear F-Curve snapshots (Ghosts) for active Graph Editor

`bpy.ops.graph.ghost_curves_create` ()  
Create snapshot (Ghosts) of selected F-Curves as background aid for active Graph Editor

`bpy.ops.graph.handle_type` (*type='FREE'*)  
Set type of handle for selected keyframes

**Parameters** **type** (*enum in ['FREE', 'VECTOR', 'ALIGNED', 'AUTO', 'AUTO\_CLAMPED'], (optional)*) – Type

- **FREE** Free.
- **VECTOR** Vector.
- **ALIGNED** Aligned.
- **AUTO** Automatic.
- **AUTO\_CLAMPED** Auto Clamped, Auto handles clamped to not overshoot.

`bpy.ops.graph.handles_view_toggle` ()  
Toggle whether handles are drawn on all keyframes that need them

`bpy.ops.graph.interpolation_type` (*type='CONSTANT'*)  
Set interpolation mode for the F-Curve segments starting from the selected keyframes

**Parameters** **type** (*enum in ['CONSTANT', 'LINEAR', 'BEZIER'], (optional)*) – Type

`bpy.ops.graph.keyframe_insert` (*type='ALL'*)  
Insert keyframes for the specified channels

**Parameters** **type** (*enum in ['ALL', 'SEL'], (optional)*) – Type

`bpy.ops.graph.mirror` (*type='CFRA'*)  
Flip selected keyframes over the selected mirror line

**Parameters** `type` (*enum in ['CFRA', 'VALUE', 'YAXIS', 'XAXIS', 'MARKER'], (optional)*) – Type

`bpy.ops.graph.paste` (*offset='START', merge='MIX'*)

Paste keyframes from copy/paste buffer for the selected channels, starting on the current frame

#### Parameters

- **offset** (*enum in ['START', 'END', 'RELATIVE', 'NONE'], (optional)*) – Offset, Paste time offset of keys
  - `START` Frame Start, Paste keys starting at current frame.
  - `END` Frame End, Paste keys ending at current frame.
  - `RELATIVE` Frame Relative, Paste keys relative to the current frame when copying.
  - `NONE` No Offset, Paste keys from original time.
- **merge** (*enum in ['MIX', 'OVER\_ALL', 'OVER\_RANGE', 'OVER\_RANGE\_ALL'], (optional)*) – Type, Method of merging pasted keys and existing
  - `MIX` Mix, Overlay existing with new keys.
  - `OVER_ALL` Overwrite All, Replace all keys.
  - `OVER_RANGE` Overwrite Range, Overwrite keys in pasted range.
  - `OVER_RANGE_ALL` Overwrite Entire Range, Overwrite keys in pasted range, using the range of all copied keys.

`bpy.ops.graph.previewrange_set` ()

Automatically set Preview Range based on range of keyframes

`bpy.ops.graph.properties` ()

Toggle display properties panel

`bpy.ops.graph.sample` ()

Add keyframes on every frame between the selected keyframes

`bpy.ops.graph.select_all_toggle` (*invert=False*)

Toggle selection of all keyframes

**Parameters** `invert` (*boolean, (optional)*) – Invert

`bpy.ops.graph.select_border` (*gesture\_mode=0, xmin=0, xmax=0, ymin=0, ymax=0, axis\_range=False, include\_handles=False*)

Select all keyframes within the specified region

#### Parameters

- **gesture\_mode** (*int in [-inf, inf], (optional)*) – Gesture Mode
- **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
- **axis\_range** (*boolean, (optional)*) – Axis Range
- **include\_handles** (*boolean, (optional)*) – Include Handles, Are handles tested individually against the selection criteria

`bpy.ops.graph.select_column` (*mode='KEYS'*)

Select all keyframes on the specified frame(s)

**Parameters mode** (*enum in ['KEYS', 'CFRA', 'MARKERS\_COLUMN', 'MARKERS\_BETWEEN'], (optional)*) – Mode

`bpy.ops.graph.select_leftright` (*mode='CHECK', extend=False*)  
Select keyframes to the left or the right of the current frame

**Parameters**

- **mode** (*enum in ['CHECK', 'LEFT', 'RIGHT'], (optional)*) – Mode
- **extend** (*boolean, (optional)*) – Extend Select

`bpy.ops.graph.select_less` ()  
Deselect keyframes on ends of selection islands

`bpy.ops.graph.select_linked` ()  
Select keyframes occurring in the same F-Curves as selected ones

`bpy.ops.graph.select_more` ()  
Select keyframes beside already selected ones

`bpy.ops.graph.smooth` ()  
Apply weighted moving means to make selected F-Curves less bumpy

`bpy.ops.graph.snap` (*type='CFRA'*)  
Snap selected keyframes to the chosen times/values

**Parameters type** (*enum in ['CFRA', 'VALUE', 'NEAREST\_FRAME', 'NEAREST\_SECOND', 'NEAREST\_MARKER', 'HORIZONTAL'], (optional)*) – Type

`bpy.ops.graph.sound_bake` (*filepath=""*, *filter\_blender=False*, *filter\_image=False*, *filter\_movie=True*, *filter\_python=False*, *filter\_font=False*, *filter\_sound=True*, *filter\_text=False*, *filter\_btx=False*, *filter\_collada=False*, *filter\_folder=True*, *filemode=9*, *low=0.0*, *high=100000.0*, *attack=0.005*, *release=0.2*, *threshold=0.0*, *accumulate=False*, *use\_additive=False*, *square=False*, *sthreshold=0.1*)  
Bakes a sound wave to selected F-Curves

**Parameters**

- **filepath** (*string, (optional)*) – File Path, Path to file
- **filter\_blender** (*boolean, (optional)*) – Filter .blend files
- **filter\_image** (*boolean, (optional)*) – Filter image files
- **filter\_movie** (*boolean, (optional)*) – Filter movie files
- **filter\_python** (*boolean, (optional)*) – Filter python files
- **filter\_font** (*boolean, (optional)*) – Filter font files
- **filter\_sound** (*boolean, (optional)*) – Filter sound files
- **filter\_text** (*boolean, (optional)*) – Filter text files
- **filter\_btx** (*boolean, (optional)*) – Filter btx files
- **filter\_collada** (*boolean, (optional)*) – Filter COLLADA files
- **filter\_folder** (*boolean, (optional)*) – Filter folders
- **filemode** (*int in [1, 9], (optional)*) – File Browser Mode, The setting for the file browser mode to load a .blend file, a library or a special file
- **low** (*float in [0, 100000], (optional)*) – Lowest frequency

- **high** (*float in [0, 100000], (optional)*) – Highest frequency
- **attack** (*float in [0, 2], (optional)*) – Attack time
- **release** (*float in [0, 5], (optional)*) – Release time
- **threshold** (*float in [0, 1], (optional)*) – Threshold
- **accumulate** (*boolean, (optional)*) – Accumulate
- **use\_additive** (*boolean, (optional)*) – Additive
- **square** (*boolean, (optional)*) – Square
- **sthreshold** (*float in [0, 1], (optional)*) – Square Threshold

`bpy.ops.graph.view_all()`  
Reset viewable area to show full keyframe range

`bpy.ops.graph.view_selected()`  
Reset viewable area to show selected keyframe range

## Group Operators

`bpy.ops.group.create(name="Group")`  
Create an object group from selected objects

**Parameters** `name` (*string, (optional)*) – Name, Name of the new group

`bpy.ops.group.objects_add_active()`  
Add the object to an object group that contains the active object

`bpy.ops.group.objects_remove()`  
Remove selected objects from all groups

`bpy.ops.group.objects_remove_active()`  
Remove the object from an object group that contains the active object

## Image Operators

`bpy.ops.image.curves_point_set(point='BLACK_POINT')`  
Undocumented (contribute)

**Parameters** `point` (*enum in ['BLACK\_POINT', 'WHITE\_POINT'], (optional)*) – Point, Set black point or white point for curves

`bpy.ops.image.cycle_render_slot(reverse=False)`  
Undocumented (contribute)

**Parameters** `reverse` (*boolean, (optional)*) – Cycle in Reverse

`bpy.ops.image.external_edit(filepath="")`  
Edit image in an external application

**Parameters** `filepath` (*string, (optional)*) – File Path, Path to an image file

**File** `startup/bl_operators/image.py:62`

`bpy.ops.image.invert(invert_r=False, invert_g=False, invert_b=False, invert_a=False)`  
Undocumented (contribute)

**Parameters**

- **invert\_r** (*boolean, (optional)*) – Red, Invert Red Channel

- **invert\_g** (*boolean, (optional)*) – Green, Invert Green Channel
- **invert\_b** (*boolean, (optional)*) – Blue, Invert Blue Channel
- **invert\_a** (*boolean, (optional)*) – Alpha, Invert Alpha Channel

`bpy.ops.image.new` (*name="untitled", width=1024, height=1024, color=(0.0, 0.0, 0.0, 1.0), alpha=True, uv\_test\_grid=False, float=False*)

Create a new image

#### Parameters

- **name** (*string, (optional)*) – Name, Image datablock name
- **width** (*int in [1, inf], (optional)*) – Width, Image width
- **height** (*int in [1, inf], (optional)*) – Height, Image height
- **color** (*float array of 4 items in [0, inf], (optional)*) – Color, Default fill color
- **alpha** (*boolean, (optional)*) – Alpha, Create an image with an alpha channel
- **uv\_test\_grid** (*boolean, (optional)*) – UV Test Grid, Fill the image with a grid for UV map testing
- **float** (*boolean, (optional)*) – 32 bit Float, Create image with 32 bit floating point bit depth

`bpy.ops.image.open` (*filepath="", filter\_blender=False, filter\_image=True, filter\_movie=True, filter\_python=False, filter\_font=False, filter\_sound=False, filter\_text=False, filter\_btx=False, filter\_collada=False, filter\_folder=True, filemode=9, relative\_path=True*)

Open image

#### Parameters

- **filepath** (*string, (optional)*) – File Path, Path to file
- **filter\_blender** (*boolean, (optional)*) – Filter .blend files
- **filter\_image** (*boolean, (optional)*) – Filter image files
- **filter\_movie** (*boolean, (optional)*) – Filter movie files
- **filter\_python** (*boolean, (optional)*) – Filter python files
- **filter\_font** (*boolean, (optional)*) – Filter font files
- **filter\_sound** (*boolean, (optional)*) – Filter sound files
- **filter\_text** (*boolean, (optional)*) – Filter text files
- **filter\_btx** (*boolean, (optional)*) – Filter btx files
- **filter\_collada** (*boolean, (optional)*) – Filter COLLADA files
- **filter\_folder** (*boolean, (optional)*) – Filter folders
- **filemode** (*int in [1, 9], (optional)*) – File Browser Mode, The setting for the file browser mode to load a .blend file, a library or a special file
- **relative\_path** (*boolean, (optional)*) – Relative Path, Select the file relative to the blend file

`bpy.ops.image.pack` (*as\_png=False*)

Pack an image as embedded data into the .blend file

**Parameters** *as\_png* (*boolean, (optional)*) – Pack As PNG, Pack image as lossless PNG

`bpy.ops.image.project_apply` ()

Project edited image back onto the object



**File** `startup/bl_operators/image.py:211`

`bpy.ops.image.project_edit()`  
Edit a snapshot of the viewport in an external image editor

**File** `startup/bl_operators/image.py:140`

`bpy.ops.image.properties()`  
Toggle display properties panel

`bpy.ops.image.record_composite()`  
Undocumented ([contribute](#))

`bpy.ops.image.reload()`  
Undocumented ([contribute](#))

`bpy.ops.image.replace(filepath="", filter_blender=False, filter_image=True, filter_movie=True, filter_python=False, filter_font=False, filter_sound=False, filter_text=False, filter_btx=False, filter_collada=False, filter_folder=True, filemode=9, relative_path=True)`  
Undocumented ([contribute](#))

#### Parameters

- **filepath** (*string, (optional)*) – File Path, Path to file
- **filter\_blender** (*boolean, (optional)*) – Filter .blend files
- **filter\_image** (*boolean, (optional)*) – Filter image files
- **filter\_movie** (*boolean, (optional)*) – Filter movie files
- **filter\_python** (*boolean, (optional)*) – Filter python files
- **filter\_font** (*boolean, (optional)*) – Filter font files
- **filter\_sound** (*boolean, (optional)*) – Filter sound files
- **filter\_text** (*boolean, (optional)*) – Filter text files
- **filter\_btx** (*boolean, (optional)*) – Filter btx files
- **filter\_collada** (*boolean, (optional)*) – Filter COLLADA files
- **filter\_folder** (*boolean, (optional)*) – Filter folders
- **filemode** (*int in [1, 9], (optional)*) – File Browser Mode, The setting for the file browser mode to load a .blend file, a library or a special file
- **relative\_path** (*boolean, (optional)*) – Relative Path, Select the file relative to the blend file

`bpy.ops.image.sample()`  
Undocumented ([contribute](#))

`bpy.ops.image.sample_line(xstart=0, xend=0, ystart=0, yend=0, cursor=1002)`  
Undocumented ([contribute](#))

#### Parameters

- **xstart** (*int in [-inf, inf], (optional)*) – X Start
- **xend** (*int in [-inf, inf], (optional)*) – X End
- **ystart** (*int in [-inf, inf], (optional)*) – Y Start
- **yend** (*int in [-inf, inf], (optional)*) – Y End

- **cursor** (*int in [0, inf], (optional)*) – Cursor, Mouse cursor style to use during the modal operator

`bpy.ops.image.save()`

Undocumented ([contribute](#))

`bpy.ops.image.save_as(file_format='PNG', color_mode='RGB', file_quality=90, filepath="", check_existing=True, filter_blender=False, filter_image=True, filter_movie=True, filter_python=False, filter_font=False, filter_sound=False, filter_text=False, filter_btx=False, filter_collada=False, filter_folder=True, filemode=9, relative_path=True, copy=False)`

Undocumented ([contribute](#))

### Parameters

- **file\_format** (*enum in ['TARGA', 'TARGA RAW', 'PNG', 'BMP', 'JPEG', 'IRIS', 'TIFF', 'OPENEXR', 'MULTILAYER'], (optional)*) – File Type, File type to save image as
- **color\_mode** (*enum in ['BW', 'RGB', 'RGBA'], (optional)*) – Channels, Image channels to save
  - BW BW, Images get saved in 8 bits grayscale (only PNG, JPEG, TGA, TIF).
  - RGB RGB, Images are saved with RGB (color) data.
  - RGBA RGBA, Images are saved with RGB and Alpha data (if supported).
- **file\_quality** (*int in [0, 100], (optional)*) – Quality
- **filepath** (*string, (optional)*) – File Path, Path to file
- **check\_existing** (*boolean, (optional)*) – Check Existing, Check and warn on overwriting existing files
- **filter\_blender** (*boolean, (optional)*) – Filter .blend files
- **filter\_image** (*boolean, (optional)*) – Filter image files
- **filter\_movie** (*boolean, (optional)*) – Filter movie files
- **filter\_python** (*boolean, (optional)*) – Filter python files
- **filter\_font** (*boolean, (optional)*) – Filter font files
- **filter\_sound** (*boolean, (optional)*) – Filter sound files
- **filter\_text** (*boolean, (optional)*) – Filter text files
- **filter\_btx** (*boolean, (optional)*) – Filter btx files
- **filter\_collada** (*boolean, (optional)*) – Filter COLLADA files
- **filter\_folder** (*boolean, (optional)*) – Filter folders
- **filemode** (*int in [1, 9], (optional)*) – File Browser Mode, The setting for the file browser mode to load a .blend file, a library or a special file
- **relative\_path** (*boolean, (optional)*) – Relative Path, Select the file relative to the blend file
- **copy** (*boolean, (optional)*) – Copy, Create a new image file without modifying the current image in blender

`bpy.ops.image.save_dirty()`

Save all modified textures

**File** `startup/bl_operators/image.py:115`

`bpy.ops.image.save_sequence()`  
Undocumented (contribute)

`bpy.ops.image.scopes()`  
Toggle display scopes panel

`bpy.ops.image.unpack(method='USE_LOCAL', id='')`  
Save an image packed in the .blend file to disk

#### Parameters

- **method** (enum in [`'USE_LOCAL'`, `'WRITE_LOCAL'`, `'USE_ORIGINAL'`, `'WRITE_ORIGINAL'`], (optional)) – Method, How to unpack
- **id** (string, (optional)) – Image Name, Image datablock name to unpack

`bpy.ops.image.view_all()`  
Undocumented (contribute)

`bpy.ops.image.view_ndof()`  
Undocumented (contribute)

`bpy.ops.image.view_pan(offset=(0.0, 0.0))`  
Undocumented (contribute)

**Parameters** **offset** (float array of 2 items in  $[-inf, inf]$ , (optional)) – Offset, Offset in floating point units, 1.0 is the width and height of the image

`bpy.ops.image.view_selected()`  
Undocumented (contribute)

`bpy.ops.image.view_zoom(factor=0.0)`  
Undocumented (contribute)

**Parameters** **factor** (float in  $[0, inf]$ , (optional)) – Factor, Zoom factor, values higher than 1.0 zoom in, lower values zoom out

`bpy.ops.image.view_zoom_in()`  
Undocumented (contribute)

`bpy.ops.image.view_zoom_out()`  
Undocumented (contribute)

`bpy.ops.image.view_zoom_ratio(ratio=0.0)`  
Undocumented (contribute)

**Parameters** **ratio** (float in  $[0, inf]$ , (optional)) – Ratio, Zoom ratio, 1.0 is 1:1, higher is zoomed in, lower is zoomed out

## Import Anim Operators

`bpy.ops.import_anim.bvh(filepath="", filter_glob="*.bvh", target='ARMATURE', global_scale=1.0, frame_start=1, use_cyclic=False, rotate_mode='NATIVE', axis_forward='-Z', axis_up='Y')`  
Load a BVH motion capture file

#### Parameters

- **filepath** (string, (optional)) – File Path, Filepath used for importing the file
- **target** (enum in [`'ARMATURE'`, `'OBJECT'`], (optional)) – Target, Import target type
- **global\_scale** (float in  $[0.0001, 1e+06]$ , (optional)) – Scale, Scale the BVH by this value

- **frame\_start** (*int in [-inf, inf], (optional)*) – Start Frame, Starting frame for the animation
- **use\_cyclic** (*boolean, (optional)*) – Loop, Loop the animation playback
- **rotate\_mode** (*enum in ['QUATERNION', 'NATIVE', 'XYZ', 'XZY', 'YXZ', 'YZX', 'ZXY', 'ZYZ'], (optional)*) – Rotation, Rotation conversion
  - QUATERNION Quaternion, Convert rotations to quaternions.
  - NATIVE Euler (Native), Use the rotation order defined in the BVH file.
  - XYZ Euler (XYZ), Convert rotations to euler XYZ.
  - XZY Euler (XZY), Convert rotations to euler XZY.
  - YXZ Euler (YXZ), Convert rotations to euler YXZ.
  - YZX Euler (YZX), Convert rotations to euler YZX.
  - ZXY Euler (ZXY), Convert rotations to euler ZXY.
  - ZYZ Euler (ZYZ), Convert rotations to euler ZYZ.
- **axis\_forward** (*enum in ['X', 'Y', 'Z', '-X', '-Y', '-Z'], (optional)*) – Forward
- **axis\_up** (*enum in ['X', 'Y', 'Z', '-X', '-Y', '-Z'], (optional)*) – Up

**File** `addons/io_anim_bvh/__init__.py:130`

## Import Curve Operators

```
bpy.ops.import_curve.svg(filepath="", filter_glob="*.svg")
```

Load a SVG file

**Parameters** **filepath** (*string, (optional)*) – File Path, Filepath used for importing the file

**File** `addons/io_curve_svg/__init__.py:58`

## Import Mesh Operators

```
bpy.ops.import_mesh.ply(filepath="", files=None, directory="", filter_glob="*.ply")
```

Load a PLY geometry file

### Parameters

- **filepath** (*string, (optional)*) – File Path, Filepath used for importing the file
- **files** (*bpy\_prop\_collection of OperatorFileListElement, (optional)*) – File Path, File path used for importing the PLY file

**File** `addons/io_mesh_ply/__init__.py:67`

```
bpy.ops.import_mesh.stl(filepath="", filter_glob="*.stl", files=None, directory="")
```

Load STL triangle mesh data

### Parameters

- **filepath** (*string, (optional)*) – File Path, Filepath used for importing the file
- **files** (*bpy\_prop\_collection of OperatorFileListElement, (optional)*) – File Path

**File** `addons/io_mesh_stl/__init__.py:86`

## Import Scene Operators

```
bpy.ops.import_scene.autodesk_3ds(filepath="", filter_glob="*.3ds", constrain_size=10.0,
                                  use_image_search=True, use_apply_transform=True,
                                  axis_forward='Y', axis_up='Z')
```

Import from 3DS file format (.3ds)

### Parameters

- **filepath** (*string, (optional)*) – File Path, Filepath used for importing the file
- **constrain\_size** (*float in [0, 1000], (optional)*) – Size Constraint, Scale the model by 10 until it reaches the size constraint. Zero Disables
- **use\_image\_search** (*boolean, (optional)*) – Image Search, Search subdirectories for any associated images (Warning, may be slow)
- **use\_apply\_transform** (*boolean, (optional)*) – Apply Transform, Workaround for object transformations importing incorrectly
- **axis\_forward** (*enum in ['X', 'Y', 'Z', '-X', '-Y', '-Z'], (optional)*) – Forward
- **axis\_up** (*enum in ['X', 'Y', 'Z', '-X', '-Y', '-Z'], (optional)*) – Up

**File** `addons/io_scene_3ds/__init__.py:107`

```
bpy.ops.import_scene.obj(filepath="", filter_glob="*.obj;*.mtl", use_ngons=True,
                        use_edges=True, use_smooth_groups=True, use_split_objects=True,
                        use_split_groups=True, use_groups_as_vgroups=False,
                        use_image_search=True, split_mode='ON', global_clamp_size=0.0,
                        axis_forward='-Z', axis_up='Y')
```

Load a Wavefront OBJ File

### Parameters

- **filepath** (*string, (optional)*) – File Path, Filepath used for importing the file
- **use\_ngons** (*boolean, (optional)*) – NGons, Import faces with more than 4 verts as fgons
- **use\_edges** (*boolean, (optional)*) – Lines, Import lines and faces with 2 verts as edge
- **use\_smooth\_groups** (*boolean, (optional)*) – Smooth Groups, Surround smooth groups by sharp edges
- **use\_split\_objects** (*boolean, (optional)*) – Object, Import OBJ Objects into Blender Objects
- **use\_split\_groups** (*boolean, (optional)*) – Group, Import OBJ Groups into Blender Objects
- **use\_groups\_as\_vgroups** (*boolean, (optional)*) – Poly Groups, Import OBJ groups as vertex groups
- **use\_image\_search** (*boolean, (optional)*) – Image Search, Search subdirs for any associated images (Warning, may be slow)
- **split\_mode** (*enum in ['ON', 'OFF'], (optional)*) – Split
  - ON Split, Split geometry, omits unused verts.
  - OFF Keep Vert Order, Keep vertex order from file.
- **global\_clamp\_size** (*float in [0, 1000], (optional)*) – Clamp Scale, Clamp the size to this maximum (Zero to Disable)
- **axis\_forward** (*enum in ['X', 'Y', 'Z', '-X', '-Y', '-Z'], (optional)*) – Forward
- **axis\_up** (*enum in ['X', 'Y', 'Z', '-X', '-Y', '-Z'], (optional)*) – Up

**File** `addons/io_scene_obj/__init__.py:147`

```
bpy.ops.import_scene.x3d(filepath="", filter_glob="*.x3d;*.wrl", axis_forward='Z',  
axis_up='Y')
```

Import and X3D or VRML file

#### Parameters

- **filepath** (*string, (optional)*) – File Path, Filepath used for importing the file
- **axis\_forward** (*enum in ['X', 'Y', 'Z', '-X', '-Y', '-Z'], (optional)*) – Forward
- **axis\_up** (*enum in ['X', 'Y', 'Z', '-X', '-Y', '-Z'], (optional)*) – Up

**File** `addons/io_scene_x3d/__init__.py:84`

## Info Operators

```
bpy.ops.info.report_copy()
```

Copy selected reports to Clipboard

```
bpy.ops.info.report_delete()
```

Delete selected reports

```
bpy.ops.info.report_replay()
```

Replay selected reports

```
bpy.ops.info.reports_display_update()
```

Undocumented ([contribute](#))

```
bpy.ops.info.select_all_toggle()
```

(de)select all reports

```
bpy.ops.info.select_border(gesture_mode=0, xmin=0, xmax=0, ymin=0, ymax=0)
```

Toggle border selection

#### Parameters

- **gesture\_mode** (*int in [-inf, inf], (optional)*) – Gesture Mode
- **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

```
bpy.ops.info.select_pick(report_index=0)
```

Select reports by index

**Parameters** **report\_index** (*int in [0, inf], (optional)*) – Report, Index of the report

## Lamp Operators

```
bpy.ops.lamp.sunsky_preset_add(name="", remove_active=False)
```

Add a Sky & Atmosphere Preset

**Parameters** **name** (*string, (optional)*) – Name, Name of the preset, used to make the path name

**File** `startup/bl_operators/presets.py:50`

## Lattice Operators

`bpy.ops.lattice.make_regular()`

Set UVW control points a uniform distance apart

`bpy.ops.lattice.select_all(action='TOGGLE')`

Change selection of all UVW control 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.

## Logic Operators

`bpy.ops.logic.actuator_add(type='', name="", object="")`

Add an actuator to the active object

### Parameters

- `type` (*enum in [], (optional)*) – Type, Type of actuator to add
- `name` (*string, (optional)*) – Name, Name of the Actuator to add
- `object` (*string, (optional)*) – Object, Name of the Object to add the Actuator to

`bpy.ops.logic.actuator_move(actuator="", object="", direction='UP')`

Move Actuator

### Parameters

- `actuator` (*string, (optional)*) – Actuator, Name of the actuator to edit
- `object` (*string, (optional)*) – Object, Name of the object the actuator belongs to
- `direction` (*enum in ['UP', 'DOWN'], (optional)*) – Direction, Move Up or Down

`bpy.ops.logic.actuator_remove(actuator="", object="")`

Remove an actuator from the active object

### Parameters

- `actuator` (*string, (optional)*) – Actuator, Name of the actuator to edit
- `object` (*string, (optional)*) – Object, Name of the object the actuator belongs to

`bpy.ops.logic.controller_add(type='LOGIC_AND', name="", object="")`

Add a controller to the active object

### Parameters

- `type` (*enum in ['LOGIC\_AND', 'LOGIC\_OR', 'LOGIC\_NAND', 'LOGIC\_NOR', 'LOGIC\_XOR', 'LOGIC\_XNOR', 'EXPRESSION', 'PYTHON'], (optional)*) – Type, Type of controller to add
  - `LOGIC_AND` And, Logic And.
  - `LOGIC_OR` Or, Logic Or.
  - `LOGIC_NAND` Nand, Logic Nand.

- LOGIC\_NOR Nor, Logic Nor.
- LOGIC\_XOR Xor, Logic Xor.
- LOGIC\_XNOR Xnor, Logic Xnor.
- EXPRESSION Expression.
- PYTHON Python.
- **name** (*string, (optional)*) – Name, Name of the Controller to add
- **object** (*string, (optional)*) – Object, Name of the Object to add the Controller to

`bpy.ops.logic.controller_move` (*controller=""*, *object=""*, *direction='UP'*)  
Move Controller

**Parameters**

- **controller** (*string, (optional)*) – Controller, Name of the controller to edit
- **object** (*string, (optional)*) – Object, Name of the object the controller belongs to
- **direction** (*enum in ['UP', 'DOWN'], (optional)*) – Direction, Move Up or Down

`bpy.ops.logic.controller_remove` (*controller=""*, *object=""*)  
Remove a controller from the active object

**Parameters**

- **controller** (*string, (optional)*) – Controller, Name of the controller to edit
- **object** (*string, (optional)*) – Object, Name of the object the controller belongs to

`bpy.ops.logic.links_cut` (*path=None*, *cursor=9*)  
Remove logic brick connections

**Parameters**

- **path** (*bpy\_prop\_collection of OperatorMousePath, (optional)*) – path
- **cursor** (*int in [0, inf], (optional)*) – Cursor

`bpy.ops.logic.properties` ()  
Toggle display properties panel

`bpy.ops.logic.sensor_add` (*type=''*, *name=""*, *object=""*)  
Add a sensor to the active object

**Parameters**

- **type** (*enum in [], (optional)*) – Type, Type of sensor to add
- **name** (*string, (optional)*) – Name, Name of the Sensor to add
- **object** (*string, (optional)*) – Object, Name of the Object to add the Sensor to

`bpy.ops.logic.sensor_move` (*sensor=""*, *object=""*, *direction='UP'*)  
Move Sensor

**Parameters**

- **sensor** (*string, (optional)*) – Sensor, Name of the sensor to edit
- **object** (*string, (optional)*) – Object, Name of the object the sensor belongs to
- **direction** (*enum in ['UP', 'DOWN'], (optional)*) – Direction, Move Up or Down



`bpy.ops.logic.sensor_remove` (*sensor=""*, *object=""*)  
Remove a sensor from the active object

**Parameters**

- **sensor** (*string, (optional)*) – Sensor, Name of the sensor to edit
- **object** (*string, (optional)*) – Object, Name of the object the sensor belongs to

`bpy.ops.logic.texface_convert` ()  
Convert old texface settings into material. It may create new materials if needed

## Marker Operators

`bpy.ops.marker.add` ()  
Add a new time marker

`bpy.ops.marker.camera_bind` ()  
Bind the active camera to selected markers(s)

`bpy.ops.marker.delete` ()  
Delete selected time marker(s)

`bpy.ops.marker.duplicate` (*frames=0*)  
Duplicate selected time marker(s)

**Parameters** **frames** (*int in [-inf, inf], (optional)*) – Frames

`bpy.ops.marker.make_links_scene` (*scene=''*)  
Copy selected markers to another scene

**Parameters** **scene** (*enum in [], (optional)*) – Scene

`bpy.ops.marker.move` (*frames=0*)  
Move selected time marker(s)

**Parameters** **frames** (*int in [-inf, inf], (optional)*) – Frames

`bpy.ops.marker.rename` (*name="RenamedMarker"*)  
Rename first selected time marker

**Parameters** **name** (*string, (optional)*) – Name, New name for marker

`bpy.ops.marker.select` (*extend=False, camera=False*)  
Select time marker(s)

**Parameters**

- **extend** (*boolean, (optional)*) – Extend, extend the selection
- **camera** (*boolean, (optional)*) – Camera, Select the camera

`bpy.ops.marker.select_all` (*action='TOGGLE'*)  
Change selection of all time markers

**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.marker.select_border` (*gesture\_mode=0, xmin=0, xmax=0, ymin=0, ymax=0*)  
Select all time markers using border selection

**Parameters**

- **gesture\_mode** (*int in [-inf, inf], (optional)*) – Gesture Mode
- **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

## Material Operators

`bpy.ops.material.copy` ()  
Copy the material settings and nodes

`bpy.ops.material.new` ()  
Add a new material

`bpy.ops.material.paste` ()  
Paste the material settings and nodes

`bpy.ops.material.sss_preset_add` (*name=""*, *remove\_active=False*)  
Add a Subsurface Scattering Preset

**Parameters** **name** (*string, (optional)*) – Name, Name of the preset, used to make the path name

**File** `startup/bl_operators/presets.py:50`

## Mball Operators

`bpy.ops.mball.delete_metaelems` ()  
Delete selected metaelement(s)

`bpy.ops.mball.duplicate_metaelems` (*mode='TRANSLATION'*)  
Delete selected metaelement(s)

**Parameters** **mode** (*enum in ['INIT', 'DUMMY', 'TRANSLATION', 'ROTATION', 'RESIZE', 'TOSPHERE', 'SHEAR', 'WARP', 'SHRINKFATTEN', 'TILT', 'TRACKBALL', 'PUSHPULL', 'CREASE', 'MIRROR', 'BONE\_SIZE', 'BONE\_ENVELOPE', 'CURVE\_SHRINKFATTEN', 'BONE\_ROLL', 'TIME\_TRANSLATE', 'TIME\_SLIDE', 'TIME\_SCALE', 'TIME\_EXTEND', 'BAKE\_TIME', 'BEVEL', 'BWEIGHT', 'ALIGN', 'EDGESLIDE', 'SEQSLIDE'], (optional)*) – Mode

`bpy.ops.mball.hide_metaelems` (*unselected=False*)  
Hide (un)selected metaelement(s)

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

`bpy.ops.mball.reveal_metaelems` ()  
Reveal all hidden metaelements

`bpy.ops.mball.select_all` (*action='TOGGLE'*)  
Change selection of all meta elements

**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.mball.select_inverse_metaelems()`  
Select inverse of (un)selected metaelements

`bpy.ops.mball.select_random_metaelems(percent=0.5)`  
Randomly select metaelements

**Parameters** `percent` (*float in [0, 1], (optional)*) – Percent, Percentage of metaelems to select randomly

## Mesh Operators

`bpy.ops.mesh.beautify_fill()`  
Rearrange geometry on a selected surface to avoid skinny faces

`bpy.ops.mesh.blend_from_shape(shape='', blend=1.0, add=False)`  
Blend in shape from a shape key

### Parameters

- **shape** (*enum in [], (optional)*) – Shape, Shape key to use for blending
- **blend** (*float in [-inf, inf], (optional)*) – Blend, Blending factor
- **add** (*boolean, (optional)*) – Add, Add rather than blend between shapes

`bpy.ops.mesh.colors_mirror(axis='X')`  
Mirror UV/image color layer

**Parameters** `axis` (*enum in ['X', 'Y'], (optional)*) – Axis, Axis to mirror colors around

`bpy.ops.mesh.colors_rotate(direction='CW')`  
Rotate UV/image color layer

**Parameters** `direction` (*enum in ['CW', 'CCW'], (optional)*) – Direction, Direction to rotate edge around

`bpy.ops.mesh.delete(type='VERT')`  
Delete selected vertices, edges or faces

**Parameters** `type` (*enum in ['VERT', 'EDGE', 'FACE', 'ALL', 'EDGE\_FACE', 'ONLY\_FACE', 'EDGE\_LOOP'], (optional)*) – Type, Method used for deleting mesh data

`bpy.ops.mesh.delete_edgeloop()`  
Delete an edge loop by merging the faces on each side to a single face loop

**File** `startup/bl_operators/wm.py:38`

`bpy.ops.mesh.drop_named_image(name="Image", filepath="Path")`  
Assigns Image to active UV layer, or creates a UV layer

### Parameters

- **name** (*string, (optional)*) – Name, Image name to assign
- **filepath** (*string, (optional)*) – Filepath, Path to image file

`bpy.ops.mesh.dupli_extrude_cursor` (*rotate\_source=True*)

Duplicate and extrude selected vertices, edges or faces towards 3D Cursor

**Parameters** `rotate_source` (*boolean, (optional)*) – Rotate Source, Rotate initial selection giving better shape

`bpy.ops.mesh.duplicate` (*mode='TRANSLATION'*)

Duplicate selected vertices, edges or faces

**Parameters** `mode` (*enum in ['INIT', 'DUMMY', 'TRANSLATION', 'ROTATION', 'RESIZE', 'TOSPHERE', 'SHEAR', 'WARP', 'SHRINKFATTEN', 'TILT', 'TRACKBALL', 'PUSHPULL', 'CREASE', 'MIRROR', 'BONE\_SIZE', 'BONE\_ENVELOPE', 'CURVE\_SHRINKFATTEN', 'BONE\_ROLL', 'TIME\_TRANSLATE', 'TIME\_SLIDE', 'TIME\_SCALE', 'TIME\_EXTEND', 'BAKE\_TIME', 'BEVEL', 'BWEIGHT', 'ALIGN', 'EDGESLIDE', 'SEQSLIDE'], (optional)*) – Mode

`bpy.ops.mesh.duplicate_move` (*MESH\_OT\_duplicate=None, TRANSFORM\_OT\_translate=None*)

Undocumented ([contribute](#))

**Parameters**

- **MESH\_OT\_duplicate** (*MESH\_OT\_duplicate, (optional)*) – Duplicate Mesh, Duplicate selected vertices, edges or faces
- **TRANSFORM\_OT\_translate** (*TRANSFORM\_OT\_translate, (optional)*) – Translate, Translate selected items

`bpy.ops.mesh.edge_face_add` ()

Add an edge or face to selected

`bpy.ops.mesh.edge_flip` ()

Flip selected edge or adjoining faces

`bpy.ops.mesh.edge_rotate` (*direction='CW'*)

Rotate selected edge or adjoining faces

**Parameters** `direction` (*enum in ['CW', 'CCW'], (optional)*) – Direction, Direction to rotate the edge around

`bpy.ops.mesh.edgering_select` (*extend=False*)

Select an edge ring

**Parameters** `extend` (*boolean, (optional)*) – Extend, Extend the selection

`bpy.ops.mesh.edges_select_sharp` (*sharpness=0.01*)

Marked selected edges as sharp

**Parameters** `sharpness` (*float in [0, inf], (optional)*) – sharpness

`bpy.ops.mesh.extrude` (*type='REGION'*)

Extrude selected vertices, edges or faces

**Parameters** `type` (*enum in ['REGION', 'FACES', 'EDGES', 'VERTS'], (optional)*) – Type

`bpy.ops.mesh.extrude_edges_move` (*MESH\_OT\_extrude=None, TRANSFORM\_OT\_translate=None*)

Undocumented ([contribute](#))

**Parameters**

- **MESH\_OT\_extrude** (*MESH\_OT\_extrude, (optional)*) – Extrude, Extrude selected vertices, edges or faces
- **TRANSFORM\_OT\_translate** (*TRANSFORM\_OT\_translate, (optional)*) – Translate, Translate selected items

`bpy.ops.mesh.extrude_faces_move` (*MESH\_OT\_extrude=None, TRANS-*  
*FORM\_OT\_shrink\_fatten=None*)

Undocumented ([contribute](#))

#### Parameters

- **MESH\_OT\_extrude** (*MESH\_OT\_extrude, (optional)*) – Extrude, Extrude selected vertices, edges or faces
- **TRANSFORM\_OT\_shrink\_fatten** (*TRANSFORM\_OT\_shrink\_fatten, (optional)*) – Shrink/Fatten, Shrink/fatten selected vertices along normals

`bpy.ops.mesh.extrude_region_move` (*MESH\_OT\_extrude=None, TRANS-*  
*FORM\_OT\_translate=None*)

Undocumented ([contribute](#))

#### Parameters

- **MESH\_OT\_extrude** (*MESH\_OT\_extrude, (optional)*) – Extrude, Extrude selected vertices, edges or faces
- **TRANSFORM\_OT\_translate** (*TRANSFORM\_OT\_translate, (optional)*) – Translate, Translate selected items

`bpy.ops.mesh.extrude_repeat` (*offset=2.0, steps=10, direction=(0.0, 0.0, 0.0)*)

Extrude selected vertices, edges or faces repeatedly

#### Parameters

- **offset** (*float in [0, 100], (optional)*) – Offset
- **steps** (*int in [0, 180], (optional)*) – Steps
- **direction** (*float array of 3 items in [-inf, inf], (optional)*) – Direction, Direction of extrude

`bpy.ops.mesh.extrude_vertices_move` (*MESH\_OT\_extrude=None, TRANS-*  
*FORM\_OT\_translate=None*)

Undocumented ([contribute](#))

#### Parameters

- **MESH\_OT\_extrude** (*MESH\_OT\_extrude, (optional)*) – Extrude, Extrude selected vertices, edges or faces
- **TRANSFORM\_OT\_translate** (*TRANSFORM\_OT\_translate, (optional)*) – Translate, Translate selected items

`bpy.ops.mesh.faces_mirror_uv` (*direction='POSITIVE'*)

Copy mirror UV coordinates on the X axis based on a mirrored mesh

**Parameters** *direction* (*enum in ['POSITIVE', 'NEGATIVE'], (optional)*) – Axis Direction

**File** `startup/bl_operators/mesh.py:88`

`bpy.ops.mesh.faces_select_interior` ()

Select faces where all edges have more then 2 face users

**File** `startup/bl_operators/mesh.py:39`

`bpy.ops.mesh.faces_select_linked_flat` (*sharpness=135.0*)

Select linked faces by angle

**Parameters** *sharpness* (*float in [0, inf], (optional)*) – sharpness

`bpy.ops.mesh.faces_shade_flat` ()

Display faces 'flat'

`bpy.ops.mesh.faces_shade_smooth()`  
Display faces 'smooth' (using vertex normals)

`bpy.ops.mesh.fgon_clear()`  
Clear fgon from selected face

`bpy.ops.mesh.fgon_make()`  
Make fgon from selected faces

`bpy.ops.mesh.fill()`  
Create a segment, edge or face

`bpy.ops.mesh.flip_normals()`  
Toggle the direction of selected face's vertex and face normals

`bpy.ops.mesh.hide(unselected=False)`  
Hide (un)selected vertices, edges or faces

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

`bpy.ops.mesh.knife_cut(type='EXACT', path=None, num_cuts=1, cursor=9)`  
Cut selected edges and faces into parts

**Parameters**

- `type` (*enum in ['EXACT', 'MIDPOINTS', 'MULTICUT'], (optional)*) – Type
- `path` (*bpy\_prop\_collection of OperatorMousePath, (optional)*) – path
- `num_cuts` (*int in [1, 256], (optional)*) – Number of Cuts, Only for Multi-Cut
- `cursor` (*int in [0, inf], (optional)*) – Cursor

`bpy.ops.mesh.loop_multi_select(ring=False)`  
Select a loop of connected edges by connection type

**Parameters** `ring` (*boolean, (optional)*) – Ring

`bpy.ops.mesh.loop_select(extend=False, ring=False)`  
Select a loop of connected edges

**Parameters**

- `extend` (*boolean, (optional)*) – Extend Select
- `ring` (*boolean, (optional)*) – Select Ring

`bpy.ops.mesh.loop_to_region()`  
Select a loop of connected edges as a region

`bpy.ops.mesh.loopcut(number_cuts=1)`  
Add a new loop between existing loops

**Parameters** `number_cuts` (*int in [1, inf], (optional)*) – Number of Cuts

`bpy.ops.mesh.loopcut_slide(MESH_OT_loopcut=None, TRANSFORM_OT_edge_slide=None)`  
Undocumented ([contribute](#))

**Parameters**

- `MESH_OT_loopcut` (*MESH\_OT\_loopcut, (optional)*) – Loop Cut, Add a new loop between existing loops
- `TRANSFORM_OT_edge_slide` (*TRANSFORM\_OT\_edge\_slide, (optional)*) – Edge Slide, Slide an edge loop along a mesh

`bpy.ops.mesh.mark_seam` (*clear=False*)  
(un)mark selected edges as a seam

**Parameters** `clear` (*boolean, (optional)*) – Clear

`bpy.ops.mesh.mark_sharp` (*clear=False*)  
(un)mark selected edges as sharp

**Parameters** `clear` (*boolean, (optional)*) – Clear

`bpy.ops.mesh.merge` (*type='CENTER', uvs=False*)  
Merge selected vertices

**Parameters**

- **type** (*enum in ['FIRST', 'LAST', 'CENTER', 'CURSOR', 'COLLAPSE'], (optional)*) – Type, Merge method to use
- **uvs** (*boolean, (optional)*) – UVs, Move UVs according to merge

`bpy.ops.mesh.noise` (*factor=0.1*)  
Use vertex coordinate as texture coordinate

**Parameters** `factor` (*float in [-inf, inf], (optional)*) – Factor

`bpy.ops.mesh.normals_make_consistent` (*inside=False*)  
Flip all selected vertex and face normals in a consistent direction

**Parameters** `inside` (*boolean, (optional)*) – Inside

`bpy.ops.mesh.primitive_circle_add` (*vertices=32, radius=1.0, fill=False, view\_align=False, enter\_editmode=False, location=(0.0, 0.0, 0.0), rotation=(0.0, 0.0, 0.0), layers=(False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False)*)

Construct a circle mesh

**Parameters**

- **vertices** (*int in [3, inf], (optional)*) – Vertices
- **radius** (*float in [0, inf], (optional)*) – Radius
- **fill** (*boolean, (optional)*) – Fill
- **view\_align** (*boolean, (optional)*) – Align to View, Align the new object to the view
- **enter\_editmode** (*boolean, (optional)*) – Enter Editmode, Enter editmode when adding this object
- **location** (*float array of 3 items in [-inf, inf], (optional)*) – Location, Location for the newly added object
- **rotation** (*float array of 3 items in [-inf, inf], (optional)*) – Rotation, Rotation for the newly added object
- **layers** (*boolean array of 20 items, (optional)*) – Layer

`bpy.ops.mesh.primitive_cone_add` (*vertices=32, radius=1.0, depth=2.0, cap\_end=True, view\_align=False, enter\_editmode=False, location=(0.0, 0.0, 0.0), rotation=(0.0, 0.0, 0.0), layers=(False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False)*)

Construct a conic mesh (ends filled)

**Parameters**

- **vertices** (*int in [2, inf], (optional)*) – Vertices
- **radius** (*float in [0, inf], (optional)*) – Radius
- **depth** (*float in [0, inf], (optional)*) – Depth
- **cap\_end** (*boolean, (optional)*) – Cap End
- **view\_align** (*boolean, (optional)*) – Align to View, Align the new object to the view
- **enter\_editmode** (*boolean, (optional)*) – Enter Editmode, Enter editmode when adding this object
- **location** (*float array of 3 items in [-inf, inf], (optional)*) – Location, Location for the newly added object
- **rotation** (*float array of 3 items in [-inf, inf], (optional)*) – Rotation, Rotation for the newly added object
- **layers** (*boolean array of 20 items, (optional)*) – Layer

```
bpy.ops.mesh.primitive_cube_add (view_align=False, enter_editmode=False, location=(0.0, 0.0, 0.0), rotation=(0.0, 0.0, 0.0), layers=(False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False))
```

Construct a cube mesh

#### Parameters

- **view\_align** (*boolean, (optional)*) – Align to View, Align the new object to the view
- **enter\_editmode** (*boolean, (optional)*) – Enter Editmode, Enter editmode when adding this object
- **location** (*float array of 3 items in [-inf, inf], (optional)*) – Location, Location for the newly added object
- **rotation** (*float array of 3 items in [-inf, inf], (optional)*) – Rotation, Rotation for the newly added object
- **layers** (*boolean array of 20 items, (optional)*) – Layer

```
bpy.ops.mesh.primitive_cylinder_add (vertices=32, radius=1.0, depth=2.0, cap_ends=True, view_align=False, enter_editmode=False, location=(0.0, 0.0, 0.0), rotation=(0.0, 0.0, 0.0), layers=(False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False))
```

Construct a cylinder mesh

#### Parameters

- **vertices** (*int in [2, inf], (optional)*) – Vertices
- **radius** (*float in [0, inf], (optional)*) – Radius
- **depth** (*float in [0, inf], (optional)*) – Depth
- **cap\_ends** (*boolean, (optional)*) – Cap Ends
- **view\_align** (*boolean, (optional)*) – Align to View, Align the new object to the view
- **enter\_editmode** (*boolean, (optional)*) – Enter Editmode, Enter editmode when adding this object



- **location** (*float array of 3 items in [-inf, inf], (optional)*) – Location, Location for the newly added object
- **rotation** (*float array of 3 items in [-inf, inf], (optional)*) – Rotation, Rotation for the newly added object
- **layers** (*boolean array of 20 items, (optional)*) – Layer

```
bpy.ops.mesh.primitive_grid_add(x_subdivisions=10, y_subdivisions=10, size=1.0,
view_align=False, enter_editmode=False, location=(0.0,
0.0, 0.0), rotation=(0.0, 0.0, 0.0), layers=(False, False, False,
False, False, False, False, False, False, False, False,
False, False, False, False, False, False, False, False))
```

Construct a grid mesh

#### Parameters

- **x\_subdivisions** (*int in [3, inf], (optional)*) – X Subdivisions
- **y\_subdivisions** (*int in [3, inf], (optional)*) – Y Subdivisions
- **size** (*float in [0, inf], (optional)*) – Size
- **view\_align** (*boolean, (optional)*) – Align to View, Align the new object to the view
- **enter\_editmode** (*boolean, (optional)*) – Enter Editmode, Enter editmode when adding this object
- **location** (*float array of 3 items in [-inf, inf], (optional)*) – Location, Location for the newly added object
- **rotation** (*float array of 3 items in [-inf, inf], (optional)*) – Rotation, Rotation for the newly added object
- **layers** (*boolean array of 20 items, (optional)*) – Layer

```
bpy.ops.mesh.primitive_ico_sphere_add(subdivisions=2, size=1.0, view_align=False,
enter_editmode=False, location=(0.0, 0.0, 0.0), rotation=(0.0, 0.0, 0.0), layers=(False, False, False, False,
False, False, False, False, False, False, False, False,
False, False, False, False, False, False, False, False))
```

Construct an Icosphere mesh

#### Parameters

- **subdivisions** (*int in [1, inf], (optional)*) – Subdivisions
- **size** (*float in [0, inf], (optional)*) – Size
- **view\_align** (*boolean, (optional)*) – Align to View, Align the new object to the view
- **enter\_editmode** (*boolean, (optional)*) – Enter Editmode, Enter editmode when adding this object
- **location** (*float array of 3 items in [-inf, inf], (optional)*) – Location, Location for the newly added object
- **rotation** (*float array of 3 items in [-inf, inf], (optional)*) – Rotation, Rotation for the newly added object
- **layers** (*boolean array of 20 items, (optional)*) – Layer

`bpy.ops.mesh.primitive_monkey_add` (*view\_align=False, enter\_editmode=False, location=(0.0, 0.0, 0.0), rotation=(0.0, 0.0, 0.0), layers=(False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False)*)

Construct a Suzanne mesh

**Parameters**

- **view\_align** (*boolean, (optional)*) – Align to View, Align the new object to the view
- **enter\_editmode** (*boolean, (optional)*) – Enter Editmode, Enter editmode when adding this object
- **location** (*float array of 3 items in [-inf, inf], (optional)*) – Location, Location for the newly added object
- **rotation** (*float array of 3 items in [-inf, inf], (optional)*) – Rotation, Rotation for the newly added object
- **layers** (*boolean array of 20 items, (optional)*) – Layer

`bpy.ops.mesh.primitive_plane_add` (*view\_align=False, enter\_editmode=False, location=(0.0, 0.0, 0.0), rotation=(0.0, 0.0, 0.0), layers=(False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False)*)

Construct a filled planar mesh with 4 vertices

**Parameters**

- **view\_align** (*boolean, (optional)*) – Align to View, Align the new object to the view
- **enter\_editmode** (*boolean, (optional)*) – Enter Editmode, Enter editmode when adding this object
- **location** (*float array of 3 items in [-inf, inf], (optional)*) – Location, Location for the newly added object
- **rotation** (*float array of 3 items in [-inf, inf], (optional)*) – Rotation, Rotation for the newly added object
- **layers** (*boolean array of 20 items, (optional)*) – Layer

`bpy.ops.mesh.primitive_torus_add` (*major\_radius=1.0, minor\_radius=0.25, major\_segments=48, minor\_segments=12, use\_abso=False, abso\_major\_rad=1.0, abso\_minor\_rad=0.5, view\_align=False, location=(0.0, 0.0, 0.0), rotation=(0.0, 0.0, 0.0)*)

Add a torus mesh

**Parameters**

- **major\_radius** (*float in [0.01, 100], (optional)*) – Major Radius, Radius from the origin to the center of the cross sections
- **minor\_radius** (*float in [0.01, 100], (optional)*) – Minor Radius, Radius of the torus' cross section
- **major\_segments** (*int in [3, 256], (optional)*) – Major Segments, Number of segments for the main ring of the torus
- **minor\_segments** (*int in [3, 256], (optional)*) – Minor Segments, Number of segments for the minor ring of the torus
- **use\_abso** (*boolean, (optional)*) – Use Int+Ext Controls, Use the Int / Ext controls for torus dimensions

- **abso\_major\_rad** (*float in [0.01, 100], (optional)*) – Exterior Radius, Total Exterior Radius of the torus
- **abso\_minor\_rad** (*float in [0.01, 100], (optional)*) – Inside Radius, Total Interior Radius of the torus
- **view\_align** (*boolean, (optional)*) – Align to View
- **location** (*float array of 3 items in [-inf, inf], (optional)*) – Location
- **rotation** (*float array of 3 items in [-inf, inf], (optional)*) – Rotation

**File** `startup/bl_operators/add_mesh_torus.py:148`

```
bpy.ops.mesh.primitive_uv_sphere_add (segments=32,          ring_count=16,          size=1.0,
                                       view_align=False,     enter_editmode=False,  loca-
                                       tion=(0.0, 0.0, 0.0), rotation=(0.0, 0.0, 0.0), lay-
                                       ers=(False, False, False, False, False, False, False,
                                       False, False, False, False, False, False, False,
                                       False, False, False, False, False))
```

Construct a UV sphere mesh

#### Parameters

- **segments** (*int in [3, inf], (optional)*) – Segments
- **ring\_count** (*int in [3, inf], (optional)*) – Rings
- **size** (*float in [0, inf], (optional)*) – Size
- **view\_align** (*boolean, (optional)*) – Align to View, Align the new object to the view
- **enter\_editmode** (*boolean, (optional)*) – Enter Editmode, Enter editmode when adding this object
- **location** (*float array of 3 items in [-inf, inf], (optional)*) – Location, Location for the newly added object
- **rotation** (*float array of 3 items in [-inf, inf], (optional)*) – Rotation, Rotation for the newly added object
- **layers** (*boolean array of 20 items, (optional)*) – Layer

```
bpy.ops.mesh.quads_convert_to_tris ()
  Convert selected quads to triangles
```

```
bpy.ops.mesh.region_to_loop ()
  Select a region as a loop of connected edges
```

```
bpy.ops.mesh.remove_doubles (limit=0.0001)
  Remove duplicate vertices
```

**Parameters** **limit** (*float in [1e-06, 50], (optional)*) – Merge Threshold, Minimum distance between merged verts

```
bpy.ops.mesh.reveal ()
  Reveal all hidden vertices, edges and faces
```

```
bpy.ops.mesh.rip (constraint_axis=(False, False, False), constraint_orientation='', mirror=False, re-
                 lease_confirm=False)
  Rip selection from mesh (quads only)
```

#### Parameters

- **constraint\_axis** (*boolean array of 3 items, (optional)*) – Constraint Axis

- **constraint\_orientation** (*enum in [], (optional)*) – Orientation, Transformation orientation
- **mirror** (*boolean, (optional)*) – Mirror Editing
- **release\_confirm** (*boolean, (optional)*) – Confirm on Release, Always confirm operation when releasing button

`bpy.ops.mesh.rip_move` (*MESH\_OT\_rip=None, TRANSFORM\_OT\_translate=None*)  
Undocumented ([contribute](#))

#### Parameters

- **MESH\_OT\_rip** (*MESH\_OT\_rip, (optional)*) – Rip, Rip selection from mesh (quads only)
- **TRANSFORM\_OT\_translate** (*TRANSFORM\_OT\_translate, (optional)*) – Translate, Translate selected items

`bpy.ops.mesh.screw` (*steps=9, turns=1, center=(0.0, 0.0, 0.0), axis=(0.0, 0.0, 0.0)*)  
Extrude selected vertices in screw-shaped rotation around the cursor in indicated viewport

#### Parameters

- **steps** (*int in [0, inf], (optional)*) – Steps, Steps
- **turns** (*int in [0, inf], (optional)*) – Turns, Turns
- **center** (*float array of 3 items in [-inf, inf], (optional)*) – Center, Center in global view space
- **axis** (*float array of 3 items in [-1, 1], (optional)*) – Axis, Axis in global view space

`bpy.ops.mesh.select_all` (*action='TOGGLE'*)  
Change selection of all vertices, edges or faces

**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.mesh.select_axis` (*mode='POSITIVE', axis='X\_AXIS'*)  
Select all data in the mesh on a single axis

#### Parameters

- **mode** (*enum in ['POSITIVE', 'NEGATIVE', 'ALIGNED'], (optional)*) – Axis Mode, Axis side to use when selecting
- **axis** (*enum in ['X\_AXIS', 'Y\_AXIS', 'Z\_AXIS'], (optional)*) – Axis, Select the axis to compare each vertex on

`bpy.ops.mesh.select_by_number_vertices` (*type='TRIANGLES'*)  
Select vertices or faces by vertex count

**Parameters** **type** (*enum in ['TRIANGLES', 'QUADS', 'OTHER'], (optional)*) – Type, Type of elements to select

`bpy.ops.mesh.select_inverse` ()  
Select inverse of (un)selected vertices, edges or faces

`bpy.ops.mesh.select_less` ()  
Select less vertices, edges or faces connected to initial selection

`bpy.ops.mesh.select_linked` (*limit=False*)

Select all vertices linked to the active mesh

**Parameters** `limit` (*boolean, (optional)*) – Limit by Seams, Limit selection by seam boundaries (faces only)

`bpy.ops.mesh.select_linked_pick` (*deselect=False, limit=False*)

(un)select all vertices linked to the active mesh

**Parameters**

- `deselect` (*boolean, (optional)*) – Deselect
- `limit` (*boolean, (optional)*) – Limit by Seams, Limit selection by seam boundaries (faces only)

`bpy.ops.mesh.select_mirror` (*extend=False*)

Select mesh items at mirrored locations

**Parameters** `extend` (*boolean, (optional)*) – Extend, Extend the existing selection

`bpy.ops.mesh.select_more` ()

Select more vertices, edges or faces connected to initial selection

`bpy.ops.mesh.select_non_manifold` ()

Select all non-manifold vertices or edges

`bpy.ops.mesh.select_nth` (*nth=2*)

Undocumented ([contribute](#))

**Parameters** `nth` (*int in [2, 100], (optional)*) – Nth Selection

`bpy.ops.mesh.select_random` (*percent=50.0, extend=False*)

Randomly select vertices

**Parameters**

- `percent` (*float in [0, 100], (optional)*) – Percent, Percentage of elements to select randomly
- `extend` (*boolean, (optional)*) – Extend Selection, Extend selection instead of deselecting everything first

`bpy.ops.mesh.select_shortest_path` (*extend=False*)

Select shortest path between two selections

**Parameters** `extend` (*boolean, (optional)*) – Extend Select

`bpy.ops.mesh.select_similar` (*type='NORMAL', threshold=0.01*)

Select similar vertices, edges or faces by property types

**Parameters**

- `type` (*enum in ['NORMAL', 'FACE', 'VGROUP', 'LENGTH', 'DIR', 'FACE', 'FACE\_ANGLE', 'CREASE', 'SEAM', 'SHARP', 'MATERIAL', 'IMAGE', 'AREA', 'PERIMETER', 'NORMAL', 'COPLANAR'], (optional)*) – Type
- `threshold` (*float in [0, inf], (optional)*) – Threshold

`bpy.ops.mesh.select_vertex_path` (*type='EDGE\_LENGTH'*)

Select shortest path between two vertices by distance type

**Parameters** `type` (*enum in ['EDGE\_LENGTH', 'TOPOLOGICAL'], (optional)*) – Type, Method to compute distance

`bpy.ops.mesh.separate` (*type='SELECTED'*)

Separate selected geometry into a new mesh

**Parameters type** (*enum in ['SELECTED', 'MATERIAL', 'LOOSE'], (optional)*) – Type

`bpy.ops.mesh.shape_propagate_to_all()`  
Apply selected vertex locations to all other shape keys

`bpy.ops.mesh.solidify(thickness=0.01)`  
Create a solid skin by extruding, compensating for sharp angles

**Parameters thickness** (*float in [-inf, inf], (optional)*) – Thickness

`bpy.ops.mesh.sort_faces(type='VIEW_AXIS')`  
The faces of the active Mesh Object are sorted, based on the current view

**Parameters type** (*enum in ['VIEW\_AXIS', 'CURSOR\_DISTANCE', 'MATERIAL', 'SELECTED', 'RANDOMIZE'], (optional)*) – Type

`bpy.ops.mesh.spin(steps=9, dupli=False, degrees=90.0, center=(0.0, 0.0, 0.0), axis=(0.0, 0.0, 0.0))`  
Extrude selected vertices in a circle around the cursor in indicated viewport

#### Parameters

- **steps** (*int in [0, inf], (optional)*) – Steps, Steps
- **dupli** (*boolean, (optional)*) – Dupli, Make Duplicates
- **degrees** (*float in [-inf, inf], (optional)*) – Degrees, Degrees
- **center** (*float array of 3 items in [-inf, inf], (optional)*) – Center, Center in global view space
- **axis** (*float array of 3 items in [-1, 1], (optional)*) – Axis, Axis in global view space

`bpy.ops.mesh.split()`  
Split selected geometry into separate disconnected mesh

`bpy.ops.mesh.sticky_add()`  
Add sticky UV texture layer

`bpy.ops.mesh.sticky_remove()`  
Remove sticky UV texture layer

`bpy.ops.mesh.subdivide(number_cuts=1, smoothness=0.0, fractal=0.0, corner_cut_pattern='INNER_VERTEX')`  
Subdivide selected edges

#### Parameters

- **number\_cuts** (*int in [1, inf], (optional)*) – Number of Cuts
- **smoothness** (*float in [0, inf], (optional)*) – Smoothness, Smoothness factor
- **fractal** (*float in [0, inf], (optional)*) – Fractal, Fractal randomness factor
- **corner\_cut\_pattern** (*enum in ['PATH', 'INNER\_VERTEX', 'FAN'], (optional)*) – Corner Cut Pattern, Topology pattern to use to fill a face after cutting across its corner

`bpy.ops.mesh.tris_convert_to_quads()`  
Convert selected triangles to quads

`bpy.ops.mesh.uv_texture_add()`  
Add UV texture layer

`bpy.ops.mesh.uv_texture_remove()`  
Remove UV texture layer

`bpy.ops.mesh.uvs_mirror(axis='X')`  
Mirror selected UVs

**Parameters** `axis` (*enum in ['X', 'Y'], (optional)*) – Axis, Axis to mirror UVs around

`bpy.ops.mesh.uvs_rotate` (*direction='CW'*)  
Rotate selected UVs

**Parameters** `direction` (*enum in ['CW', 'CCW'], (optional)*) – Direction, Direction to rotate UVs around

`bpy.ops.mesh.vertex_color_add` ()  
Add vertex color layer

`bpy.ops.mesh.vertex_color_remove` ()  
Remove vertex color layer

`bpy.ops.mesh.vertices_randomize` ()  
Randomize vertex order

`bpy.ops.mesh.vertices_smooth` (*repeat=1, xaxis=True, yaxis=True, zaxis=True*)  
Flatten angles of selected vertices

#### Parameters

- **repeat** (*int in [1, 100], (optional)*) – Smooth Iterations
- **xaxis** (*boolean, (optional)*) – X-Axis, Smooth along the X axis
- **yaxis** (*boolean, (optional)*) – Y-Axis, Smooth along the Y axis
- **zaxis** (*boolean, (optional)*) – Z-Axis, Smooth along the Z axis

`bpy.ops.mesh.vertices_sort` ()  
Sort vertex order

## Nla Operators

`bpy.ops.nla.action_sync_length` (*active=True*)  
Synchronise the length of the referenced Action with the length used in the strip

**Parameters** `active` (*boolean, (optional)*) – Active Strip Only, Only sync the active length for the active strip

`bpy.ops.nla.actionclip_add` (*action=''*)  
Add an Action-Clip strip (i.e. an NLA Strip referencing an Action) to the active track

**Parameters** `action` (*enum in [], (optional)*) – Action

`bpy.ops.nla.apply_scale` ()  
Apply scaling of selected strips to their referenced Actions

`bpy.ops.nla.bake` (*frame\_start=1, frame\_end=250, step=1, only\_selected=True, clear\_constraints=False, bake\_types={'POSE'}*)  
Bake animation to an Action

#### Parameters

- **frame\_start** (*int in [0, 300000], (optional)*) – Start Frame, Start frame for baking
- **frame\_end** (*int in [1, 300000], (optional)*) – End Frame, End frame for baking
- **step** (*int in [1, 120], (optional)*) – Frame Step, Frame Step
- **only\_selected** (*boolean, (optional)*) – Only Selected
- **clear\_constraints** (*boolean, (optional)*) – Clear Constraints

- **bake\_types** (*enum set in {'POSE', 'OBJECT'}, (optional)*) – Bake Data

**File** `startup/bl_operators/anim.py:204`

`bpy.ops.nla.channels_click` (*extend=False*)  
Handle clicks to select NLA channels

**Parameters** **extend** (*boolean, (optional)*) – Extend Select

`bpy.ops.nla.clear_scale` ()  
Reset scaling of selected strips

`bpy.ops.nla.click_select` (*extend=False*)  
Handle clicks to select NLA Strips

**Parameters** **extend** (*boolean, (optional)*) – Extend Select

`bpy.ops.nla.delete` ()  
Delete selected strips

`bpy.ops.nla.delete_tracks` ()  
Delete selected NLA-Tracks and the strips they contain

`bpy.ops.nla.duplicate` (*mode='TRANSLATION'*)  
Duplicate selected NLA-Strips, adding the new strips in new tracks above the originals

**Parameters** **mode** (*enum in ['INIT', 'DUMMY', 'TRANSLATION', 'ROTATION', 'RESIZE', 'TOSPHERE', 'SHEAR', 'WARP', 'SHRINKFATTEN', 'TILT', 'TRACKBALL', 'PUSHPULL', 'CREASE', 'MIRROR', 'BONE\_SIZE', 'BONE\_ENVELOPE', 'CURVE\_SHRINKFATTEN', 'BONE\_ROLL', 'TIME\_TRANSLATE', 'TIME\_SLIDE', 'TIME\_SCALE', 'TIME\_EXTEND', 'BAKE\_TIME', 'BEVEL', 'BWEIGHT', 'ALIGN', 'EDGESLIDE', 'SEQSLIDE'], (optional)*) – Mode

`bpy.ops.nla.fmodifier_add` (*type='NULL', only\_active=False*)  
Add a F-Modifier of the specified type to the selected NLA-Strips

**Parameters**

- **type** (*enum in ['NULL', 'GENERATOR', 'FNGENERATOR', 'ENVELOPE', 'CYCLES', 'NOISE', 'FILTER', 'LIMITS', 'STEPPED'], (optional)*) – Type
- **only\_active** (*boolean, (optional)*) – Only Active, Only add a F-Modifier of the specified type to the active strip

`bpy.ops.nla.fmodifier_copy` ()  
Copy the F-Modifier(s) of the active NLA-Strip

`bpy.ops.nla.fmodifier_paste` ()  
Add copied F-Modifiers to the selected NLA-Strips

`bpy.ops.nla.meta_add` ()  
Add new meta-strips incorporating the selected strips

`bpy.ops.nla.meta_remove` ()  
Separate out the strips held by the selected meta-strips

`bpy.ops.nla.move_down` ()  
Move selected strips down a track if there's room

`bpy.ops.nla.move_up` ()  
Move selected strips up a track if there's room

`bpy.ops.nla.mute_toggle` ()  
Mute or un-mute selected strips



`bpy.ops.nla.properties()`  
Toggle display properties panel

`bpy.ops.nla.select_all_toggle(invert=False)`  
(De)Select all NLA-Strips

**Parameters** `invert` (*boolean, (optional)*) – Invert

`bpy.ops.nla.select_border(gesture_mode=0, xmin=0, xmax=0, ymin=0, ymax=0, axis_range=False)`  
Use box selection to grab NLA-Strips

**Parameters**

- `gesture_mode` (*int in [-inf, inf], (optional)*) – Gesture Mode
- `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
- `axis_range` (*boolean, (optional)*) – Axis Range

`bpy.ops.nla.select_leftright(mode='CHECK', extend=False)`  
Select strips to the left or the right of the current frame

**Parameters**

- `mode` (*enum in ['CHECK', 'LEFT', 'RIGHT'], (optional)*) – Mode
- `extend` (*boolean, (optional)*) – Extend Select

`bpy.ops.nla.snap(type='CFRA')`  
Move start of strips to specified time

**Parameters** `type` (*enum in ['CFRA', 'NEAREST\_FRAME', 'NEAREST\_SECOND', 'NEAREST\_MARKER'], (optional)*) – Type

`bpy.ops.nla.soundclip_add()`  
Add a strip for controlling when speaker plays its sound clip

`bpy.ops.nla.split()`  
Split selected strips at their midpoints

`bpy.ops.nla.swap()`  
Swap order of selected strips within tracks

`bpy.ops.nla.tracks_add(above_selected=False)`  
Add NLA-Tracks above/after the selected tracks

**Parameters** `above_selected` (*boolean, (optional)*) – Above Selected, Add a new NLA Track above every existing selected one

`bpy.ops.nla.transition_add()`  
Add a transition strip between two adjacent selected strips

`bpy.ops.nla.tweakmode_enter()`  
Enter tweaking mode for the action referenced by the active strip

`bpy.ops.nla.tweakmode_exit()`  
Exit tweaking mode for the action referenced by the active strip

`bpy.ops.nla.view_all()`  
Reset viewable area to show full strips range

`bpy.ops.nla.view_selected()`  
Reset viewable area to show selected strips range

## Node Operators

`bpy.ops.node.add_file` (*filepath=""*, *filter\_blender=False*, *filter\_image=True*, *filter\_movie=False*,  
*filter\_python=False*, *filter\_font=False*, *filter\_sound=False*, *filter\_text=False*,  
*filter\_btx=False*, *filter\_collada=False*, *filter\_folder=True*, *filemode=9*,  
*name="Image"*)  
Add a file node to the current node editor

### Parameters

- **filepath** (*string, (optional)*) – File Path, Path to file
- **filter\_blender** (*boolean, (optional)*) – Filter .blend files
- **filter\_image** (*boolean, (optional)*) – Filter image files
- **filter\_movie** (*boolean, (optional)*) – Filter movie files
- **filter\_python** (*boolean, (optional)*) – Filter python files
- **filter\_font** (*boolean, (optional)*) – Filter font files
- **filter\_sound** (*boolean, (optional)*) – Filter sound files
- **filter\_text** (*boolean, (optional)*) – Filter text files
- **filter\_btx** (*boolean, (optional)*) – Filter btx files
- **filter\_collada** (*boolean, (optional)*) – Filter COLLADA files
- **filter\_folder** (*boolean, (optional)*) – Filter folders
- **filemode** (*int in [1, 9], (optional)*) – File Browser Mode, The setting for the file browser mode to load a .blend file, a library or a special file
- **name** (*string, (optional)*) – Name, Datablock name to assign

`bpy.ops.node.backimage_move` ()  
Move Node backdrop

`bpy.ops.node.backimage_sample` ()  
Undocumented ([contribute](#))

`bpy.ops.node.backimage_zoom` (*factor=1.2*)  
Undocumented ([contribute](#))

**Parameters** **factor** (*float in [0, 10], (optional)*) – Factor

`bpy.ops.node.delete` ()  
Delete selected nodes

`bpy.ops.node.delete_reconnect` ()  
Delete nodes; will reconnect nodes as if deletion was muted

`bpy.ops.node.duplicate` (*keep\_inputs=False*)  
Duplicate selected nodes

**Parameters** **keep\_inputs** (*boolean, (optional)*) – Keep Inputs, Keep the input links to duplicated nodes

`bpy.ops.node.duplicate_move` (*NODE\_OT\_duplicate=None, TRANSFORM\_OT\_translate=None*)  
Undocumented ([contribute](#))

**Parameters**

- **NODE\_OT\_duplicate** (`NODE_OT_duplicate`, (optional)) – Duplicate Nodes, Duplicate selected nodes
- **TRANSFORM\_OT\_translate** (`TRANSFORM_OT_translate`, (optional)) – Translate, Translate selected items

`bpy.ops.node.duplicate_move_keep_inputs` (`NODE_OT_duplicate=None`, `TRANSFORM_OT_translate=None`)

Undocumented ([contribute](#))

**Parameters**

- **NODE\_OT\_duplicate** (`NODE_OT_duplicate`, (optional)) – Duplicate Nodes, Duplicate selected nodes
- **TRANSFORM\_OT\_translate** (`TRANSFORM_OT_translate`, (optional)) – Translate, Translate selected items

`bpy.ops.node.group_edit` ()  
Edit node group

`bpy.ops.node.group_make` ()  
Make group from selected nodes

`bpy.ops.node.group_socket_add` (`in_out='SOCK_IN'`, `name=""`, `type='VALUE'`)  
Add node group socket

**Parameters**

- **in\_out** (*enum in ['SOCK\_IN', 'SOCK\_OUT']*, (optional)) – Socket Type, Input or Output
- **name** (*string*, (optional)) – Name, Group socket name
- **type** (*enum in ['VALUE', 'VECTOR', 'RGBA']*, (optional)) – Type, Type of the group socket

`bpy.ops.node.group_socket_move_down` (`index=0`, `in_out='SOCK_IN'`)  
Move down node group socket

**Parameters**

- **index** (*int in [0, inf]*, (optional)) – Index
- **in\_out** (*enum in ['SOCK\_IN', 'SOCK\_OUT']*, (optional)) – Socket Type, Input or Output

`bpy.ops.node.group_socket_move_up` (`index=0`, `in_out='SOCK_IN'`)  
Move up node group socket

**Parameters**

- **index** (*int in [0, inf]*, (optional)) – Index
- **in\_out** (*enum in ['SOCK\_IN', 'SOCK\_OUT']*, (optional)) – Socket Type, Input or Output

`bpy.ops.node.group_socket_remove` (`index=0`, `in_out='SOCK_IN'`)  
Remove a node group socket

**Parameters**

- **index** (*int in [0, inf]*, (optional)) – Index
- **in\_out** (*enum in ['SOCK\_IN', 'SOCK\_OUT']*, (optional)) – Socket Type, Input or Output

`bpy.ops.node.group_ungroup` ()  
Ungroup selected nodes

`bpy.ops.node.hide_socket_toggle()`  
Toggle unused node socket display

`bpy.ops.node.hide_toggle()`  
Toggle hiding of selected nodes

`bpy.ops.node.link()`  
Undocumented (contribute)

`bpy.ops.node.link_make(replace=False)`  
Makes a link between selected output in input sockets

**Parameters** `replace` (*boolean, (optional)*) – Replace, Replace socket connections with the new links

`bpy.ops.node.link_viewer()`  
Link to viewer node

`bpy.ops.node.links_cut(path=None, cursor=9)`  
Undocumented (contribute)

**Parameters**

- `path` (*bpy\_prop\_collection of OperatorMousePath, (optional)*) – path
- `cursor` (*int in [0, inf], (optional)*) – Cursor

`bpy.ops.node.mute_toggle()`  
Toggle muting of the nodes

`bpy.ops.node.new_node_tree(type='COMPOSITING', name="NodeTree")`  
Undocumented (contribute)

**Parameters**

- `type` (*enum in ['SHADER', 'TEXTURE', 'COMPOSITING'], (optional)*) – Tree Type
  - SHADER Shader, Shader nodes.
  - TEXTURE Texture, Texture nodes.
  - COMPOSITING Compositing, Compositing nodes.
- `name` (*string, (optional)*) – Name

`bpy.ops.node.preview_toggle()`  
Toggle preview display for selected nodes

`bpy.ops.node.properties()`  
Toggles the properties panel display

`bpy.ops.node.read_fullsamplelayers()`  
Undocumented (contribute)

`bpy.ops.node.read_renderlayers()`  
Undocumented (contribute)

`bpy.ops.node.render_changed()`  
Undocumented (contribute)

`bpy.ops.node.resize()`  
Undocumented (contribute)

`bpy.ops.node.select(mouse_x=0, mouse_y=0, extend=False)`  
Select the node under the cursor

**Parameters**

- **mouse\_x** (*int in [-inf, inf], (optional)*) – Mouse X
- **mouse\_y** (*int in [-inf, inf], (optional)*) – Mouse Y
- **extend** (*boolean, (optional)*) – Extend

`bpy.ops.node.select_all()`  
(De)select all nodes

`bpy.ops.node.select_border` (*gesture\_mode=0, xmin=0, xmax=0, ymin=0, ymax=0, tweak=False*)  
Use box selection to select nodes

#### Parameters

- **gesture\_mode** (*int in [-inf, inf], (optional)*) – Gesture Mode
- **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
- **tweak** (*boolean, (optional)*) – Tweak, Only activate when mouse is not over a node - useful for tweak gesture

`bpy.ops.node.select_link_viewer` (*NODE\_OT\_select=None, NODE\_OT\_link\_viewer=None*)  
Undocumented ([contribute](#))

#### Parameters

- **NODE\_OT\_select** (*NODE\_OT\_select, (optional)*) – Select, Select the node under the cursor
- **NODE\_OT\_link\_viewer** (*NODE\_OT\_link\_viewer, (optional)*) – Link to Viewer Node, Link to viewer node

`bpy.ops.node.select_linked_from()`  
Select nodes linked from the selected ones

`bpy.ops.node.select_linked_to()`  
Select nodes linked to the selected ones

`bpy.ops.node.select_same_type()`  
Select all the nodes of the same type

`bpy.ops.node.select_same_type_next()`  
Select the next node of the same type

`bpy.ops.node.select_same_type_prev()`  
Select the prev node of the same type

`bpy.ops.node.show_cyclic_dependencies()`  
Sort the nodes and show the cyclic dependencies between the nodes

`bpy.ops.node.view_all()`  
Resize view so you can see all nodes

`bpy.ops.node.visibility_toggle` (*mouse\_x=0, mouse\_y=0*)  
Handle clicks on node header buttons

#### Parameters

- **mouse\_x** (*int in [-inf, inf], (optional)*) – Mouse X
- **mouse\_y** (*int in [-inf, inf], (optional)*) – Mouse Y

## Object Operators

```
bpy.ops.object.add (type='EMPTY', view_align=False, enter_editmode=False, location=(0.0, 0.0, 0.0), rotation=(0.0, 0.0, 0.0), layers=(False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False))
```

Add an object to the scene

### Parameters

- **type** (*enum in ['MESH', 'CURVE', 'SURFACE', 'META', 'FONT', 'ARMATURE', 'LATTICE', 'EMPTY', 'CAMERA', 'LAMP', 'SPEAKER'], (optional)*) – Type
- **view\_align** (*boolean, (optional)*) – Align to View, Align the new object to the view
- **enter\_editmode** (*boolean, (optional)*) – Enter Editmode, Enter editmode when adding this object
- **location** (*float array of 3 items in [-inf, inf], (optional)*) – Location, Location for the newly added object
- **rotation** (*float array of 3 items in [-inf, inf], (optional)*) – Rotation, Rotation for the newly added object
- **layers** (*boolean array of 20 items, (optional)*) – Layer

```
bpy.ops.object.add_named (linked=False, name="Cube")
```

Add named object

### Parameters

- **linked** (*boolean, (optional)*) – Linked, Duplicate object but not object data, linking to the original data
- **name** (*string, (optional)*) – Name, Object name to add

```
bpy.ops.object.add_named_cursor (name="Cube", VIEW3D_OT_cursor3d=None, OBJECT_OT_add_named=None)
```

Undocumented ([contribute](#))

### Parameters

- **name** (*string, (optional)*) – Name, Object name to add
- **VIEW3D\_OT\_cursor3d** (*VIEW3D\_OT\_cursor3d, (optional)*) – Set 3D Cursor, Set the location of the 3D cursor
- **OBJECT\_OT\_add\_named** (*OBJECT\_OT\_add\_named, (optional)*) – Add Named Object, Add named object

```
bpy.ops.object.align (bb_quality=True, align_mode='OPT_2', relative_to='OPT_4', align_axis=set())
```

Align Objects

### Parameters

- **bb\_quality** (*boolean, (optional)*) – High Quality, Enables high quality calculation of the bounding box for perfect results on complex shape meshes with rotation/scale (Slow)
- **align\_mode** (*enum in ['OPT\_1', 'OPT\_2', 'OPT\_3'], (optional)*) – Align Mode:
- **relative\_to** (*enum in ['OPT\_1', 'OPT\_2', 'OPT\_3', 'OPT\_4'], (optional)*) – Relative To:
- **align\_axis** (*enum set in {'X', 'Y', 'Z'}, (optional)*) – Align, Align to axis

**File** [startup/bl\\_operators/object\\_align.py:387](#)

`bpy.ops.object.anim_transforms_to_deltas()`

Convert object animation for normal transforms to delta transforms

**File** `startup/bl_operators/object.py:701`

`bpy.ops.object.armature_add`(*view\_align=False, enter\_editmode=False, location=(0.0, 0.0, 0.0), rotation=(0.0, 0.0, 0.0), layers=(False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False)*)

Add an armature object to the scene

#### Parameters

- **view\_align** (*boolean, (optional)*) – Align to View, Align the new object to the view
- **enter\_editmode** (*boolean, (optional)*) – Enter Editmode, Enter editmode when adding this object
- **location** (*float array of 3 items in [-inf, inf], (optional)*) – Location, Location for the newly added object
- **rotation** (*float array of 3 items in [-inf, inf], (optional)*) – Rotation, Rotation for the newly added object
- **layers** (*boolean array of 20 items, (optional)*) – Layer

`bpy.ops.object.bake_image()`

Bake image textures of selected objects

`bpy.ops.object.camera_add`(*view\_align=False, enter\_editmode=False, location=(0.0, 0.0, 0.0), rotation=(0.0, 0.0, 0.0), layers=(False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False)*)

Add a camera object to the scene

#### Parameters

- **view\_align** (*boolean, (optional)*) – Align to View, Align the new object to the view
- **enter\_editmode** (*boolean, (optional)*) – Enter Editmode, Enter editmode when adding this object
- **location** (*float array of 3 items in [-inf, inf], (optional)*) – Location, Location for the newly added object
- **rotation** (*float array of 3 items in [-inf, inf], (optional)*) – Rotation, Rotation for the newly added object
- **layers** (*boolean array of 20 items, (optional)*) – Layer

`bpy.ops.object.constraint_add`(*type=''*)

Add a constraint to the active object

**Parameters type** (*enum in ['COPY\_LOCATION', 'COPY\_ROTATION', 'COPY\_SCALE', 'COPY\_TRANSFORMS', 'LIMIT\_DISTANCE', 'LIMIT\_LOCATION', 'LIMIT\_ROTATION', 'LIMIT\_SCALE', 'MAINTAIN\_VOLUME', 'TRANSFORM', 'CLAMP\_TO', 'DAMPED\_TRACK', 'IK', 'LOCKED\_TRACK', 'SPLINE\_IK', 'STRETCH\_TO', 'TRACK\_TO', 'ACTION', 'CHILD\_OF', 'FLOOR', 'FOLLOW\_PATH', 'PIVOT', 'RIGID\_BODY\_JOINT', 'SCRIPT', 'SHRINKWRAP']*, (*optional*)) – Type

- `COPY_LOCATION` Copy Location.
- `COPY_ROTATION` Copy Rotation.
- `COPY_SCALE` Copy Scale.

- COPY\_TRANSFORMS Copy Transforms.
- LIMIT\_DISTANCE Limit Distance.
- LIMIT\_LOCATION Limit Location.
- LIMIT\_ROTATION Limit Rotation.
- LIMIT\_SCALE Limit Scale.
- MAINTAIN\_VOLUME Maintain Volume.
- TRANSFORM Transformation.
- CLAMP\_TO Clamp To.
- DAMPED\_TRACK Damped Track, Tracking by taking the shortest path.
- IK Inverse Kinematics.
- LOCKED\_TRACK Locked Track, Tracking along a single axis.
- SPLINE\_IK Spline IK.
- STRETCH\_TO Stretch To.
- TRACK\_TO Track To, Legacy tracking constraint prone to twisting artifacts.
- ACTION Action.
- CHILD\_OF Child Of.
- FLOOR Floor.
- FOLLOW\_PATH Follow Path.
- PIVOT Pivot.
- RIGID\_BODY\_JOINT Rigid Body Joint.
- SCRIPT Script.
- SHRINKWRAP Shrinkwrap.

`bpy.ops.object.constraint_add_with_targets` (*type=''*)

Add a constraint to the active object, with target (where applicable) set to the selected Objects/Bones

**Parameters** *type* (*enum in* [`'COPY_LOCATION'`, `'COPY_ROTATION'`, `'COPY_SCALE'`, `'COPY_TRANSFORMS'`, `'LIMIT_DISTANCE'`, `'LIMIT_LOCATION'`, `'LIMIT_ROTATION'`, `'LIMIT_SCALE'`, `'MAINTAIN_VOLUME'`, `'TRANSFORM'`, `'CLAMP_TO'`, `'DAMPED_TRACK'`, `'IK'`, `'LOCKED_TRACK'`, `'SPLINE_IK'`, `'STRETCH_TO'`, `'TRACK_TO'`, `'ACTION'`, `'CHILD_OF'`, `'FLOOR'`, `'FOLLOW_PATH'`, `'PIVOT'`, `'RIGID_BODY_JOINT'`, `'SCRIPT'`, `'SHRINKWRAP'`], (*optional*)) – Type

- COPY\_LOCATION Copy Location.
- COPY\_ROTATION Copy Rotation.
- COPY\_SCALE Copy Scale.
- COPY\_TRANSFORMS Copy Transforms.
- LIMIT\_DISTANCE Limit Distance.
- LIMIT\_LOCATION Limit Location.
- LIMIT\_ROTATION Limit Rotation.
- LIMIT\_SCALE Limit Scale.



- `MAINTAIN_VOLUME` Maintain Volume.
- `TRANSFORM` Transformation.
- `CLAMP_TO` Clamp To.
- `DAMPED_TRACK` Damped Track, Tracking by taking the shortest path.
- `IK` Inverse Kinematics.
- `LOCKED_TRACK` Locked Track, Tracking along a single axis.
- `SPLINE_IK` Spline IK.
- `STRETCH_TO` Stretch To.
- `TRACK_TO` Track To, Legacy tracking constraint prone to twisting artifacts.
- `ACTION` Action.
- `CHILD_OF` Child Of.
- `FLOOR` Floor.
- `FOLLOW_PATH` Follow Path.
- `PIVOT` Pivot.
- `RIGID_BODY_JOINT` Rigid Body Joint.
- `SCRIPT` Script.
- `SHRINKWRAP` Shrinkwrap.

`bpy.ops.object.constraints_clear()`  
Clear all the constraints for the active Object only

`bpy.ops.object.constraints_copy()`  
Copy constraints to other selected objects

`bpy.ops.object.convert(target='MESH', keep_original=False)`  
Convert selected objects to another type

#### Parameters

- **target** (*enum in ['CURVE', 'MESH'], (optional)*) – Target, Type of object to convert to
- **keep\_original** (*boolean, (optional)*) – Keep Original, Keep original objects instead of replacing them

`bpy.ops.object.delete()`  
Delete selected objects

`bpy.ops.object.drop_named_material(name="Material")`  
Undocumented ([contribute](#))

**Parameters** **name** (*string, (optional)*) – Name, Material name to assign

`bpy.ops.object.duplicate(linked=False, mode='TRANSLATION')`  
Duplicate selected objects

#### Parameters

- **linked** (*boolean, (optional)*) – Linked, Duplicate object but not object data, linking to the original data

- **mode** (*enum in ['INIT', 'DUMMY', 'TRANSLATION', 'ROTATION', 'RESIZE', 'TOSPHERE', 'SHEAR', 'WARP', 'SHRINKFATTEN', 'TILT', 'TRACKBALL', 'PUSHPULL', 'CREASE', 'MIRROR', 'BONE\_SIZE', 'BONE\_ENVELOPE', 'CURVE\_SHRINKFATTEN', 'BONE\_ROLL', 'TIME\_TRANSLATE', 'TIME\_SLIDE', 'TIME\_SCALE', 'TIME\_EXTEND', 'BAKE\_TIME', 'BEVEL', 'BWEIGHT', 'ALIGN', 'EDGESLIDE', 'SEQSLIDE'], (optional)*) – Mode

`bpy.ops.object.duplicate_move` (*OBJECT\_OT\_duplicate=None, TRANSFORM\_OT\_translate=None*)

Undocumented ([contribute](#))

#### Parameters

- **OBJECT\_OT\_duplicate** (*OBJECT\_OT\_duplicate, (optional)*) – Duplicate Objects, Duplicate selected objects
- **TRANSFORM\_OT\_translate** (*TRANSFORM\_OT\_translate, (optional)*) – Translate, Translate selected items

`bpy.ops.object.duplicate_move_linked` (*OBJECT\_OT\_duplicate=None, TRANSFORM\_OT\_translate=None*)

Undocumented ([contribute](#))

#### Parameters

- **OBJECT\_OT\_duplicate** (*OBJECT\_OT\_duplicate, (optional)*) – Duplicate Objects, Duplicate selected objects
- **TRANSFORM\_OT\_translate** (*TRANSFORM\_OT\_translate, (optional)*) – Translate, Translate selected items

`bpy.ops.object.duplicates_make_real` ()  
Make dupli objects attached to this object real

`bpy.ops.object.editmode_toggle` ()  
Toggle object's editmode

`bpy.ops.object.effector_add` (*type='FORCE', view\_align=False, enter\_editmode=False, location=(0.0, 0.0, 0.0), rotation=(0.0, 0.0, 0.0), layers=(False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False)*)

Add an empty object with a physics effector to the scene

#### Parameters

- **type** (*enum in ['FORCE', 'WIND', 'VORTEX', 'MAGNET', 'HARMONIC', 'CHARGE', 'LENNARDJ', 'TEXTURE', 'GUIDE', 'BOID', 'TURBULENCE', 'DRAG'], (optional)*) – Type
- **view\_align** (*boolean, (optional)*) – Align to View, Align the new object to the view
- **enter\_editmode** (*boolean, (optional)*) – Enter Editmode, Enter editmode when adding this object
- **location** (*float array of 3 items in [-inf, inf], (optional)*) – Location, Location for the newly added object
- **rotation** (*float array of 3 items in [-inf, inf], (optional)*) – Rotation, Rotation for the newly added object
- **layers** (*boolean array of 20 items, (optional)*) – Layer

`bpy.ops.object.explode_refresh` (*modifier=""*)  
Refresh data in the Explode modifier

**Parameters** `modifier` (*string, (optional)*) – Modifier, Name of the modifier to edit

`bpy.ops.object.forcefield_toggle()`  
Toggle object's force field

`bpy.ops.object.game_property_clear()`  
Undocumented ([contribute](#))

`bpy.ops.object.game_property_copy(operation='COPY', property='')`  
Undocumented ([contribute](#))

#### Parameters

- **operation** (*enum in ['REPLACE', 'MERGE', 'COPY'], (optional)*) – Operation
- **property** (*enum in [], (optional)*) – Property, Properties to copy

`bpy.ops.object.game_property_new()`  
Create a new property available to the game engine

`bpy.ops.object.game_property_remove(index=0)`  
Remove game property

**Parameters** `index` (*int in [0, inf], (optional)*) – Index, Property index to remove

`bpy.ops.object.group_add()`  
Add an object to a new group

`bpy.ops.object.group_instance_add(group='', view_align=False, location=(0.0, 0.0, 0.0), rotation=(0.0, 0.0, 0.0), layers=(False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False))`

Add a dupligroup instance

#### Parameters

- **group** (*enum in [], (optional)*) – Group
- **view\_align** (*boolean, (optional)*) – Align to View, Align the new object to the view
- **location** (*float array of 3 items in [-inf, inf], (optional)*) – Location, Location for the newly added object
- **rotation** (*float array of 3 items in [-inf, inf], (optional)*) – Rotation, Rotation for the newly added object
- **layers** (*boolean array of 20 items, (optional)*) – Layer

`bpy.ops.object.group_link(group='')`  
Add an object to an existing group

**Parameters** `group` (*enum in [], (optional)*) – Group

`bpy.ops.object.group_remove()`  
Undocumented ([contribute](#))

`bpy.ops.object.hide_render_clear()`  
Reveal the render object by setting the hide render flag

`bpy.ops.object.hide_render_clear_all()`  
Reveal all render objects by setting the hide render flag

**File** `startup/bl_operators/object.py:684`

`bpy.ops.object.hide_render_set(unselected=False)`  
Hide the render object by setting the hide render flag

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

`bpy.ops.object.hide_view_clear()`  
Reveal the object by setting the hide flag

`bpy.ops.object.hide_view_set(unselected=False)`  
Hide the object by setting the hide flag

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

`bpy.ops.object.hook_add_newob()`  
Hook selected vertices to the first selected Object

`bpy.ops.object.hook_add_selob()`  
Hook selected vertices to the first selected Object

`bpy.ops.object.hook_assign(modifier='')`  
Assign the selected vertices to a hook

**Parameters modifier** (*enum in [], (optional)*) – Modifier, Modifier number to assign to

`bpy.ops.object.hook_recenter(modifier='')`  
Set hook center to cursor position

**Parameters modifier** (*enum in [], (optional)*) – Modifier, Modifier number to assign to

`bpy.ops.object.hook_remove(modifier='')`  
Remove a hook from the active object

**Parameters modifier** (*enum in [], (optional)*) – Modifier, Modifier number to remove

`bpy.ops.object.hook_reset(modifier='')`  
Recalculate and clear offset transformation

**Parameters modifier** (*enum in [], (optional)*) – Modifier, Modifier number to assign to

`bpy.ops.object.hook_select(modifier='')`  
Select affected vertices on mesh

**Parameters modifier** (*enum in [], (optional)*) – Modifier, Modifier number to remove

`bpy.ops.object.isolate_type_render()`  
Hide unselected render objects of same type as active by setting the hide render flag

**File** `startup/bl_operators/object.py:664`

`bpy.ops.object.join()`  
Join selected objects into active object

`bpy.ops.object.join_shapes()`  
Merge selected objects to shapes of active object

`bpy.ops.object.join_uv()`  
Copy UV Layout to objects with matching geometry

**File** `startup/bl_operators/object.py:578`

`bpy.ops.object.lamp_add(type='POINT', view_align=False, location=(0.0, 0.0, 0.0), rotation=(0.0, 0.0, 0.0), layers=(False, False, False, False, False, False, False, False, False, False, False, False, False, False, False))`

Add a lamp object to the scene

**Parameters**

- **type** (*enum in ['POINT', 'SUN', 'SPOT', 'HEMI', 'AREA'], (optional)*) – Type
  - POINT Point, Omnidirectional point light source.
  - SUN Sun, Constant direction parallel ray light source.
  - SPOT Spot, Directional cone light source.
  - HEMI Hemi, 180 degree constant light source.
  - AREA Area, Directional area light source.
- **view\_align** (*boolean, (optional)*) – Align to View, Align the new object to the view
- **location** (*float array of 3 items in [-inf, inf], (optional)*) – Location, Location for the newly added object
- **rotation** (*float array of 3 items in [-inf, inf], (optional)*) – Rotation, Rotation for the newly added object
- **layers** (*boolean array of 20 items, (optional)*) – Layer

`bpy.ops.object.location_clear()`  
Clear the object's location

`bpy.ops.object.logic_bricks_copy()`  
Copy logic bricks to other selected objects

`bpy.ops.object.make_dupli_face()`  
Make linked objects into dupli-faces

**File** `startup/bl_operators/object.py:652`

`bpy.ops.object.make_links_data(type='OBDATA')`  
Make links from the active object to other selected objects

**Parameters** `type` (*enum in ['OBDATA', 'MATERIAL', 'ANIMATION', 'DUPLIGROUP', 'MODIFIERS'], (optional)*) – Type

`bpy.ops.object.make_links_scene(scene='')`  
Link selection to another scene

**Parameters** `scene` (*enum in [], (optional)*) – Scene

`bpy.ops.object.make_local(type='SELECTED_OBJECTS')`  
Make library linked datablocks local to this file

**Parameters** `type` (*enum in ['SELECTED\_OBJECTS', 'SELECTED\_OBJECTS\_DATA', 'ALL'], (optional)*) – Type

`bpy.ops.object.make_single_user(type='SELECTED_OBJECTS', object=False, obdata=False, material=False, texture=False, animation=False)`  
Make linked data local to each object

#### Parameters

- **type** (*enum in ['SELECTED\_OBJECTS', 'ALL'], (optional)*) – Type
- **object** (*boolean, (optional)*) – Object, Make single user objects
- **obdata** (*boolean, (optional)*) – Object Data, Make single user object data
- **material** (*boolean, (optional)*) – Materials, Make materials local to each datablock
- **texture** (*boolean, (optional)*) – Textures, Make textures local to each material
- **animation** (*boolean, (optional)*) – Object Animation, Make animation data local to each object

`bpy.ops.object.material_slot_add()`  
Add a new material slot

`bpy.ops.object.material_slot_assign()`  
Assign the material in the selected material slot to the selected vertices

`bpy.ops.object.material_slot_copy()`  
Copies materials to other selected objects

`bpy.ops.object.material_slot_deselect()`  
Deselect vertices assigned to the selected material slot

`bpy.ops.object.material_slot_remove()`  
Remove the selected material slot

`bpy.ops.object.material_slot_select()`  
Select vertices assigned to the selected material slot

`bpy.ops.object.meshdeform_bind(modifier="")`  
Bind mesh to cage in mesh deform modifier

**Parameters** `modifier` (*string, (optional)*) – Modifier, Name of the modifier to edit

`bpy.ops.object.metaball_add(type='BALL', view_align=False, enter_editmode=False, location=(0.0, 0.0, 0.0), rotation=(0.0, 0.0, 0.0), layers=(False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False))`

Add an metaball object to the scene

#### Parameters

- **type** (*enum in ['BALL', 'CAPSULE', 'PLANE', 'ELLIPSOID', 'CUBE'], (optional)*) – Primitive
- **view\_align** (*boolean, (optional)*) – Align to View, Align the new object to the view
- **enter\_editmode** (*boolean, (optional)*) – Enter Editmode, Enter editmode when adding this object
- **location** (*float array of 3 items in [-inf, inf], (optional)*) – Location, Location for the newly added object
- **rotation** (*float array of 3 items in [-inf, inf], (optional)*) – Rotation, Rotation for the newly added object
- **layers** (*boolean array of 20 items, (optional)*) – Layer

`bpy.ops.object.mode_set(mode='OBJECT', toggle=False)`  
Sets the object interaction mode

#### Parameters

- **mode** (*enum in ['OBJECT', 'EDIT', 'SCULPT', 'VERTEX\_PAINT', 'WEIGHT\_PAINT', 'TEXTURE\_PAINT', 'PARTICLE\_EDIT', 'POSE'], (optional)*) – Mode
- **toggle** (*boolean, (optional)*) – Toggle

`bpy.ops.object.modifier_add(type='SUBSURF')`  
Add a modifier to the active object

**Parameters** `type` (*enum in ['UV\_PROJECT', 'VERTEX\_WEIGHT\_EDIT', 'VERTEX\_WEIGHT\_MIX', 'VERTEX\_WEIGHT\_PROXIMITY', 'ARRAY', 'BEVEL', 'BOOLEAN', 'BUILD', 'DECIMATE', 'EDGE\_SPLIT', 'MASK', 'MIRROR', 'MULTIRES', 'SCREW', 'SOLIDIFY', 'SUBSURF', 'ARMATURE', 'CAST', 'CURVE', 'DISPLACE', 'HOOK', 'LATTICE', 'MESH\_DEFORM', 'SHRINKWRAP', 'SIMPLE\_DEFORM', 'SMOOTH', 'WARP', 'WAVE',*

`'CLOTH', 'COLLISION', 'EXPLODE', 'FLUID_SIMULATION', 'PARTICLE_INSTANCE', 'PARTICLE_SYSTEM', 'SMOKE', 'SOFT_BODY', 'SURFACE']`, (optional)) – Type

`bpy.ops.object.modifier_apply` (*apply\_as*='DATA', *modifier*="")

Apply modifier and remove from the stack

#### Parameters

- **apply\_as** (*enum in ['DATA', 'SHAPE']*, (optional)) – Apply as, How to apply the modifier to the geometry
  - DATA Object Data, Apply modifier to the object's data.
  - SHAPE New Shape, Apply deform-only modifier to a new shape on this object.
- **modifier** (*string*, (optional)) – Modifier, Name of the modifier to edit

`bpy.ops.object.modifier_convert` (*modifier*="")

Convert particles to a mesh object

**Parameters** **modifier** (*string*, (optional)) – Modifier, Name of the modifier to edit

`bpy.ops.object.modifier_copy` (*modifier*="")

Duplicate modifier at the same position in the stack

**Parameters** **modifier** (*string*, (optional)) – Modifier, Name of the modifier to edit

`bpy.ops.object.modifier_move_down` (*modifier*="")

Move modifier down in the stack

**Parameters** **modifier** (*string*, (optional)) – Modifier, Name of the modifier to edit

`bpy.ops.object.modifier_move_up` (*modifier*="")

Move modifier up in the stack

**Parameters** **modifier** (*string*, (optional)) – Modifier, Name of the modifier to edit

`bpy.ops.object.modifier_remove` (*modifier*="")

Remove a modifier from the active object

**Parameters** **modifier** (*string*, (optional)) – Modifier, Name of the modifier to edit

`bpy.ops.object.move_to_layer` (*layers*=(False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False))

Move the object to different layers

**Parameters** **layers** (*boolean array of 20 items*, (optional)) – Layer

`bpy.ops.object.multires_base_apply` (*modifier*="")

Modify the base mesh to conform to the displaced mesh

**Parameters** **modifier** (*string*, (optional)) – Modifier, Name of the modifier to edit

`bpy.ops.object.multires_external_pack` ()

Pack displacements from an external file

`bpy.ops.object.multires_external_save` (*filepath*="", *check\_existing*=True, *filter\_blender*=False, *filter\_image*=False, *filter\_movie*=False, *filter\_python*=False, *filter\_font*=False, *filter\_sound*=False, *filter\_text*=False, *filter\_btx*=True, *filter\_collada*=False, *filter\_folder*=True, *filemode*=9, *relative\_path*=True, *modifier*="")

Save displacements to an external file

**Parameters**

- **filepath** (*string, (optional)*) – File Path, Path to file
- **check\_existing** (*boolean, (optional)*) – Check Existing, Check and warn on overwriting existing files
- **filter\_blender** (*boolean, (optional)*) – Filter .blend files
- **filter\_image** (*boolean, (optional)*) – Filter image files
- **filter\_movie** (*boolean, (optional)*) – Filter movie files
- **filter\_python** (*boolean, (optional)*) – Filter python files
- **filter\_font** (*boolean, (optional)*) – Filter font files
- **filter\_sound** (*boolean, (optional)*) – Filter sound files
- **filter\_text** (*boolean, (optional)*) – Filter text files
- **filter\_btx** (*boolean, (optional)*) – Filter btx files
- **filter\_collada** (*boolean, (optional)*) – Filter COLLADA files
- **filter\_folder** (*boolean, (optional)*) – Filter folders
- **filemode** (*int in [1, 9], (optional)*) – File Browser Mode, The setting for the file browser mode to load a .blend file, a library or a special file
- **relative\_path** (*boolean, (optional)*) – Relative Path, Select the file relative to the blend file
- **modifier** (*string, (optional)*) – Modifier, Name of the modifier to edit

`bpy.ops.object.multires_higher_levels_delete(modifier="")`  
Deletes the higher resolution mesh, potential loss of detail

**Parameters** **modifier** (*string, (optional)*) – Modifier, Name of the modifier to edit

`bpy.ops.object.multires_reshape(modifier="")`  
Copy vertex coordinates from other object

**Parameters** **modifier** (*string, (optional)*) – Modifier, Name of the modifier to edit

`bpy.ops.object.multires_subdivide(modifier="")`  
Add a new level of subdivision

**Parameters** **modifier** (*string, (optional)*) – Modifier, Name of the modifier to edit

`bpy.ops.object.origin_clear()`  
Clear the object's origin

`bpy.ops.object.origin_set(type='GEOMETRY_ORIGIN', center='MEDIAN')`  
Set the object's origin, by either moving the data, or set to center of data, or use 3d cursor

**Parameters**

- **type** (*enum in ['GEOMETRY\_ORIGIN', 'ORIGIN\_GEOMETRY', 'ORIGIN\_CURSOR'], (optional)*) – Type
  - `GEOMETRY_ORIGIN` Geometry to Origin, Move object geometry to object origin.
  - `ORIGIN_GEOMETRY` Origin to Geometry, Move object origin to center of object geometry.
  - `ORIGIN_CURSOR` Origin to 3D Cursor, Move object origin to position of the 3d cursor.
- **center** (*enum in ['MEDIAN', 'BOUNDS'], (optional)*) – Center



`bpy.ops.object.parent_clear` (*type*='CLEAR')  
Clear the object's parenting

**Parameters** *type* (*enum in* ['CLEAR', 'CLEAR\_KEEP\_TRANSFORM', 'CLEAR\_INVERSE'], (*optional*)) – Type

`bpy.ops.object.parent_no_inverse_set` ()  
Set the object's parenting without setting the inverse parent correction

`bpy.ops.object.parent_set` (*type*='OBJECT')  
Set the object's parenting

**Parameters** *type* (*enum in* ['OBJECT', 'ARMATURE', 'ARMATURE\_NAME', 'ARMATURE\_AUTO', 'ARMATURE\_ENVELOPE', 'BONE', 'CURVE', 'FOLLOW', 'PATH\_CONST', 'LATTICE', 'VERTEX', 'TRIA'], (*optional*)) – Type

`bpy.ops.object.particle_system_add` ()  
Add a particle system

`bpy.ops.object.particle_system_remove` ()  
Remove the selected particle system

`bpy.ops.object.paths_calculate` ()  
Calculate paths for the selected bones

`bpy.ops.object.paths_clear` ()  
Clear path caches for selected bones

`bpy.ops.object.posemode_toggle` ()  
Enable or disable posing/selecting bones

`bpy.ops.object.proxy_make` (*object*="", *type*='DEFAULT')  
Add empty object to become local replacement data of a library-linked object

#### Parameters

- **object** (*string, optional*) – Proxy Object, Name of lib-linked/grouped object to make a proxy for
- **type** (*enum in* ['DEFAULT'], (*optional*)) – Type, Group object

`bpy.ops.object.quick_explode` (*style*='EXPLODE', *amount*=100, *frame\_duration*=50, *frame\_start*=1, *frame\_end*=10, *velocity*=1.0, *fade*=True)

Undocumented ([contribute](#))

#### Parameters

- **style** (*enum in* ['EXPLODE', 'BLEND'], (*optional*)) – Explode Style
- **amount** (*int in* [2, 10000], (*optional*)) – Amount of pieces
- **frame\_duration** (*int in* [1, 300000], (*optional*)) – Duration
- **frame\_start** (*int in* [1, 300000], (*optional*)) – Start Frame
- **frame\_end** (*int in* [1, 300000], (*optional*)) – End Frame
- **velocity** (*float in* [0, 300000], (*optional*)) – Outwards Velocity
- **fade** (*boolean, optional*) – Fade, Fade the pieces over time

**File** `startup/bl_operators/object_quick_effects.py:164`

`bpy.ops.object.quick_fluid` (*style*='BASIC', *initial\_velocity*=(0.0, 0.0, 0.0), *show\_flows*=False, *start\_baking*=False)

Undocumented ([contribute](#))

**Parameters**

- **style** (*enum in ['INFLOW', 'BASIC'], (optional)*) – Fluid Style
- **initial\_velocity** (*float array of 3 items in [-100, 100], (optional)*) – Initial Velocity, Initial velocity of the fluid
- **show\_flows** (*boolean, (optional)*) – Render Fluid Objects, Keep the fluid objects visible during rendering
- **start\_baking** (*boolean, (optional)*) – Start Fluid Bake, Start baking the fluid immediately after creating the domain object

**File** `startup/bl_operators/object_quick_effects.py:441`

`bpy.ops.object.quick_fur` (*density='MEDIUM', view\_percentage=10, length=0.1*)  
Undocumented ([contribute](#))

**Parameters**

- **density** (*enum in ['LIGHT', 'MEDIUM', 'HEAVY'], (optional)*) – Fur Density
- **view\_percentage** (*int in [1, 100], (optional)*) – View %
- **length** (*float in [0.001, 100], (optional)*) – Length

**File** `startup/bl_operators/object_quick_effects.py:74`

`bpy.ops.object.quick_smoke` (*style='STREAM', show\_flows=False*)  
Undocumented ([contribute](#))

**Parameters**

- **style** (*enum in ['STREAM', 'PUFF', 'FIRE'], (optional)*) – Smoke Style
- **show\_flows** (*boolean, (optional)*) – Render Smoke Objects, Keep the smoke objects visible during rendering

**File** `startup/bl_operators/object_quick_effects.py:314`

`bpy.ops.object.randomize_transform` (*random\_seed=0, use\_delta=False, use\_loc=True, loc=(0.0, 0.0, 0.0), use\_rot=True, rot=(0.0, 0.0, 0.0), use\_scale=True, scale\_even=False, scale=(0.0, 0.0, 0.0)*)

Randomize objects loc/rot/scale

**Parameters**

- **random\_seed** (*int in [0, 1000], (optional)*) – Random Seed, Seed value for the random generator
- **use\_delta** (*boolean, (optional)*) – Transform Delta, Randomize delta transform values instead of regular transform
- **use\_loc** (*boolean, (optional)*) – Randomize Location, Randomize the location values
- **loc** (*float array of 3 items in [-100, 100], (optional)*) – Location, Maximum distance the objects can spread over each axis
- **use\_rot** (*boolean, (optional)*) – Randomize Rotation, Randomize the rotation values
- **rot** (*float array of 3 items in [-180, 180], (optional)*) – Rotation, Maximum rotation over each axis
- **use\_scale** (*boolean, (optional)*) – Randomize Scale, Randomize the scale values
- **scale\_even** (*boolean, (optional)*) – Scale Even, Use the same scale value for all axis

- **scale** (*float array of 3 items in [-100, 100], (optional)*) – Scale, Maximum scale randomization over each axis

**File** `startup/bl_operators/object_randomize_transform.py:164`

`bpy.ops.object.rotation_clear()`  
Clear the object's rotation

`bpy.ops.object.scale_clear()`  
Clear the object's scale

`bpy.ops.object.select_all(action='TOGGLE')`  
Change selection of all visible objects in scene

**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.object.select_by_layer(extend=False, layers=1)`  
Select all visible objects on a layer

#### Parameters

- **extend** (*boolean, (optional)*) – Extend, Extend selection instead of deselecting everything first
- **layers** (*int in [1, 20], (optional)*) – Layer

`bpy.ops.object.select_by_type(extend=False, type='MESH')`  
Select all visible objects that are of a type

#### Parameters

- **extend** (*boolean, (optional)*) – Extend, Extend selection instead of deselecting everything first
- **type** (*enum in ['MESH', 'CURVE', 'SURFACE', 'META', 'FONT', 'ARMATURE', 'LATTICE', 'EMPTY', 'CAMERA', 'LAMP', 'SPEAKER'], (optional)*) – Type

`bpy.ops.object.select_camera()`  
Select object matching a naming pattern

**File** `startup/bl_operators/object.py:113`

`bpy.ops.object.select_grouped(extend=False, type='CHILDREN_RECURSIVE')`  
Select all visible objects grouped by various properties

#### Parameters

- **extend** (*boolean, (optional)*) – Extend, Extend selection instead of deselecting everything first
- **type** (*enum in ['CHILDREN\_RECURSIVE', 'CHILDREN', 'PARENT', 'SIBLINGS', 'TYPE', 'LAYER', 'GROUP', 'HOOK', 'PASS', 'COLOR', 'PROPERTIES', 'KEYINGSET'], (optional)*) – Type
  - CHILDREN\_RECURSIVE Children.
  - CHILDREN Immediate Children.

- PARENT Parent.
- SIBLINGS Siblings, Shared Parent.
- TYPE Type, Shared object type.
- LAYER Layer, Shared layers.
- GROUP Group, Shared group.
- HOOK Hook.
- PASS Pass, Render pass Index.
- COLOR Color, Object Color.
- PROPERTIES Properties, Game Properties.
- KEYINGSET Keying Set, Objects included in active Keying Set.

`bpy.ops.object.select_hierarchy` (*direction='PARENT', extend=False*)  
Select object relative to the active object's position in the hierarchy

#### Parameters

- **direction** (*enum in ['PARENT', 'CHILD'], (optional)*) – Direction, Direction to select in the hierarchy
- **extend** (*boolean, (optional)*) – Extend, Extend the existing selection

**File** `startup/bl_operators/object.py:149`

`bpy.ops.object.select_inverse` ()  
Invert selection of all visible objects

`bpy.ops.object.select_linked` (*extend=False, type='OBDATA'*)  
Select all visible objects that are linked

#### Parameters

- **extend** (*boolean, (optional)*) – Extend, Extend selection instead of deselecting everything first
- **type** (*enum in ['OBDATA', 'MATERIAL', 'TEXTURE', 'DUPGROUP', 'PARTICLE', 'LIBRARY', 'LIBRARY\_OBDATA'], (optional)*) – Type

`bpy.ops.object.select_mirror` (*extend=False*)  
Select the Mirror objects of the selected object eg. L.sword -> R.sword

**Parameters** **extend** (*boolean, (optional)*) – Extend, Extend selection instead of deselecting everything first

`bpy.ops.object.select_name` (*name="" , extend=False*)  
Select an object with this name

#### Parameters

- **name** (*string, (optional)*) – Name, Object name to select
- **extend** (*boolean, (optional)*) – Extend, Extend selection instead of deselecting everything first

`bpy.ops.object.select_pattern` (*pattern="\*\*", case\_sensitive=False, extend=True*)  
Select object matching a naming pattern

#### Parameters

- **pattern** (*string, (optional)*) – Pattern, Name filter using '\*' and '?' wildcard chars

- **case\_sensitive** (*boolean, (optional)*) – Case Sensitive, Do a case sensitive compare
- **extend** (*boolean, (optional)*) – Extend, Extend the existing selection

**File** `startup/bl_operators/object.py:49`

`bpy.ops.object.select_random(percent=50.0, extend=False)`

Set select on random visible objects

#### Parameters

- **percent** (*float in [0, 100], (optional)*) – Percent, Percentage of objects to select randomly
- **extend** (*boolean, (optional)*) – Extend Selection, Extend selection instead of deselecting everything first

`bpy.ops.object.select_same_group(group="")`

Select object in the same group

**Parameters** **group** (*string, (optional)*) – Group, Name of the group to select

`bpy.ops.object.shade_flat()`

Display faces 'flat'

`bpy.ops.object.shade_smooth()`

Display faces 'smooth' (using vertex normals)

`bpy.ops.object.shape_key_add(from_mix=True)`

Add shape key to the object

**Parameters** **from\_mix** (*boolean, (optional)*) – From Mix, Create the new shape key from the existing mix of keys

`bpy.ops.object.shape_key_clear()`

Clear weights for all shape keys

`bpy.ops.object.shape_key_mirror()`

Undocumented ([contribute](#))

`bpy.ops.object.shape_key_move(type='UP')`

Undocumented ([contribute](#))

**Parameters** **type** (*enum in ['UP', 'DOWN'], (optional)*) – Type

`bpy.ops.object.shape_key_remove()`

Remove shape key from the object

`bpy.ops.object.shape_key_transfer(mode='OFFSET', use_clamp=False)`

Copy another selected objects active shape to this one by applying the relative offsets

#### Parameters

- **mode** (*enum in ['OFFSET', 'RELATIVE\_FACE', 'RELATIVE\_EDGE'], (optional)*) – Transformation Mode, Relative shape positions to the new shape method
  - `OFFSET` Offset, Apply the relative positional offset.
  - `RELATIVE_FACE` Relative Face, Calculate relative position (using faces).
  - `RELATIVE_EDGE` Relative Edge, Calculate relative position (using edges).
- **use\_clamp** (*boolean, (optional)*) – Clamp Offset, Clamp the transformation to the distance each vertex moves in the original shape

**File** `startup/bl_operators/object.py:491`

`bpy.ops.object.slow_parent_clear()`  
Clear the object's slow parent

`bpy.ops.object.slow_parent_set()`  
Set the object's slow parent

`bpy.ops.object.speaker_add` (*view\_align=False, enter\_editmode=False, location=(0.0, 0.0, 0.0), rotation=(0.0, 0.0, 0.0), layers=(False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False)*)

Add a speaker object to the scene

#### Parameters

- **view\_align** (*boolean, (optional)*) – Align to View, Align the new object to the view
- **enter\_editmode** (*boolean, (optional)*) – Enter Editmode, Enter editmode when adding this object
- **location** (*float array of 3 items in [-inf, inf], (optional)*) – Location, Location for the newly added object
- **rotation** (*float array of 3 items in [-inf, inf], (optional)*) – Rotation, Rotation for the newly added object
- **layers** (*boolean array of 20 items, (optional)*) – Layer

`bpy.ops.object.subdivision_set` (*level=1, relative=False*)  
Sets a Subdivision Surface Level (1-5)

#### Parameters

- **level** (*int in [-100, 100], (optional)*) – Level
- **relative** (*boolean, (optional)*) – Relative, Apply the subsurf level as an offset relative to the current level

**File** `startup/bl_operators/object.py:217`

`bpy.ops.object.text_add` (*view\_align=False, enter\_editmode=False, location=(0.0, 0.0, 0.0), rotation=(0.0, 0.0, 0.0), layers=(False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False)*)

Add a text object to the scene

#### Parameters

- **view\_align** (*boolean, (optional)*) – Align to View, Align the new object to the view
- **enter\_editmode** (*boolean, (optional)*) – Enter Editmode, Enter editmode when adding this object
- **location** (*float array of 3 items in [-inf, inf], (optional)*) – Location, Location for the newly added object
- **rotation** (*float array of 3 items in [-inf, inf], (optional)*) – Rotation, Rotation for the newly added object
- **layers** (*boolean array of 20 items, (optional)*) – Layer

`bpy.ops.object.track_clear` (*type='CLEAR'*)  
Clear tracking constraint or flag from object

**Parameters** **type** (*enum in ['CLEAR', 'CLEAR\_KEEP\_TRANSFORM'], (optional)*) – Type

`bpy.ops.object.track_set` (*type='DAMPTRACK'*)

Make the object track another object, either by constraint or old way or locked track

**Parameters** *type* (*enum in ['DAMPTRACK', 'TRACKTO', 'LOCKTRACK'], (optional)*) – Type

`bpy.ops.object.transform_apply` (*location=False, rotation=False, scale=False*)

Apply the object's transformation to its data

**Parameters**

- **location** (*boolean, (optional)*) – Location
- **rotation** (*boolean, (optional)*) – Rotation
- **scale** (*boolean, (optional)*) – Scale

`bpy.ops.object.vertex_group_add` ()

Undocumented ([contribute](#))

`bpy.ops.object.vertex_group_assign` (*new=False*)

Undocumented ([contribute](#))

**Parameters** *new* (*boolean, (optional)*) – New, Assign vertex to new vertex group

`bpy.ops.object.vertex_group_blend` ()

Undocumented ([contribute](#))

`bpy.ops.object.vertex_group_clean` (*limit=0.01, all\_groups=False, keep\_single=False*)

Remove Vertex Group assignments which aren't required

**Parameters**

- **limit** (*float in [0, 1], (optional)*) – Limit, Remove weights under this limit
- **all\_groups** (*boolean, (optional)*) – All Groups, Clean all vertex groups
- **keep\_single** (*boolean, (optional)*) – Keep Single, Keep verts assigned to at least one group when cleaning

`bpy.ops.object.vertex_group_copy` ()

Undocumented ([contribute](#))

`bpy.ops.object.vertex_group_copy_to_linked` ()

Copy Vertex Groups to all users of the same Geometry data

`bpy.ops.object.vertex_group_copy_to_selected` ()

Copy Vertex Groups to other selected objects with matching indices

`bpy.ops.object.vertex_group_deselect` ()

Undocumented ([contribute](#))

`bpy.ops.object.vertex_group_fix` (*dist=0.0, strength=1.0, accuracy=1.0*)

Modify the position of selected vertices by changing only their respective groups' weights (this tool may be slow for many vertices)

**Parameters**

- **dist** (*float in [-inf, inf], (optional)*) – Distance, The distance to move to
- **strength** (*float in [-2, inf], (optional)*) – Strength, The distance moved can be changed by this multiplier
- **accuracy** (*float in [0.05, inf], (optional)*) – Change Sensitivity, Change the amount weights are altered with each iteration: lower values are slower

`bpy.ops.object.vertex_group_invert` (*auto\_assign=True, auto\_remove=True*)  
Undocumented (contribute)

**Parameters**

- **auto\_assign** (*boolean, (optional)*) – Add Weights, Add verts from groups that have zero weight before inverting
- **auto\_remove** (*boolean, (optional)*) – Remove Weights, Remove verts from groups that have zero weight after inverting

`bpy.ops.object.vertex_group_levels` (*offset=0.0, gain=1.0*)  
Undocumented (contribute)

**Parameters**

- **offset** (*float in [-1, 1], (optional)*) – Offset, Value to add to weights
- **gain** (*float in [0, inf], (optional)*) – Gain, Value to multiply weights by

`bpy.ops.object.vertex_group_lock` (*action='TOGGLE'*)  
Undocumented (contribute)

**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.object.vertex_group_mirror` (*mirror\_weights=True, flip\_group\_names=True, all\_groups=False*)

Mirror all vertex groups, flip weights and/or names, editing only selected vertices, flipping when both sides are selected otherwise copy from unselected

**Parameters**

- **mirror\_weights** (*boolean, (optional)*) – Mirror Weights, Mirror weights
- **flip\_group\_names** (*boolean, (optional)*) – Flip Groups, Flip vertex group names
- **all\_groups** (*boolean, (optional)*) – All Groups, Mirror all vertex groups weights

`bpy.ops.object.vertex_group_move` (*direction='UP'*)  
Undocumented (contribute)

**Parameters** **direction** (*enum in ['UP', 'DOWN'], (optional)*) – Direction, Direction to move, UP or DOWN

`bpy.ops.object.vertex_group_normalize` ()  
Undocumented (contribute)

`bpy.ops.object.vertex_group_normalize_all` (*lock\_active=True*)  
Undocumented (contribute)

**Parameters** **lock\_active** (*boolean, (optional)*) – Lock Active, Keep the values of the active group while normalizing others

`bpy.ops.object.vertex_group_remove` (*all=False*)  
Undocumented (contribute)

**Parameters** **all** (*boolean, (optional)*) – All, Remove from all vertex groups



`bpy.ops.object.vertex_group_remove_from` (*all=False*)  
Undocumented (contribute)

**Parameters** `all` (*boolean, (optional)*) – All, Remove from all vertex groups

`bpy.ops.object.vertex_group_select` ()  
Undocumented (contribute)

`bpy.ops.object.vertex_group_set_active` (*group=''*)  
Set the active vertex group

**Parameters** `group` (*enum in [], (optional)*) – Group, Vertex group to set as active

`bpy.ops.object.vertex_group_sort` ()  
Sorts vertex groups alphabetically

`bpy.ops.object.vertex_parent_set` ()  
Parent selected objects to the selected vertices

`bpy.ops.object.visual_transform_apply` ()  
Apply the object's visual transformation to its data

## Outliner Operators

`bpy.ops.outliner.action_set` (*action=''*)  
Change the active action used

**Parameters** `action` (*enum in [], (optional)*) – Action

`bpy.ops.outliner.animdata_operation` (*type='SET\_ACT'*)  
Undocumented (contribute)

**Parameters** `type` (*enum in ['SET\_ACT', 'CLEAR\_ACT', 'REFRESH\_DRIVERS', 'CLEAR\_DRIVERS'], (optional)*) – Animation Operation

`bpy.ops.outliner.data_operation` (*type='SELECT'*)  
Undocumented (contribute)

**Parameters** `type` (*enum in ['SELECT', 'DESELECT', 'HIDE', 'UNHIDE'], (optional)*) – Data Operation

`bpy.ops.outliner.drivers_add_selected` ()  
Add drivers to selected items

`bpy.ops.outliner.drivers_delete_selected` ()  
Delete drivers assigned to selected items

`bpy.ops.outliner.expanded_toggle` ()  
Expand/Collapse all items

`bpy.ops.outliner.group_operation` (*type='UNLINK'*)  
Undocumented (contribute)

**Parameters** `type` (*enum in ['UNLINK', 'LOCAL', 'LINK', 'TOGVIS', 'TOGSEL', 'TOGREN', 'RENAME'], (optional)*) – Group Operation

`bpy.ops.outliner.id_operation` (*type='UNLINK'*)  
Undocumented (contribute)

**Parameters** `type` (*enum in ['UNLINK', 'LOCAL', 'SINGLE', 'ADD\_FAKE', 'CLEAR\_FAKE', 'RENAME'], (optional)*) – ID data Operation

- UNLINK Unlink.

- LOCAL Make Local.
- SINGLE Make Single User.
- ADD\_FAKE Add Fake User, Ensure datablock gets saved even if it isn't in use (e.g. for motion and material libraries).
- CLEAR\_FAKE Clear Fake User.
- RENAME Rename.

`bpy.ops.outliner.item_activate` (*extend=True*)  
Handle mouse clicks to activate/select items

**Parameters** `extend` (*boolean, (optional)*) – Extend, Extend selection for activation

`bpy.ops.outliner.item_openclose` (*all=True*)  
Toggle whether item under cursor is enabled or closed

**Parameters** `all` (*boolean, (optional)*) – All, Close or open all items

`bpy.ops.outliner.item_rename` ()  
Rename item under cursor

`bpy.ops.outliner.keyingset_add_selected` ()  
Add selected items (blue-grey rows) to active Keying Set

`bpy.ops.outliner.keyingset_remove_selected` ()  
Remove selected items (blue-grey rows) from active Keying Set

`bpy.ops.outliner.object_operation` (*type='SELECT'*)  
Undocumented ([contribute](#))

**Parameters** `type` (*enum in ['SELECT', 'DESELECT', 'DELETE', 'TOGVIS', 'TOGSEL', 'TOGREN', 'RENAME'], (optional)*) – Object Operation

`bpy.ops.outliner.operation` ()  
Context menu for item operations

`bpy.ops.outliner.renderability_toggle` ()  
Toggle the renderability of selected items

`bpy.ops.outliner.scroll_page` (*up=False*)  
Scroll page up or down

**Parameters** `up` (*boolean, (optional)*) – Up, Scroll up one page

`bpy.ops.outliner.selectability_toggle` ()  
Toggle the selectability

`bpy.ops.outliner.selected_toggle` ()  
Toggle the Outliner selection of items

`bpy.ops.outliner.show_active` ()  
Adjust the view so that the active Object is shown centered

`bpy.ops.outliner.show_hierarchy` ()  
Open all object entries and close all others

`bpy.ops.outliner.show_one_level` (*open=True*)  
Expand/collapse all entries by one level

**Parameters** `open` (*boolean, (optional)*) – Open, Expand all entries one level deep

`bpy.ops.outliner.visibility_toggle` ()  
Toggle the visibility of selected items

## Paint Operators

`bpy.ops.paint.clone_cursor_set` (*location*=(0.0, 0.0, 0.0))  
Undocumented ([contribute](#))

**Parameters** *location* (*float array of 3 items in [-inf, inf], (optional)*) – Location, Cursor location in world space coordinates

`bpy.ops.paint.face_select_all` (*action*='TOGGLE')  
Change selection for all faces

**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.paint.face_select_hide` (*unselected*=False)  
Hide selected faces

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

`bpy.ops.paint.face_select_inverse` ()  
Invert selection of faces

`bpy.ops.paint.face_select_linked` ()  
Select linked faces

`bpy.ops.paint.face_select_linked_pick` (*extend*=False)  
Select linked faces

**Parameters** *extend* (*boolean, (optional)*) – Extend, Extend the existing selection

`bpy.ops.paint.face_select_reveal` (*unselected*=False)  
Reveal hidden faces

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

`bpy.ops.paint.grab_clone` (*delta*=(0.0, 0.0))  
Undocumented ([contribute](#))

**Parameters** *delta* (*float array of 2 items in [-inf, inf], (optional)*) – Delta, Delta offset of clone image in 0.0..1.0 coordinates

`bpy.ops.paint.image_from_view` (*filepath*=""")  
Make an image from the current 3D view for re-projection

**Parameters** *filepath* (*string, (optional)*) – File Path, Name of the file

`bpy.ops.paint.image_paint` (*stroke*=None)  
Undocumented ([contribute](#))

**Parameters** *stroke* (*bpy\_prop\_collection of OperatorStrokeElement, (optional)*) – Stroke

`bpy.ops.paint.project_image` (*image*='')  
Project an edited render from the active camera back onto the object

**Parameters image** (*enum in [], (optional)*) – Image

`bpy.ops.paint.sample_color` (*location=(0, 0)*)  
Undocumented ([contribute](#))

**Parameters location** (*int array of 2 items in [0, inf], (optional)*) – Location, Cursor location in region coordinates

`bpy.ops.paint.texture_paint_toggle` ()  
Undocumented ([contribute](#))

`bpy.ops.paint.vert_select_all` (*action='TOGGLE'*)  
Change selection for all vertices

**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.paint.vert_select_inverse` ()  
Invert selection of vertices

`bpy.ops.paint.vertex_color_dirt` (*blur\_strength=1.0, blur\_iterations=1, clean\_angle=180.0, dirt\_angle=0.0, dirt\_only=False*)  
Undocumented ([contribute](#))

#### Parameters

- **blur\_strength** (*float in [0.01, 1], (optional)*) – Blur Strength, Blur strength per iteration
- **blur\_iterations** (*int in [0, 40], (optional)*) – Blur Iterations, Number of times to blur the colors (higher blurs more)
- **clean\_angle** (*float in [0, 180], (optional)*) – Highlight Angle, Less than 90 limits the angle used in the tonal range
- **dirt\_angle** (*float in [0, 180], (optional)*) – Dirt Angle, Less than 90 limits the angle used in the tonal range
- **dirt\_only** (*boolean, (optional)*) – Dirt Only, Dont calculate cleans for convex areas

**File** `startup/bl_operators/vertexpaint_dirt.py:184`

`bpy.ops.paint.vertex_color_set` ()  
Undocumented ([contribute](#))

`bpy.ops.paint.vertex_paint` (*stroke=None*)  
Undocumented ([contribute](#))

**Parameters stroke** (*bpy\_prop\_collection of OperatorStrokeElement, (optional)*) – Stroke

`bpy.ops.paint.vertex_paint_toggle` ()  
Undocumented ([contribute](#))

`bpy.ops.paint.weight_from_bones` (*type='AUTOMATIC'*)  
Undocumented ([contribute](#))

**Parameters type** (*enum in ['AUTOMATIC', 'ENVELOPES'], (optional)*) – Type, Method to use for assigning weights

- **AUTOMATIC** Automatic, Automatic weights froms bones.
- **ENVELOPES** From Envelopes, Weights from envelopes with user defined radius.

`bpy.ops.paint.weight_paint` (*stroke=None*)  
Undocumented (contribute)

**Parameters stroke** (*bpy\_prop\_collection of OperatorStrokeElement, (optional)*) – Stroke

`bpy.ops.paint.weight_paint_toggle` ()  
Undocumented (contribute)

`bpy.ops.paint.weight_sample` ()  
Undocumented (contribute)

`bpy.ops.paint.weight_sample_group` (*group='DEFAULT'*)  
Undocumented (contribute)

**Parameters group** (*enum in ['DEFAULT'], (optional)*) – Keying Set, The Keying Set to use

`bpy.ops.paint.weight_set` ()  
Undocumented (contribute)

## Particle Operators

`bpy.ops.particle.brush_edit` (*stroke=None*)  
Undocumented (contribute)

**Parameters stroke** (*bpy\_prop\_collection of OperatorStrokeElement, (optional)*) – Stroke

`bpy.ops.particle.connect_hair` (*all=False*)  
Connect hair to the emitter mesh

**Parameters all** (*boolean, (optional)*) – All hair, Connect all hair systems to the emitter mesh

`bpy.ops.particle.delete` (*type='PARTICLE'*)  
Undocumented (contribute)

**Parameters type** (*enum in ['PARTICLE', 'KEY'], (optional)*) – Type, Delete a full particle or only keys

`bpy.ops.particle.disconnect_hair` (*all=False*)  
Disconnect hair from the emitter mesh

**Parameters all** (*boolean, (optional)*) – All hair, Disconnect all hair systems from the emitter mesh

`bpy.ops.particle.dupliobj_copy` ()  
Duplicate the current dupliobject

`bpy.ops.particle.dupliobj_move_down` ()  
Move dupli object down in the list

`bpy.ops.particle.dupliobj_move_up` ()  
Move dupli object up in the list

`bpy.ops.particle.dupliobj_remove` ()  
Remove the selected dupliobject

`bpy.ops.particle.edited_clear` ()  
Undocumented (contribute)

`bpy.ops.particle.hide` (*unselected=False*)  
Undocumented (contribute)

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

`bpy.ops.particle.mirror` ()  
Undocumented (contribute)

`bpy.ops.particle.new` ()  
Add new particle settings

`bpy.ops.particle.new_target` ()  
Add a new particle target

`bpy.ops.particle.particle_edit_toggle` ()  
Undocumented (contribute)

`bpy.ops.particle.rekey` (*keys=2*)  
Undocumented (contribute)

**Parameters** `keys` (*int in [2, inf], (optional)*) – Number of Keys

`bpy.ops.particle.remove_doubles` (*threshold=0.0002*)  
Undocumented (contribute)

**Parameters** `threshold` (*float in [0, inf], (optional)*) – Threshold, Threshold distance withing which particles are removed

`bpy.ops.particle.reveal` ()  
Undocumented (contribute)

`bpy.ops.particle.select_all` (*action='TOGGLE'*)  
Undocumented (contribute)

**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.particle.select_inverse` ()  
Undocumented (contribute)

`bpy.ops.particle.select_less` ()  
Undocumented (contribute)

`bpy.ops.particle.select_linked` (*deselect=False, location=(0, 0)*)  
Undocumented (contribute)

**Parameters**

- `deselect` (*boolean, (optional)*) – Deselect, Deselect linked keys rather than selecting them
- `location` (*int array of 2 items in [0, inf], (optional)*) – Location

`bpy.ops.particle.select_more` ()  
Undocumented (contribute)

`bpy.ops.particle.select_roots` ()  
Undocumented (contribute)



- **percentage** (*float in [0, 1], (optional)*) – Percentage, Weighting factor for the sliding operation

`bpy.ops.pose.constraint_add` (*type=''*)

Add a constraint to the active bone

**Parameters type** (*enum in ['COPY\_LOCATION', 'COPY\_ROTATION', 'COPY\_SCALE', 'COPY\_TRANSFORMS', 'LIMIT\_DISTANCE', 'LIMIT\_LOCATION', 'LIMIT\_ROTATION', 'LIMIT\_SCALE', 'MAINTAIN\_VOLUME', 'TRANSFORM', 'CLAMP\_TO', 'DAMPED\_TRACK', 'IK', 'LOCKED\_TRACK', 'SPLINE\_IK', 'STRETCH\_TO', 'TRACK\_TO', 'ACTION', 'CHILD\_OF', 'FLOOR', 'FOLLOW\_PATH', 'PIVOT', 'RIGID\_BODY\_JOINT', 'SCRIPT', 'SHRINKWRAP'], (optional)*) – Type

- `COPY_LOCATION` Copy Location.
- `COPY_ROTATION` Copy Rotation.
- `COPY_SCALE` Copy Scale.
- `COPY_TRANSFORMS` Copy Transforms.
- `LIMIT_DISTANCE` Limit Distance.
- `LIMIT_LOCATION` Limit Location.
- `LIMIT_ROTATION` Limit Rotation.
- `LIMIT_SCALE` Limit Scale.
- `MAINTAIN_VOLUME` Maintain Volume.
- `TRANSFORM` Transformation.
- `CLAMP_TO` Clamp To.
- `DAMPED_TRACK` Damped Track, Tracking by taking the shortest path.
- `IK` Inverse Kinematics.
- `LOCKED_TRACK` Locked Track, Tracking along a single axis.
- `SPLINE_IK` Spline IK.
- `STRETCH_TO` Stretch To.
- `TRACK_TO` Track To, Legacy tracking constraint prone to twisting artifacts.
- `ACTION` Action.
- `CHILD_OF` Child Of.
- `FLOOR` Floor.
- `FOLLOW_PATH` Follow Path.
- `PIVOT` Pivot.
- `RIGID_BODY_JOINT` Rigid Body Joint.
- `SCRIPT` Script.
- `SHRINKWRAP` Shrinkwrap.

`bpy.ops.pose.constraint_add_with_targets` (*type=''*)

Add a constraint to the active bone, with target (where applicable) set to the selected Objects/Bones



**Parameters type** (*enum in* [`'COPY_LOCATION'`, `'COPY_ROTATION'`, `'COPY_SCALE'`, `'COPY_TRANSFORMS'`, `'LIMIT_DISTANCE'`, `'LIMIT_LOCATION'`, `'LIMIT_ROTATION'`, `'LIMIT_SCALE'`, `'MAINTAIN_VOLUME'`, `'TRANSFORM'`, `'CLAMP_TO'`, `'DAMPED_TRACK'`, `'IK'`, `'LOCKED_TRACK'`, `'SPLINE_IK'`, `'STRETCH_TO'`, `'TRACK_TO'`, `'ACTION'`, `'CHILD_OF'`, `'FLOOR'`, `'FOLLOW_PATH'`, `'PIVOT'`, `'RIGID_BODY_JOINT'`, `'SCRIPT'`, `'SHRINKWRAP'`], (*optional*)) – Type

- `COPY_LOCATION` Copy Location.
- `COPY_ROTATION` Copy Rotation.
- `COPY_SCALE` Copy Scale.
- `COPY_TRANSFORMS` Copy Transforms.
- `LIMIT_DISTANCE` Limit Distance.
- `LIMIT_LOCATION` Limit Location.
- `LIMIT_ROTATION` Limit Rotation.
- `LIMIT_SCALE` Limit Scale.
- `MAINTAIN_VOLUME` Maintain Volume.
- `TRANSFORM` Transformation.
- `CLAMP_TO` Clamp To.
- `DAMPED_TRACK` Damped Track, Tracking by taking the shortest path.
- `IK` Inverse Kinematics.
- `LOCKED_TRACK` Locked Track, Tracking along a single axis.
- `SPLINE_IK` Spline IK.
- `STRETCH_TO` Stretch To.
- `TRACK_TO` Track To, Legacy tracking constraint prone to twisting artifacts.
- `ACTION` Action.
- `CHILD_OF` Child Of.
- `FLOOR` Floor.
- `FOLLOW_PATH` Follow Path.
- `PIVOT` Pivot.
- `RIGID_BODY_JOINT` Rigid Body Joint.
- `SCRIPT` Script.
- `SHRINKWRAP` Shrinkwrap.

`bpy.ops.pose.constraints_clear()`  
Clear all the constraints for the selected bones

`bpy.ops.pose.constraints_copy()`  
Copy constraints to other selected bones

`bpy.ops.pose.copy()`  
Copies the current pose of the selected bones to copy/paste buffer

`bpy.ops.pose.flip_names()`  
Flips (and corrects) the axis suffixes of the the names of selected bones

`bpy.ops.pose.group_add()`  
Add a new bone group

`bpy.ops.pose.group_assign (type=0)`  
Add selected bones to the chosen bone group

**Parameters** `type` (*int in [0, 10], (optional)*) – Bone Group Index

`bpy.ops.pose.group_deselect ()`  
Deselect bones of active Bone Group

`bpy.ops.pose.group_move (direction='UP')`  
Change position of active Bone Group in list of Bone Groups

**Parameters** `direction` (*enum in ['UP', 'DOWN'], (optional)*) – Direction, Direction to move, UP or DOWN

`bpy.ops.pose.group_remove ()`  
Removes the active bone group

`bpy.ops.pose.group_select ()`  
Select bones in active Bone Group

`bpy.ops.pose.group_sort ()`  
Sort Bone Groups by their names in ascending order

`bpy.ops.pose.group_unassign ()`  
Remove selected bones from all bone groups

`bpy.ops.pose.hide (unselected=False)`  
Tag selected bones to not be visible in Pose Mode

**Parameters** `unselected` (*boolean, (optional)*) – Unselected

`bpy.ops.pose.ik_add (with_targets=True)`  
Add IK Constraint to the active Bone

**Parameters** `with_targets` (*boolean, (optional)*) – With Targets, Assign IK Constraint with targets derived from the select bones/objects

`bpy.ops.pose.ik_clear ()`  
Remove all IK Constraints from selected bones

`bpy.ops.pose.loc_clear ()`  
Reset locations of selected bones to their default values

`bpy.ops.pose.paste (flipped=False, selected_mask=False)`  
Paste the stored pose on to the current pose

**Parameters**

- **flipped** (*boolean, (optional)*) – Flipped on X-Axis, Paste the stored pose flipped on to current pose
- **selected\_mask** (*boolean, (optional)*) – On Selected Only, Only paste the stored pose on to selected bones in the current pose

`bpy.ops.pose.paths_calculate ()`  
Calculate paths for the selected bones

`bpy.ops.pose.paths_clear ()`  
Clear path caches for selected bones

`bpy.ops.pose.propagate (mode='WHILE_HELD', end_frame=250.0)`  
Copy selected aspects of the current pose to subsequent poses already keyframed

**Parameters**

- **mode** (*enum in ['WHILE\_HELD', 'NEXT\_KEY', 'LAST\_KEY', 'BEFORE\_FRAME', 'BEFORE\_END', 'SELECTED\_MARKERS'], (optional)*) – Terminate Mode, Method used to determine when to stop propagating pose to keyframes
  - `WHILE_HELD` While Held, Propagate pose to all keyframes after current frame that don't change (Default behaviour).
  - `NEXT_KEY` To Next Keyframe, Propagate pose to first keyframe following the current frame only.
  - `LAST_KEY` To Last Keyframe, Propagate pose to the last keyframe only (i.e. making action cyclic).
  - `BEFORE_FRAME` Before Frame, Propagate pose to all keyframes between current frame and 'Frame' property.
  - `BEFORE_END` Before Last Keyframe, Propagate pose to all keyframes from current frame until no more are found.
  - `SELECTED_MARKERS` On Selected Markers, Propagate pose to all keyframes occurring on frames with Scene Markers after the current frame.
- **end\_frame** (*float in [1.17549e-38, inf], (optional)*) – End Frame, Frame to stop propagating frames to (for 'Before Frame' mode)

`bpy.ops.pose.push` (*prev\_frame=0, next\_frame=0, percentage=0.5*)

Exaggerate the current pose

**Parameters**

- **prev\_frame** (*int in [-300000, 300000], (optional)*) – Previous Keyframe, Frame number of keyframe immediately before the current frame
- **next\_frame** (*int in [-300000, 300000], (optional)*) – Next Keyframe, Frame number of keyframe immediately after the current frame
- **percentage** (*float in [0, 1], (optional)*) – Percentage, Weighting factor for the sliding operation

`bpy.ops.pose.quaternions_flip` ()

Flip quaternion values to achieve desired rotations, while maintaining the same orientations

`bpy.ops.pose.relax` (*prev\_frame=0, next\_frame=0, percentage=0.5*)

Make the current pose more similar to its surrounding ones

**Parameters**

- **prev\_frame** (*int in [-300000, 300000], (optional)*) – Previous Keyframe, Frame number of keyframe immediately before the current frame
- **next\_frame** (*int in [-300000, 300000], (optional)*) – Next Keyframe, Frame number of keyframe immediately after the current frame
- **percentage** (*float in [0, 1], (optional)*) – Percentage, Weighting factor for the sliding operation

`bpy.ops.pose.reveal` ()

Unhide all bones that have been tagged to be hidden in Pose Mode

`bpy.ops.pose.rot_clear` ()

Reset rotations of selected bones to their default values

`bpy.ops.pose.rotation_mode_set` (*type*='QUATERNION')

Set the rotation representation used by selected bones

**Parameters** *type* (*enum in ['QUATERNION', 'XYZ', 'XZY', 'YXZ', 'YZX', 'ZXY', 'ZYX', 'AXIS\_ANGLE']*, (*optional*)) – Rotation Mode

- QUATERNION Quaternion (WXYZ), No Gimbal Lock (default).
- XYZ XYZ Euler, XYZ Rotation Order (prone to Gimbal Lock).
- XZY XZY Euler, XZY Rotation Order (prone to Gimbal Lock).
- YXZ YXZ Euler, YXZ Rotation Order (prone to Gimbal Lock).
- YZX YZX Euler, YZX Rotation Order (prone to Gimbal Lock).
- ZXY ZXY Euler, ZXY Rotation Order (prone to Gimbal Lock).
- ZYX ZYX Euler, ZYX Rotation Order (prone to Gimbal Lock).
- AXIS\_ANGLE Axis Angle, Axis Angle (W+XYZ), defines a rotation around some axis defined by 3D-Vector.

`bpy.ops.pose.scale_clear` ()

Reset scaling of selected bones to their default values

`bpy.ops.pose.select_all` (*action*='TOGGLE')

Toggle selection status of all bones

**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.pose.select_constraint_target` ()

Select bones used as targets for the currently selected bones

`bpy.ops.pose.select_flip_active` ()

Activate the bone with a flipped name

`bpy.ops.pose.select_grouped` (*extend=False*, *type='LAYER'*)

Select all visible bones grouped by similar properties

**Parameters**

- **extend** (*boolean*, (*optional*)) – Extend, Extend selection instead of deselecting everything first
- **type** (*enum in ['LAYER', 'GROUP', 'KEYINGSET']*, (*optional*)) – Type
  - LAYER Layer, Shared layers.
  - GROUP Group, Shared group.
  - KEYINGSET Keying Set, All bones affected by active Keying Set.

`bpy.ops.pose.select_hierarchy` (*direction='PARENT'*, *extend=False*)

Select immediate parent/children of selected bones

**Parameters**

- **direction** (*enum in ['PARENT', 'CHILD']*, (*optional*)) – Direction

- **extend** (*boolean, (optional)*) – Add to Selection

`bpy.ops.pose.select_inverse()`  
Flip the selection status of bones (selected -> unselected, unselected -> selected)

`bpy.ops.pose.select_linked(extend=False)`  
Select bones related to selected ones by parent/child relationships

**Parameters** **extend** (*boolean, (optional)*) – Extend, Extend selection instead of deselecting everything first

`bpy.ops.pose.select_parent()`  
Select bones that are parents of the currently selected bones

`bpy.ops.pose.transforms_clear()`  
Reset location, rotation, and scaling of selected bones to their default values

`bpy.ops.pose.user_transforms_clear()`  
Reset pose on selected bones to keyframed state

`bpy.ops.pose.visual_transform_apply()`  
Apply final constrained position of pose bones to their transform

## Poselib Operators

`bpy.ops.poselib.action_sanitise()`  
Make action suitable for use as a Pose Library

`bpy.ops.poselib.apply_pose(pose_index=-1)`  
Apply specified Pose Library pose to the rig

**Parameters** **pose\_index** (*int in [-2, inf], (optional)*) – Pose, Index of the pose to apply (-2 for no change to pose, -1 for poselib active pose)

`bpy.ops.poselib.browse_interactive(pose_index=-1)`  
Interactively browse poses in 3D-View

**Parameters** **pose\_index** (*int in [-2, inf], (optional)*) – Pose, Index of the pose to apply (-2 for no change to pose, -1 for poselib active pose)

`bpy.ops.poselib.new()`  
Add New Pose Library to active Object

`bpy.ops.poselib.pose_add(frame=1, name="Pose")`  
Add the current Pose to the active Pose Library

### Parameters

- **frame** (*int in [0, inf], (optional)*) – Frame, Frame to store pose on
- **name** (*string, (optional)*) – Pose Name, Name of newly added Pose

`bpy.ops.poselib.pose_remove(pose='DEFAULT')`  
Remove nth pose from the active Pose Library

**Parameters** **pose** (*enum in ['DEFAULT'], (optional)*) – Pose, The pose to remove

`bpy.ops.poselib.pose_rename(name="RenamedPose", pose='')`  
Rename specified pose from the active Pose Library

### Parameters

- **name** (*string, (optional)*) – New Pose Name, New name for pose

- **pose** (*enum in [], (optional)*) – Pose, The pose to rename

`bpy.ops.poselib.unlink()`  
Remove Pose Library from active Object

## Ptcache Operators

`bpy.ops.ptcache.add()`  
Add new cache

`bpy.ops.ptcache.bake (bake=False)`  
Bake physics

**Parameters** **bake** (*boolean, (optional)*) – Bake

`bpy.ops.ptcache.bake_all (bake=True)`  
Bake all physics

**Parameters** **bake** (*boolean, (optional)*) – Bake

`bpy.ops.ptcache.bake_from_cache()`  
Bake from cache

`bpy.ops.ptcache.free_bake()`  
Free physics bake

`bpy.ops.ptcache.free_bake_all()`  
Undocumented (*contribute*)

`bpy.ops.ptcache.remove()`  
Delete current cache

## Render Operators

`bpy.ops.render.opengl (animation=False, write_still=False, view_context=True)`  
OpenGL render active viewport

### Parameters

- **animation** (*boolean, (optional)*) – Animation, Render files from the animation range of this scene
- **write\_still** (*boolean, (optional)*) – Write Image, Save rendered the image to the output path (used only when animation is disabled)
- **view\_context** (*boolean, (optional)*) – View Context, Use the current 3D view for rendering, else use scene settings

`bpy.ops.render.play_rendered_anim()`  
Play back rendered frames/movies using an external player

**File** `startup/bl_operators/screen_play_rendered_anim.py:74`

`bpy.ops.render.preset_add (name="", remove_active=False)`  
Add a Render Preset

**Parameters** **name** (*string, (optional)*) – Name, Name of the preset, used to make the path name

**File** `startup/bl_operators/presets.py:50`

`bpy.ops.render.render (animation=False, write_still=False, layer="", scene="")`  
Render active scene

### Parameters

- **animation** (*boolean, (optional)*) – Animation, Render files from the animation range of this scene
- **write\_still** (*boolean, (optional)*) – Write Image, Save rendered the image to the output path (used only when animation is disabled)
- **layer** (*string, (optional)*) – Render Layer, Single render layer to re-render (used only when animation is disabled)
- **scene** (*string, (optional)*) – Scene, Scene to render, current scene if not specified

`bpy.ops.render.view_cancel()`  
Cancel show render view

`bpy.ops.render.view_show()`  
Toggle show render view

### Scene Operators

`bpy.ops.scene.delete()`  
Delete active scene

`bpy.ops.scene.new(type='NEW')`  
Add new scene by type

**Parameters** `type` (*enum in ['NEW', 'EMPTY', 'LINK\_OBJECTS', 'LINK\_OBJECT\_DATA', 'FULL\_COPY'], (optional)*) – Type

- `NEW` New, Add new scene.
- `EMPTY` Copy Settings, Make a copy without any objects.
- `LINK_OBJECTS` Link Objects, Link to the objects from the current scene.
- `LINK_OBJECT_DATA` Link Object Data, Copy objects linked to data from the current scene.
- `FULL_COPY` Full Copy, Make a full copy of the current scene.

`bpy.ops.scene.render_layer_add()`  
Add a render layer

`bpy.ops.scene.render_layer_remove()`  
Remove the selected render layer

### Screen Operators

`bpy.ops.screen.actionzone(modifier=0)`  
Handle area action zones for mouse actions/gestures

**Parameters** `modifier` (*int in [0, 2], (optional)*) – Modifier, Modifier state

`bpy.ops.screen.animation_cancel(restore_frame=True)`  
Cancel animation, returning to the original frame

**Parameters** `restore_frame` (*boolean, (optional)*) – Restore Frame, Restore the frame when animation was initialized

`bpy.ops.screen.animation_play(reverse=False, sync=False)`  
Play animation

**Parameters**

- **reverse** (*boolean, (optional)*) – Play in Reverse, Animation is played backwards
- **sync** (*boolean, (optional)*) – Sync, Drop frames to maintain framerate

`bpy.ops.screen.animation_step()`  
Step through animation by position

`bpy.ops.screen.area_dupli()`  
Duplicate selected area into new window

`bpy.ops.screen.area_join(min_x=-100, min_y=-100, max_x=-100, max_y=-100)`  
Join selected areas into new window

**Parameters**

- **min\_x** (*int in [-inf, inf], (optional)*) – X 1
- **min\_y** (*int in [-inf, inf], (optional)*) – Y 1
- **max\_x** (*int in [-inf, inf], (optional)*) – X 2
- **max\_y** (*int in [-inf, inf], (optional)*) – Y 2

`bpy.ops.screen.area_move(x=0, y=0, delta=0)`  
Move selected area edges

**Parameters**

- **x** (*int in [-inf, inf], (optional)*) – X
- **y** (*int in [-inf, inf], (optional)*) – Y
- **delta** (*int in [-inf, inf], (optional)*) – Delta

`bpy.ops.screen.area_options()`  
Operations for splitting and merging

`bpy.ops.screen.area_split(direction='HORIZONTAL', factor=0.5, mouse_x=-100, mouse_y=-100)`  
Split selected area into new windows

**Parameters**

- **direction** (*enum in ['HORIZONTAL', 'VERTICAL'], (optional)*) – Direction
- **factor** (*float in [0, 1], (optional)*) – Factor
- **mouse\_x** (*int in [-inf, inf], (optional)*) – Mouse X
- **mouse\_y** (*int in [-inf, inf], (optional)*) – Mouse Y

`bpy.ops.screen.area_swap()`  
Swap selected areas screen positions

`bpy.ops.screen.back_to_previous()`  
Revert back to the original screen layout, before fullscreen area overlay

`bpy.ops.screen.delete()`  
Delete active screen

`bpy.ops.screen.frame_jump(end=False)`  
Jump to first/last frame in frame range

**Parameters** **end** (*boolean, (optional)*) – Last Frame, Jump to the last frame of the frame range



`bpy.ops.screen.frame_offset` (*delta=0*)  
Undocumented ([contribute](#))

**Parameters** `delta` (*int in [-inf, inf], (optional)*) – Delta

`bpy.ops.screen.header_flip` ()  
Undocumented ([contribute](#))

`bpy.ops.screen.header_toolbox` ()  
Display header region toolbox

`bpy.ops.screen.keyframe_jump` (*next=True*)  
Jump to previous/next keyframe

**Parameters** `next` (*boolean, (optional)*) – Next Keyframe

`bpy.ops.screen.new` ()  
Add a new screen

`bpy.ops.screen.redo_last` ()  
Display menu for last action performed

`bpy.ops.screen.region_flip` ()  
Undocumented ([contribute](#))

`bpy.ops.screen.region_quadview` ()  
Split selected area into camera, front, right & top views

`bpy.ops.screen.region_scale` ()  
Scale selected area

`bpy.ops.screen.repeat_history` (*index=0*)  
Display menu for previous actions performed

**Parameters** `index` (*int in [0, inf], (optional)*) – Index

`bpy.ops.screen.repeat_last` ()  
Repeat last action

`bpy.ops.screen.screen_full_area` ()  
Toggle display selected area as fullscreen

`bpy.ops.screen.screen_set` (*delta=0*)  
Cycle through available screens

**Parameters** `delta` (*int in [-inf, inf], (optional)*) – Delta

`bpy.ops.screen.screencast` (*filepath=""*, *full=True*)  
Undocumented ([contribute](#))

**Parameters**

- `filepath` (*string, (optional)*) – filepath
- `full` (*boolean, (optional)*) – Full Screen

`bpy.ops.screen.screenshot` (*filepath=""*, *check\_existing=True*, *filter\_blender=False*, *filter\_image=True*, *filter\_movie=False*, *filter\_python=False*, *filter\_font=False*, *filter\_sound=False*, *filter\_text=False*, *filter\_btx=False*, *filter\_collada=False*, *filter\_folder=True*, *filemode=9*, *full=True*)

Undocumented ([contribute](#))

**Parameters**

- `filepath` (*string, (optional)*) – File Path, Path to file

- **check\_existing** (*boolean, (optional)*) – Check Existing, Check and warn on overwriting existing files
- **filter\_blender** (*boolean, (optional)*) – Filter .blend files
- **filter\_image** (*boolean, (optional)*) – Filter image files
- **filter\_movie** (*boolean, (optional)*) – Filter movie files
- **filter\_python** (*boolean, (optional)*) – Filter python files
- **filter\_font** (*boolean, (optional)*) – Filter font files
- **filter\_sound** (*boolean, (optional)*) – Filter sound files
- **filter\_text** (*boolean, (optional)*) – Filter text files
- **filter\_btx** (*boolean, (optional)*) – Filter btx files
- **filter\_collada** (*boolean, (optional)*) – Filter COLLADA files
- **filter\_folder** (*boolean, (optional)*) – Filter folders
- **filemode** (*int in [1, 9], (optional)*) – File Browser Mode, The setting for the file browser mode to load a .blend file, a library or a special file
- **full** (*boolean, (optional)*) – Full Screen

`bpy.ops.screen.spacedata_cleanup()`  
Remove unused settings for invisible editors

`bpy.ops.screen.userpref_show()`  
Show/hide user preferences

## Script Operators

`bpy.ops.script.execute_preset(filepath="", menu_idname="")`  
Execute a preset

### Parameters

- **filepath** (*string, (optional)*) – Path, Path of the Python file to execute
- **menu\_idname** (*string, (optional)*) – Menu ID Name, ID name of the menu this was called from

**File** `startup/bl_operators/presets.py:159`

`bpy.ops.script.python_file_run(filepath="")`  
Run Python file

**Parameters** **filepath** (*string, (optional)*) – Path

`bpy.ops.script.reload()`  
Reload Scripts

## Sculpt Operators

`bpy.ops.sculpt.brush_stroke(stroke=None, mode='NORMAL', ignore_background_click=False)`  
Undocumented ([contribute](#))

**Parameters**

- **stroke** (*bpy\_prop\_collection* of *OperatorStrokeElement*, (optional)) – Stroke
- **mode** (*enum* in [*'NORMAL'*, *'INVERT'*, *'SMOOTH'*], (optional)) – Sculpt Stroke Mode, Action taken when a sculpt stroke is made
  - *NORMAL* Normal, Apply brush normally.
  - *INVERT* Invert, Invert action of brush for duration of stroke.
  - *SMOOTH* Smooth, Switch brush to smooth mode for duration of stroke.
- **ignore\_background\_click** (*boolean*, (optional)) – Ignore Background Click, Clicks on the background do not start the stroke

`bpy.ops.sculpt.sculptmode_toggle()`  
Undocumented ([contribute](#))

`bpy.ops.sculpt.set_persistent_base()`  
Undocumented ([contribute](#))

## Sequencer Operators

`bpy.ops.sequencer.change_effect_input` (*swap*='A\_B')  
Undocumented ([contribute](#))

**Parameters** *swap* (*enum* in [*'A\_B'*, *'B\_C'*, *'A\_C'*], (optional)) – Swap, The effect inputs to swap

`bpy.ops.sequencer.change_effect_type` (*type*='CROSS')  
Undocumented ([contribute](#))

**Parameters** *type* (*enum* in [*'CROSS'*, *'ADD'*, *'SUBTRACT'*, *'ALPHA\_OVER'*, *'ALPHA\_UNDER'*, *'GAMMA\_CROSS'*, *'MULTIPLY'*, *'OVER\_DROP'*, *'PLUGIN'*, *'WIPE'*, *'GLOW'*, *'TRANSFORM'*, *'COLOR'*, *'SPEED'*, *'MULTICAM'*, *'ADJUSTMENT'*], (optional)) – Type, Sequencer effect type

- *CROSS* Crossfade, Crossfade effect strip type.
- *ADD* Add, Add effect strip type.
- *SUBTRACT* Subtract, Subtract effect strip type.
- *ALPHA\_OVER* Alpha Over, Alpha Over effect strip type.
- *ALPHA\_UNDER* Alpha Under, Alpha Under effect strip type.
- *GAMMA\_CROSS* Gamma Cross, Gamma Cross effect strip type.
- *MULTIPLY* Multiply, Multiply effect strip type.
- *OVER\_DROP* Alpha Over Drop, Alpha Over Drop effect strip type.
- *PLUGIN* Plugin, Plugin effect strip type.
- *WIPE* Wipe, Wipe effect strip type.
- *GLOW* Glow, Glow effect strip type.
- *TRANSFORM* Transform, Transform effect strip type.
- *COLOR* Color, Color effect strip type.
- *SPEED* Speed, Color effect strip type.
- *MULTICAM* Multicam Selector.

- ADJUSTMENT Adjustment Layer.

`bpy.ops.sequencer.change_path` (*filepath=""*, *directory=""*, *files=None*, *filter\_blender=False*, *filter\_image=True*, *filter\_movie=True*, *filter\_python=False*, *filter\_font=False*, *filter\_sound=False*, *filter\_text=False*, *filter\_btx=False*, *filter\_collada=False*, *filter\_folder=True*, *filemode=9*, *relative\_path=True*)

Undocumented ([contribute](#))

#### Parameters

- **filepath** (*string, (optional)*) – File Path, Path to file
- **directory** (*string, (optional)*) – Directory, Directory of the file
- **files** (*bpy\_prop\_collection of OperatorFileListElement, (optional)*) – Files
- **filter\_blender** (*boolean, (optional)*) – Filter .blend files
- **filter\_image** (*boolean, (optional)*) – Filter image files
- **filter\_movie** (*boolean, (optional)*) – Filter movie files
- **filter\_python** (*boolean, (optional)*) – Filter python files
- **filter\_font** (*boolean, (optional)*) – Filter font files
- **filter\_sound** (*boolean, (optional)*) – Filter sound files
- **filter\_text** (*boolean, (optional)*) – Filter text files
- **filter\_btx** (*boolean, (optional)*) – Filter btx files
- **filter\_collada** (*boolean, (optional)*) – Filter COLLADA files
- **filter\_folder** (*boolean, (optional)*) – Filter folders
- **filemode** (*int in [1, 9], (optional)*) – File Browser Mode, The setting for the file browser mode to load a .blend file, a library or a special file
- **relative\_path** (*boolean, (optional)*) – Relative Path, Select the file relative to the blend file

`bpy.ops.sequencer.copy` ()

Undocumented ([contribute](#))

`bpy.ops.sequencer.crossfade_sounds` ()

Do crossfading volume animation of two selected sound strips

**File** `startup/bl_operators/sequencer.py:41`

`bpy.ops.sequencer.cut` (*frame=0*, *type='SOFT'*, *side='BOTH'*)

Cut the selected strips

#### Parameters

- **frame** (*int in [-inf, inf], (optional)*) – Frame, Frame where selected strips will be cut
- **type** (*enum in ['SOFT', 'HARD'], (optional)*) – Type, The type of cut operation to perform on strips
- **side** (*enum in ['LEFT', 'RIGHT', 'BOTH'], (optional)*) – Side, The side that remains selected after cutting

`bpy.ops.sequencer.cut_multicam` (*camera=1*)

Cut multicam strip and select camera

**Parameters** **camera** (*int in [1, 32], (optional)*) – Camera

**File** `startup/bl_operators/sequencer.py:99`

`bpy.ops.sequencer.deinterlace_selected_movies()`  
Deinterlace all selected movie sources

**File** `startup/bl_operators/sequencer.py:134`

`bpy.ops.sequencer.delete()`  
Erase selected strips from the sequencer

`bpy.ops.sequencer.duplicate(mode='TRANSLATION')`  
Duplicate the selected strips

**Parameters** `mode` (*enum in ['INIT', 'DUMMY', 'TRANSLATION', 'ROTATION', 'RESIZE', 'TOSPHERE', 'SHEAR', 'WARP', 'SHRINKFATTEN', 'TILT', 'TRACKBALL', 'PUSHPULL', 'CREASE', 'MIRROR', 'BONE\_SIZE', 'BONE\_ENVELOPE', 'CURVE\_SHRINKFATTEN', 'BONE\_ROLL', 'TIME\_TRANSLATE', 'TIME\_SLIDE', 'TIME\_SCALE', 'TIME\_EXTEND', 'BAKE\_TIME', 'BEVEL', 'BWEIGHT', 'ALIGN', 'EDGESLIDE', 'SEQSLIDE']*, (optional)) – Mode

`bpy.ops.sequencer.effect_strip_add(filepath="", filter_blender=False, filter_image=False, filter_movie=False, filter_python=False, filter_font=False, filter_sound=False, filter_text=False, filter_btx=False, filter_collada=False, filter_folder=False, filemode=9, relative_path=True, frame_start=0, frame_end=0, channel=1, replace_sel=True, overlap=False, type='CROSS', color=(0.0, 0.0, 0.0))`

Add an effect to the sequencer, most are applied on top of existing strips

#### Parameters

- **filepath** (*string, (optional)*) – File Path, Path to file
- **filter\_blender** (*boolean, (optional)*) – Filter .blend files
- **filter\_image** (*boolean, (optional)*) – Filter image files
- **filter\_movie** (*boolean, (optional)*) – Filter movie files
- **filter\_python** (*boolean, (optional)*) – Filter python files
- **filter\_font** (*boolean, (optional)*) – Filter font files
- **filter\_sound** (*boolean, (optional)*) – Filter sound files
- **filter\_text** (*boolean, (optional)*) – Filter text files
- **filter\_btx** (*boolean, (optional)*) – Filter btx files
- **filter\_collada** (*boolean, (optional)*) – Filter COLLADA files
- **filter\_folder** (*boolean, (optional)*) – Filter folders
- **filemode** (*int in [1, 9], (optional)*) – File Browser Mode, The setting for the file browser mode to load a .blend file, a library or a special file
- **relative\_path** (*boolean, (optional)*) – Relative Path, Select the file relative to the blend file
- **frame\_start** (*int in [-inf, inf], (optional)*) – Start Frame, Start frame of the sequence strip
- **frame\_end** (*int in [-inf, inf], (optional)*) – End Frame, End frame for the color strip
- **channel** (*int in [1, 32], (optional)*) – Channel, Channel to place this strip into
- **replace\_sel** (*boolean, (optional)*) – Replace Selection, replace the current selection

- **overlap** (*boolean, (optional)*) – Allow Overlap, Don't correct overlap on new sequence strips
- **type** (*enum in ['CROSS', 'ADD', 'SUBTRACT', 'ALPHA\_OVER', 'ALPHA\_UNDER', 'GAMMA\_CROSS', 'MULTIPLY', 'OVER\_DROP', 'PLUGIN', 'WIPE', 'GLOW', 'TRANSFORM', 'COLOR', 'SPEED', 'MULTICAM', 'ADJUSTMENT'], (optional)*) – Type, Sequencer effect type
  - CROSS Crossfade, Crossfade effect strip type.
  - ADD Add, Add effect strip type.
  - SUBTRACT Subtract, Subtract effect strip type.
  - ALPHA\_OVER Alpha Over, Alpha Over effect strip type.
  - ALPHA\_UNDER Alpha Under, Alpha Under effect strip type.
  - GAMMA\_CROSS Gamma Cross, Gamma Cross effect strip type.
  - MULTIPLY Multiply, Multiply effect strip type.
  - OVER\_DROP Alpha Over Drop, Alpha Over Drop effect strip type.
  - PLUGIN Plugin, Plugin effect strip type.
  - WIPE Wipe, Wipe effect strip type.
  - GLOW Glow, Glow effect strip type.
  - TRANSFORM Transform, Transform effect strip type.
  - COLOR Color, Color effect strip type.
  - SPEED Speed, Color effect strip type.
  - MULTICAM Multicam Selector.
  - ADJUSTMENT Adjustment Layer.
- **color** (*float array of 3 items in [0, 1], (optional)*) – Color, Initialize the strip with this color (only used when type='COLOR')

```
bpy.ops.sequencer.image_strip_add(directory="", files=None, filter_blender=False, filter_image=True, filter_movie=False, filter_python=False, filter_font=False, filter_sound=False, filter_text=False, filter_btx=False, filter_collada=False, filter_folder=True, filemode=9, relative_path=True, frame_start=0, frame_end=0, channel=1, replace_sel=True, overlap=False)
```

Add an image or image sequence to the sequencer

#### Parameters

- **directory** (*string, (optional)*) – Directory, Directory of the file
- **files** (*bpy\_prop\_collection of OperatorFileListElement, (optional)*) – Files
- **filter\_blender** (*boolean, (optional)*) – Filter .blend files
- **filter\_image** (*boolean, (optional)*) – Filter image files
- **filter\_movie** (*boolean, (optional)*) – Filter movie files
- **filter\_python** (*boolean, (optional)*) – Filter python files
- **filter\_font** (*boolean, (optional)*) – Filter font files
- **filter\_sound** (*boolean, (optional)*) – Filter sound files

- **filter\_text** (*boolean, (optional)*) – Filter text files
- **filter\_btx** (*boolean, (optional)*) – Filter btx files
- **filter\_collada** (*boolean, (optional)*) – Filter COLLADA files
- **filter\_folder** (*boolean, (optional)*) – Filter folders
- **filemode** (*int in [1, 9], (optional)*) – File Browser Mode, The setting for the file browser mode to load a .blend file, a library or a special file
- **relative\_path** (*boolean, (optional)*) – Relative Path, Select the file relative to the blend file
- **frame\_start** (*int in [-inf, inf], (optional)*) – Start Frame, Start frame of the sequence strip
- **frame\_end** (*int in [-inf, inf], (optional)*) – End Frame, End frame for the color strip
- **channel** (*int in [1, 32], (optional)*) – Channel, Channel to place this strip into
- **replace\_sel** (*boolean, (optional)*) – Replace Selection, replace the current selection
- **overlap** (*boolean, (optional)*) – Allow Overlap, Don't correct overlap on new sequence strips

`bpy.ops.sequencer.images_separate` (*length=1*)

On image sequence strips, it returns a strip for each image

**Parameters** **length** (*int in [1, 1000], (optional)*) – Length, Length of each frame

`bpy.ops.sequencer.lock` ()

Lock the active strip so that it can't be transformed

`bpy.ops.sequencer.meta_make` ()

Group selected strips into a metastrip

`bpy.ops.sequencer.meta_separate` ()

Put the contents of a metastrip back in the sequencer

`bpy.ops.sequencer.meta_toggle` ()

Toggle a metastrip (to edit enclosed strips)

`bpy.ops.sequencer.movie_strip_add` (*filepath=""*, *files=None*, *filter\_blender=False*, *filter\_image=False*, *filter\_movie=True*, *filter\_python=False*, *filter\_font=False*, *filter\_sound=False*, *filter\_text=False*, *filter\_btx=False*, *filter\_collada=False*, *filter\_folder=True*, *filemode=9*, *relative\_path=True*, *frame\_start=0*, *channel=1*, *replace\_sel=True*, *overlap=False*, *sound=True*)

Add a movie strip to the sequencer

#### Parameters

- **filepath** (*string, (optional)*) – File Path, Path to file
- **files** (*bpy\_prop\_collection of OperatorFileListElement, (optional)*) – Files
- **filter\_blender** (*boolean, (optional)*) – Filter .blend files
- **filter\_image** (*boolean, (optional)*) – Filter image files
- **filter\_movie** (*boolean, (optional)*) – Filter movie files
- **filter\_python** (*boolean, (optional)*) – Filter python files
- **filter\_font** (*boolean, (optional)*) – Filter font files
- **filter\_sound** (*boolean, (optional)*) – Filter sound files
- **filter\_text** (*boolean, (optional)*) – Filter text files

- **filter\_btx** (*boolean, (optional)*) – Filter btx files
- **filter\_collada** (*boolean, (optional)*) – Filter COLLADA files
- **filter\_folder** (*boolean, (optional)*) – Filter folders
- **filemode** (*int in [1, 9], (optional)*) – File Browser Mode, The setting for the file browser mode to load a .blend file, a library or a special file
- **relative\_path** (*boolean, (optional)*) – Relative Path, Select the file relative to the blend file
- **frame\_start** (*int in [-inf, inf], (optional)*) – Start Frame, Start frame of the sequence strip
- **channel** (*int in [1, 32], (optional)*) – Channel, Channel to place this strip into
- **replace\_sel** (*boolean, (optional)*) – Replace Selection, replace the current selection
- **overlap** (*boolean, (optional)*) – Allow Overlap, Don't correct overlap on new sequence strips
- **sound** (*boolean, (optional)*) – Sound, Load sound with the movie

`bpy.ops.sequencer.mute` (*unselected=False*)  
Mute selected strips

**Parameters** `unselected` (*boolean, (optional)*) – Unselected, Mute unselected rather than selected strips

`bpy.ops.sequencer.next_edit` ()  
Move frame to next edit point

`bpy.ops.sequencer.offset_clear` ()  
Clear strip offsets from the start and end frames

`bpy.ops.sequencer.paste` ()  
Undocumented ([contribute](#))

`bpy.ops.sequencer.previous_edit` ()  
Move frame to previous edit point

`bpy.ops.sequencer.properties` ()  
Open sequencer properties panel

`bpy.ops.sequencer.reassign_inputs` ()  
Reassign the inputs for the effect strip

`bpy.ops.sequencer.rebuild_proxy` ()  
Rebuild all selected proxies and timecode indices using the job system

`bpy.ops.sequencer.refresh_all` ()  
Refresh the sequencer editor

`bpy.ops.sequencer.reload` ()  
Reload strips in the sequencer

`bpy.ops.sequencer.rendersize` ()  
Set render size and aspect from active sequence

`bpy.ops.sequencer.scene_strip_add` (*frame\_start=0, channel=1, replace\_sel=True, overlap=False, scene=''*)  
Add a strip to the sequencer using a blender scene as a source

**Parameters**

- **frame\_start** (*int in [-inf, inf], (optional)*) – Start Frame, Start frame of the sequence strip
- **channel** (*int in [1, 32], (optional)*) – Channel, Channel to place this strip into



- **replace\_sel** (*boolean, (optional)*) – Replace Selection, replace the current selection
- **overlap** (*boolean, (optional)*) – Allow Overlap, Don't correct overlap on new sequence strips
- **scene** (*enum in [], (optional)*) – Scene

`bpy.ops.sequencer.select` (*extend=False, linked\_handle=False, left\_right=False, linked\_time=False*)  
Select a strip (last selected becomes the “active strip”)

#### Parameters

- **extend** (*boolean, (optional)*) – Extend, Extend the selection
- **linked\_handle** (*boolean, (optional)*) – Linked Handle, Select handles next to the active strip
- **left\_right** (*boolean, (optional)*) – Left/Right, Select based on the current frame side the cursor is on
- **linked\_time** (*boolean, (optional)*) – Linked Time, Select other strips at the same time

`bpy.ops.sequencer.select_active_side` (*side='BOTH'*)  
Select strips on the nominated side of the active strip

**Parameters** *side* (*enum in ['LEFT', 'RIGHT', 'BOTH'], (optional)*) – Side, The side of the handle that is selected

`bpy.ops.sequencer.select_all_toggle` ()  
Select or deselect all strips

`bpy.ops.sequencer.select_border` (*gesture\_mode=0, xmin=0, xmax=0, ymin=0, ymax=0*)  
Enable border select mode

#### Parameters

- **gesture\_mode** (*int in [-inf, inf], (optional)*) – Gesture Mode
- **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

`bpy.ops.sequencer.select_grouped` (*extend=False, type='TYPE'*)  
Select all strips grouped by various properties

#### Parameters

- **extend** (*boolean, (optional)*) – Extend, Extend selection instead of deselecting everything first
- **type** (*enum in ['TYPE', 'TYPE\_BASIC', 'TYPE\_EFFECT', 'DATA', 'EFFECT', 'EFFECT\_LINK', 'OVERLAP'], (optional)*) – Type
  - `TYPE` Type, Shared strip type.
  - `TYPE_BASIC` Global Type, All strips of same basic type (Graphical or Sound).
  - `TYPE_EFFECT` Effect Type, Shared strip effect type (if active strip is not an effect one, select all non-effect strips).
  - `DATA` Data, Shared data (scene, image, sound, etc.).
  - `EFFECT` Effect, Shared effects.

- `EFFECT_LINK` Effect/Linked, Other strips affected by the active one (sharing some time, and below or effect-assigned).
- `OVERLAP` Overlap, Overlapping time.

`bpy.ops.sequencer.select_handles` (*side='BOTH'*)  
Select manipulator handles on the sides of the selected strip

**Parameters** `side` (*enum in ['LEFT', 'RIGHT', 'BOTH'], (optional)*) – Side, The side of the handle that is selected

`bpy.ops.sequencer.select_inverse` ()  
Select unselected strips

`bpy.ops.sequencer.select_less` ()  
Shrink the current selection of adjacent selected strips

`bpy.ops.sequencer.select_linked` ()  
Select all strips adjacent to the current selection

`bpy.ops.sequencer.select_linked_pick` (*extend=False*)  
Select a chain of linked strips nearest to the mouse pointer

**Parameters** `extend` (*boolean, (optional)*) – Extend, extend the selection

`bpy.ops.sequencer.select_more` ()  
Select more strips adjacent to the current selection

`bpy.ops.sequencer.snap` (*frame=0*)  
Frame where selected strips will be snapped

**Parameters** `frame` (*int in [-inf, inf], (optional)*) – Frame, Frame where selected strips will be snapped

`bpy.ops.sequencer.sound_strip_add` (*filepath=""*, *files=None*, *filter\_blender=False*, *filter\_image=False*, *filter\_movie=False*, *filter\_python=False*, *filter\_font=False*, *filter\_sound=True*, *filter\_text=False*, *filter\_btx=False*, *filter\_collada=False*, *filter\_folder=True*, *filemode=9*, *relative\_path=True*, *frame\_start=0*, *channel=1*, *replace\_sel=True*, *overlap=False*, *cache=False*)

Add a sound strip to the sequencer

**Parameters**

- `filepath` (*string, (optional)*) – File Path, Path to file
- `files` (*bpy\_prop\_collection of OperatorFileListElement, (optional)*) – Files
- `filter_blender` (*boolean, (optional)*) – Filter .blend files
- `filter_image` (*boolean, (optional)*) – Filter image files
- `filter_movie` (*boolean, (optional)*) – Filter movie files
- `filter_python` (*boolean, (optional)*) – Filter python files
- `filter_font` (*boolean, (optional)*) – Filter font files
- `filter_sound` (*boolean, (optional)*) – Filter sound files
- `filter_text` (*boolean, (optional)*) – Filter text files
- `filter_btx` (*boolean, (optional)*) – Filter btx files
- `filter_collada` (*boolean, (optional)*) – Filter COLLADA files
- `filter_folder` (*boolean, (optional)*) – Filter folders

- **filemode** (*int in [1, 9], (optional)*) – File Browser Mode, The setting for the file browser mode to load a .blend file, a library or a special file
- **relative\_path** (*boolean, (optional)*) – Relative Path, Select the file relative to the blend file
- **frame\_start** (*int in [-inf, inf], (optional)*) – Start Frame, Start frame of the sequence strip
- **channel** (*int in [1, 32], (optional)*) – Channel, Channel to place this strip into
- **replace\_sel** (*boolean, (optional)*) – Replace Selection, replace the current selection
- **overlap** (*boolean, (optional)*) – Allow Overlap, Don't correct overlap on new sequence strips
- **cache** (*boolean, (optional)*) – Cache, Cache the sound in memory

`bpy.ops.sequencer.swap` (*side='RIGHT'*)  
Swap active strip with strip to the right or left

**Parameters** *side* (*enum in ['LEFT', 'RIGHT'], (optional)*) – Side, Side of the strip to swap

`bpy.ops.sequencer.swap_data` ()  
Swap 2 sequencer strips

`bpy.ops.sequencer.swap_inputs` ()  
Swap the first two inputs for the effect strip

`bpy.ops.sequencer.unlock` ()  
Unlock the active strip so that it can't be transformed

`bpy.ops.sequencer.unmute` (*unselected=False*)  
Un-Mute unselected rather than selected strips

**Parameters** *unselected* (*boolean, (optional)*) – Unselected, UnMute unselected rather than selected strips

`bpy.ops.sequencer.view_all` ()  
View all the strips in the sequencer

`bpy.ops.sequencer.view_all_preview` ()  
Zoom preview to fit in the area

`bpy.ops.sequencer.view_ghost_border` (*gesture\_mode=0, xmin=0, xmax=0, ymin=0, ymax=0*)  
Enable border select mode

#### Parameters

- **gesture\_mode** (*int in [-inf, inf], (optional)*) – Gesture Mode
- **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

`bpy.ops.sequencer.view_selected` ()  
Zoom the sequencer on the selected strips

`bpy.ops.sequencer.view_toggle` ()  
Toggle between sequencer views (sequence, preview, both)

`bpy.ops.sequencer.view_zoom_ratio` (*ratio=1.0*)  
Change zoom ratio of sequencer preview

**Parameters** **ratio** (*float in [0, inf], (optional)*) – Ratio, Zoom ratio, 1.0 is 1:1, higher is zoomed in, lower is zoomed out

## Sketch Operators

`bpy.ops.sketch.cancel_stroke()`  
Undocumented (contribute)

`bpy.ops.sketch.convert()`  
Undocumented (contribute)

`bpy.ops.sketch.delete()`  
Undocumented (contribute)

`bpy.ops.sketch.draw_preview(snap=False)`  
Undocumented (contribute)

**Parameters** **snap** (*boolean, (optional)*) – Snap

`bpy.ops.sketch.draw_stroke(snap=False)`  
Undocumented (contribute)

**Parameters** **snap** (*boolean, (optional)*) – Snap

`bpy.ops.sketch.finish_stroke()`  
Undocumented (contribute)

`bpy.ops.sketch.gesture(snap=False)`  
Undocumented (contribute)

**Parameters** **snap** (*boolean, (optional)*) – Snap

`bpy.ops.sketch.select()`  
Undocumented (contribute)

## Sound Operators

`bpy.ops.sound.bake_animation()`  
Updates the audio animation cache so that it's up to date

`bpy.ops.sound.mixdown(filepath="", check_existing=True, filter_blender=False, filter_image=False, filter_movie=False, filter_python=False, filter_font=False, filter_sound=True, filter_text=False, filter_btx=False, filter_collada=False, filter_folder=True, filemode=9, accuracy=1024, container='FLAC', codec='FLAC', format='S16', bitrate=192)`

Mixes the scene's audio to a sound file

### Parameters

- **filepath** (*string, (optional)*) – File Path, Path to file
- **check\_existing** (*boolean, (optional)*) – Check Existing, Check and warn on overwriting existing files
- **filter\_blender** (*boolean, (optional)*) – Filter .blend files
- **filter\_image** (*boolean, (optional)*) – Filter image files
- **filter\_movie** (*boolean, (optional)*) – Filter movie files
- **filter\_python** (*boolean, (optional)*) – Filter python files

- **filter\_font** (*boolean, (optional)*) – Filter font files
- **filter\_sound** (*boolean, (optional)*) – Filter sound files
- **filter\_text** (*boolean, (optional)*) – Filter text files
- **filter\_btx** (*boolean, (optional)*) – Filter btx files
- **filter\_collada** (*boolean, (optional)*) – Filter COLLADA files
- **filter\_folder** (*boolean, (optional)*) – Filter folders
- **filemode** (*int in [1, 9], (optional)*) – File Browser Mode, The setting for the file browser mode to load a .blend file, a library or a special file
- **accuracy** (*int in [1, inf], (optional)*) – Accuracy, Sample accuracy, important for animation data (the lower the value, the more accurate)
- **container** (*enum in ['FLAC', 'OGG', 'WAV'], (optional)*) – Container, File format
  - `FLAC flac`, Free Lossless Audio Codec.
  - `OGG ogg`, Xiph.Org Ogg Container.
  - `WAV wav`, Waveform Audio File Format.
- **codec** (*enum in ['FLAC', 'PCM', 'VORBIS'], (optional)*) – Codec, Audio Codec
  - `FLAC FLAC`, Free Lossless Audio Codec.
  - `PCM PCM`, Pulse Code Modulation (RAW).
  - `VORBIS Vorbis`, Xiph.Org Vorbis Codec.
- **format** (*enum in ['U8', 'S16', 'S24', 'S32', 'F32', 'F64'], (optional)*) – Format, Sample format
  - `U8 U8`, 8 bit unsigned.
  - `S16 S16`, 16 bit signed.
  - `S24 S24`, 24 bit signed.
  - `S32 S32`, 32 bit signed.
  - `F32 F32`, 32 bit floating point.
  - `F64 F64`, 64 bit floating point.
- **bitrate** (*int in [32, 512], (optional)*) – Bitrate, Bitrate in kbit/s

```
bpy.ops.sound.open (filepath="", filter_blender=False, filter_image=False, filter_movie=True,
                    filter_python=False, filter_font=False, filter_sound=True, filter_text=False,
                    filter_btx=False, filter_collada=False, filter_folder=True, filemode=9, rela-
                    tive_path=True, cache=False, mono=False)
```

Load a sound file

#### Parameters

- **filepath** (*string, (optional)*) – File Path, Path to file
- **filter\_blender** (*boolean, (optional)*) – Filter .blend files
- **filter\_image** (*boolean, (optional)*) – Filter image files
- **filter\_movie** (*boolean, (optional)*) – Filter movie files
- **filter\_python** (*boolean, (optional)*) – Filter python files
- **filter\_font** (*boolean, (optional)*) – Filter font files

- **filter\_sound** (*boolean, (optional)*) – Filter sound files
- **filter\_text** (*boolean, (optional)*) – Filter text files
- **filter\_btx** (*boolean, (optional)*) – Filter btx files
- **filter\_collada** (*boolean, (optional)*) – Filter COLLADA files
- **filter\_folder** (*boolean, (optional)*) – Filter folders
- **filemode** (*int in [1, 9], (optional)*) – File Browser Mode, The setting for the file browser mode to load a .blend file, a library or a special file
- **relative\_path** (*boolean, (optional)*) – Relative Path, Select the file relative to the blend file
- **cache** (*boolean, (optional)*) – Cache, Cache the sound in memory
- **mono** (*boolean, (optional)*) – Mono, Mixdown the sound to mono

```
bpy.ops.sound.open_mono(filepath="", filter_blender=False, filter_image=False, filter_movie=True,
                        filter_python=False, filter_font=False, filter_sound=True, filter_text=False,
                        filter_btx=False, filter_collada=False, filter_folder=True, filemode=9,
                        relative_path=True, cache=False, mono=True)
```

Load a sound file as mono

#### Parameters

- **filepath** (*string, (optional)*) – File Path, Path to file
- **filter\_blender** (*boolean, (optional)*) – Filter .blend files
- **filter\_image** (*boolean, (optional)*) – Filter image files
- **filter\_movie** (*boolean, (optional)*) – Filter movie files
- **filter\_python** (*boolean, (optional)*) – Filter python files
- **filter\_font** (*boolean, (optional)*) – Filter font files
- **filter\_sound** (*boolean, (optional)*) – Filter sound files
- **filter\_text** (*boolean, (optional)*) – Filter text files
- **filter\_btx** (*boolean, (optional)*) – Filter btx files
- **filter\_collada** (*boolean, (optional)*) – Filter COLLADA files
- **filter\_folder** (*boolean, (optional)*) – Filter folders
- **filemode** (*int in [1, 9], (optional)*) – File Browser Mode, The setting for the file browser mode to load a .blend file, a library or a special file
- **relative\_path** (*boolean, (optional)*) – Relative Path, Select the file relative to the blend file
- **cache** (*boolean, (optional)*) – Cache, Cache the sound in memory
- **mono** (*boolean, (optional)*) – Mono, Mixdown the sound to mono

```
bpy.ops.sound.pack()
```

Pack the sound into the current blend file

```
bpy.ops.sound.unpack(method='USE_LOCAL', id="")
```

Unpack the sound to the samples filename

#### Parameters

- **method** (*enum in ['USE\_LOCAL', 'WRITE\_LOCAL', 'USE\_ORIGINAL', 'WRITE\_ORIGINAL'], (optional)*) – Method, How to unpack

- **id** (*string, (optional)*) – Sound Name, Sound datablock name to unpack

```
bpy.ops.sound.update_animation_flags ()
    Update animation flags
```

## Surface Operators

```
bpy.ops.surface.primitive_nurbs_surface_circle_add (view_align=False,          enter_
                                                    editmode=False,          loca-
                                                    tion=(0.0, 0.0, 0.0), rotation=(0.0,
                                                    0.0, 0.0), layers=(False, False,
                                                    False, False, False, False, False,
                                                    False, False, False, False, False,
                                                    False, False, False, False, False,
                                                    False, False, False))
```

Construct a Nurbs surface Circle

### Parameters

- **view\_align** (*boolean, (optional)*) – Align to View, Align the new object to the view
- **enter\_editmode** (*boolean, (optional)*) – Enter Editmode, Enter editmode when adding this object
- **location** (*float array of 3 items in [-inf, inf], (optional)*) – Location, Location for the newly added object
- **rotation** (*float array of 3 items in [-inf, inf], (optional)*) – Rotation, Rotation for the newly added object
- **layers** (*boolean array of 20 items, (optional)*) – Layer

```
bpy.ops.surface.primitive_nurbs_surface_curve_add (view_align=False,          enter_
                                                    editmode=False, location=(0.0,
                                                    0.0, 0.0), rotation=(0.0, 0.0, 0.0),
                                                    layers=(False, False, False, False,
                                                    False, False, False, False, False,
                                                    False, False, False, False, False,
                                                    False, False, False, False, False,
                                                    False))
```

Construct a Nurbs surface Curve

### Parameters

- **view\_align** (*boolean, (optional)*) – Align to View, Align the new object to the view
- **enter\_editmode** (*boolean, (optional)*) – Enter Editmode, Enter editmode when adding this object
- **location** (*float array of 3 items in [-inf, inf], (optional)*) – Location, Location for the newly added object
- **rotation** (*float array of 3 items in [-inf, inf], (optional)*) – Rotation, Rotation for the newly added object
- **layers** (*boolean array of 20 items, (optional)*) – Layer

```
bpy.ops.surface.primitive_nurbs_surface_cylinder_add (view_align=False,          enter_
                                                    editmode=False,          loca-
                                                    tion=(0.0, 0.0, 0.0), ro-
                                                    tation=(0.0, 0.0, 0.0), lay-
                                                    ers=(False, False, False, False,
                                                    False, False, False, False,
                                                    False, False, False, False,
                                                    False, False, False, False,
                                                    False, False, False, False))
```

Construct a Nurbs surface Cylinder

#### Parameters

- **view\_align** (*boolean, (optional)*) – Align to View, Align the new object to the view
- **enter\_editmode** (*boolean, (optional)*) – Enter Editmode, Enter editmode when adding this object
- **location** (*float array of 3 items in [-inf, inf], (optional)*) – Location, Location for the newly added object
- **rotation** (*float array of 3 items in [-inf, inf], (optional)*) – Rotation, Rotation for the newly added object
- **layers** (*boolean array of 20 items, (optional)*) – Layer

```
bpy.ops.surface.primitive_nurbs_surface_sphere_add (view_align=False,          enter_
                                                    editmode=False,          loca-
                                                    tion=(0.0, 0.0, 0.0), ro-
                                                    tation=(0.0, 0.0, 0.0), lay-
                                                    ers=(False, False, False, False, False,
                                                    False, False, False, False, False,
                                                    False, False, False, False, False,
                                                    False, False, False, False, False,
                                                    False, False, False))
```

Construct a Nurbs surface Sphere

#### Parameters

- **view\_align** (*boolean, (optional)*) – Align to View, Align the new object to the view
- **enter\_editmode** (*boolean, (optional)*) – Enter Editmode, Enter editmode when adding this object
- **location** (*float array of 3 items in [-inf, inf], (optional)*) – Location, Location for the newly added object
- **rotation** (*float array of 3 items in [-inf, inf], (optional)*) – Rotation, Rotation for the newly added object
- **layers** (*boolean array of 20 items, (optional)*) – Layer

```
bpy.ops.surface.primitive_nurbs_surface_surface_add (view_align=False,          enter_
                                                    editmode=False,          loca-
                                                    tion=(0.0, 0.0, 0.0), ro-
                                                    tation=(0.0, 0.0, 0.0), lay-
                                                    ers=(False, False, False, False, False,
                                                    False, False, False, False, False,
                                                    False, False, False, False, False,
                                                    False, False, False, False, False,
                                                    False, False, False))
```

Construct a Nurbs surface Patch



**Parameters**

- **view\_align** (*boolean, (optional)*) – Align to View, Align the new object to the view
- **enter\_editmode** (*boolean, (optional)*) – Enter Editmode, Enter editmode when adding this object
- **location** (*float array of 3 items in [-inf, inf], (optional)*) – Location, Location for the newly added object
- **rotation** (*float array of 3 items in [-inf, inf], (optional)*) – Rotation, Rotation for the newly added object
- **layers** (*boolean array of 20 items, (optional)*) – Layer

```
bpy.ops.surface.primitive_nurbs_surface_torus_add (view_align=False,
                                                    enter_editmode=False, location=(0.0,
                                                    0.0, 0.0), rotation=(0.0, 0.0, 0.0),
                                                    layers=(False, False, False, False,
                                                    False, False, False, False, False,
                                                    False, False, False, False, False,
                                                    False, False, False, False, False,
                                                    False))
```

Construct a Nurbs surface Torus

**Parameters**

- **view\_align** (*boolean, (optional)*) – Align to View, Align the new object to the view
- **enter\_editmode** (*boolean, (optional)*) – Enter Editmode, Enter editmode when adding this object
- **location** (*float array of 3 items in [-inf, inf], (optional)*) – Location, Location for the newly added object
- **rotation** (*float array of 3 items in [-inf, inf], (optional)*) – Rotation, Rotation for the newly added object
- **layers** (*boolean array of 20 items, (optional)*) – Layer

**Text Operators**

```
bpy.ops.text.comment ()
    Convert selected text to comment
```

```
bpy.ops.text.convert_whitespace (type='SPACES')
    Convert whitespaces by type
```

**Parameters** **type** (*enum in ['SPACES', 'TABS'], (optional)*) – Type, Type of whitespace to convert to

```
bpy.ops.text.copy ()
    Copy selected text to clipboard
```

```
bpy.ops.text.cursor_set (x=0, y=0)
    Set cursor position
```

**Parameters**

- **x** (*int in [-inf, inf], (optional)*) – X
- **y** (*int in [-inf, inf], (optional)*) – Y

`bpy.ops.text.cut()`  
Cut selected text to clipboard

`bpy.ops.text.delete (type='NEXT_CHARACTER')`  
Delete text by cursor position

**Parameters** `type` (enum in ['NEXT\_CHARACTER', 'PREVIOUS\_CHARACTER', 'NEXT\_WORD', 'PREVIOUS\_WORD'], (optional)) – Type, Which part of the text to delete

`bpy.ops.text.find()`  
Find specified text

`bpy.ops.text.find_set_selected()`  
Find specified text and set as selected

`bpy.ops.text.indent()`  
Indent selected text

`bpy.ops.text.insert (text="")`  
Insert text at cursor position

**Parameters** `text` (string, (optional)) – Text, Text to insert at the cursor position

`bpy.ops.text.jump (line=1)`  
Jump cursor to line

**Parameters** `line` (int in [1, inf], (optional)) – Line, Line number to jump to

`bpy.ops.text.line_break()`  
Insert line break at cursor position

`bpy.ops.text.line_number()`  
The current line number

`bpy.ops.text.make_internal()`  
Make active text file internal

`bpy.ops.text.mark_all()`  
Mark all specified text

`bpy.ops.text.markers_clear()`  
Clear all markers

`bpy.ops.text.move (type='LINE_BEGIN')`  
Move cursor to position type

**Parameters** `type` (enum in ['LINE\_BEGIN', 'LINE\_END', 'FILE\_TOP', 'FILE\_BOTTOM', 'PREVIOUS\_CHARACTER', 'NEXT\_CHARACTER', 'PREVIOUS\_WORD', 'NEXT\_WORD', 'PREVIOUS\_LINE', 'NEXT\_LINE', 'PREVIOUS\_PAGE', 'NEXT\_PAGE'], (optional)) – Type, Where to move cursor to

`bpy.ops.text.move_select (type='LINE_BEGIN')`  
Make selection from current cursor position to new cursor position type

**Parameters** `type` (enum in ['LINE\_BEGIN', 'LINE\_END', 'FILE\_TOP', 'FILE\_BOTTOM', 'PREVIOUS\_CHARACTER', 'NEXT\_CHARACTER', 'PREVIOUS\_WORD', 'NEXT\_WORD', 'PREVIOUS\_LINE', 'NEXT\_LINE', 'PREVIOUS\_PAGE', 'NEXT\_PAGE'], (optional)) – Type, Where to move cursor to, to make a selection

`bpy.ops.text.new()`  
Create a new text data block

`bpy.ops.text.next_marker()`  
Move to next marker

`bpy.ops.text.open` (*filepath=""*, *filter\_blender=False*, *filter\_image=False*, *filter\_movie=False*, *filter\_python=True*, *filter\_font=False*, *filter\_sound=False*, *filter\_text=True*, *filter\_btx=False*, *filter\_collada=False*, *filter\_folder=True*, *filemode=9*, *internal=False*)

Open a new text data block

#### Parameters

- **filepath** (*string, (optional)*) – File Path, Path to file
- **filter\_blender** (*boolean, (optional)*) – Filter .blend files
- **filter\_image** (*boolean, (optional)*) – Filter image files
- **filter\_movie** (*boolean, (optional)*) – Filter movie files
- **filter\_python** (*boolean, (optional)*) – Filter python files
- **filter\_font** (*boolean, (optional)*) – Filter font files
- **filter\_sound** (*boolean, (optional)*) – Filter sound files
- **filter\_text** (*boolean, (optional)*) – Filter text files
- **filter\_btx** (*boolean, (optional)*) – Filter btx files
- **filter\_collada** (*boolean, (optional)*) – Filter COLLADA files
- **filter\_folder** (*boolean, (optional)*) – Filter folders
- **filemode** (*int in [1, 9], (optional)*) – File Browser Mode, The setting for the file browser mode to load a .blend file, a library or a special file
- **internal** (*boolean, (optional)*) – Make internal, Make text file internal after loading

`bpy.ops.text.override_toggle` ()  
Toggle overwrite while typing

`bpy.ops.text.paste` (*selection=False*)  
Paste text from clipboard

**Parameters** *selection* (*boolean, (optional)*) – Selection, Paste text selected elsewhere rather than copied (X11 only)

`bpy.ops.text.previous_marker` ()  
Move to previous marker

`bpy.ops.text.properties` ()  
Toggle text properties panel

`bpy.ops.text.refresh_pyconstraints` ()  
Refresh all pyconstraints

`bpy.ops.text.reload` ()  
Reload active text data block from its file

`bpy.ops.text.replace` ()  
Replace text with the specified text

`bpy.ops.text.replace_set_selected` ()  
Replace text with specified text and set as selected

`bpy.ops.text.resolve_conflict` (*resolution='IGNORE'*)  
When external text is out of sync, resolve the conflict

**Parameters** *resolution* (*enum in ['IGNORE', 'RELOAD', 'SAVE', 'MAKE\_INTERNAL'], (optional)*) – Resolution, How to solve conflict due to differences in internal and external text

`bpy.ops.text.run_script()`  
Run active script

`bpy.ops.text.save()`  
Save active text data block

`bpy.ops.text.save_as(filepath="", check_existing=True, filter_blender=False, filter_image=False, filter_movie=False, filter_python=True, filter_font=False, filter_sound=False, filter_text=True, filter_btx=False, filter_collada=False, filter_folder=True, filemode=9)`  
Save active text file with options

**Parameters**

- **filepath** (*string, (optional)*) – File Path, Path to file
- **check\_existing** (*boolean, (optional)*) – Check Existing, Check and warn on overwriting existing files
- **filter\_blender** (*boolean, (optional)*) – Filter .blend files
- **filter\_image** (*boolean, (optional)*) – Filter image files
- **filter\_movie** (*boolean, (optional)*) – Filter movie files
- **filter\_python** (*boolean, (optional)*) – Filter python files
- **filter\_font** (*boolean, (optional)*) – Filter font files
- **filter\_sound** (*boolean, (optional)*) – Filter sound files
- **filter\_text** (*boolean, (optional)*) – Filter text files
- **filter\_btx** (*boolean, (optional)*) – Filter btx files
- **filter\_collada** (*boolean, (optional)*) – Filter COLLADA files
- **filter\_folder** (*boolean, (optional)*) – Filter folders
- **filemode** (*int in [1, 9], (optional)*) – File Browser Mode, The setting for the file browser mode to load a .blend file, a library or a special file

`bpy.ops.text.scroll(lines=1)`  
Scroll text screen

**Parameters** **lines** (*int in [-inf, inf], (optional)*) – Lines, Number of lines to scroll

`bpy.ops.text.scroll_bar(lines=1)`  
Scroll text screen

**Parameters** **lines** (*int in [-inf, inf], (optional)*) – Lines, Number of lines to scroll

`bpy.ops.text.select_all()`  
Select all text

`bpy.ops.text.select_line()`  
Select text by line

`bpy.ops.text.select_word()`  
Select word under cursor

`bpy.ops.text.selection_set(select=False)`  
Set cursor selection

**Parameters** **select** (*boolean, (optional)*) – Select, Set selection end rather than cursor

`bpy.ops.text.to_3d_object` (*split\_lines=False*)  
Create 3d text object from active text data block

**Parameters** `split_lines` (*boolean, (optional)*) – Split Lines, Create one object per line in the text

`bpy.ops.text.uncomment` ()  
Convert selected comment to text

`bpy.ops.text.unindent` ()  
Unindent selected text

`bpy.ops.text.unlink` ()  
Unlink active text data block

## Texture Operators

`bpy.ops.texture.envmap_clear` ()  
Discard the environment map and free it from memory

`bpy.ops.texture.envmap_clear_all` ()  
Discard all environment maps in the .blend file and free them from memory

`bpy.ops.texture.envmap_save` (*layout=(0.0, 0.0, 1.0, 0.0, 2.0, 0.0, 0.0, 1.0, 1.0, 1.0, 2.0, 1.0), filepath=""*, *check\_existing=True, filter\_blender=False, filter\_image=True, filter\_movie=True, filter\_python=False, filter\_font=False, filter\_sound=False, filter\_text=False, filter\_btx=False, filter\_collada=False, filter\_folder=True, filemode=9*)  
Save the current generated Environment map to an image file

### Parameters

- **layout** (*float array of 12 items in [-inf, inf], (optional)*) – File layout, Flat array describing the X,Y position of each cube face in the output image, where 1 is the size of a face - order is [+Z -Z +Y -X -Y +X] (use -1 to skip a face)
- **filepath** (*string, (optional)*) – File Path, Path to file
- **check\_existing** (*boolean, (optional)*) – Check Existing, Check and warn on overwriting existing files
- **filter\_blender** (*boolean, (optional)*) – Filter .blend files
- **filter\_image** (*boolean, (optional)*) – Filter image files
- **filter\_movie** (*boolean, (optional)*) – Filter movie files
- **filter\_python** (*boolean, (optional)*) – Filter python files
- **filter\_font** (*boolean, (optional)*) – Filter font files
- **filter\_sound** (*boolean, (optional)*) – Filter sound files
- **filter\_text** (*boolean, (optional)*) – Filter text files
- **filter\_btx** (*boolean, (optional)*) – Filter btx files
- **filter\_collada** (*boolean, (optional)*) – Filter COLLADA files
- **filter\_folder** (*boolean, (optional)*) – Filter folders
- **filemode** (*int in [1, 9], (optional)*) – File Browser Mode, The setting for the file browser mode to load a .blend file, a library or a special file

`bpy.ops.texture.new()`  
Add a new texture

`bpy.ops.texture.slot_copy()`  
Copy the material texture settings and nodes

`bpy.ops.texture.slot_move(type='UP')`  
Move texture slots up and down

**Parameters** `type` (enum in ['UP', 'DOWN'], (optional)) – Type

`bpy.ops.texture.slot_paste()`  
Copy the texture settings and nodes

## Time Operators

`bpy.ops.time.end_frame_set()`  
Set the end frame

`bpy.ops.time.start_frame_set()`  
Set the start frame

`bpy.ops.time.view_all()`  
Show the entire playable frame range

## Transform Operators

`bpy.ops.transform.create_orientation(name="", use=False, overwrite=False)`  
Create transformation orientation from selection

### Parameters

- **name** (string, (optional)) – Name, Text to insert at the cursor position
- **use** (boolean, (optional)) – Use after creation, Select orientation after its creation
- **overwrite** (boolean, (optional)) – Overwrite previous, Overwrite previously created orientation with same name

`bpy.ops.transform.delete_orientation()`  
Delete transformation orientation

`bpy.ops.transform.edge_crease(value=0.0, snap=False, snap_target='CLOSEST', snap_point=(0.0, 0.0, 0.0), snap_align=False, snap_normal=(0.0, 0.0, 0.0), release_confirm=False)`

Change the crease of edges

### Parameters

- **value** (float in [-1, 1], (optional)) – Factor
- **snap** (boolean, (optional)) – Use Snapping Options
- **snap\_target** (enum in ['CLOSEST', 'CENTER', 'MEDIAN', 'ACTIVE'], (optional)) – Target
  - CLOSEST Closest, Snap closest point onto target.
  - CENTER Center, Snap center onto target.
  - MEDIAN Median, Snap median onto target.
  - ACTIVE Active, Snap active onto target.

- **snap\_point** (*float array of 3 items in [-inf, inf], (optional)*) – Point
- **snap\_align** (*boolean, (optional)*) – Align with Point Normal
- **snap\_normal** (*float array of 3 items in [-inf, inf], (optional)*) – Normal
- **release\_confirm** (*boolean, (optional)*) – Confirm on Release, Always confirm operation when releasing button

`bpy.ops.transform.edge_slide` (*value=0.0, mirror=False, snap=False, snap\_target='CLOSEST', snap\_point=(0.0, 0.0, 0.0), snap\_align=False, snap\_normal=(0.0, 0.0, 0.0), correct\_uv=False, release\_confirm=False*)

Slide an edge loop along a mesh

#### Parameters

- **value** (*float in [-1, 1], (optional)*) – Factor
- **mirror** (*boolean, (optional)*) – Mirror Editing
- **snap** (*boolean, (optional)*) – Use Snapping Options
- **snap\_target** (*enum in ['CLOSEST', 'CENTER', 'MEDIAN', 'ACTIVE'], (optional)*) – Target
  - CLOSEST Closest, Snap closest point onto target.
  - CENTER Center, Snap center onto target.
  - MEDIAN Median, Snap median onto target.
  - ACTIVE Active, Snap active onto target.
- **snap\_point** (*float array of 3 items in [-inf, inf], (optional)*) – Point
- **snap\_align** (*boolean, (optional)*) – Align with Point Normal
- **snap\_normal** (*float array of 3 items in [-inf, inf], (optional)*) – Normal
- **correct\_uv** (*boolean, (optional)*) – Correct UV coords when transforming
- **release\_confirm** (*boolean, (optional)*) – Confirm on Release, Always confirm operation when releasing button

`bpy.ops.transform.mirror` (*constraint\_axis=(False, False, False), constraint\_orientation='', proportional='DISABLED', proportional\_edit\_falloff='SMOOTH', proportional\_size=1.0, release\_confirm=False*)

Mirror selected vertices around one or more axes

#### Parameters

- **constraint\_axis** (*boolean array of 3 items, (optional)*) – Constraint Axis
- **constraint\_orientation** (*enum in [], (optional)*) – Orientation, Transformation orientation
- **proportional** (*enum in ['DISABLED', 'ENABLED', 'CONNECTED'], (optional)*) – Proportional Editing
  - DISABLED Disable, Proportional Editing disabled.
  - ENABLED Enable, Proportional Editing enabled.
  - CONNECTED Connected, Proportional Editing using connected geometry only.
- **proportional\_edit\_falloff** (*enum in ['SMOOTH', 'SPHERE', 'ROOT', 'SHARP', 'LINEAR', 'CONSTANT', 'RANDOM'], (optional)*) – Proportional Editing Falloff, Falloff type for proportional editing mode

- SMOOTH Smooth, Smooth falloff.
- SPHERE Sphere, Spherical falloff.
- ROOT Root, Root falloff.
- SHARP Sharp, Sharp falloff.
- LINEAR Linear, Linear falloff.
- CONSTANT Constant, Consant falloff.
- RANDOM Random, Random falloff.
- **proportional\_size** (*float in [1e-05, inf], (optional)*) – Proportional Size
- **release\_confirm** (*boolean, (optional)*) – Confirm on Release, Always confirm operation when releasing button

```
bpy.ops.transform.push_pull (value=0.0, mirror=False, proportional='DISABLED', proportional_edit_falloff='SMOOTH', proportional_size=1.0, snap=False, snap_target='CLOSEST', snap_point=(0.0, 0.0, 0.0), snap_align=False, snap_normal=(0.0, 0.0, 0.0), release_confirm=False)
```

Push/Pull selected items

#### Parameters

- **value** (*float in [-inf, inf], (optional)*) – Distance
- **mirror** (*boolean, (optional)*) – Mirror Editing
- **proportional** (*enum in ['DISABLED', 'ENABLED', 'CONNECTED'], (optional)*) – Proportional Editing
  - DISABLED Disable, Proportional Editing disabled.
  - ENABLED Enable, Proportional Editing enabled.
  - CONNECTED Connected, Proportional Editing using connected geometry only.
- **proportional\_edit\_falloff** (*enum in ['SMOOTH', 'SPHERE', 'ROOT', 'SHARP', 'LINEAR', 'CONSTANT', 'RANDOM'], (optional)*) – Proportional Editing Falloff, Falloff type for proportional editing mode
  - SMOOTH Smooth, Smooth falloff.
  - SPHERE Sphere, Spherical falloff.
  - ROOT Root, Root falloff.
  - SHARP Sharp, Sharp falloff.
  - LINEAR Linear, Linear falloff.
  - CONSTANT Constant, Consant falloff.
  - RANDOM Random, Random falloff.
- **proportional\_size** (*float in [1e-05, inf], (optional)*) – Proportional Size
- **snap** (*boolean, (optional)*) – Use Snapping Options
- **snap\_target** (*enum in ['CLOSEST', 'CENTER', 'MEDIAN', 'ACTIVE'], (optional)*) – Target
  - CLOSEST Closest, Snap closest point onto target.
  - CENTER Center, Snap center onto target.



- **MEDIAN** Median, Snap median onto target.
- **ACTIVE** Active, Snap active onto target.
- **snap\_point** (*float array of 3 items in [-inf, inf], (optional)*) – Point
- **snap\_align** (*boolean, (optional)*) – Align with Point Normal
- **snap\_normal** (*float array of 3 items in [-inf, inf], (optional)*) – Normal
- **release\_confirm** (*boolean, (optional)*) – Confirm on Release, Always confirm operation when releasing button

`bpy.ops.transform.resize` (*value=(1.0, 1.0, 1.0), constraint\_axis=(False, False, False), constraint\_orientation='', mirror=False, proportional='DISABLED', proportional\_edit\_falloff='SMOOTH', proportional\_size=1.0, snap=False, snap\_target='CLOSEST', snap\_point=(0.0, 0.0, 0.0), snap\_align=False, snap\_normal=(0.0, 0.0, 0.0), texture\_space=False, release\_confirm=False*)

Resize selected items

#### Parameters

- **value** (*float array of 3 items in [-inf, inf], (optional)*) – Vector
- **constraint\_axis** (*boolean array of 3 items, (optional)*) – Constraint Axis
- **constraint\_orientation** (*enum in [], (optional)*) – Orientation, Transformation orientation
- **mirror** (*boolean, (optional)*) – Mirror Editing
- **proportional** (*enum in ['DISABLED', 'ENABLED', 'CONNECTED'], (optional)*) – Proportional Editing
  - **DISABLED** Disable, Proportional Editing disabled.
  - **ENABLED** Enable, Proportional Editing enabled.
  - **CONNECTED** Connected, Proportional Editing using connected geometry only.
- **proportional\_edit\_falloff** (*enum in ['SMOOTH', 'SPHERE', 'ROOT', 'SHARP', 'LINEAR', 'CONSTANT', 'RANDOM'], (optional)*) – Proportional Editing Falloff, Falloff type for proportional editing mode
  - **SMOOTH** Smooth, Smooth falloff.
  - **SPHERE** Sphere, Spherical falloff.
  - **ROOT** Root, Root falloff.
  - **SHARP** Sharp, Sharp falloff.
  - **LINEAR** Linear, Linear falloff.
  - **CONSTANT** Constant, Consant falloff.
  - **RANDOM** Random, Random falloff.
- **proportional\_size** (*float in [1e-05, inf], (optional)*) – Proportional Size
- **snap** (*boolean, (optional)*) – Use Snapping Options
- **snap\_target** (*enum in ['CLOSEST', 'CENTER', 'MEDIAN', 'ACTIVE'], (optional)*) – Target
  - **CLOSEST** Closest, Snap closest point onto target.
  - **CENTER** Center, Snap center onto target.

- **MEDIAN** Median, Snap median onto target.
- **ACTIVE** Active, Snap active onto target.
- **snap\_point** (*float array of 3 items in [-inf, inf], (optional)*) – Point
- **snap\_align** (*boolean, (optional)*) – Align with Point Normal
- **snap\_normal** (*float array of 3 items in [-inf, inf], (optional)*) – Normal
- **texture\_space** (*boolean, (optional)*) – Edit Object data texture space
- **release\_confirm** (*boolean, (optional)*) – Confirm on Release, Always confirm operation when releasing button

```
bpy.ops.transform.rotate (value=(0.0), axis=(0.0, 0.0, 0.0), constraint_axis=(False, False, False),
                           constraint_orientation='', mirror=False, proportional='DISABLED',
                           proportional_edit_falloff='SMOOTH', proportional_size=1.0,
                           snap=False, snap_target='CLOSEST', snap_point=(0.0, 0.0, 0.0),
                           snap_align=False, snap_normal=(0.0, 0.0, 0.0), release_confirm=False)
```

Rotate selected items

### Parameters

- **value** (*float array of 1 items in [-inf, inf], (optional)*) – Angle
- **axis** (*float array of 3 items in [-inf, inf], (optional)*) – Axis, The axis around which the transformation occurs
- **constraint\_axis** (*boolean array of 3 items, (optional)*) – Constraint Axis
- **constraint\_orientation** (*enum in [], (optional)*) – Orientation, Transformation orientation
- **mirror** (*boolean, (optional)*) – Mirror Editing
- **proportional** (*enum in ['DISABLED', 'ENABLED', 'CONNECTED'], (optional)*) – Proportional Editing
  - **DISABLED** Disable, Proportional Editing disabled.
  - **ENABLED** Enable, Proportional Editing enabled.
  - **CONNECTED** Connected, Proportional Editing using connected geometry only.
- **proportional\_edit\_falloff** (*enum in ['SMOOTH', 'SPHERE', 'ROOT', 'SHARP', 'LINEAR', 'CONSTANT', 'RANDOM'], (optional)*) – Proportional Editing Falloff, Falloff type for proportional editing mode
  - **SMOOTH** Smooth, Smooth falloff.
  - **SPHERE** Sphere, Spherical falloff.
  - **ROOT** Root, Root falloff.
  - **SHARP** Sharp, Sharp falloff.
  - **LINEAR** Linear, Linear falloff.
  - **CONSTANT** Constant, Consant falloff.
  - **RANDOM** Random, Random falloff.
- **proportional\_size** (*float in [1e-05, inf], (optional)*) – Proportional Size
- **snap** (*boolean, (optional)*) – Use Snapping Options
- **snap\_target** (*enum in ['CLOSEST', 'CENTER', 'MEDIAN', 'ACTIVE'], (optional)*) – Target

- CLOSEST Closest, Snap closest point onto target.
- CENTER Center, Snap center onto target.
- MEDIAN Median, Snap median onto target.
- ACTIVE Active, Snap active onto target.
- **snap\_point** (*float array of 3 items in [-inf, inf], (optional)*) – Point
- **snap\_align** (*boolean, (optional)*) – Align with Point Normal
- **snap\_normal** (*float array of 3 items in [-inf, inf], (optional)*) – Normal
- **release\_confirm** (*boolean, (optional)*) – Confirm on Release, Always confirm operation when releasing button

`bpy.ops.transform.select_orientation` (*orientation=''*)  
Select transformation orientation

**Parameters** **orientation** (*enum in [], (optional)*) – Orientation, Transformation orientation

`bpy.ops.transform.seq_slide` (*value=(1.0, 1.0), snap=False, snap\_target='CLOSEST', snap\_point=(0.0, 0.0, 0.0), snap\_align=False, snap\_normal=(0.0, 0.0, 0.0), release\_confirm=False*)

Slide a sequence strip in time

**Parameters**

- **value** (*float array of 2 items in [-inf, inf], (optional)*) – Angle
- **snap** (*boolean, (optional)*) – Use Snapping Options
- **snap\_target** (*enum in ['CLOSEST', 'CENTER', 'MEDIAN', 'ACTIVE'], (optional)*) – Target
  - CLOSEST Closest, Snap closest point onto target.
  - CENTER Center, Snap center onto target.
  - MEDIAN Median, Snap median onto target.
  - ACTIVE Active, Snap active onto target.
- **snap\_point** (*float array of 3 items in [-inf, inf], (optional)*) – Point
- **snap\_align** (*boolean, (optional)*) – Align with Point Normal
- **snap\_normal** (*float array of 3 items in [-inf, inf], (optional)*) – Normal
- **release\_confirm** (*boolean, (optional)*) – Confirm on Release, Always confirm operation when releasing button

`bpy.ops.transform.shear` (*value=0.0, mirror=False, proportional='DISABLED', proportional\_edit\_falloff='SMOOTH', proportional\_size=1.0, snap=False, snap\_target='CLOSEST', snap\_point=(0.0, 0.0, 0.0), snap\_align=False, snap\_normal=(0.0, 0.0, 0.0), release\_confirm=False*)

Shear selected items along the horizontal screen axis

**Parameters**

- **value** (*float in [-inf, inf], (optional)*) – Offset
- **mirror** (*boolean, (optional)*) – Mirror Editing
- **proportional** (*enum in ['DISABLED', 'ENABLED', 'CONNECTED'], (optional)*) – Proportional Editing

- `DISABLED` Disable, Proportional Editing disabled.
- `ENABLED` Enable, Proportional Editing enabled.
- `CONNECTED` Connected, Proportional Editing using connected geometry only.
- **proportional\_edit\_falloff** (*enum in ['SMOOTH', 'SPHERE', 'ROOT', 'SHARP', 'LINEAR', 'CONSTANT', 'RANDOM'], (optional)*) – Proportional Editing Falloff, Falloff type for proportional editing mode
  - `SMOOTH` Smooth, Smooth falloff.
  - `SPHERE` Sphere, Spherical falloff.
  - `ROOT` Root, Root falloff.
  - `SHARP` Sharp, Sharp falloff.
  - `LINEAR` Linear, Linear falloff.
  - `CONSTANT` Constant, Constant falloff.
  - `RANDOM` Random, Random falloff.
- **proportional\_size** (*float in [1e-05, inf], (optional)*) – Proportional Size
- **snap** (*boolean, (optional)*) – Use Snapping Options
- **snap\_target** (*enum in ['CLOSEST', 'CENTER', 'MEDIAN', 'ACTIVE'], (optional)*) – Target
  - `CLOSEST` Closest, Snap closest point onto target.
  - `CENTER` Center, Snap center onto target.
  - `MEDIAN` Median, Snap median onto target.
  - `ACTIVE` Active, Snap active onto target.
- **snap\_point** (*float array of 3 items in [-inf, inf], (optional)*) – Point
- **snap\_align** (*boolean, (optional)*) – Align with Point Normal
- **snap\_normal** (*float array of 3 items in [-inf, inf], (optional)*) – Normal
- **release\_confirm** (*boolean, (optional)*) – Confirm on Release, Always confirm operation when releasing button

```
bpy.ops.transform.shrink_fatten (value=0.0, mirror=False, proportional='DISABLED', proportional_edit_falloff='SMOOTH', proportional_size=1.0, snap=False, snap_target='CLOSEST', snap_point=(0.0, 0.0, 0.0), snap_align=False, snap_normal=(0.0, 0.0, 0.0), release_confirm=False)
```

Shrink/fatten selected vertices along normals

#### Parameters

- **value** (*float in [-inf, inf], (optional)*) – Offset
- **mirror** (*boolean, (optional)*) – Mirror Editing
- **proportional** (*enum in ['DISABLED', 'ENABLED', 'CONNECTED'], (optional)*) – Proportional Editing
  - `DISABLED` Disable, Proportional Editing disabled.
  - `ENABLED` Enable, Proportional Editing enabled.
  - `CONNECTED` Connected, Proportional Editing using connected geometry only.

- **proportional\_edit\_falloff** (*enum in ['SMOOTH', 'SPHERE', 'ROOT', 'SHARP', 'LINEAR', 'CONSTANT', 'RANDOM'], (optional)*) – Proportional Editing Falloff, Falloff type for proportional editing mode
  - SMOOTH Smooth, Smooth falloff.
  - SPHERE Sphere, Spherical falloff.
  - ROOT Root, Root falloff.
  - SHARP Sharp, Sharp falloff.
  - LINEAR Linear, Linear falloff.
  - CONSTANT Constant, Consant falloff.
  - RANDOM Random, Random falloff.
- **proportional\_size** (*float in [1e-05, inf], (optional)*) – Proportional Size
- **snap** (*boolean, (optional)*) – Use Snapping Options
- **snap\_target** (*enum in ['CLOSEST', 'CENTER', 'MEDIAN', 'ACTIVE'], (optional)*) – Target
  - CLOSEST Closest, Snap closest point onto target.
  - CENTER Center, Snap center onto target.
  - MEDIAN Median, Snap median onto target.
  - ACTIVE Active, Snap active onto target.
- **snap\_point** (*float array of 3 items in [-inf, inf], (optional)*) – Point
- **snap\_align** (*boolean, (optional)*) – Align with Point Normal
- **snap\_normal** (*float array of 3 items in [-inf, inf], (optional)*) – Normal
- **release\_confirm** (*boolean, (optional)*) – Confirm on Release, Always confirm operation when releasing button

`bpy.ops.transform.snap_type` (*type='INCREMENT'*)  
Set the snap element type

**Parameters** **type** (*enum in ['INCREMENT', 'VERTEX', 'EDGE', 'FACE', 'VOLUME'], (optional)*)

– Type, Set the snap element type

- INCREMENT Increment, Snap to increments of grid.
- VERTEX Vertex, Snap to vertices.
- EDGE Edge, Snap to edges.
- FACE Face, Snap to faces.
- VOLUME Volume, Snap to volume.

`bpy.ops.transform.tilt` (*value=(0.0), constraint\_axis=(False, False, False), constraint\_orientation=',', mirror=False, proportional='DISABLED', proportional\_edit\_falloff='SMOOTH', proportional\_size=1.0, snap=False, snap\_target='CLOSEST', snap\_point=(0.0, 0.0, 0.0), snap\_align=False, snap\_normal=(0.0, 0.0, 0.0), release\_confirm=False*)

Tilt selected control vertices of 3d curve

**Parameters**

- **value** (*float array of 1 items in [-inf, inf], (optional)*) – Angle

- **constraint\_axis** (*boolean array of 3 items, (optional)*) – Constraint Axis
- **constraint\_orientation** (*enum in [], (optional)*) – Orientation, Transformation orientation
- **mirror** (*boolean, (optional)*) – Mirror Editing
- **proportional** (*enum in ['DISABLED', 'ENABLED', 'CONNECTED'], (optional)*) – Proportional Editing
  - DISABLED Disable, Proportional Editing disabled.
  - ENABLED Enable, Proportional Editing enabled.
  - CONNECTED Connected, Proportional Editing using connected geometry only.
- **proportional\_edit\_falloff** (*enum in ['SMOOTH', 'SPHERE', 'ROOT', 'SHARP', 'LINEAR', 'CONSTANT', 'RANDOM'], (optional)*) – Proportional Editing Falloff, Falloff type for proportional editing mode
  - SMOOTH Smooth, Smooth falloff.
  - SPHERE Sphere, Spherical falloff.
  - ROOT Root, Root falloff.
  - SHARP Sharp, Sharp falloff.
  - LINEAR Linear, Linear falloff.
  - CONSTANT Constant, Consant falloff.
  - RANDOM Random, Random falloff.
- **proportional\_size** (*float in [1e-05, inf], (optional)*) – Proportional Size
- **snap** (*boolean, (optional)*) – Use Snapping Options
- **snap\_target** (*enum in ['CLOSEST', 'CENTER', 'MEDIAN', 'ACTIVE'], (optional)*) – Target
  - CLOSEST Closest, Snap closest point onto target.
  - CENTER Center, Snap center onto target.
  - MEDIAN Median, Snap median onto target.
  - ACTIVE Active, Snap active onto target.
- **snap\_point** (*float array of 3 items in [-inf, inf], (optional)*) – Point
- **snap\_align** (*boolean, (optional)*) – Align with Point Normal
- **snap\_normal** (*float array of 3 items in [-inf, inf], (optional)*) – Normal
- **release\_confirm** (*boolean, (optional)*) – Confirm on Release, Always confirm operation when releasing button

```
bpy.ops.transform.tosphere (value=0.0, mirror=False, proportional='DISABLED', proportional_edit_falloff='SMOOTH', proportional_size=1.0, snap=False, snap_target='CLOSEST', snap_point=(0.0, 0.0, 0.0), snap_align=False, snap_normal=(0.0, 0.0, 0.0), release_confirm=False)
```

Move selected vertices outward in a spherical shape around mesh center

#### Parameters

- **value** (*float in [0, 1], (optional)*) – Factor
- **mirror** (*boolean, (optional)*) – Mirror Editing

- **proportional** (*enum in ['DISABLED', 'ENABLED', 'CONNECTED'], (optional)*) – Proportional Editing
  - `DISABLED` Disable, Proportional Editing disabled.
  - `ENABLED` Enable, Proportional Editing enabled.
  - `CONNECTED` Connected, Proportional Editing using connected geometry only.
- **proportional\_edit\_falloff** (*enum in ['SMOOTH', 'SPHERE', 'ROOT', 'SHARP', 'LINEAR', 'CONSTANT', 'RANDOM'], (optional)*) – Proportional Editing Falloff, Falloff type for proportional editing mode
  - `SMOOTH` Smooth, Smooth falloff.
  - `SPHERE` Sphere, Spherical falloff.
  - `ROOT` Root, Root falloff.
  - `SHARP` Sharp, Sharp falloff.
  - `LINEAR` Linear, Linear falloff.
  - `CONSTANT` Constant, Constant falloff.
  - `RANDOM` Random, Random falloff.
- **proportional\_size** (*float in [1e-05, inf], (optional)*) – Proportional Size
- **snap** (*boolean, (optional)*) – Use Snapping Options
- **snap\_target** (*enum in ['CLOSEST', 'CENTER', 'MEDIAN', 'ACTIVE'], (optional)*) – Target
  - `CLOSEST` Closest, Snap closest point onto target.
  - `CENTER` Center, Snap center onto target.
  - `MEDIAN` Median, Snap median onto target.
  - `ACTIVE` Active, Snap active onto target.
- **snap\_point** (*float array of 3 items in [-inf, inf], (optional)*) – Point
- **snap\_align** (*boolean, (optional)*) – Align with Point Normal
- **snap\_normal** (*float array of 3 items in [-inf, inf], (optional)*) – Normal
- **release\_confirm** (*boolean, (optional)*) – Confirm on Release, Always confirm operation when releasing button

```
bpy.ops.transform.trackball (value=(1.0, 1.0), mirror=False, proportional='DISABLED',
                             proportional_edit_falloff='SMOOTH', proportional_size=1.0,
                             snap=False, snap_target='CLOSEST', snap_point=(0.0, 0.0,
                             0.0), snap_align=False, snap_normal=(0.0, 0.0, 0.0), re-
                             lease_confirm=False)
```

Trackball style rotation of selected items

#### Parameters

- **value** (*float array of 2 items in [-inf, inf], (optional)*) – angle
- **mirror** (*boolean, (optional)*) – Mirror Editing
- **proportional** (*enum in ['DISABLED', 'ENABLED', 'CONNECTED'], (optional)*) – Proportional Editing
  - `DISABLED` Disable, Proportional Editing disabled.

- `ENABLED` Enable, Proportional Editing enabled.
- `CONNECTED` Connected, Proportional Editing using connected geometry only.
- **proportional\_edit\_falloff** (*enum in ['SMOOTH', 'SPHERE', 'ROOT', 'SHARP', 'LINEAR', 'CONSTANT', 'RANDOM'], (optional)*) – Proportional Editing Falloff, Falloff type for proportional editing mode
  - `SMOOTH` Smooth, Smooth falloff.
  - `SPHERE` Sphere, Spherical falloff.
  - `ROOT` Root, Root falloff.
  - `SHARP` Sharp, Sharp falloff.
  - `LINEAR` Linear, Linear falloff.
  - `CONSTANT` Constant, Consant falloff.
  - `RANDOM` Random, Random falloff.
- **proportional\_size** (*float in [1e-05, inf], (optional)*) – Proportional Size
- **snap** (*boolean, (optional)*) – Use Snapping Options
- **snap\_target** (*enum in ['CLOSEST', 'CENTER', 'MEDIAN', 'ACTIVE'], (optional)*) – Target
  - `CLOSEST` Closest, Snap closest point onto target.
  - `CENTER` Center, Snap center onto target.
  - `MEDIAN` Median, Snap median onto target.
  - `ACTIVE` Active, Snap active onto target.
- **snap\_point** (*float array of 3 items in [-inf, inf], (optional)*) – Point
- **snap\_align** (*boolean, (optional)*) – Align with Point Normal
- **snap\_normal** (*float array of 3 items in [-inf, inf], (optional)*) – Normal
- **release\_confirm** (*boolean, (optional)*) – Confirm on Release, Always confirm operation when releasing button

```
bpy.ops.transform.transform(mode='TRANSLATION', value=(0.0, 0.0, 0.0, 0.0), axis=(0.0, 0.0, 0.0), constraint_axis=(False, False, False), constraint_orientation='', mirror=False, proportional='DISABLED', proportional_edit_falloff='SMOOTH', proportional_size=1.0, snap=False, snap_target='CLOSEST', snap_point=(0.0, 0.0, 0.0), snap_align=False, snap_normal=(0.0, 0.0, 0.0), release_confirm=False)
```

Transform selected items by mode type

### Parameters

- **mode** (*enum in ['INIT', 'DUMMY', 'TRANSLATION', 'ROTATION', 'RESIZE', 'TOSPHERE', 'SHEAR', 'WARP', 'SHRINKFATTEN', 'TILT', 'TRACKBALL', 'PUSHPULL', 'CREASE', 'MIRROR', 'BONE\_SIZE', 'BONE\_ENVELOPE', 'CURVE\_SHRINKFATTEN', 'BONE\_ROLL', 'TIME\_TRANSLATE', 'TIME\_SLIDE', 'TIME\_SCALE', 'TIME\_EXTEND', 'BAKE\_TIME', 'BEVEL', 'BWEIGHT', 'ALIGN', 'EDGESLIDE', 'SEQSLIDE'], (optional)*) – Mode
- **value** (*float array of 4 items in [-inf, inf], (optional)*) – Values



- **axis** (*float array of 3 items in [-inf, inf], (optional)*) – Axis, The axis around which the transformation occurs
- **constraint\_axis** (*boolean array of 3 items, (optional)*) – Constraint Axis
- **constraint\_orientation** (*enum in [], (optional)*) – Orientation, Transformation orientation
- **mirror** (*boolean, (optional)*) – Mirror Editing
- **proportional** (*enum in ['DISABLED', 'ENABLED', 'CONNECTED'], (optional)*) – Proportional Editing
  - `DISABLED` Disable, Proportional Editing disabled.
  - `ENABLED` Enable, Proportional Editing enabled.
  - `CONNECTED` Connected, Proportional Editing using connected geometry only.
- **proportional\_edit\_falloff** (*enum in ['SMOOTH', 'SPHERE', 'ROOT', 'SHARP', 'LINEAR', 'CONSTANT', 'RANDOM'], (optional)*) – Proportional Editing Falloff, Falloff type for proportional editing mode
  - `SMOOTH` Smooth, Smooth falloff.
  - `SPHERE` Sphere, Spherical falloff.
  - `ROOT` Root, Root falloff.
  - `SHARP` Sharp, Sharp falloff.
  - `LINEAR` Linear, Linear falloff.
  - `CONSTANT` Constant, Consant falloff.
  - `RANDOM` Random, Random falloff.
- **proportional\_size** (*float in [1e-05, inf], (optional)*) – Proportional Size
- **snap** (*boolean, (optional)*) – Use Snapping Options
- **snap\_target** (*enum in ['CLOSEST', 'CENTER', 'MEDIAN', 'ACTIVE'], (optional)*) – Target
  - `CLOSEST` Closest, Snap closest point onto target.
  - `CENTER` Center, Snap center onto target.
  - `MEDIAN` Median, Snap median onto target.
  - `ACTIVE` Active, Snap active onto target.
- **snap\_point** (*float array of 3 items in [-inf, inf], (optional)*) – Point
- **snap\_align** (*boolean, (optional)*) – Align with Point Normal
- **snap\_normal** (*float array of 3 items in [-inf, inf], (optional)*) – Normal
- **release\_confirm** (*boolean, (optional)*) – Confirm on Release, Always confirm operation when releasing button

```
bpy.ops.transform.translate (value=(0.0, 0.0, 0.0), constraint_axis=(False, False, False), constraint_orientation='', mirror=False, proportional='DISABLED', proportional_edit_falloff='SMOOTH', proportional_size=1.0, snap=False, snap_target='CLOSEST', snap_point=(0.0, 0.0, 0.0), snap_align=False, snap_normal=(0.0, 0.0, 0.0), texture_space=False, release_confirm=False)
```

Translate selected items

**Parameters**

- **value** (*float array of 3 items in [-inf, inf], (optional)*) – Vector
- **constraint\_axis** (*boolean array of 3 items, (optional)*) – Constraint Axis
- **constraint\_orientation** (*enum in [], (optional)*) – Orientation, Transformation orientation
- **mirror** (*boolean, (optional)*) – Mirror Editing
- **proportional** (*enum in ['DISABLED', 'ENABLED', 'CONNECTED'], (optional)*) – Proportional Editing
  - **DISABLED** Disable, Proportional Editing disabled.
  - **ENABLED** Enable, Proportional Editing enabled.
  - **CONNECTED** Connected, Proportional Editing using connected geometry only.
- **proportional\_edit\_falloff** (*enum in ['SMOOTH', 'SPHERE', 'ROOT', 'SHARP', 'LINEAR', 'CONSTANT', 'RANDOM'], (optional)*) – Proportional Editing Falloff, Falloff type for proportional editing mode
  - **SMOOTH** Smooth, Smooth falloff.
  - **SPHERE** Sphere, Spherical falloff.
  - **ROOT** Root, Root falloff.
  - **SHARP** Sharp, Sharp falloff.
  - **LINEAR** Linear, Linear falloff.
  - **CONSTANT** Constant, Consant falloff.
  - **RANDOM** Random, Random falloff.
- **proportional\_size** (*float in [1e-05, inf], (optional)*) – Proportional Size
- **snap** (*boolean, (optional)*) – Use Snapping Options
- **snap\_target** (*enum in ['CLOSEST', 'CENTER', 'MEDIAN', 'ACTIVE'], (optional)*) – Target
  - **CLOSEST** Closest, Snap closest point onto target.
  - **CENTER** Center, Snap center onto target.
  - **MEDIAN** Median, Snap median onto target.
  - **ACTIVE** Active, Snap active onto target.
- **snap\_point** (*float array of 3 items in [-inf, inf], (optional)*) – Point
- **snap\_align** (*boolean, (optional)*) – Align with Point Normal
- **snap\_normal** (*float array of 3 items in [-inf, inf], (optional)*) – Normal
- **texture\_space** (*boolean, (optional)*) – Edit Object data texture space
- **release\_confirm** (*boolean, (optional)*) – Confirm on Release, Always confirm operation when releasing button

```
bpy.ops.transform.warp (value=(0.0), mirror=False, proportional='DISABLED', proportional_edit_falloff='SMOOTH', proportional_size=1.0, snap=False, snap_target='CLOSEST', snap_point=(0.0, 0.0, 0.0), snap_align=False, snap_normal=(0.0, 0.0, 0.0), release_confirm=False)
```

Warp selected items around the cursor

**Parameters**

- **value** (*float array of 1 items in [-inf, inf], (optional)*) – Angle
- **mirror** (*boolean, (optional)*) – Mirror Editing
- **proportional** (*enum in ['DISABLED', 'ENABLED', 'CONNECTED'], (optional)*) – Proportional Editing
  - DISABLED Disable, Proportional Editing disabled.
  - ENABLED Enable, Proportional Editing enabled.
  - CONNECTED Connected, Proportional Editing using connected geometry only.
- **proportional\_edit\_falloff** (*enum in ['SMOOTH', 'SPHERE', 'ROOT', 'SHARP', 'LINEAR', 'CONSTANT', 'RANDOM'], (optional)*) – Proportional Editing Falloff, Falloff type for proportional editing mode
  - SMOOTH Smooth, Smooth falloff.
  - SPHERE Sphere, Spherical falloff.
  - ROOT Root, Root falloff.
  - SHARP Sharp, Sharp falloff.
  - LINEAR Linear, Linear falloff.
  - CONSTANT Constant, Constant falloff.
  - RANDOM Random, Random falloff.
- **proportional\_size** (*float in [1e-05, inf], (optional)*) – Proportional Size
- **snap** (*boolean, (optional)*) – Use Snapping Options
- **snap\_target** (*enum in ['CLOSEST', 'CENTER', 'MEDIAN', 'ACTIVE'], (optional)*) – Target
  - CLOSEST Closest, Snap closest point onto target.
  - CENTER Center, Snap center onto target.
  - MEDIAN Median, Snap median onto target.
  - ACTIVE Active, Snap active onto target.
- **snap\_point** (*float array of 3 items in [-inf, inf], (optional)*) – Point
- **snap\_align** (*boolean, (optional)*) – Align with Point Normal
- **snap\_normal** (*float array of 3 items in [-inf, inf], (optional)*) – Normal
- **release\_confirm** (*boolean, (optional)*) – Confirm on Release, Always confirm operation when releasing button

**Ui Operators**

`bpy.ops.ui.copy_data_path_button()`  
Copy the RNA data path for this property to the clipboard

`bpy.ops.ui.copy_to_selected_button(all=True)`  
Copy property from this object to selected objects or bones

**Parameters** `all` (*boolean, (optional)*) – All, Reset to default values all elements of the array

`bpy.ops.ui.editsource()`  
Edit source code for a button

`bpy.ops.ui.eyedropper()`  
Sample a color from the Blender Window to store in a property

`bpy.ops.ui.reports_to_textblock()`  
Write the reports

`bpy.ops.ui.reset_default_button(all=True)`  
Reset this property's value to its default value

**Parameters** `all` (*boolean, (optional)*) – All, Reset to default values all elements of the array

`bpy.ops.ui.reset_default_theme()`  
Reset to the default theme colors

## Uv Operators

`bpy.ops.uv.align(axis='ALIGN_AUTO')`  
Align selected UV vertices to an axis

**Parameters** `axis` (*enum in ['ALIGN\_S', 'ALIGN\_T', 'ALIGN\_U', 'ALIGN\_AUTO', 'ALIGN\_X', 'ALIGN\_Y'], (optional)*) – Axis, Axis to align UV locations on

- `ALIGN_S` Straighten, Align UVs along the line defined by the endpoints.
- `ALIGN_T` Straighten X, Align UVs along the line defined by the endpoints along the X axis.
- `ALIGN_U` Straighten Y, Align UVs along the line defined by the endpoints along the Y axis.
- `ALIGN_AUTO` Align Auto, Automatically choose the axis on which there is most alignment already.
- `ALIGN_X` Align X, Align UVs on X axis.
- `ALIGN_Y` Align Y, Align UVs on Y axis.

`bpy.ops.uv.average_islands_scale()`  
Undocumented ([contribute](#))

`bpy.ops.uv.circle_select(x=0, y=0, radius=0, gesture_mode=0)`  
Select UV vertices using circle selection

### Parameters

- `x` (*int in [-inf, inf], (optional)*) – X
- `y` (*int in [-inf, inf], (optional)*) – Y
- `radius` (*int in [-inf, inf], (optional)*) – Radius
- `gesture_mode` (*int in [-inf, inf], (optional)*) – Gesture Mode

`bpy.ops.uv.cube_project(cube_size=1.0, correct_aspect=True, clip_to_bounds=False, scale_to_bounds=False)`  
Undocumented ([contribute](#))

### Parameters

- `cube_size` (*float in [0, inf], (optional)*) – Cube Size, Size of the cube to project on
- `correct_aspect` (*boolean, (optional)*) – Correct Aspect, Map UVs taking image aspect ratio into account

- **clip\_to\_bounds** (*boolean, (optional)*) – Clip to Bounds, Clip UV coordinates to bounds after unwrapping
- **scale\_to\_bounds** (*boolean, (optional)*) – Scale to Bounds, Scale UV coordinates to bounds after unwrapping

`bpy.ops.uv.cursor_set` (*location=(0.0, 0.0)*)

Set 2D cursor location

**Parameters** **location** (*float array of 2 items in [-inf, inf], (optional)*) – Location, Cursor location in normalised (0.0-1.0) coordinates

`bpy.ops.uv.cylinder_project` (*direction='VIEW\_ON\_EQUATOR', align='POLAR\_ZX', radius=1.0, correct\_aspect=True, clip\_to\_bounds=False, scale\_to\_bounds=False*)

Undocumented ([contribute](#))

#### Parameters

- **direction** (*enum in ['VIEW\_ON\_EQUATOR', 'VIEW\_ON\_POLES', 'ALIGN\_TO\_OBJECT'], (optional)*) – Direction, Direction of the sphere or cylinder
  - `VIEW_ON_EQUATOR` View on Equator, 3D view is on the equator.
  - `VIEW_ON_POLES` View on Poles, 3D view is on the poles.
  - `ALIGN_TO_OBJECT` Align to Object, Align according to object transform.
- **align** (*enum in ['POLAR\_ZX', 'POLAR\_ZY'], (optional)*) – Align, How to determine rotation around the pole
  - `POLAR_ZX` Polar ZX, Polar 0 is X.
  - `POLAR_ZY` Polar ZY, Polar 0 is Y.
- **radius** (*float in [0, inf], (optional)*) – Radius, Radius of the sphere or cylinder
- **correct\_aspect** (*boolean, (optional)*) – Correct Aspect, Map UVs taking image aspect ratio into account
- **clip\_to\_bounds** (*boolean, (optional)*) – Clip to Bounds, Clip UV coordinates to bounds after unwrapping
- **scale\_to\_bounds** (*boolean, (optional)*) – Scale to Bounds, Scale UV coordinates to bounds after unwrapping

`bpy.ops.uv.export_layout` (*filepath="", check\_existing=True, export\_all=False, mode='PNG', size=(1024, 1024), opacity=0.25*)

Export UV layout to file

#### Parameters

- **filepath** (*string, (optional)*) – File Path, File path used for exporting the SVG file
- **check\_existing** (*boolean, (optional)*) – Check Existing, Check and warn on overwriting existing files
- **export\_all** (*boolean, (optional)*) – All UVs, Export all UVs in this mesh (not just visible ones)
- **mode** (*enum in ['SVG', 'EPS', 'PNG'], (optional)*) – Format, File format to export the UV layout to
  - `SVG` Scalable Vector Graphic (.svg), Export the UV layout to a vector SVG file.
  - `EPS` Encapsulate PostScript (.eps), Export the UV layout to a vector EPS file.

- PNG PNG Image (.png), Export the UV layout to a bitmap image.
- **size** (*int array of 2 items in [8, 32768], (optional)*) – Dimensions of the exported file
- **opacity** (*float in [0, 1], (optional)*) – Fill Opacity

**File** `addons/io_mesh_uv_layout/__init__.py:167`

`bpy.ops.uv.follow_active_quads` (*mode='LENGTH'*)  
Follow UVs from active quads along continuous face loops

**Parameters** **mode** (*enum in ['EVEN', 'LENGTH'], (optional)*) – Edge Length Mode, Method to space UV edge loops

- **EVEN** Even, Space all UVs evenly.
- **LENGTH** Length, Average space UVs edge length of each loop.

**File** `startup/bl_operators/uvcalc_follow_active.py:246`

`bpy.ops.uv.hide` (*unselected=False*)  
Hide (un)selected UV vertices

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

`bpy.ops.uv.lightmap_pack` (*PREF\_CONTEXT='SEL\_FACES', PREF\_PACK\_IN\_ONE=True, PREF\_NEW\_UVLAYER=False, PREF\_APPLY\_IMAGE=False, PREF\_IMG\_PX\_SIZE=512, PREF\_BOX\_DIV=12, PREF\_MARGIN\_DIV=0.1*)  
Follow UVs from active quads along continuous face loops

#### Parameters

- **PREF\_CONTEXT** (*enum in ['SEL\_FACES', 'ALL\_FACES', 'ALL\_OBJECTS'], (optional)*) – Selection
  - **SEL\_FACES** Selected Faces, Space all UVs evenly.
  - **ALL\_FACES** All Faces, Average space UVs edge length of each loop.
  - **ALL\_OBJECTS** Selected Mesh Object, Average space UVs edge length of each loop.
- **PREF\_PACK\_IN\_ONE** (*boolean, (optional)*) – Share Tex Space, Objects Share texture space, map all objects into 1 uvmap
- **PREF\_NEW\_UVLAYER** (*boolean, (optional)*) – New UV Layer, Create a new UV layer for every mesh packed
- **PREF\_APPLY\_IMAGE** (*boolean, (optional)*) – New Image, Assign new images for every mesh (only one if shared tex space enabled)
- **PREF\_IMG\_PX\_SIZE** (*int in [64, 5000], (optional)*) – Image Size, Width and Height for the new image
- **PREF\_BOX\_DIV** (*int in [1, 48], (optional)*) – Pack Quality, Pre Packing before the complex boxpack
- **PREF\_MARGIN\_DIV** (*float in [0.001, 1], (optional)*) – Margin, Size of the margin as a division of the UV

**File** `startup/bl_operators/uvcalc_lightmap.py:599`

`bpy.ops.uv.minimize_stretch` (*fill\_holes=True, blend=0.0, iterations=0*)  
Reduce UV stretching by relaxing angles

#### Parameters

- **fill\_holes** (*boolean, (optional)*) – Fill Holes, Virtual fill holes in mesh before unwrapping, to better avoid overlaps and preserve symmetry
- **blend** (*float in [0, 1], (optional)*) – Blend, Blend factor between stretch minimized and original
- **iterations** (*int in [0, inf], (optional)*) – Iterations, Number of iterations to run, 0 is unlimited when run interactively

`bpy.ops.uv.pack_islands` (*margin=0.0*)

Undocumented ([contribute](#))

**Parameters** **margin** (*float in [0, 1], (optional)*) – Margin, Space between islands

`bpy.ops.uv.pin` (*clear=False*)

Set/clear selected UV vertices as anchored between multiple unwrap operations

**Parameters** **clear** (*boolean, (optional)*) – Clear, Clear pinning for the selection instead of setting it

`bpy.ops.uv.project_from_view` (*orthographic=False, correct\_aspect=True, clip\_to\_bounds=False, scale\_to\_bounds=False*)

Undocumented ([contribute](#))

#### Parameters

- **orthographic** (*boolean, (optional)*) – Orthographic, Use orthographic projection
- **correct\_aspect** (*boolean, (optional)*) – Correct Aspect, Map UVs taking image aspect ratio into account
- **clip\_to\_bounds** (*boolean, (optional)*) – Clip to Bounds, Clip UV coordinates to bounds after unwrapping
- **scale\_to\_bounds** (*boolean, (optional)*) – Scale to Bounds, Scale UV coordinates to bounds after unwrapping

`bpy.ops.uv.reset` ()

Undocumented ([contribute](#))

`bpy.ops.uv.reveal` ()

Reveal all hidden UV vertices

`bpy.ops.uv.select` (*extend=False, location=(0.0, 0.0)*)

Select UV vertices

#### Parameters

- **extend** (*boolean, (optional)*) – Extend, Extend selection rather than clearing the existing selection
- **location** (*float array of 2 items in [-inf, inf], (optional)*) – Location, Mouse location in normalized coordinates, 0.0 to 1.0 is within the image bounds

`bpy.ops.uv.select_all` (*action='TOGGLE'*)

Change selection of all UV vertices

**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.uv.select_border` (*pinned=False, gesture\_mode=0, xmin=0, xmax=0, ymin=0, ymax=0*)  
Select UV vertices using border selection

**Parameters**

- **pinned** (*boolean, (optional)*) – Pinned, Border select pinned UVs only
- **gesture\_mode** (*int in [-inf, inf], (optional)*) – Gesture Mode
- **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

`bpy.ops.uv.select_linked` (*extend=False*)  
Select all UV vertices linked to the active UV map

**Parameters** **extend** (*boolean, (optional)*) – Extend, Extend selection rather than clearing the existing selection

`bpy.ops.uv.select_linked_pick` (*extend=False, location=(0.0, 0.0)*)  
Select all UV vertices linked under the mouse

**Parameters**

- **extend** (*boolean, (optional)*) – Extend, Extend selection rather than clearing the existing selection
- **location** (*float array of 2 items in [-inf, inf], (optional)*) – Location, Mouse location in normalized coordinates, 0.0 to 1.0 is within the image bounds

`bpy.ops.uv.select_loop` (*extend=False, location=(0.0, 0.0)*)  
Select a loop of connected UV vertices

**Parameters**

- **extend** (*boolean, (optional)*) – Extend, Extend selection rather than clearing the existing selection
- **location** (*float array of 2 items in [-inf, inf], (optional)*) – Location, Mouse location in normalized coordinates, 0.0 to 1.0 is within the image bounds

`bpy.ops.uv.select_pinned` ()  
Select all pinned UV vertices

`bpy.ops.uv.smart_project` (*angle\_limit=66.0, island\_margin=0.0, user\_area\_weight=0.0*)  
This script projection unwraps the selected faces of a mesh (it operates on all selected mesh objects, and can be used to unwrap selected faces, or all faces)

**Parameters**

- **angle\_limit** (*float in [1, 89], (optional)*) – Angle Limit, lower for more projection groups, higher for less distortion
- **island\_margin** (*float in [0, 1], (optional)*) – Island Margin, Margin to reduce bleed from adjacent islands
- **user\_area\_weight** (*float in [0, 1], (optional)*) – Area Weight, Weight projections vector by faces with larger areas

**File** `startup/bl_operators/uvcalc_smart_project.py:1137`



`bpy.ops.uv.snap_cursor` (*target='PIXELS'*)  
Snap cursor to target type

**Parameters** **target** (*enum in ['PIXELS', 'SELECTED'], (optional)*) – Target, Target to snap the selected UVs to

`bpy.ops.uv.snap_selected` (*target='PIXELS'*)  
Snap selected UV vertices to target type

**Parameters** **target** (*enum in ['PIXELS', 'CURSOR', 'ADJACENT\_UNSELECTED'], (optional)*) – Target, Target to snap the selected UVs to

`bpy.ops.uv.sphere_project` (*direction='VIEW\_ON\_EQUATOR', align='POLAR\_ZX', correct\_aspect=True, clip\_to\_bounds=False, scale\_to\_bounds=False*)  
Undocumented ([contribute](#))

#### Parameters

- **direction** (*enum in ['VIEW\_ON\_EQUATOR', 'VIEW\_ON\_POLES', 'ALIGN\_TO\_OBJECT'], (optional)*) – Direction, Direction of the sphere or cylinder
  - VIEW\_ON\_EQUATOR View on Equator, 3D view is on the equator.
  - VIEW\_ON\_POLES View on Poles, 3D view is on the poles.
  - ALIGN\_TO\_OBJECT Align to Object, Align according to object transform.
- **align** (*enum in ['POLAR\_ZX', 'POLAR\_ZY'], (optional)*) – Align, How to determine rotation around the pole
  - POLAR\_ZX Polar ZX, Polar 0 is X.
  - POLAR\_ZY Polar ZY, Polar 0 is Y.
- **correct\_aspect** (*boolean, (optional)*) – Correct Aspect, Map UVs taking image aspect ratio into account
- **clip\_to\_bounds** (*boolean, (optional)*) – Clip to Bounds, Clip UV coordinates to bounds after unwrapping
- **scale\_to\_bounds** (*boolean, (optional)*) – Scale to Bounds, Scale UV coordinates to bounds after unwrapping

`bpy.ops.uv.stitch` (*use\_limit=True, limit=0.01*)  
Stitch selected UV vertices by proximity

#### Parameters

- **use\_limit** (*boolean, (optional)*) – Use Limit, Stitch UVs within a specified limit distance
- **limit** (*float in [0, inf], (optional)*) – Limit, Limit distance in normalized coordinates

`bpy.ops.uv.tile_set` (*tile=(0, 0)*)  
Set UV image tile coordinates

**Parameters** **tile** (*int array of 2 items in [0, inf], (optional)*) – Tile, Tile coordinate

`bpy.ops.uv.unlink_selected` ()  
Unlink selected UV vertices from active UV map

`bpy.ops.uv.unwrap` (*method='ANGLE\_BASED', fill\_holes=True, correct\_aspect=True*)  
Unwrap the mesh of the object being edited

#### Parameters

- **method** (*enum in ['ANGLE\_BASED', 'CONFORMAL'], (optional)*) – Method, Unwrapping method (Angle Based usually gives better results than Conformal, while being somewhat slower)
- **fill\_holes** (*boolean, (optional)*) – Fill Holes, Virtual fill holes in mesh before unwrapping, to better avoid overlaps and preserve symmetry
- **correct\_aspect** (*boolean, (optional)*) – Correct Aspect, Map UVs taking image aspect ratio into account

`bpy.ops.uv.weld()`  
Weld selected UV vertices together

## View2D Operators

`bpy.ops.view2d.pan(deltax=0, deltay=0)`  
Pan the view

### Parameters

- **deltax** (*int in [-inf, inf], (optional)*) – Delta X
- **deltay** (*int in [-inf, inf], (optional)*) – Delta Y

`bpy.ops.view2d.reset()`  
Reset the view

`bpy.ops.view2d.scroll_down(deltax=0, deltay=0, page=False)`  
Scroll the view down

### Parameters

- **deltax** (*int in [-inf, inf], (optional)*) – Delta X
- **deltay** (*int in [-inf, inf], (optional)*) – Delta Y
- **page** (*boolean, (optional)*) – Page, Scroll down one page

`bpy.ops.view2d.scroll_left(deltax=0, deltay=0)`  
Scroll the view left

### Parameters

- **deltax** (*int in [-inf, inf], (optional)*) – Delta X
- **deltay** (*int in [-inf, inf], (optional)*) – Delta Y

`bpy.ops.view2d.scroll_right(deltax=0, deltay=0)`  
Scroll the view right

### Parameters

- **deltax** (*int in [-inf, inf], (optional)*) – Delta X
- **deltay** (*int in [-inf, inf], (optional)*) – Delta Y

`bpy.ops.view2d.scroll_up(deltax=0, deltay=0, page=False)`  
Scroll the view up

### Parameters

- **deltax** (*int in [-inf, inf], (optional)*) – Delta X
- **deltay** (*int in [-inf, inf], (optional)*) – Delta Y
- **page** (*boolean, (optional)*) – Page, Scroll up one page

`bpy.ops.view2d.scroller_activate()`  
 Scroll view by mouse click and drag

`bpy.ops.view2d.zoom(deltax=0.0, deltax=0.0)`  
 Zoom in/out the view

#### Parameters

- **deltax** (*float in [-inf, inf], (optional)*) – Delta X
- **deltay** (*float in [-inf, inf], (optional)*) – Delta Y

`bpy.ops.view2d.zoom_border(gesture_mode=0, xmin=0, xmax=0, ymin=0, ymax=0)`  
 Zoom in the view to the nearest item contained in the border

#### Parameters

- **gesture\_mode** (*int in [-inf, inf], (optional)*) – Gesture Mode
- **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

`bpy.ops.view2d.zoom_in(zoomfacx=0.0, zoomfacy=0.0)`  
 Zoom in the view

#### Parameters

- **zoomfacx** (*float in [-inf, inf], (optional)*) – Zoom Factor X
- **zoomfacy** (*float in [-inf, inf], (optional)*) – Zoom Factor Y

`bpy.ops.view2d.zoom_out(zoomfacx=0.0, zoomfacy=0.0)`  
 Zoom out the view

#### Parameters

- **zoomfacx** (*float in [-inf, inf], (optional)*) – Zoom Factor X
- **zoomfacy** (*float in [-inf, inf], (optional)*) – Zoom Factor Y

## View3D Operators

`bpy.ops.view3d.background_image_add(name="Image", filepath="Path")`  
 Add a new background image

#### Parameters

- **name** (*string, (optional)*) – Name, Image name to assign
- **filepath** (*string, (optional)*) – Filepath, Path to image file

`bpy.ops.view3d.background_image_remove(index=0)`  
 Remove a background image from the 3D view

**Parameters** **index** (*int in [0, inf], (optional)*) – Index, Background image index to remove

`bpy.ops.view3d.camera_to_view()`  
 Set camera view to active view

`bpy.ops.view3d.clip_border(xmin=0, xmax=0, ymin=0, ymax=0)`  
 Set the view clipping border

**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

`bpy.ops.view3d.cursor3d()`  
Set the location of the 3D cursor

`bpy.ops.view3d.dolly(delta=0, mx=0, my=0)`  
Dolly in/out in the view

**Parameters**

- **delta** (*int in [-inf, inf], (optional)*) – Delta
- **mx** (*int in [0, inf], (optional)*) – Zoom Position X
- **my** (*int in [0, inf], (optional)*) – Zoom Position Y

`bpy.ops.view3d.edit_mesh_extrude_individual_move()`  
Extrude individual elements and move

**File** `startup/bl_operators/view3d.py:30`

`bpy.ops.view3d.edit_mesh_extrude_move_normal()`  
Extrude and move along normals

**File** `startup/bl_operators/view3d.py:63`

`bpy.ops.view3d.enable_manipulator(translate=False, rotate=False, scale=False)`  
Enable the transform manipulator for use

**Parameters**

- **translate** (*boolean, (optional)*) – Translate, Enable the translate manipulator
- **rotate** (*boolean, (optional)*) – Rotate, Enable the rotate manipulator
- **scale** (*boolean, (optional)*) – Scale, Enable the scale manipulator

`bpy.ops.view3d.fly()`  
Interactively fly around the scene

`bpy.ops.view3d.game_start()`  
Start game engine

`bpy.ops.view3d.layers(nr=1, extend=False, toggle=True)`  
Toggle layer(s) visibility

**Parameters**

- **nr** (*int in [0, 20], (optional)*) – Number, The layer number to set, zero for all layers
- **extend** (*boolean, (optional)*) – Extend, Add this layer to the current view layers
- **toggle** (*boolean, (optional)*) – Toggle, Toggle the layer

`bpy.ops.view3d.localview()`  
Toggle display of selected object(s) separately and centered in view

`bpy.ops.view3d.manipulator(constraint_axis=(False, False, False), constraint_orientation='', release_confirm=False)`  
Manipulate selected item by axis

**Parameters**

- **constraint\_axis** (*boolean array of 3 items, (optional)*) – Constraint Axis
- **constraint\_orientation** (*enum in [], (optional)*) – Orientation, Transformation orientation
- **release\_confirm** (*boolean, (optional)*) – Confirm on Release, Always confirm operation when releasing button

`bpy.ops.view3d.move()`  
Move the view

`bpy.ops.view3d.ndof_orbit()`  
Explore every angle of an object using the 3D mouse

`bpy.ops.view3d.ndof_pan()`  
Position your viewpoint with the 3D mouse

`bpy.ops.view3d.object_as_camera()`  
Set the active object as the active camera for this view or scene

`bpy.ops.view3d.properties()`  
Toggles the properties panel display

`bpy.ops.view3d.render_border` (*xmin=0, xmax=0, ymin=0, ymax=0*)  
Set the boundaries of the border render and enables border render

**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

`bpy.ops.view3d.rotate()`  
Rotate the view

`bpy.ops.view3d.select` (*extend=False, center=False, enumerate=False, object=False*)  
Activate/select item(s)

**Parameters**

- **extend** (*boolean, (optional)*) – Extend, Extend selection instead of deselecting everything first
- **center** (*boolean, (optional)*) – Center, Use the object center when selecting, in editmode used to extend object selection
- **enumerate** (*boolean, (optional)*) – Enumerate, List objects under the mouse (object mode only)
- **object** (*boolean, (optional)*) – Object, Use object selection (editmode only)

`bpy.ops.view3d.select_border` (*gesture\_mode=0, xmin=0, xmax=0, ymin=0, ymax=0, extend=True*)  
Select items using border selection

**Parameters**

- **gesture\_mode** (*int in [-inf, inf], (optional)*) – Gesture Mode
- **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
- **extend** (*boolean, (optional)*) – Extend, Extend selection instead of deselecting everything first

`bpy.ops.view3d.select_circle` (*x=0, y=0, radius=0, gesture\_mode=0*)  
Select items using circle selection

**Parameters**

- **x** (*int in [-inf, inf], (optional)*) – X
- **y** (*int in [-inf, inf], (optional)*) – Y
- **radius** (*int in [-inf, inf], (optional)*) – Radius
- **gesture\_mode** (*int in [-inf, inf], (optional)*) – Event Type

`bpy.ops.view3d.select_lasso` (*path=None, deselect=False, extend=True*)  
Select items using lasso selection

**Parameters**

- **path** (*bpy\_prop\_collection of OperatorMousePath, (optional)*) – Path
- **deselect** (*boolean, (optional)*) – Deselect, Deselect rather than select items
- **extend** (*boolean, (optional)*) – Extend, Extend selection instead of deselecting everything first

`bpy.ops.view3d.smoothview` ()  
The time to animate the change of view (in milliseconds)

`bpy.ops.view3d.snap_cursor_to_active` ()  
Snap cursor to active item

`bpy.ops.view3d.snap_cursor_to_center` ()  
Snap cursor to the Center

`bpy.ops.view3d.snap_cursor_to_grid` ()  
Snap cursor to nearest grid node

`bpy.ops.view3d.snap_cursor_to_selected` ()  
Snap cursor to center of selected item(s)

`bpy.ops.view3d.snap_selected_to_cursor` ()  
Snap selected item(s) to cursor

`bpy.ops.view3d.snap_selected_to_grid` ()  
Snap selected item(s) to nearest grid node

`bpy.ops.view3d.toolshelf` ()  
Toggles tool shelf display

`bpy.ops.view3d.view_all` (*center=False*)  
View all objects in scene

**Parameters** **center** (*boolean, (optional)*) – Center

`bpy.ops.view3d.view_center_camera` ()  
Center the camera view

`bpy.ops.view3d.view_center_cursor` ()  
Center the view so that the cursor is in the middle of the view

`bpy.ops.view3d.view_orbit` (*type='ORBITLEFT'*)

Orbit the view

**Parameters** *type* (*enum in ['ORBITLEFT', 'ORBITRIGHT', 'ORBITUP', 'ORBITDOWN'], (optional)*) – Orbit, Direction of View Orbit

- ORBITLEFT Orbit Left, Orbit the view around to the Left.
- ORBITRIGHT Orbit Right, Orbit the view around to the Right.
- ORBITUP Orbit Up, Orbit the view Up.
- ORBITDOWN Orbit Down, Orbit the view Down.

`bpy.ops.view3d.view_pan` (*type='PANLEFT'*)

Pan the view

**Parameters** *type* (*enum in ['PANLEFT', 'PANRIGHT', 'PANUP', 'PANDOWN'], (optional)*) – Pan, Direction of View Pan

- PANLEFT Pan Left, Pan the view to the Left.
- PANRIGHT Pan Right, Pan the view to the Right.
- PANUP Pan Up, Pan the view Up.
- PANDOWN Pan Down, Pan the view Down.

`bpy.ops.view3d.view_persportho` ()

Switch the current view from perspective/orthographic

`bpy.ops.view3d.view_selected` ()

Move the view to the selection center

`bpy.ops.view3d.viewnumpad` (*type='FRONT', align\_active=False*)

Set the view

**Parameters**

- *type* (*enum in ['FRONT', 'BACK', 'LEFT', 'RIGHT', 'TOP', 'BOTTOM', 'CAMERA'], (optional)*) – View, The Type of view
  - FRONT Front, View From the Front.
  - BACK Back, View From the Back.
  - LEFT Left, View From the Left.
  - RIGHT Right, View From the Right.
  - TOP Top, View From the Top.
  - BOTTOM Bottom, View From the Bottom.
  - CAMERA Camera, View From the active camera.
- *align\_active* (*boolean, (optional)*) – Align Active, Align to the active object's axis

`bpy.ops.view3d.zoom` (*delta=0, mx=0, my=0*)

Zoom in/out in the view

**Parameters**

- *delta* (*int in [-inf, inf], (optional)*) – Delta
- *mx* (*int in [0, inf], (optional)*) – Zoom Position X
- *my* (*int in [0, inf], (optional)*) – Zoom Position Y

`bpy.ops.view3d.zoom_border` (*xmin=0, xmax=0, ymin=0, ymax=0*)  
Zoom in the view to the nearest object contained in the border

**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

`bpy.ops.view3d.zoom_camera_1_to_1` ()  
Match the camera to 1:1 to the render output

## Wm Operators

`bpy.ops.wm.addon_disable` (*module=""*)  
Disable an addon

**Parameters** **module** (*string, (optional)*) – Module, Module name of the addon to disable

**File** `startup/bl_operators/wm.py:1492`

`bpy.ops.wm.addon_enable` (*module=""*)  
Enable an addon

**Parameters** **module** (*string, (optional)*) – Module, Module name of the addon to enable

**File** `startup/bl_operators/wm.py:1461`

`bpy.ops.wm.addon_expand` (*module=""*)  
Display more information on this addon

**Parameters** **module** (*string, (optional)*) – Module, Module name of the addon to expand

**File** `startup/bl_operators/wm.py:1734`

`bpy.ops.wm.addon_install` (*overwrite=True, target='DEFAULT', filepath="", filter\_folder=True, filter\_python=True, filter\_glob="\*.py;\*.zip"*)  
Install an addon

**Parameters**

- **overwrite** (*boolean, (optional)*) – Overwrite, Remove existing addons with the same ID
- **target** (*enum in ['DEFAULT', 'PREFS'], (optional)*) – Target Path
- **filepath** (*string, (optional)*) – File Path, File path to write file to
- **filter\_folder** (*boolean, (optional)*) – Filter folders
- **filter\_python** (*boolean, (optional)*) – Filter python

**File** `startup/bl_operators/wm.py:1548`

`bpy.ops.wm.addon_remove` (*module=""*)  
Disable an addon

**Parameters** **module** (*string, (optional)*) – Module, Module name of the addon to remove

**File** `startup/bl_operators/wm.py:1692`

`bpy.ops.wm.appconfig_activate` (*filepath=""*)  
Undocumented ([contribute](#))



**Parameters** `filepath` (*string, (optional)*) – File Path

**File** `startup/bl_operators/wm.py:1123`

`bpy.ops.wm.appconfig_default` ()  
Undocumented ([contribute](#))

**File** `startup/bl_operators/wm.py:1101`

`bpy.ops.wm.call_menu` (*name=""*)  
Undocumented ([contribute](#))

**Parameters** `name` (*string, (optional)*) – Name, Name of the menu

`bpy.ops.wm.context_collection_boolean_set` (*data\_path\_iter=""*, *data\_path\_item=""*,  
*type='TOGGLE'*)

Set boolean values for a collection of items

**Parameters**

- `data_path_iter` (*string, (optional)*) – The data path relative to the context, must point to an iterable
- `data_path_item` (*string, (optional)*) – The data path from each iterable to the value (int or float)
- `type` (*enum in ['TOGGLE', 'ENABLE', 'DISABLE'], (optional)*) – Type

**File** `startup/bl_operators/wm.py:593`

`bpy.ops.wm.context_cycle_array` (*data\_path=""*, *reverse=False*)

**Set a context array value.** Useful for cycling the active mesh edit mode

**Parameters**

- `data_path` (*string, (optional)*) – Context Attributes, rna context string
- `reverse` (*boolean, (optional)*) – Reverse, Cycle backwards

**File** `startup/bl_operators/wm.py:469`

`bpy.ops.wm.context_cycle_enum` (*data\_path=""*, *reverse=False*)  
Toggle a context value

**Parameters**

- `data_path` (*string, (optional)*) – Context Attributes, rna context string
- `reverse` (*boolean, (optional)*) – Reverse, Cycle backwards

**File** `startup/bl_operators/wm.py:416`

`bpy.ops.wm.context_cycle_int` (*data\_path=""*, *reverse=False*)  
Set a context value. Useful for cycling active material,

**Parameters**

- `data_path` (*string, (optional)*) – Context Attributes, rna context string
- `reverse` (*boolean, (optional)*) – Reverse, Cycle backwards

**File** `startup/bl_operators/wm.py:382`

`bpy.ops.wm.context_menu_enum` (*data\_path=""*)  
Undocumented ([contribute](#))

**Parameters** `data_path` (*string, (optional)*) – Context Attributes, rna context string

**File** `startup/bl_operators/wm.py:513`

`bpy.ops.wm.context_modal_mouse` (*data\_path\_iter=""*, *data\_path\_item=""*, *input\_scale=0.01*, *invert=False*, *initial\_x=0*)

Adjust arbitrary values with mouse input

#### Parameters

- **data\_path\_iter** (*string, (optional)*) – The data path relative to the context, must point to an iterable
- **data\_path\_item** (*string, (optional)*) – The data path from each iterable to the value (int or float)
- **input\_scale** (*float in [-inf, inf], (optional)*) – Scale the mouse movement by this value before applying the delta
- **invert** (*boolean, (optional)*) – Invert the mouse input

**File** `startup/bl_operators/wm.py:714`

`bpy.ops.wm.context_scale_int` (*data\_path=""*, *value=1.0*, *always\_step=True*)

Scale an int context value

#### Parameters

- **data\_path** (*string, (optional)*) – Context Attributes, rna context string
- **value** (*float in [-inf, inf], (optional)*) – Value, Assign value
- **always\_step** (*boolean, (optional)*) – Always Step, Always adjust the value by a minimum of 1 when 'value' is not 1.0

**File** `startup/bl_operators/wm.py:225`

`bpy.ops.wm.context_set_boolean` (*data\_path=""*, *value=True*)

Set a context value

#### Parameters

- **data\_path** (*string, (optional)*) – Context Attributes, rna context string
- **value** (*boolean, (optional)*) – Value, Assignment value

**File** `startup/bl_operators/wm.py:127`

`bpy.ops.wm.context_set_enum` (*data\_path=""*, *value=""*)

Set a context value

#### Parameters

- **data\_path** (*string, (optional)*) – Context Attributes, rna context string
- **value** (*string, (optional)*) – Value, Assignment value (as a string)

**File** `startup/bl_operators/wm.py:127`

`bpy.ops.wm.context_set_float` (*data\_path=""*, *value=0.0*, *relative=False*)

Set a context value

#### Parameters

- **data\_path** (*string, (optional)*) – Context Attributes, rna context string
- **value** (*float in [-inf, inf], (optional)*) – Value, Assignment value
- **relative** (*boolean, (optional)*) – Relative, Apply relative to the current value (delta)

**File** `startup/bl_operators/wm.py:127`

`bpy.ops.wm.context_set_id` (*data\_path=""*, *value=""*)  
Toggle a context value

**Parameters**

- **data\_path** (*string, (optional)*) – Context Attributes, rna context string
- **value** (*string, (optional)*) – Value, Assign value

**File** `startup/bl_operators/wm.py:533`

`bpy.ops.wm.context_set_int` (*data\_path=""*, *value=0*, *relative=False*)  
Set a context value

**Parameters**

- **data\_path** (*string, (optional)*) – Context Attributes, rna context string
- **value** (*int in [-inf, inf], (optional)*) – Value, Assign value
- **relative** (*boolean, (optional)*) – Relative, Apply relative to the current value (delta)

**File** `startup/bl_operators/wm.py:127`

`bpy.ops.wm.context_set_string` (*data\_path=""*, *value=""*)  
Set a context value

**Parameters**

- **data\_path** (*string, (optional)*) – Context Attributes, rna context string
- **value** (*string, (optional)*) – Value, Assign value

**File** `startup/bl_operators/wm.py:127`

`bpy.ops.wm.context_set_value` (*data\_path=""*, *value=""*)  
Set a context value

**Parameters**

- **data\_path** (*string, (optional)*) – Context Attributes, rna context string
- **value** (*string, (optional)*) – Value, Assignment value (as a string)

**File** `startup/bl_operators/wm.py:312`

`bpy.ops.wm.context_toggle` (*data\_path=""*)  
Toggle a context value

**Parameters** **data\_path** (*string, (optional)*) – Context Attributes, rna context string

**File** `startup/bl_operators/wm.py:328`

`bpy.ops.wm.context_toggle_enum` (*data\_path=""*, *value\_1=""*, *value\_2=""*)  
Toggle a context value

**Parameters**

- **data\_path** (*string, (optional)*) – Context Attributes, rna context string
- **value\_1** (*string, (optional)*) – Value, Toggle enum
- **value\_2** (*string, (optional)*) – Value, Toggle enum

**File** `startup/bl_operators/wm.py:357`

`bpy.ops.wm.copy_prev_settings` ()  
Copy settings from previous version

**File** `startup/bl_operators/wm.py:1151`

`bpy.ops.wm.debug_menu` (*debug\_value=0*)  
Open a popup to set the debug level

**Parameters** `debug_value` (*int in [-10000, 10000], (optional)*) – Debug Value

`bpy.ops.wm.doc_edit` (*doc\_id=""*, *doc\_new=""*)  
Load online reference docs

**Parameters**

- `doc_id` (*string, (optional)*) – Doc ID
- `doc_new` (*string, (optional)*) – Edit Description

**File** `startup/bl_operators/wm.py:856`

`bpy.ops.wm.doc_view` (*doc\_id=""*)  
Load online reference docs

**Parameters** `doc_id` (*string, (optional)*) – Doc ID

**File** `startup/bl_operators/wm.py:803`

`bpy.ops.wm.interaction_preset_add` (*name=""*, *remove\_active=False*)  
Add an Application Interaction Preset

**Parameters** `name` (*string, (optional)*) – Name, Name of the preset, used to make the path name

**File** `startup/bl_operators/presets.py:50`

`bpy.ops.wm.keyconfig_activate` (*filepath=""*)  
Undocumented ([contribute](#))

**Parameters** `filepath` (*string, (optional)*) – File Path

**File** `startup/bl_operators/wm.py:1092`

`bpy.ops.wm.keyconfig_export` (*filepath="keymap.py"*, *filter\_folder=True*, *filter\_text=True*, *filter\_python=True*)  
Export key configuration to a python script

**Parameters**

- `filepath` (*string, (optional)*) – File Path, Filepath to write file to
- `filter_folder` (*boolean, (optional)*) – Filter folders
- `filter_text` (*boolean, (optional)*) – Filter text
- `filter_python` (*boolean, (optional)*) – Filter python

**File** `startup/bl_operators/wm.py:1293`

`bpy.ops.wm.keyconfig_import` (*filepath="keymap.py"*, *filter\_folder=True*, *filter\_text=True*, *filter\_python=True*, *keep\_original=True*)  
Import key configuration from a python script

**Parameters**

- `filepath` (*string, (optional)*) – File Path, Filepath to write file to
- `filter_folder` (*boolean, (optional)*) – Filter folders
- `filter_text` (*boolean, (optional)*) – Filter text
- `filter_python` (*boolean, (optional)*) – Filter python

- **keep\_original** (*boolean, (optional)*) – Keep original, Keep original file after copying to configuration folder

**File** `startup/bl_operators/wm.py:1231`

`bpy.ops.wm.keyconfig_preset_add` (*name=""*, *remove\_active=False*)  
Add a Keyconfig Preset

**Parameters** **name** (*string, (optional)*) – Name, Name of the preset, used to make the path name

**File** `startup/bl_operators/presets.py:50`

`bpy.ops.wm.keyconfig_remove` ()  
Remove key config

**File** `startup/bl_operators/wm.py:1419`

`bpy.ops.wm.keyconfig_test` ()  
Test keyconfig for conflicts

**File** `startup/bl_operators/wm.py:1188`

`bpy.ops.wm.keyitem_add` ()  
Add key map item

**File** `startup/bl_operators/wm.py:1370`

`bpy.ops.wm.keyitem_remove` (*item\_id=0*)  
Remove key map item

**Parameters** **item\_id** (*int in [-inf, inf], (optional)*) – Item Identifier, Identifier of the item to remove

**File** `startup/bl_operators/wm.py:1401`

`bpy.ops.wm.keyitem_restore` (*item\_id=0*)  
Restore key map item

**Parameters** **item\_id** (*int in [-inf, inf], (optional)*) – Item Identifier, Identifier of the item to remove

**File** `startup/bl_operators/wm.py:1355`

`bpy.ops.wm.keymap_restore` (*all=False*)  
Restore key map(s)

**Parameters** **all** (*boolean, (optional)*) – All Keymaps, Restore all keymaps to default

**File** `startup/bl_operators/wm.py:1327`

`bpy.ops.wm.link_append` (*filepath=""*, *directory=""*, *filename=""*, *files=None*, *filter\_blender=True*,  
*filter\_image=False*, *filter\_movie=False*, *filter\_python=False*, *filter\_font=False*, *filter\_sound=False*, *filter\_text=False*, *filter\_btx=False*,  
*filter\_collada=False*, *filter\_folder=True*, *filemode=1*, *relative\_path=True*,  
*link=True*, *autoselect=True*, *active\_layer=True*, *instance\_groups=True*)

Link or Append from a Library .blend file

#### Parameters

- **filepath** (*string, (optional)*) – File Path, Path to file
- **directory** (*string, (optional)*) – Directory, Directory of the file
- **filename** (*string, (optional)*) – File Name, Name of the file
- **files** (*bpy\_prop\_collection of OperatorFileListElement, (optional)*) – Files
- **filter\_blender** (*boolean, (optional)*) – Filter .blend files
- **filter\_image** (*boolean, (optional)*) – Filter image files

- **filter\_movie** (*boolean, (optional)*) – Filter movie files
- **filter\_python** (*boolean, (optional)*) – Filter python files
- **filter\_font** (*boolean, (optional)*) – Filter font files
- **filter\_sound** (*boolean, (optional)*) – Filter sound files
- **filter\_text** (*boolean, (optional)*) – Filter text files
- **filter\_btx** (*boolean, (optional)*) – Filter btx files
- **filter\_collada** (*boolean, (optional)*) – Filter COLLADA files
- **filter\_folder** (*boolean, (optional)*) – Filter folders
- **filemode** (*int in [1, 9], (optional)*) – File Browser Mode, The setting for the file browser mode to load a .blend file, a library or a special file
- **relative\_path** (*boolean, (optional)*) – Relative Path, Select the file relative to the blend file
- **link** (*boolean, (optional)*) – Link, Link the objects or datablocks rather than appending
- **autoselect** (*boolean, (optional)*) – Select, Select the linked objects
- **active\_layer** (*boolean, (optional)*) – Active Layer, Put the linked objects on the active layer
- **instance\_groups** (*boolean, (optional)*) – Instance Groups, Create instances for each group as a DupliGroup

`bpy.ops.wm.memory_statistics()`  
Print memory statistics to the console

`bpy.ops.wm.ndof_sensitivity_change(decrease=True, fast=False)`  
Change NDOF sensitivity

#### Parameters

- **decrease** (*boolean, (optional)*) – Decrease NDOF sensitivity, If true then action decreases NDOF sensitivity instead of increasing
- **fast** (*boolean, (optional)*) – Fast NDOF sensitivity change, If true then sensitivity changes 50%, otherwise 10%

`bpy.ops.wm.open_mainfile(filepath="", filter_blender=True, filter_image=False, filter_movie=False, filter_python=False, filter_font=False, filter_sound=False, filter_text=False, filter_btx=False, filter_collada=False, filter_folder=True, filemode=8, load_ui=True, use_scripts=True)`

Open a Blender file

#### Parameters

- **filepath** (*string, (optional)*) – File Path, Path to file
- **filter\_blender** (*boolean, (optional)*) – Filter .blend files
- **filter\_image** (*boolean, (optional)*) – Filter image files
- **filter\_movie** (*boolean, (optional)*) – Filter movie files
- **filter\_python** (*boolean, (optional)*) – Filter python files
- **filter\_font** (*boolean, (optional)*) – Filter font files
- **filter\_sound** (*boolean, (optional)*) – Filter sound files
- **filter\_text** (*boolean, (optional)*) – Filter text files

- **filter\_btx** (*boolean, (optional)*) – Filter btx files
- **filter\_collada** (*boolean, (optional)*) – Filter COLLADA files
- **filter\_folder** (*boolean, (optional)*) – Filter folders
- **filemode** (*int in [1, 9], (optional)*) – File Browser Mode, The setting for the file browser mode to load a .blend file, a library or a special file
- **load\_ui** (*boolean, (optional)*) – Load UI, Load user interface setup in the .blend file
- **use\_scripts** (*boolean, (optional)*) – Trusted Source, Allow blend file execute scripts automatically, default available from system preferences

`bpy.ops.wm.operator_cheat_sheet` ()  
Undocumented ([contribute](#))

**File** [startup/bl\\_operators/wm.py:1430](#)

`bpy.ops.wm.operator_preset_add` (*name=""*, *remove\_active=False*, *operator=""*)  
Add an Application Interaction Preset

#### Parameters

- **name** (*string, (optional)*) – Name, Name of the preset, used to make the path name
- **operator** (*string, (optional)*) – Operator

**File** [startup/bl\\_operators/presets.py:50](#)

`bpy.ops.wm.path_open` (*filepath=""*)  
Open a path in a file browser

**Parameters** **filepath** (*string, (optional)*) – File Path

**File** [startup/bl\\_operators/wm.py:756](#)

`bpy.ops.wm.properties_add` (*data\_path=""*)  
Internal use (edit a property data\_path)

**Parameters** **data\_path** (*string, (optional)*) – Property Edit, Property data\_path edit

**File** [startup/bl\\_operators/wm.py:1033](#)

`bpy.ops.wm.properties_context_change` (*context=""*)  
Change the context tab in a Properties Window

**Parameters** **context** (*string, (optional)*) – Context

**File** [startup/bl\\_operators/wm.py:1063](#)

`bpy.ops.wm.properties_edit` (*data\_path=""*, *property=""*, *value=""*, *min=0.0*, *max=1.0*, *description=""*)  
Internal use (edit a property data\_path)

#### Parameters

- **data\_path** (*string, (optional)*) – Property Edit, Property data\_path edit
- **property** (*string, (optional)*) – Property Name, Property name edit
- **value** (*string, (optional)*) – Property Value, Property value edit
- **min** (*float in [-inf, inf], (optional)*) – Min
- **max** (*float in [-inf, inf], (optional)*) – Max
- **description** (*string, (optional)*) – Tip

**File** `startup/bl_operators/wm.py:954`

`bpy.ops.wm.properties_remove` (`data_path=""`, `property=""`)  
Internal use (edit a property `data_path`)

#### Parameters

- **data\_path** (*string, (optional)*) – Property Edit, Property `data_path` edit
- **property** (*string, (optional)*) – Property Name, Property name edit

**File** `startup/bl_operators/wm.py:1076`

`bpy.ops.wm.quit_blender` ()  
Quit Blender

`bpy.ops.wm.radial_control` (`data_path=""`, `rotation_path=""`, `color_path=""`, `fill_color_path=""`,  
`zoom_path=""`, `image_id=""`)  
Undocumented ([contribute](#))

#### Parameters

- **data\_path** (*string, (optional)*) – Data Path, Path of property to be set by the radial control
- **rotation\_path** (*string, (optional)*) – Rotation Path, Path of property used to rotate the texture display
- **color\_path** (*string, (optional)*) – Color Path, Path of property used to set the color of the control
- **fill\_color\_path** (*string, (optional)*) – Fill Color Path, Path of property used to set the fill color of the control
- **zoom\_path** (*string, (optional)*) – Zoom Path, Path of property used to set the zoom level for the control
- **image\_id** (*string, (optional)*) – Image ID, Path of ID that is used to generate an image for the control

`bpy.ops.wm.read_factory_settings` ()  
Load default file and user preferences

`bpy.ops.wm.read_homefile` ()  
Open the default file (doesn't save the current file)

`bpy.ops.wm.recover_auto_save` (`filepath=""`, `filter_blender=True`, `filter_image=False`, `filter_movie=False`,  
`filter_python=False`, `filter_font=False`, `filter_sound=False`, `filter_text=False`, `filter_btx=False`, `filter_collada=False`,  
`filter_folder=False`, `filemode=8`)  
Open an automatically saved file to recover it

#### Parameters

- **filepath** (*string, (optional)*) – File Path, Path to file
- **filter\_blender** (*boolean, (optional)*) – Filter .blend files
- **filter\_image** (*boolean, (optional)*) – Filter image files
- **filter\_movie** (*boolean, (optional)*) – Filter movie files
- **filter\_python** (*boolean, (optional)*) – Filter python files
- **filter\_font** (*boolean, (optional)*) – Filter font files
- **filter\_sound** (*boolean, (optional)*) – Filter sound files



- **filter\_text** (*boolean, (optional)*) – Filter text files
- **filter\_btx** (*boolean, (optional)*) – Filter btx files
- **filter\_collada** (*boolean, (optional)*) – Filter COLLADA files
- **filter\_folder** (*boolean, (optional)*) – Filter folders
- **filemode** (*int in [1, 9], (optional)*) – File Browser Mode, The setting for the file browser mode to load a .blend file, a library or a special file

`bpy.ops.wm.recover_last_session()`  
Open the last closed file (“quit.blend”)

`bpy.ops.wm.redraw_timer` (*type='DRAW', iterations=10*)  
Simple redraw timer to test the speed of updating the interface

#### Parameters

- **type** (*enum in ['DRAW', 'DRAW\_SWAP', 'DRAW\_WIN', 'DRAW\_WIN\_SWAP', 'ANIM\_STEP', 'ANIM\_PLAY', 'UNDO'], (optional)*) – Type
  - DRAW Draw Region, Draw Region.
  - DRAW\_SWAP Draw Region + Swap, Draw Region and Swap.
  - DRAW\_WIN Draw Window, Draw Window.
  - DRAW\_WIN\_SWAP Draw Window + Swap, Draw Window and Swap.
  - ANIM\_STEP Anim Step, Animation Steps.
  - ANIM\_PLAY Anim Play, Animation Playback.
  - UNDO Undo/Redo, Undo/Redo.
- **iterations** (*int in [1, inf], (optional)*) – Iterations, Number of times to redraw

`bpy.ops.wm.save_as_mainfile` (*filepath=""*, *check\_existing=True*, *filter\_blender=True*, *filter\_image=False*, *filter\_movie=False*, *filter\_python=False*, *filter\_font=False*, *filter\_sound=False*, *filter\_text=False*, *filter\_btx=False*, *filter\_collada=False*, *filter\_folder=True*, *filemode=8*, *compress=False*, *relative\_remap=True*, *copy=False*)

Save the current file in the desired location

#### Parameters

- **filepath** (*string, (optional)*) – File Path, Path to file
- **check\_existing** (*boolean, (optional)*) – Check Existing, Check and warn on overwriting existing files
- **filter\_blender** (*boolean, (optional)*) – Filter .blend files
- **filter\_image** (*boolean, (optional)*) – Filter image files
- **filter\_movie** (*boolean, (optional)*) – Filter movie files
- **filter\_python** (*boolean, (optional)*) – Filter python files
- **filter\_font** (*boolean, (optional)*) – Filter font files
- **filter\_sound** (*boolean, (optional)*) – Filter sound files
- **filter\_text** (*boolean, (optional)*) – Filter text files
- **filter\_btx** (*boolean, (optional)*) – Filter btx files
- **filter\_collada** (*boolean, (optional)*) – Filter COLLADA files

- **filter\_folder** (*boolean, (optional)*) – Filter folders
- **filemode** (*int in [1, 9], (optional)*) – File Browser Mode, The setting for the file browser mode to load a .blend file, a library or a special file
- **compress** (*boolean, (optional)*) – Compress, Write compressed .blend file
- **relative\_remap** (*boolean, (optional)*) – Remap Relative, Remap relative paths when saving in a different directory
- **copy** (*boolean, (optional)*) – Save Copy, Save a copy of the actual working state but does not make saved file active

`bpy.ops.wm.save_homefile()`

Make the current file the default .blend file

`bpy.ops.wm.save_mainfile(filepath=""`, `check_existing=True`, `filter_blender=True`, `filter_image=False`, `filter_movie=False`, `filter_python=False`, `filter_font=False`, `filter_sound=False`, `filter_text=False`, `filter_btx=False`, `filter_collada=False`, `filter_folder=True`, `filemode=8`, `compress=False`, `relative_remap=False`)

Save the current Blender file

#### Parameters

- **filepath** (*string, (optional)*) – File Path, Path to file
- **check\_existing** (*boolean, (optional)*) – Check Existing, Check and warn on overwriting existing files
- **filter\_blender** (*boolean, (optional)*) – Filter .blend files
- **filter\_image** (*boolean, (optional)*) – Filter image files
- **filter\_movie** (*boolean, (optional)*) – Filter movie files
- **filter\_python** (*boolean, (optional)*) – Filter python files
- **filter\_font** (*boolean, (optional)*) – Filter font files
- **filter\_sound** (*boolean, (optional)*) – Filter sound files
- **filter\_text** (*boolean, (optional)*) – Filter text files
- **filter\_btx** (*boolean, (optional)*) – Filter btx files
- **filter\_collada** (*boolean, (optional)*) – Filter COLLADA files
- **filter\_folder** (*boolean, (optional)*) – Filter folders
- **filemode** (*int in [1, 9], (optional)*) – File Browser Mode, The setting for the file browser mode to load a .blend file, a library or a special file
- **compress** (*boolean, (optional)*) – Compress, Write compressed .blend file
- **relative\_remap** (*boolean, (optional)*) – Remap Relative, Remap relative paths when saving in a different directory

`bpy.ops.wm.search_menu()`

Undocumented ([contribute](#))

`bpy.ops.wm.splash()`

Opens a blocking popup region with release info

`bpy.ops.wm.sysinfo()`

Generate System Info

**File** startup/bl\_operators/wm.py:1140

`bpy.ops.wm.url_open(url="")`  
Open a website in the Webbrowser

**Parameters** `url` (*string, (optional)*) – URL, URL to open

**File** startup/bl\_operators/wm.py:739

`bpy.ops.wm.window_duplicate()`  
Duplicate the current Blender window

`bpy.ops.wm.window_fullscreen_toggle()`  
Toggle the current window fullscreen

## World Operators

`bpy.ops.world.new()`  
Add a new world

## 2.4 Types (bpy.types)

### 2.4.1 Action(ID)

base classes — `bpy_struct`, `ID`

**class** `bpy.types.Action` (*ID*)  
A collection of F-Curves for animation

**fcurves**

The individual F-Curves that make up the Action

**Type** `ActionFCurves` `bpy_prop_collection` of `FCurve`, (readonly)

**frame\_range**

The final frame range of all F-Curves within this action

**Type** float array of 2 items in `[-inf, inf]`, default `(0.0, 0.0)`, (readonly)

**groups**

Convenient groupings of F-Curves

**Type** `ActionGroups` `bpy_prop_collection` of `ActionGroup`, (readonly)

**id\_root**

Type of ID-block that action can be used on - DO NOT CHANGE UNLESS YOU KNOW WHAT YOU'RE DOING

**Type** enum in `['ACTION', 'ARMATURE', 'BRUSH', 'CAMERA', 'CURVE', 'FONT', 'GREASEPENCIL', 'GROUP', 'IMAGE', 'KEY', 'LAMP', 'LIBRARY', 'LATTICE', 'MATERIAL', 'META', 'MESH', 'NODETREE', 'OBJECT', 'PARTICLE', 'SCENE', 'SCREEN', 'SPEAKER', 'SOUND', 'TEXT', 'TEXTURE', 'WORLD', 'WINDOWMANAGER']`, default `'ACTION'`

**pose\_markers**

Markers specific to this Action, for labeling poses

**Type** `ActionPoseMarkers` `bpy_prop_collection` of `TimelineMarker`, (readonly)

### Inherited Properties

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`

### References

- `ActionActuator.action`
- `ActionConstraint.action`
- `AnimData.action`
- `BlendData.actions`
- `BlendDataActions.new`
- `BlendDataActions.remove`
- `NlaStrip.action`
- `NlaStrips.new`
- `Object.pose_library`
- `ShapeActionActuator.action`
- `SpaceDopeSheetEditor.action`

## 2.4.2 ActionActuator(Actuator)

base classes — `bpy_struct`, `Actuator`

**class** `bpy.types.ActionActuator` (*Actuator*)

Actuator to control the object movement

**action**

**Type** `Action`

**apply\_to\_children**

Update Action on all children Objects as well

**Type** `boolean`, default `False`

**frame\_blend\_in**

Number of frames of motion blending

**Type** `int` in `[0, 32767]`, default `0`

**frame\_end**

**Type** `float` in `[-inf, inf]`, default `0.0`

**frame\_property**

Assign the action's current frame number to this property

**Type** `string`, default `""`

**frame\_start**

**Type** `float` in `[-inf, inf]`, default `0.0`

**layer**

The animation layer to play the action on

**Type** `int` in `[0, 7]`, default `0`

**layer\_weight**

How much of the previous layer to blend into this one (0 = add mode)

**Type** `float` in `[0, 1]`, default `0.0`

**play\_mode**

Action playback type

**Type** `enum` in `['PLAY', 'PINGPONG', 'FLIPPER', 'LOOPSTOP', 'LOOPEND', 'PROPERTY']`, default `'PLAY'`

**priority**

Execution priority - lower numbers will override actions with higher numbers. With 2 or more actions at once, the overriding channels must be lower in the stack

**Type** `int` in `[0, 100]`, default `0`

**property**

Use this property to define the Action position

**Type** `string`, default `""`

**use\_additive**

Action is added to the current loc/rot/scale in global or local coordinate according to Local flag

**Type** `boolean`, default `False`

**use\_continue\_last\_frame**

Restore last frame when switching on/off, otherwise play from the start each time

**Type** boolean, default False

**use\_force**

Apply Action as a global or local force depending on the local option (dynamic objects only)

**Type** boolean, default False

**use\_local**

Let the Action act in local coordinates, used in Force and Add mode

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `Actuator.name`
- `Actuator.show_expanded`
- `Actuator.pin`
- `Actuator.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Actuator.link`
- `Actuator.unlink`

## 2.4.3 ActionConstraint(Constraint)

base classes — `bpy_struct`, `Constraint`

**class** `bpy.types.ActionConstraint` (*Constraint*)

Map an action to the transform axes of a bone

**action**

The constraining action

**Type** `Action`

**frame\_end**

Last frame of the Action to use

**Type** int in [-300000, 300000], default 0

**frame\_start**

First frame of the Action to use

**Type** int in [-300000, 300000], default 0

**max**

Maximum value for target channel range

**Type** float in [-1000, 1000], default 0.0

**min**

Minimum value for target channel range

**Type** float in [-1000, 1000], default 0.0

**subtarget**

**Type** string, default ""

**target**

Target Object

**Type** Object

**transform\_channel**

Transformation channel from the target that is used to key the Action

**Type** enum in ['LOCATION\_X', 'LOCATION\_Y', 'LOCATION\_Z', 'ROTATION\_X', 'ROTATION\_Y', 'ROTATION\_Z', 'SCALE\_X', 'SCALE\_Y', 'SCALE\_Z'], default 'ROTATION\_X'

**Inherited Properties**

- bpy\_struct.id\_data
- Constraint.name
- Constraint.active
- Constraint.mute
- Constraint.show\_expanded
- Constraint.influence
- Constraint.error\_location
- Constraint.owner\_space
- Constraint.is\_proxy\_local
- Constraint.error\_rotation
- Constraint.target\_space
- Constraint.type
- Constraint.is\_valid

**Inherited Functions**

- bpy\_struct.as\_pointer
- bpy\_struct.callback\_add
- bpy\_struct.callback\_remove
- bpy\_struct.driver\_add
- bpy\_struct.driver\_remove

- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.4 ActionFCurves(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.ActionFCurves` (*bpy\_struct*)

Collection of action F-Curves

**new** (*data\_path*, *index=0*, *action\_group=""*)

Add a keyframe to the F-Curve

### Parameters

- **data\_path** (*string*) – Data Path, F-Curve data path to use
- **index** (*int in [0, inf], (optional)*) – Index, Array index
- **action\_group** (*string, (optional)*) – Action Group, Action group to add this F-Curve into

**Returns** Newly created F-Curve

**Return type** `FCurve`

**remove** (*fcurve*)

Remove action group

**Parameters** **fcurve** (`FCurve`, (never None)) – F-Curve to remove

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`



- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Action.fcurves`

## 2.4.5 ActionGroup(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.ActionGroup` (*bpy\_struct*)  
Groups of F-Curves

### **channels**

F-Curves in this group

**Type** `bpy_prop_collection` of `FCurve`, (readonly)

### **custom\_color**

Index of custom color set

**Type** `int` in `[-inf, inf]`, default 0

### **lock**

Action Group is locked

**Type** `boolean`, default `False`

### **name**

**Type** `string`, default ""

### **select**

Action Group is selected

**Type** `boolean`, default `False`

### **show\_expanded**

Action Group is expanded

**Type** `boolean`, default `False`

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`

- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Action.groups`
- `ActionGroups.new`
- `ActionGroups.remove`
- `FCurve.group`

## 2.4.6 ActionGroups(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.ActionGroups` (*bpy\_struct*)

Collection of action groups

**new** (*name*)

Add a keyframe to the curve

**Parameters** `name` (*string*) – New name for the action group

**Returns** Newly created action group

**Return type** `ActionGroup`

**remove** (*action\_group*)

Remove action group

**Parameters** `action_group` (`ActionGroup`, (never `None`)) – Action group to remove

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`

- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Action.groups`

## 2.4.7 ActionPoseMarkers(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.ActionPoseMarkers` (*bpy\_struct*)

Collection of timeline markers

### **active**

Active pose marker for this Action

**Type** `TimelineMarker`

### **active\_index**

Index of active pose marker

**Type** `int` in `[-inf, inf]`, default 0

### **new** (*name*)

Add a pose marker to the action

**Parameters** **name** (*string*) – New name for the marker (not unique)

**Returns** Newly created marker

**Return type** `TimelineMarker`

### **remove** (*marker*)

Remove a timeline marker

**Parameters** **marker** (`TimelineMarker`, (never `None`)) – Timeline marker to remove

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`

- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Action.pose_markers`

## 2.4.8 Actuator(`bpy_struct`)

base class — `bpy_struct`

subclasses — `ShapeActionActuator`, `VisibilityActuator`, `MessageActuator`, `RandomActuator`, `Filter2DActuator`, `GameActuator`, `CameraActuator`, `SteeringActuator`, `ArmatureActuator`, `SoundActuator`, `ParentActuator`, `SceneActuator`, `StateActuator`, `ActionActuator`, `ConstraintActuator`, `PropertyActuator`, `ObjectActuator`, `EditObjectActuator`

**class** `bpy.types.Actuator` (*`bpy_struct`*)

Actuator to apply actions in the game engine

**name**

**Type** string, default ""

**pin**

Display when not linked to a visible states controller

**Type** boolean, default False

**show\_expanded**

Set actuator expanded in the user interface

**Type** boolean, default False

**type**

**Type** enum in ['ACTION', 'ARMATURE', 'CAMERA', 'CONSTRAINT', 'EDIT\_OBJECT', 'FILTER\_2D', 'GAME', 'MESSAGE', 'MOTION', 'PARENT', 'PROPERTY', 'RANDOM', 'SCENE', 'SOUND', 'STATE', 'VISIBILITY', 'STEERING'], default 'MOTION'

**link** (*`controller`*)

Link the actuator to a controller

**Parameters** `controller` (`Controller`) – Controller to link to

**unlink** (*`controller`*)

Unlink the actuator from a controller

**Parameters** `controller` (`Controller`) – Controller to unlink from

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Controller.link`
- `Controller.unlink`
- `GameObjectSettings.actuators`

## 2.4.9 ActuatorSensor(Sensor)

base classes — `bpy_struct`, `Sensor`

**class** `bpy.types.ActuatorSensor` (*Sensor*)  
Sensor to detect state modifications of actuators

**actuator**

Actuator name, actuator active state modifications will be detected

**Type** string, default ""

### Inherited Properties

- `bpy_struct.id_data`
- `Sensor.name`
- `Sensor.show_expanded`
- `Sensor.frequency`
- `Sensor.invert`
- `Sensor.use_level`
- `Sensor.pin`
- `Sensor.use_pulse_false_level`
- `Sensor.use_pulse_true_level`

- `Sensor.use_tap`
- `Sensor.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Sensor.link`
- `Sensor.unlink`

## 2.4.10 Addon(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.Addon` (*bpy\_struct*)  
Python addons to be loaded automatically

**module**

Module name

**Type** string, default ""

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`

- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Addons.new`
- `Addons.remove`
- `UserPreferences.addons`

### 2.4.11 Addons(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.Addons` (*bpy\_struct*)

Collection of addons

**classmethod** `new` ()

Add a new addon

**Returns** Addon datablock

**Return type** `Addon`

**classmethod** `remove` (*addon*)

Remove addon

**Parameters** `addon` (`Addon`, (never `None`)) – Addon to remove

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`

- `bpy_struct.values`

## References

- `UserPreferences.addons`

### 2.4.12 AdjustmentSequence(Sequence)

base classes — `bpy_struct`, `Sequence`

**class** `bpy.types.AdjustmentSequence` (*Sequence*)  
Sequence strip to perform filter adjustments to layers below

**animation\_offset\_end**

Animation end offset (trim end)

**Type** int in [0, inf], default 0

**animation\_offset\_start**

Animation start offset (trim start)

**Type** int in [0, inf], default 0

**color\_balance**

**Type** `SequenceColorBalance`, (readonly)

**color\_multiply**

**Type** float in [0, 20], default 0.0

**color\_saturation**

**Type** float in [0, 20], default 0.0

**crop**

**Type** `SequenceCrop`, (readonly)

**proxy**

**Type** `SequenceProxy`, (readonly)

**strobe**

Only display every nth frame

**Type** float in [1, 30], default 0.0

**transform**

**Type** `SequenceTransform`, (readonly)

**use\_color\_balance**

(3-Way color correction) on input

**Type** boolean, default False

**use\_crop**

Crop image before processing

**Type** boolean, default False

**use\_deinterlace**

For video movies to remove fields



**Type** boolean, default False

**use\_flip\_x**

Flip on the X axis

**Type** boolean, default False

**use\_flip\_y**

Flip on the Y axis

**Type** boolean, default False

**use\_float**

Convert input to float data

**Type** boolean, default False

**use\_premultiply**

Convert RGB from key alpha to premultiplied alpha

**Type** boolean, default False

**use\_proxy**

Use a preview proxy and/or timecode index for this strip

**Type** boolean, default False

**use\_proxy\_custom\_directory**

Use a custom directory to store data

**Type** boolean, default False

**use\_proxy\_custom\_file**

Use a custom file to read proxy data from

**Type** boolean, default False

**use\_reverse\_frames**

Reverse frame order

**Type** boolean, default False

**use\_translation**

Translate image before processing

**Type** boolean, default False

## Inherited Properties

- `bpy_struct.id_data`
- `Sequence.name`
- `Sequence.blend_type`
- `Sequence.blend_alpha`
- `Sequence.channel`
- `Sequence.waveform`
- `Sequence.effect_fader`
- `Sequence.frame_final_end`
- `Sequence.frame_offset_end`
- `Sequence.frame_still_end`
- `Sequence.input_1`
- `Sequence.input_2`
- `Sequence.input_3`

- `Sequence.select_left_handle`
- `Sequence.frame_final_duration`
- `Sequence.frame_duration`
- `Sequence.lock`
- `Sequence.mute`
- `Sequence.select_right_handle`
- `Sequence.select`
- `Sequence.speed_factor`
- `Sequence.frame_start`
- `Sequence.frame_final_start`
- `Sequence.frame_offset_start`
- `Sequence.frame_still_start`
- `Sequence.type`
- `Sequence.use_default_fade`
- `Sequence.input_count`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Sequence.getStripElem`
- `Sequence.swap`

### 2.4.13 AlwaysSensor(Sensor)

base classes — `bpy_struct`, `Sensor`

**class** `bpy.types.AlwaysSensor` (*Sensor*)  
Sensor to generate continuous pulses

### Inherited Properties

- `bpy_struct.id_data`
- `Sensor.name`
- `Sensor.show_expanded`
- `Sensor.frequency`
- `Sensor.invert`
- `Sensor.use_level`

- `Sensor.pin`
- `Sensor.use_pulse_false_level`
- `Sensor.use_pulse_true_level`
- `Sensor.use_tap`
- `Sensor.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Sensor.link`
- `Sensor.unlink`

## 2.4.14 AndController(Controller)

base classes — `bpy_struct`, `Controller`

**class** `bpy.types.AndController` (*Controller*)  
Controller passing on events based on a logical AND operation

### Inherited Properties

- `bpy_struct.id_data`
- `Controller.name`
- `Controller.states`
- `Controller.show_expanded`
- `Controller.use_priority`
- `Controller.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`

- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Controller.link`
- `Controller.unlink`

### 2.4.15 AnimData(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.AnimData` (*`bpy_struct`*)  
Animation data for datablock

**action**

Active Action for this datablock

**Type** `Action`

**action\_blend\_type**

Method used for combining Active Action's result with result of NLA stack

- **REPLACE** Replace, Result strip replaces the accumulated results by amount specified by influence.
- **ADD** Add, Weighted result of strip is added to the accumulated results.
- **SUBTRACT** Subtract, Weighted result of strip is removed from the accumulated results.
- **MULTIPLY** Multiply, Weighted result of strip is multiplied with the accumulated results.

**Type** enum in ['REPLACE', 'ADD', 'SUBTRACT', 'MULTIPLY'], default 'REPLACE'

**action\_extrapolation**

Action to take for gaps past the Active Action's range (when evaluating with NLA)

- **NOTHING** Nothing, Strip has no influence past its extents.
- **HOLD** Hold, Hold the first frame if no previous strips in track, and always hold last frame.
- **HOLD\_FORWARD** Hold Forward, Only hold last frame.

**Type** enum in ['NOTHING', 'HOLD', 'HOLD\_FORWARD'], default 'HOLD'

**action\_influence**

Amount the Active Action contributes to the result of the NLA stack

**Type** float in [0, 1], default 1.0

**drivers**

The Drivers/Expressions for this datablock

**Type** `AnimDataDrivers` `bpy_prop_collection` of `FCurve`, (readonly)

**nla\_tracks**

NLA Tracks (i.e. Animation Layers)

**Type** NlaTracks bpy\_prop\_collection of NlaTrack, (readonly)

**use\_nla**

NLA stack is evaluated when evaluating this block

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Armature.animation_data`
- `Camera.animation_data`
- `Curve.animation_data`
- `ID.animation_data_create`
- `Key.animation_data`
- `Lamp.animation_data`
- `Lattice.animation_data`
- `Material.animation_data`
- `Mesh.animation_data`
- `MetaBall.animation_data`
- `NodeTree.animation_data`
- `Object.animation_data`
- `ParticleSettings.animation_data`
- `Scene.animation_data`
- `Speaker.animation_data`
- `Texture.animation_data`
- `World.animation_data`

## 2.4.16 AnimDataDrivers(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.AnimDataDrivers` (*bpy\_struct*)

Collection of Driver F-Curves

**from\_existing** (*src\_driver=None*)

Add a new driver given an existing one

**Parameters** `src_driver` (`FCurve`, optional) – Existing Driver F-Curve to use as template for a new one

**Returns** New Driver F-Curve

**Return type** `FCurve`

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `AnimData.drivers`

## 2.4.17 AnimViz(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.AnimViz` (*bpy\_struct*)

Settings for the visualisation of motion

**motion\_path**

Motion Path settings for visualisation

**Type** `AnimVizMotionPaths`, (readonly, never None)

#### **onion\_skin\_frames**

Onion Skinning (ghosting) settings for visualisation

**Type** `AnimVizOnionSkinning`, (readonly, never None)

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Object.animation_visualisation`
- `Pose.animation_visualisation`

## 2.4.18 AnimVizMotionPaths(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.AnimVizMotionPaths` (*bpy\_struct*)  
Motion Path settings for animation visualisation

#### **bake\_location**

When calculating Bone Paths, use Head or Tips

- **HEADS** Heads, Calculate bone paths from heads.
- **TAILS** Tails, Calculate bone paths from tails.

**Type** `enum` in ['HEADS', 'TAILS'], default 'TAILS'

**frame\_after**

Number of frames to show after the current frame (only for ‘Around Current Frame’ Onion-skinning method)

**Type** int in [1, 150000], default 0

**frame\_before**

Number of frames to show before the current frame (only for ‘Around Current Frame’ Onion-skinning method)

**Type** int in [1, 150000], default 0

**frame\_end**

End frame of range of paths to display/calculate (not for ‘Around Current Frame’ Onion-skinning method)

**Type** int in [-inf, inf], default 0

**frame\_start**

Starting frame of range of paths to display/calculate (not for ‘Around Current Frame’ Onion-skinning method)

**Type** int in [-inf, inf], default 0

**frame\_step**

Number of frames between paths shown (not for ‘On Keyframes’ Onion-skinning method)

**Type** int in [1, 100], default 0

**show\_frame\_numbers**

Show frame numbers on Motion Paths

**Type** boolean, default False

**show\_keyframe\_action\_all**

For bone motion paths, search whole Action for keyframes instead of in group with matching name only (is slower)

**Type** boolean, default False

**show\_keyframe\_highlight**

Emphasize position of keyframes on Motion Paths

**Type** boolean, default False

**show\_keyframe\_numbers**

Show frame numbers of Keyframes on Motion Paths

**Type** boolean, default False

**type**

Type of range to show for Motion Paths

- **CURRENT\_FRAME** Around Frame, Display Paths of poses within a fixed number of frames around the current frame.
- **RANGE** In Range, Display Paths of poses within specified range.

**Type** enum in [‘CURRENT\_FRAME’, ‘RANGE’], default ‘RANGE’

**Inherited Properties**

- `bpy_struct.id_data`



## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `AnimViz.motion_path`

### 2.4.19 AnimVizOnionSkinning(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.AnimVizOnionSkinning` (*bpy\_struct*)

Onion Skinning settings for animation visualisation

**frame\_after**

Number of frames to show after the current frame (only for ‘Around Current Frame’ Onion-skinning method)

**Type** `int` in `[0, 30]`, default `0`

**frame\_before**

Number of frames to show before the current frame (only for ‘Around Current Frame’ Onion-skinning method)

**Type** `int` in `[0, 30]`, default `0`

**frame\_end**

End frame of range of Ghosts to display (not for ‘Around Current Frame’ Onion-skinning method)

**Type** `int` in `[-inf, inf]`, default `0`

**frame\_start**

Starting frame of range of Ghosts to display (not for ‘Around Current Frame’ Onion-skinning method)

**Type** `int` in `[-inf, inf]`, default `0`

**frame\_step**

Number of frames between ghosts shown (not for ‘On Keyframes’ Onion-skinning method)

**Type** `int` in `[1, 20]`, default `0`

**show\_only\_selected**

For Pose-Mode drawing, only draw ghosts for selected bones

**Type** boolean, default False

**type**

Method used for determining what ghosts get drawn

- NONE No Ghosts, Do not show any ghosts.
- CURRENT\_FRAME Around Current Frame, Show ghosts from around the current frame.
- RANGE In Range, Show ghosts for the specified frame range.
- KEYS On Keyframes, Show ghosts on keyframes.

**Type** enum in ['NONE', 'CURRENT\_FRAME', 'RANGE', 'KEYS'], default 'NONE'

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `AnimViz.onion_skin_frames`

**2.4.20 AnyType(bpy\_struct)**

base class — `bpy_struct`

**class** `bpy.types.AnyType` (*bpy\_struct*)

RNA type used for pointers to any possible data

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `KeyingSetInfo.generate`
- `UILayout.context_pointer_set`
- `UILayout.prop`
- `UILayout.prop_enum`
- `UILayout.prop_menu_enum`
- `UILayout.prop_search`
- `UILayout.prop_search`
- `UILayout.props_enum`
- `UILayout.template_ID`
- `UILayout.template_ID_preview`
- `UILayout.template_any_ID`
- `UILayout.template_color_ramp`
- `UILayout.template_color_wheel`
- `UILayout.template_curve_mapping`
- `UILayout.template_histogram`
- `UILayout.template_image`
- `UILayout.template_layers`
- `UILayout.template_layers`
- `UILayout.template_list`
- `UILayout.template_list`
- `UILayout.template_path_builder`
- `UILayout.template_vectorscope`
- `UILayout.template_waveform`

## 2.4.21 Area(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.Area` (*bpy\_struct*)

Area in a subdivided screen, containing an editor

**height**

Area height

**Type** `int` in `[0, 32767]`, default `0`, (readonly)

**regions**

Regions this area is subdivided in

**Type** `bpy_prop_collection` of `Region`, (readonly)

**show\_menus**

Show menus in the header

**Type** `boolean`, default `False`

**spaces**

Spaces contained in this area, the first being the active space (NOTE: Useful for example to restore a previously used 3D view space in a certain area to get the old view orientation)

**Type** `AreaSpaces` `bpy_prop_collection` of `Space`, (readonly)

**type**

Space type

**Type** `enum` in `['EMPTY', 'VIEW_3D', 'GRAPH_EDITOR', 'OUTLINER', 'PROPERTIES', 'FILE_BROWSER', 'IMAGE_EDITOR', 'INFO', 'SEQUENCE_EDITOR', 'TEXT_EDITOR', 'DOPE SHEET_EDITOR', 'NLA_EDITOR', 'TIMELINE', 'NODE_EDITOR', 'LOGIC_EDITOR', 'CONSOLE', 'USER_PREFERENCES']`, default `'EMPTY'`

**width**

Area width

**Type** `int` in `[0, 32767]`, default `0`, (readonly)

**tag\_redraw()**

`tag_redraw`

**header\_text\_set** (*text=""*)

Set the header text

**Parameters** `text` (*string, optional*) – Text, New string for the header, no argument clears the text

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`

- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Context.area`
- `Screen.areas`

## 2.4.22 AreaLamp(Lamp)

base classes — `bpy_struct`, `ID`, `Lamp`

**class** `bpy.types.AreaLamp` (*Lamp*)

Directional area lamp

### **gamma**

Light gamma correction value

**Type** float in [-inf, inf], default 0.0

### **shadow\_adaptive\_threshold**

Threshold for Adaptive Sampling (Raytraced shadows)

**Type** float in [0, 1], default 0.0

### **shadow\_color**

Color of shadows cast by the lamp

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

### **shadow\_method**

Method to compute lamp shadow with

- `NOSHADOW` No Shadow.
- `RAY_SHADOW` Ray Shadow, Use ray tracing for shadow.

**Type** enum in ['NOSHADOW', 'RAY\_SHADOW'], default 'NOSHADOW'

### **shadow\_ray\_sample\_method**

Method for generating shadow samples: Adaptive QMC is fastest, Constant QMC is less noisy but slower

**Type** enum in ['ADAPTIVE\_QMC', 'CONSTANT\_QMC', 'CONSTANT\_JITTERED'], default 'CONSTANT\_JITTERED'

### **shadow\_ray\_samples\_x**

Amount of samples taken extra (samples x samples)

**Type** int in [1, 64], default 0

**shadow\_ray\_samples\_y**

Amount of samples taken extra (samples x samples)

**Type** int in [1, 64], default 0

**shadow\_soft\_size**

Light size for ray shadow sampling (Raytraced shadows)

**Type** float in [-inf, inf], default 0.0

**shape**

Shape of the area lamp

**Type** enum in ['SQUARE', 'RECTANGLE'], default 'SQUARE'

**size**

Size of the area of the area Lamp, X direction size for Rectangle shapes

**Type** float in [-inf, inf], default 0.0

**size\_y**

Size of the area of the area Lamp in the Y direction for Rectangle shapes

**Type** float in [-inf, inf], default 0.0

**use\_dither**

Use 2x2 dithering for sampling (Constant Jittered sampling)

**Type** boolean, default False

**use\_jitter**

Use noise for sampling (Constant Jittered sampling)

**Type** boolean, default False

**use\_only\_shadow**

Causes light to cast shadows only without illuminating objects

**Type** boolean, default False

**use\_shadow\_layer**

Causes only objects on the same layer to cast shadows

**Type** boolean, default False

**use\_umbra**

Emphasize parts that are fully shadowed (Constant Jittered sampling)

**Type** boolean, default False

**Inherited Properties**

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`
- `Lamp.active_texture`

- `Lamp.active_texture_index`
- `Lamp.animation_data`
- `Lamp.color`
- `Lamp.use_diffuse`
- `Lamp.distance`
- `Lamp.energy`
- `Lamp.use_own_layer`
- `Lamp.use_negative`
- `Lamp.node_tree`
- `Lamp.use_specular`
- `Lamp.texture_slots`
- `Lamp.type`
- `Lamp.use_nodes`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`

### 2.4.23 AreaSpaces(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.AreaSpaces` (*`bpy_struct`*)  
Collection of spaces

**active**

Space currently being displayed in this area

**Type** `Space`, (readonly)

### Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Area.spaces`

## 2.4.24 Armature(ID)

base classes — `bpy_struct`, `ID`

**class** `bpy.types.Armature` (*ID*)

Armature datablock containing a hierarchy of bones, usually used for rigging characters

### **animation\_data**

Animation data for this datablock

**Type** `AnimData`, (readonly)

### **bones**

**Type** `ArmatureBones` `bpy_prop_collection` of `Bone`, (readonly)

### **deform\_method**

- `BLENDER` Blender, Uses Blender's armature vertex deformation.
- `BGE_CPU` BGE, Uses vertex deformation code optimized for the BGE.

**Type** enum in [`'BLENDER'`, `'BGE_CPU'`], default `'BLENDER'`

### **draw\_type**

- `OCTAHEDRAL` Octahedral, Display bones as octahedral shape (default).
- `STICK` Stick, Display bones as simple 2D lines with dots.
- `BBONE` B-Bone, Display bones as boxes, showing subdivision and B-Splines.
- `ENVELOPE` Envelope, Display bones as extruded spheres, showing deformation influence volume.
- `WIRE` Wire, Display bones as thin wires, showing subdivision and B-Splines.



**Type** enum in ['OCTAHEDRAL', 'STICK', 'BBONE', 'ENVELOPE', 'WIRE'], default 'OCTAHEDRAL'

#### **edit\_bones**

**Type** `ArmatureEditBones bpy_prop_collection` of `EditBone`, (readonly)

#### **ghost\_frame\_end**

End frame of range of Ghosts to display (not for 'Around Current Frame' Onion-skinning method)

**Type** int in [-inf, inf], default 0

#### **ghost\_frame\_start**

Starting frame of range of Ghosts to display (not for 'Around Current Frame' Onion-skinning method)

**Type** int in [-inf, inf], default 0

#### **ghost\_size**

Frame step for Ghosts (not for 'On Keyframes' Onion-skinning method)

**Type** int in [1, 20], default 0

#### **ghost\_step**

Number of frame steps on either side of current frame to show as ghosts (only for 'Around Current Frame' Onion-skinning method)

**Type** int in [0, 30], default 0

#### **ghost\_type**

Method of Onion-skinning for active Action

- CURRENT\_FRAME Around Frame, Display Ghosts of poses within a fixed number of frames around the current frame.
- RANGE In Range, Display Ghosts of poses within specified range.
- KEYS On Keyframes, Display Ghosts of poses on Keyframes.

**Type** enum in ['CURRENT\_FRAME', 'RANGE', 'KEYS'], default 'CURRENT\_FRAME'

#### **layers**

Armature layer visibility

**Type** boolean array of 32 items, default (False, False)

#### **layers\_protected**

Protected layers in Proxy Instances are restored to Proxy settings on file reload and undo

**Type** boolean array of 32 items, default (False, False)

#### **pose\_position**

Show armature in binding pose or final posed state

- POSE Pose Position, Show armature in posed state.
- REST Rest Position, Show Armature in binding pose state. No posing possible.

**Type** enum in ['POSE', 'REST'], default 'POSE'

**show\_axes**

Draw bone axes

**Type** boolean, default False

**show\_bone\_custom\_shapes**

Draw bones with their custom shapes

**Type** boolean, default False

**show\_group\_colors**

Draw bone group colors

**Type** boolean, default False

**show\_names**

Draw bone names

**Type** boolean, default False

**show\_only\_ghost\_selected**

**Type** boolean, default False

**use\_auto\_ik**

Add temporary IK constraints while grabbing bones in Pose Mode

**Type** boolean, default False

**use\_deform\_delay**

Don't deform children when manipulating bones in Pose Mode

**Type** boolean, default False

**use\_deform\_envelopes**

Enable Bone Envelopes when defining deform

**Type** boolean, default False

**use\_deform\_preserve\_volume**

Enable deform rotation with Quaternions

**Type** boolean, default False

**use\_deform\_vertex\_groups**

Enable Vertex Groups when defining deform

**Type** boolean, default False

**use\_mirror\_x**

Apply changes to matching bone on opposite side of X-Axis

**Type** boolean, default False

**Inherited Properties**

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`

## References

- `BlendData.armatures`
- `BlendDataArmatures.new`
- `BlendDataArmatures.remove`

### 2.4.25 ArmatureActuator(Actuator)

base classes — `bpy_struct`, `Actuator`

**class** `bpy.types.ArmatureActuator` (*Actuator*)

#### **bone**

Bone on which the constraint is defined

**Type** string, default ""

#### **constraint**

Name of the constraint you want to control

**Type** string, default ""

#### **mode**

**Type** enum in ['RUN', 'ENABLE', 'DISABLE', 'SETTARGET', 'SETWEIGHT'], default 'RUN'

#### **secondary\_target**

Set this object as the secondary target of the constraint (only IK polar target at the moment)

**Type** `Object`

**target**

Set this object as the target of the constraint

**Type** `Object`

**weight**

Set weight of this constraint

**Type** float in [0, 1], default 0.0

**Inherited Properties**

- `bpy_struct.id_data`
- `Actuator.name`
- `Actuator.show_expanded`
- `Actuator.pin`
- `Actuator.type`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Actuator.link`
- `Actuator.unlink`

**2.4.26 ArmatureBones(`bpy_struct`)**

base class — `bpy_struct`

**class** `bpy.types.ArmatureBones` (*`bpy_struct`*)

Collection of armature bones

**active**

Armature's active bone

**Type** `Bone`

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Armature.bones`

### 2.4.27 ArmatureEditBones(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.ArmatureEditBones` (*bpy\_struct*)

Collection of armature edit bones

**active**

Armatures active edit bone

**Type** `EditBone`

**new** (*name*)

Add a new bone

**Parameters** **name** (*string*) – New name for the bone

**Returns** Newly created edit bone

**Return type** `EditBone`

**remove** (*bone*)

Remove an existing bone from the armature

**Parameters** **bone** (`EditBone`, (never `None`)) – `EditBone` to remove

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Armature.edit_bones`

## 2.4.28 ArmatureModifier(Modifier)

base classes — `bpy_struct`, `Modifier`

**class** `bpy.types.ArmatureModifier` (*Modifier*)  
Armature deformation modifier

**invert\_vertex\_group**

Invert vertex group influence

**Type** boolean, default False

**object**

Armature object to deform with

**Type** `Object`

**use\_bone\_envelopes**

Binds Bone envelope to armature modifier

**Type** boolean, default False

**use\_deform\_preserve\_volume**

Deform rotation interpolation with quaternions

**Type** boolean, default False

**use\_multi\_modifier**

Use same input as previous modifier, and mix results using overall vgroup

**Type** boolean, default False

**use\_vertex\_groups**

Binds vertex group to armature modifier

**Type** boolean, default False

**vertex\_group**

Name of Vertex Group which determines influence of modifier per point

**Type** string, default ""

### Inherited Properties

- `bpy_struct.id_data`
- `Modifier.name`
- `Modifier.use_apply_on_spline`
- `Modifier.show_in_editmode`
- `Modifier.show_expanded`
- `Modifier.show_on_cage`
- `Modifier.show_viewport`
- `Modifier.show_render`
- `Modifier.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.29 ArmatureSensor(Sensor)

base classes — `bpy_struct`, `Sensor`

**class** `bpy.types.ArmatureSensor` (*Sensor*)

Sensor to detect values and changes in values of IK solver

**bone**

Identify the bone to check value from

**Type** string, default ""

**constraint**

Identify the bone constraint to check value from

**Type** string, default ""

**test\_type**

Type of value and test

**Type** enum in ['STATECHG', 'LINERRORBELOW', 'LINERRORABOVE', 'ROTERROR-  
BELOW', 'ROTERRORABOVE'], default 'STATECHG'

**value**

Specify value to be used in comparison

**Type** float in [-inf, inf], default 0.0

**Inherited Properties**

- `bpy_struct.id_data`
- `Sensor.name`
- `Sensor.show_expanded`
- `Sensor.frequency`
- `Sensor.invert`
- `Sensor.use_level`
- `Sensor.pin`
- `Sensor.use_pulse_false_level`
- `Sensor.use_pulse_true_level`
- `Sensor.use_tap`
- `Sensor.type`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Sensor.link`
- `Sensor.unlink`

**2.4.30 ArrayModifier(Modifier)**

base classes — `bpy_struct`, `Modifier`



**class** `bpy.types.ArrayModifier` (*Modifier*)

Array duplication modifier

**constant\_offset\_displace**

Value for the distance between arrayed items

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**count**

Number of duplicates to make

**Type** int in [1, inf], default 0

**curve**

Curve object to fit array length to

**Type** `Object`

**end\_cap**

Mesh object to use as an end cap

**Type** `Object`

**fit\_length**

Length to fit array within

**Type** float in [0, inf], default 0.0

**fit\_type**

Array length calculation method

- `FIXED_COUNT` Fixed Count, Duplicate the object a certain number of times.
- `FIT_LENGTH` Fit Length, Duplicate the object as many times as fits in a certain length.
- `FIT_CURVE` Fit Curve, Fit the duplicated objects to a curve.

**Type** enum in ['FIXED\_COUNT', 'FIT\_LENGTH', 'FIT\_CURVE'], default 'FIXED\_COUNT'

**merge\_threshold**

Limit below which to merge vertices

**Type** float in [0, inf], default 0.0

**offset\_object**

Uses the location and rotation of another object to determine the distance and rotational change between arrayed items

**Type** `Object`

**relative\_offset\_displace**

The size of the geometry will determine the distance between arrayed items

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**start\_cap**

Mesh object to use as a start cap

**Type** `Object`

**use\_constant\_offset**

Add a constant offset

**Type** boolean, default False

**use\_merge\_vertices**

Merge vertices in adjacent duplicates

**Type** boolean, default False

**use\_merge\_vertices\_cap**

Merge vertices in first and last duplicates

**Type** boolean, default False

**use\_object\_offset**

Add another object's transformation to the total offset

**Type** boolean, default False

**use\_relative\_offset**

Add an offset relative to the object's bounding box

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `Modifier.name`
- `Modifier.use_apply_on_spline`
- `Modifier.show_in_editmode`
- `Modifier.show_expanded`
- `Modifier.show_on_cage`
- `Modifier.show_viewport`
- `Modifier.show_render`
- `Modifier.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.31 BackgroundImage(`bpy_struct`)

base class — `bpy_struct`

---

```

class bpy.types.BackgroundImage(bpy_struct)
    Image and settings for display in the 3d View background

    image
        Image displayed and edited in this space

        Type Image

    image_user
        Parameters defining which layer, pass and frame of the image is displayed

        Type ImageUser, (readonly, never None)

    offset_x
        Offset image horizontally from the world origin

        Type float in [-inf, inf], default 0.0

    offset_y
        Offset image vertically from the world origin

        Type float in [-inf, inf], default 0.0

    opacity
        Image opacity to blend the image against the background color

        Type float in [0, 1], default 0.0

    show_expanded
        Show the expanded in the user interface

        Type boolean, default False

    size
        Scaling factor for the background image

        Type float in [0, inf], default 0.0

    view_axis
        The axis to display the image on

        •LEFT Left, Show background image while looking to the left.
        •RIGHT Right, Show background image while looking to the right.
        •BACK Back, Show background image in back view.
        •FRONT Front, Show background image in front view.
        •BOTTOM Bottom, Show background image in bottom view.
        •TOP Top, Show background image in top view.
        •ALL All Views, Show background image in all views.
        •CAMERA Camera, Show background image in camera view.

        Type enum in ['LEFT', 'RIGHT', 'BACK', 'FRONT', 'BOTTOM', 'TOP', 'ALL', 'CAMERA'], default 'ALL'

```

### Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `SpaceView3D.background_images`

### 2.4.32 BevelModifier(Modifier)

base classes — `bpy_struct`, `Modifier`

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

Bevel modifier to make edges and vertices more rounded

**angle\_limit**

Angle above which to bevel edges

**Type** float in [0, 3.14159], default 0.0

**edge\_weight\_method**

What edge weight to use for weighting a vertex

**Type** enum in ['AVERAGE', 'SHARPEST', 'LARGEST'], default 'AVERAGE'

**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; apply them separately in vert/edge select mode.

**Type** enum in ['NONE', 'ANGLE', 'WEIGHT'], default 'NONE'

**use\_only\_vertices**

Bevel verts/corners, not edges

**Type** boolean, default False

**width**

Bevel value/amount

**Type** float in [0, inf], default 0.0**Inherited Properties**

- `bpy_struct.id_data`
- `Modifier.name`
- `Modifier.use_apply_on_spline`
- `Modifier.show_in_editmode`
- `Modifier.show_expanded`
- `Modifier.show_on_cage`
- `Modifier.show_viewport`
- `Modifier.show_render`
- `Modifier.type`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**2.4.33 BezierSplinePoint(`bpy_struct`)**base class — `bpy_struct`**class** `bpy.types.BezierSplinePoint` (*bpy\_struct*)

Bezier curve point with two handles

**co**

Coordinates of the control point

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)**handle\_left**

Coordinates of the first handle

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)**handle\_left\_type**

Handle types

**Type** enum in ['FREE', 'VECTOR', 'ALIGNED', 'AUTO'], default 'FREE'

**handle\_right**

Coordinates of the second handle

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**handle\_right\_type**

Handle types

**Type** enum in ['FREE', 'VECTOR', 'ALIGNED', 'AUTO'], default 'FREE'

**hide**

Visibility status

**Type** boolean, default False

**radius**

Radius for bevelling

**Type** float in [0, inf], default 0.0

**select\_control\_point**

Control point selection status

**Type** boolean, default False

**select\_left\_handle**

Handle 1 selection status

**Type** boolean, default False

**select\_right\_handle**

Handle 2 selection status

**Type** boolean, default False

**tilt**

Tilt in 3D View

**Type** float in [-inf, inf], default 0.0

**weight**

Softbody goal weight

**Type** float in [0.01, 100], default 0.0

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`

- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Spline.bezier_points`

## 2.4.34 BlendData(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.BlendData` (*bpy\_struct*)

Main data structure representing a .blend file and all its datablocks

### actions

Action datablocks

**Type** `BlendDataActions bpy_prop_collection` of `Action`, (readonly)

### armatures

Armature datablocks

**Type** `BlendDataArmatures bpy_prop_collection` of `Armature`, (readonly)

### brushes

Brush datablocks

**Type** `BlendDataBrushes bpy_prop_collection` of `Brush`, (readonly)

### cameras

Camera datablocks

**Type** `BlendDataCameras bpy_prop_collection` of `Camera`, (readonly)

### curves

Curve datablocks

**Type** `BlendDataCurves bpy_prop_collection` of `Curve`, (readonly)

### filepath

Path to the .blend file

**Type** `string`, default `""`, (readonly)

### fonts

Vector font datablocks

**Type** `BlendDataFonts bpy_prop_collection` of `VectorFont`, (readonly)

### grease\_pencil

Grease Pencil datablocks

**Type** `BlendDataGreasePencils bpy_prop_collection` of `GreasePencil`, (readonly)

**groups**

Group datablocks

**Type** `BlendDataGroups bpy_prop_collection of Group`, (readonly)

**images**

Image datablocks

**Type** `BlendDataImages bpy_prop_collection of Image`, (readonly)

**is\_dirty**

Have recent edits been saved to disk

**Type** boolean, default False, (readonly)

**is\_saved**

Has the current session been saved to disk as a .blend file

**Type** boolean, default False, (readonly)

**lamps**

Lamp datablocks

**Type** `BlendDataLamps bpy_prop_collection of Lamp`, (readonly)

**lattices**

Lattice datablocks

**Type** `BlendDataLattices bpy_prop_collection of Lattice`, (readonly)

**libraries**

Library datablocks

**Type** `BlendDataLibraries bpy_prop_collection of Library`, (readonly)

**materials**

Material datablocks

**Type** `BlendDataMaterials bpy_prop_collection of Material`, (readonly)

**meshes**

Mesh datablocks

**Type** `BlendDataMeshes bpy_prop_collection of Mesh`, (readonly)

**metaballs**

Metaball datablocks

**Type** `BlendDataMetaBalls bpy_prop_collection of MetaBall`, (readonly)

**node\_groups**

Node group datablocks

**Type** `BlendDataNodeTrees bpy_prop_collection of NodeTree`, (readonly)

**objects**

Object datablocks

**Type** `BlendDataObjects bpy_prop_collection of Object`, (readonly)

**particles**

Particle datablocks

**Type** `BlendDataParticles bpy_prop_collection of ParticleSettings`, (readonly)



**scenes**

Scene datablocks

**Type** `BlendDataScenes bpy_prop_collection` of `Scene`, (readonly)**screens**

Screen datablocks

**Type** `BlendDataScreens bpy_prop_collection` of `Screen`, (readonly)**scripts**

Script datablocks (DEPRECATED)

**Type** `bpy_prop_collection` of `ID`, (readonly)**shape\_keys**

Shape Key datablocks

**Type** `bpy_prop_collection` of `Key`, (readonly)**sounds**

Sound datablocks

**Type** `BlendDataSounds bpy_prop_collection` of `Sound`, (readonly)**speakers**

Speaker datablocks

**Type** `BlendDataSpeakers bpy_prop_collection` of `Speaker`, (readonly)**texts**

Text datablocks

**Type** `BlendDataTexts bpy_prop_collection` of `Text`, (readonly)**textures**

Texture datablocks

**Type** `BlendDataTextures bpy_prop_collection` of `Texture`, (readonly)**window\_managers**

Window manager datablocks

**Type** `BlendDataWindowManagers bpy_prop_collection` of `WindowManager`, (readonly)**worlds**

World datablocks

**Type** `BlendDataWorlds bpy_prop_collection` of `World`, (readonly)**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`

- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Context.blend_data`
- `RenderEngine.update`

## 2.4.35 BlendDataActions(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.BlendDataActions` (*bpy\_struct*)

Collection of actions

**is\_updated**

**Type** boolean, default False, (readonly)

**new** (*name*)

Add a new action to the main database

**Parameters** **name** (*string*) – New name for the datablock

**Returns** New action datablock

**Return type** `Action`

**remove** (*action*)

Remove a action from the current blendfile

**Parameters** **action** (`Action`, (never None)) – Action to remove

**tag** (*value*)

tag

**Parameters** **value** (*boolean*) – Value

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`

- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `BlendData.actions`

## 2.4.36 BlendDataArmatures(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.BlendDataArmatures` (*bpy\_struct*)  
Collection of armatures

### **is\_updated**

**Type** boolean, default False, (readonly)

### **new** (*name*)

Add a new armature to the main database

**Parameters** **name** (*string*) – New name for the datablock

**Returns** New armature datablock

**Return type** `Armature`

### **remove** (*armature*)

Remove a armature from the current blendfile

**Parameters** **armature** (`Armature`, (never None)) – Armature to remove

### **tag** (*value*)

tag

**Parameters** **value** (*boolean*) – Value

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`

- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `BlendData.armatures`

## 2.4.37 BlendDataBrushes(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.BlendDataBrushes` (*bpy\_struct*)  
Collection of brushes

### **is\_updated**

**Type** boolean, default False, (readonly)

### **new** (*name*)

Add a new brush to the main database

**Parameters** **name** (*string*) – New name for the datablock

**Returns** New brush datablock

**Return type** `Brush`

### **remove** (*brush*)

Remove a brush from the current blendfile

**Parameters** **brush** (`Brush`, (never None)) – Brush to remove

### **tag** (*value*)

tag

**Parameters** **value** (*boolean*) – Value

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `BlendData.brushes`

### 2.4.38 BlendDataCameras(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.BlendDataCameras` (*bpy\_struct*)

Collection of cameras

**is\_updated**

**Type** boolean, default False, (readonly)

**new** (*name*)

Add a new camera to the main database

**Parameters** **name** (*string*) – New name for the datablock

**Returns** New camera datablock

**Return type** `Camera`

**remove** (*camera*)

Remove a camera from the current blendfile

**Parameters** **camera** (`Camera`, (never None)) – Camera to remove

**tag** (*value*)

tag

**Parameters** **value** (*boolean*) – Value

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `BlendData.cameras`

## 2.4.39 BlendDataCurves(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.BlendDataCurves` (*bpy\_struct*)  
Collection of curves

**is\_updated**

**Type** boolean, default False, (readonly)

**new** (*name, type*)

Add a new curve to the main database

**Parameters**

- **name** (*string*) – New name for the datablock
- **type** (*enum in ['CURVE', 'SURFACE', 'FONT']*) – Type, The type of curve to add

**Returns** New curve datablock

**Return type** `Curve`

**remove** (*curve*)

Remove a curve from the current blendfile

**Parameters** **curve** (`Curve`, (never None)) – Curve to remove

**tag** (*value*)

tag

**Parameters** **value** (*boolean*) – Value

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `BlendData.curves`

### 2.4.40 BlendDataFonts(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.BlendDataFonts` (*bpy\_struct*)

Collection of fonts

**is\_updated**

**Type** boolean, default False, (readonly)

**load** (*filepath*)

Load a new font into the main database

**Parameters** **filepath** (*string*) – path of the font to load

**Returns** New font datablock

**Return type** `VectorFont`

**remove** (*vfont*)

Remove a font from the current blendfile

**Parameters** **vfont** (`VectorFont`, (never None)) – Font to remove

**tag** (*value*)

tag

**Parameters** **value** (*boolean*) – Value

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `BlendData.fonts`

## 2.4.41 BlendDataGreasePencils(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.BlendDataGreasePencils` (*bpy\_struct*)  
Collection of grease pencils

**is\_updated**

**Type** boolean, default False, (readonly)

**tag** (*value*)

tag

**Parameters** *value* (*boolean*) – Value

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`



- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `BlendData.grease_pencil`

## 2.4.42 BlendDataGroups(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.BlendDataGroups` (*bpy\_struct*)  
Collection of groups

**is\_updated**

**Type** boolean, default False, (readonly)

**new** (*name*)

Add a new group to the main database

**Parameters** **name** (*string*) – New name for the datablock

**Returns** New group datablock

**Return type** `Group`

**remove** (*group*)

Remove a group from the current blendfile

**Parameters** **group** (`Group`, (never None)) – Group to remove

**tag** (*value*)

tag

**Parameters** **value** (*boolean*) – Value

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`

- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `BlendData.groups`

## 2.4.43 BlendDataImages(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.BlendDataImages` (*bpy\_struct*)

Collection of images

**is\_updated**

**Type** boolean, default False, (readonly)

**new** (*name, width, height, alpha=False, float\_buffer=False*)

Add a new image to the main database

**Parameters**

- **name** (*string*) – New name for the datablock
- **width** (*int in [1, inf]*) – Width of the image
- **height** (*int in [1, inf]*) – Height of the image
- **alpha** (*boolean, (optional)*) – Alpha, Use alpha channel
- **float\_buffer** (*boolean, (optional)*) – Float Buffer, Create an image with floating point color

**Returns** New image datablock

**Return type** `Image`

**load** (*filepath*)

Load a new image into the main database

**Parameters** **filepath** (*string*) – path of the file to load

**Returns** New image datablock

**Return type** `Image`

**remove** (*image*)

Remove an image from the current blendfile

**Parameters** **image** (*Image*, (never None)) – Image to remove

**tag** (*value*)

tag

**Parameters** **value** (*boolean*) – Value

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `BlendData.images`

## 2.4.44 BlendDataLamps(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.BlendDataLamps` (*bpy\_struct*)

Collection of lamps

**is\_updated**

**Type** boolean, default False, (readonly)

**new** (*name*, *type*)

Add a new lamp to the main database

**Parameters**

- **name** (*string*) – New name for the datablock
- **type** (*enum in ['POINT', 'SUN', 'SPOT', 'HEMI', 'AREA']*) – Type, The type of texture to add

- POINT Point, Omnidirectional point light source.
- SUN Sun, Constant direction parallel ray light source.
- SPOT Spot, Directional cone light source.
- HEMI Hemi, 180 degree constant light source.
- AREA Area, Directional area light source.

**Returns** New lamp datablock

**Return type** `Lamp`

**remove** (*lamp*)

Remove a lamp from the current blendfile

**Parameters** **lamp** (`Lamp`, (never None)) – Lamp to remove

**tag** (*value*)

tag

**Parameters** **value** (*boolean*) – Value

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `BlendData.lamps`

## 2.4.45 `BlendDataLattices(bpy_struct)`

base class — `bpy_struct`

**class** `bpy.types.BlendDataLattices` (*bpy\_struct*)  
 Collection of lattices

**is\_updated**  
**Type** boolean, default False, (readonly)

**new** (*name*)  
 Add a new lattice to the main database

**Parameters** **name** (*string*) – New name for the datablock

**Returns** New lattices datablock

**Return type** `Lattice`

**remove** (*lattice*)  
 Remove a lattice from the current blendfile

**Parameters** **lattice** (`Lattice`, (never None)) – Lattice to remove

**tag** (*value*)  
 tag

**Parameters** **value** (*boolean*) – Value

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `BlendData.lattices`

## 2.4.46 BlendDataLibraries(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.BlendDataLibraries` (*bpy\_struct*)

Collection of libraries

**is\_updated**

**Type** boolean, default False, (readonly)

**tag** (*value*)

tag

**Parameters value** (*boolean*) – Value

**load** (*filepath*, *link=False*, *relative=False*)

Returns a context manager which exposes 2 library objects on entering. Each object has attributes matching `bpy.data` which are lists of strings to be linked.

**Parameters**

- **filepath** (*string*) – The path to a blend file.
- **link** (*bool*) – When False reference to the original file is lost.
- **relative** (*bool*) – When True the path is stored relative to the open blend file.

```
import bpy
```

```
filepath = "//link_library.blend"
```

```
# load a single scene we know the name of.
```

```
with bpy.data.libraries.load(filepath) as (data_from, data_to):  
    data_to.scenes = ["Scene"]
```

```
# load all meshes
```

```
with bpy.data.libraries.load(filepath) as (data_from, data_to):  
    data_to.meshes = data_from.meshes
```

```
# link all objects starting with 'A'
```

```
with bpy.data.libraries.load(filepath, link=True) as (data_from, data_to):  
    data_to.objects = [name for name in data_from.objects if name.startswith("A")]
```

```
# append everything
```

```
with bpy.data.libraries.load(filepath) as (data_from, data_to):  
    for attr in dir(data_to):  
        setattr(data_to, attr, getattr(data_from, attr))
```

```
# the loaded objects can be accessed from 'data_to' outside of the context
```

```
# since loading the data replaces the strings for the datablocks or None
```

```
# if the datablock could not be loaded.
```

```
with bpy.data.libraries.load(filepath) as (data_from, data_to):  
    data_to.meshes = data_from.meshes
```

```
# now operate directly on the loaded data
```

```
for mesh in data_to.meshes:
```

```
    if mesh is not None:
```

```
        print(mesh.name)
```

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `BlendData.libraries`

### 2.4.47 BlendDataMaterials(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.BlendDataMaterials` (*bpy\_struct*)

Collection of materials

#### **is\_updated**

**Type** boolean, default False, (readonly)

#### **new** (*name*)

Add a new material to the main database

**Parameters** **name** (*string*) – New name for the datablock

**Returns** New material datablock

**Return type** `Material`

#### **remove** (*material*)

Remove a material from the current blendfile

**Parameters** **material** (`Material`, (never None)) – Material to remove

#### **tag** (*value*)

tag

**Parameters** **value** (*boolean*) – Value

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `BlendData.materials`

## 2.4.48 BlendDataMeshes(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.BlendDataMeshes` (*bpy\_struct*)  
Collection of meshes

#### **is\_updated**

**Type** boolean, default False, (readonly)

#### **new** (*name*)

Add a new mesh to the main database

**Parameters** **name** (*string*) – New name for the datablock

**Returns** New mesh datablock

**Return type** `Mesh`

#### **remove** (*mesh*)

Remove a mesh from the current blendfile

**Parameters** **mesh** (`Mesh`, (never None)) – Mesh to remove

#### **tag** (*value*)

tag

**Parameters** **value** (*boolean*) – Value



## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `BlendData.meshes`

## 2.4.49 BlendDataMetaBalls(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.BlendDataMetaBalls` (*bpy\_struct*)

Collection of metaballs

**is\_updated**

**Type** boolean, default False, (readonly)

**new** (*name*)

Add a new metaball to the main database

**Parameters** **name** (*string*) – New name for the datablock

**Returns** New metaball datablock

**Return type** `MetaBall`

**remove** (*metaball*)

Remove a metaball from the current blendfile

**Parameters** **metaball** (`MetaBall`, (never None)) – `MetaBall` to remove

**tag** (*value*)

tag

**Parameters** **value** (*boolean*) – Value

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `BlendData.metaballs`

## 2.4.50 BlendDataNodeTrees(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.BlendDataNodeTrees` (*bpy\_struct*)

Collection of node trees

#### **is\_updated**

**Type** boolean, default False, (readonly)

#### **new** (*name, type*)

Add a new node tree to the main database

#### **Parameters**

- **name** (*string*) – New name for the datablock
- **type** (*enum in ['SHADER', 'COMPOSITE', 'TEXTURE']*) – Type, The type of node\_group to add

**Returns** New node tree datablock

**Return type** `NodeTree`

#### **remove** (*tree*)

Remove a node tree from the current blendfile

**Parameters** **tree** (`NodeTree`, (never None)) – Node tree to remove

**tag** (*value*)  
tag

**Parameters** **value** (*boolean*) – Value

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `BlendData.node_groups`

## 2.4.51 BlendDataObjects(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.BlendDataObjects` (*bpy\_struct*)

Collection of objects

**is\_updated**

**Type** boolean, default False, (readonly)

**new** (*name, object\_data*)

Add a new object to the main database

**Parameters**

- **name** (*string*) – New name for the datablock
- **object\_data** (*ID*) – Object data or None for an empty object

**Returns** New object datablock

**Return type** `Object`

**remove** (*object*)

Remove a object from the current blendfile

**Parameters** **object** (*Object*, (never None)) – Object to remove

**tag** (*value*)

tag

**Parameters** **value** (*boolean*) – Value

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `BlendData.objects`

## 2.4.52 BlendDataParticles(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.BlendDataParticles` (*bpy\_struct*)

Collection of particle settings

**is\_updated**

**Type** boolean, default False, (readonly)

**new** (*name*)

Add a new particle settings instance to the main database

**Parameters** **name** (*string*) – New name for the datablock

**Returns** New particle settings datablock

**Return type** `ParticleSettings`

**remove** (*particle*)

Remove a particle settings instance from the current blendfile

**Parameters** **particle** (`ParticleSettings`, (never `None`)) – Particle Settings to remove

**tag** (*value*)

tag

**Parameters** **value** (*boolean*) – Value

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `BlendData.particles`

## 2.4.53 `BlendDataScenes(bpy_struct)`

base class — `bpy_struct`

**class** `bpy.types.BlendDataScenes` (*bpy\_struct*)

Collection of scenes

**is\_updated**

**Type** `boolean`, default `False`, (readonly)

**new** (*name*)

Add a new scene to the main database

**Parameters** **name** (*string*) – New name for the datablock

**Returns** New scene datablock

**Return type** `Scene`

**remove** (*scene*)

Remove a scene from the current blendfile

**Parameters** **scene** (`Scene`, (never `None`)) – Scene to remove

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `BlendData.scenes`

## 2.4.54 BlendDataScreens(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.BlendDataScreens` (*bpy\_struct*)

Collection of screens

**is\_updated**

**Type** boolean, default `False`, (readonly)

**tag** (*value*)

tag

**Parameters** **value** (*boolean*) – Value

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `BlendData.screens`

## 2.4.55 BlendDataSounds(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.BlendDataSounds` (*bpy\_struct*)  
Collection of sounds

**`is_updated`**

**Type** boolean, default False, (readonly)

**`tag`** (*value*)

tag

**Parameters** *value* (*boolean*) – Value

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`

- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `BlendData.sounds`

## 2.4.56 BlendDataSpeakers(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.BlendDataSpeakers` (*bpy\_struct*)  
Collection of speakers

**is\_updated**

**Type** boolean, default False, (readonly)

**new** (*name*)

Add a new speaker to the main database

**Parameters** **name** (*string*) – New name for the datablock

**Returns** New speaker datablock

**Return type** `Speaker`

**remove** (*speaker*)

Remove a speaker from the current blendfile

**Parameters** **speaker** (`Speaker`, (never None)) – Speaker to remove

**tag** (*value*)

tag

**Parameters** **value** (*boolean*) – Value

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`



- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `BlendData.speakers`

## 2.4.57 BlendDataTexts(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.BlendDataTexts` (*bpy\_struct*)

Collection of texts

**is\_updated**

**Type** boolean, default False, (readonly)

**new** (*name*)

Add a new text to the main database

**Parameters** `name` (*string*) – New name for the datablock

**Returns** New text datablock

**Return type** `Text`

**remove** (*text*)

Remove a text from the current blendfile

**Parameters** `text` (`Text`, (never None)) – Text to remove

**load** (*filepath*)

Add a new text to the main database from a file

**Parameters** `filepath` (*string*) – path for the datablock

**Returns** New text datablock

**Return type** `Text`

**tag** (*value*)

tag

**Parameters** `value` (*boolean*) – Value

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `BlendData.texts`

## 2.4.58 BlendDataTextures(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.BlendDataTextures` (*bpy\_struct*)  
Collection of groups

#### **is\_updated**

**Type** boolean, default False, (readonly)

#### **new** (*name, type*)

Add a new texture to the main database

#### **Parameters**

- **name** (*string*) – New name for the datablock
- **type** (*enum in ['NONE', 'BLEND', 'CLOUDS', 'DISTORTED\_NOISE', 'ENVIRONMENT\_MAP', 'IMAGE', 'MAGIC', 'MARBLE', 'MUSGRAVE', 'NOISE', 'POINT\_DENSITY', 'STUCCI', 'VORONOI', 'VOXEL\_DATA', 'WOOD']*) – Type, The type of texture to add
  - NONE None.
  - BLEND Blend, Procedural - create a ramp texture.
  - CLOUDS Clouds, Procedural - create a cloud-like fractal noise texture.

- `DISTORTED_NOISE` Distorted Noise, Procedural - Noise texture distorted by two noise algorithms.
- `ENVIRONMENT_MAP` Environment Map, Create a render of the environment mapped to a texture.
- `IMAGE` Image or Movie, Allow for images or movies to be used as textures.
- `MAGIC` Magic, Procedural - Color texture based on trigonometric functions.
- `MARBLE` Marble, Procedural - Marble-like noise texture with wave generated bands.
- `MUSGRAVE` Musgrave, Procedural - Highly flexible fractal noise texture.
- `NOISE` Noise, Procedural - Random noise, gives a different result every time, for every frame, for every pixel.
- `POINT_DENSITY` Point Density.
- `STUCCI` Stucci, Procedural - Create a fractal noise texture.
- `VORONOI` Voronoi, Procedural - Create cell-like patterns based on Worley noise.
- `VOXEL_DATA` Voxel Data, Create a 3d texture based on volumetric data.
- `WOOD` Wood, Procedural - Wave generated bands or rings, with optional noise.

**Returns** New texture datablock

**Return type** `Texture`

**remove** (*texture*)

Remove a texture from the current blendfile

**Parameters** **texture** (`Texture`, (never None)) – Texture to remove

**tag** (*value*)

tag

**Parameters** **value** (*boolean*) – Value

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`

- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `BlendData.textures`

## 2.4.59 BlendDataWindowManagers(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.BlendDataWindowManagers` (*bpy\_struct*)  
Collection of window managers

**is\_updated**

**Type** boolean, default False, (readonly)

**tag** (*value*)

tag

**Parameters** *value* (*boolean*) – Value

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `BlendData.window_managers`

## 2.4.60 BlendDataWorlds(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.BlendDataWorlds` (*bpy\_struct*)

Collection of worlds

**is\_updated**

**Type** boolean, default False, (readonly)

**new** (*name*)

Add a new world to the main database

**Parameters** **name** (*string*) – New name for the datablock

**Returns** New world datablock

**Return type** `World`

**remove** (*world*)

Remove a world from the current blendfile

**Parameters** **world** (`World`, (never None)) – World to remove

**tag** (*value*)

tag

**Parameters** **value** (*boolean*) – Value

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `BlendData.worlds`

## 2.4.61 BlendTexture(Texture)

base classes — `bpy_struct`, `ID`, `Texture`

**class** `bpy.types.BlendTexture` (*Texture*)

Procedural color blending texture

### **progression**

Sets the style of the color blending

- **LINEAR** Linear, Create a linear progression.
- **QUADRATIC** Quadratic, Create a quadratic progression.
- **EASING** Easing, Create a progression easing from one step to the next.
- **DIAGONAL** Diagonal, Create a diagonal progression.
- **SPHERICAL** Spherical, Create a spherical progression.
- **QUADRATIC\_SPHERE** Quadratic sphere, Create a quadratic progression in the shape of a sphere.
- **RADIAL** Radial, Create a radial progression.

**Type** enum in ['LINEAR', 'QUADRATIC', 'EASING', 'DIAGONAL', 'SPHERICAL', 'QUADRATIC\_SPHERE', 'RADIAL'], default 'LINEAR'

### **use\_flip\_axis**

Flips the texture's X and Y axis

- **HORIZONTAL** Horizontal, Flips the texture's X and Y axis.
- **VERTICAL** Vertical, Flips the texture's X and Y axis.

**Type** enum in ['HORIZONTAL', 'VERTICAL'], default 'HORIZONTAL'

### **users\_material**

Materials that use this texture (readonly)

### **users\_object\_modifier**

Object modifiers that use this texture (readonly)

## Inherited Properties

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`
- `Texture.animation_data`
- `Texture.intensity`
- `Texture.color_ramp`
- `Texture.contrast`
- `Texture.factor_blue`
- `Texture.factor_green`
- `Texture.factor_red`

- `Texture.node_tree`
- `Texture.saturation`
- `Texture.use_preview_alpha`
- `Texture.type`
- `Texture.use_color_ramp`
- `Texture.use_nodes`
- `Texture.users_material`
- `Texture.users_object_modifier`
- `Texture.users_material`
- `Texture.users_object_modifier`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`
- `Texture.evaluate`

## 2.4.62 BlenderRNA(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.BlenderRNA` (*bpy\_struct*)  
Blender RNA structure definitions

### structs

**Type** `bpy_prop_collection` of `Struct`, (readonly)

### Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.63 BoidRule(bpy\_struct)

base class — `bpy_struct`

subclasses — `BoidRuleAvoidCollision`, `BoidRuleAverageSpeed`, `BoidRuleAvoid`,  
`BoidRuleFight`, `BoidRuleFollowLeader`, `BoidRuleGoal`

**class** `bpy.types.BoidRule` (*bpy\_struct*)

**name**

Boid rule name

**Type** string, default ""

**type**

- **GOAL** Goal, Go to assigned object or loudest assigned signal source.
- **AVOID** Avoid, Get away from assigned object or loudest assigned signal source.
- **AVOID\_COLLISION** Avoid Collision, Manoeuvre to avoid collisions with other boids and deflector objects in near future.
- **SEPARATE** Separate, Keep from going through other boids.
- **FLOCK** Flock, Move to center of neighbors and match their velocity.
- **FOLLOW\_LEADER** Follow Leader, Follow a boid or assigned object.
- **AVERAGE\_SPEED** Average Speed, Maintain speed, flight level or wander.
- **FIGHT** Fight, Go to closest enemy and attack when in range.

**Type** enum in ['GOAL', 'AVOID', 'AVOID\_COLLISION', 'SEPARATE', 'FLOCK', 'FOLLOW\_LEADER', 'AVERAGE\_SPEED', 'FIGHT'], default 'GOAL', (readonly)

**use\_in\_air**

Use rule when boid is flying

**Type** boolean, default False



**use\_on\_land**

Use rule when boid is on land

**Type** boolean, default False**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `BoidSettings.active_boid_state`
- `BoidState.active_boid_rule`
- `BoidState.rules`

**2.4.64 BoidRuleAverageSpeed(BoidRule)**base classes — `bpy_struct`, `BoidRule`**class** `bpy.types.BoidRuleAverageSpeed` (*BoidRule*)**level**

How much velocity's z-component is kept constant

**Type** float in [0, 1], default 0.0**speed**

Percentage of maximum speed

**Type** float in [0, 1], default 0.0**wander**

How fast velocity's direction is randomized

**Type** float in [0, 1], default 0.0

### Inherited Properties

- `bpy_struct.id_data`
- `BoidRule.name`
- `BoidRule.use_in_air`
- `BoidRule.use_on_land`
- `BoidRule.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.65 BoidRuleAvoid(BoidRule)

base classes — `bpy_struct`, `BoidRule`

**class** `bpy.types.BoidRuleAvoid` (*BoidRule*)

### **fear\_factor**

Avoid object if danger from it is above this threshold

**Type** float in [0, 100], default 0.0

### **object**

Object to avoid

**Type** `Object`

### **use\_predict**

Predict target movement

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`

- `BoidRule.name`
- `BoidRule.use_in_air`
- `BoidRule.use_on_land`
- `BoidRule.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.66 BoidRuleAvoidCollision(BoidRule)

base classes — `bpy_struct`, `BoidRule`

class `bpy.types.BoidRuleAvoidCollision` (*BoidRule*)

### **look\_ahead**

Time to look ahead in seconds

**Type** float in [0, 100], default 0.0

### **use\_avoid**

Avoid collision with other boids

**Type** boolean, default False

### **use\_avoid\_collision**

Avoid collision with deflector objects

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `BoidRule.name`
- `BoidRule.use_in_air`
- `BoidRule.use_on_land`
- `BoidRule.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.67 BoidRuleFight(BoidRule)

base classes — `bpy_struct`, `BoidRule`

**class** `bpy.types.BoidRuleFight` (*BoidRule*)

#### **distance**

Attack boids at max this distance

**Type** float in [0, 100], default 0.0

#### **flee\_distance**

Flee to this distance

**Type** float in [0, 100], default 0.0

### Inherited Properties

- `bpy_struct.id_data`
- `BoidRule.name`
- `BoidRule.use_in_air`
- `BoidRule.use_on_land`
- `BoidRule.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`

- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.68 BoidRuleFollowLeader(BoidRule)

base classes — `bpy_struct`, `BoidRule`

**class** `bpy.types.BoidRuleFollowLeader` (*BoidRule*)

### **distance**

Distance behind leader to follow

**Type** float in [0, 100], default 0.0

### **object**

Follow this object instead of a boid

**Type** `Object`

### **queue\_count**

How many boids in a line

**Type** int in [0, 100], default 0

### **use\_line**

Follow leader in a line

**Type** boolean, default False

## Inherited Properties

- `bpy_struct.id_data`
- `BoidRule.name`
- `BoidRule.use_in_air`
- `BoidRule.use_on_land`
- `BoidRule.type`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`

- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.69 BoidRuleGoal(BoidRule)

base classes — `bpy_struct`, `BoidRule`

**class** `bpy.types.BoidRuleGoal` (*BoidRule*)

### **object**

Goal object

**Type** `Object`

### **use\_predict**

Predict target movement

**Type** `boolean`, default `False`

### Inherited Properties

- `bpy_struct.id_data`
- `BoidRule.name`
- `BoidRule.use_in_air`
- `BoidRule.use_on_land`
- `BoidRule.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.70 BoidSettings(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.BoidSettings` (*bpy\_struct*)

Settings for boid physics

**accuracy**

Accuracy of attack

**Type** float in [0, 1], default 0.0

**active\_boid\_state**

**Type** `BoidRule`, (readonly)

**active\_boid\_state\_index**

**Type** int in [0, inf], default 0

**aggression**

Boid will fight this times stronger enemy

**Type** float in [0, 100], default 0.0

**air\_acc\_max**

Maximum acceleration in air (relative to maximum speed)

**Type** float in [0, 1], default 0.0

**air\_ave\_max**

Maximum angular velocity in air (relative to 180 degrees)

**Type** float in [0, 1], default 0.0

**air\_personal\_space**

Radius of boids personal space in air (% of particle size)

**Type** float in [0, 10], default 0.0

**air\_speed\_max**

Maximum speed in air

**Type** float in [0, 100], default 0.0

**air\_speed\_min**

Minimum speed in air (relative to maximum speed)

**Type** float in [0, 1], default 0.0

**bank**

Amount of rotation around velocity vector on turns

**Type** float in [0, 2], default 0.0

**health**

Initial boid health when born

**Type** float in [0, 100], default 0.0

**height**

Boid height relative to particle size

**Type** float in [0, 2], default 0.0

**land\_acc\_max**

Maximum acceleration on land (relative to maximum speed)

**Type** float in [0, 1], default 0.0

**land\_ave\_max**

Maximum angular velocity on land (relative to 180 degrees)

**Type** float in [0, 1], default 0.0

**land\_jump\_speed**

Maximum speed for jumping

**Type** float in [0, 100], default 0.0

**land\_personal\_space**

Radius of boids personal space on land (% of particle size)

**Type** float in [0, 10], default 0.0

**land\_smooth**

How smoothly the boids land

**Type** float in [0, 10], default 0.0

**land\_speed\_max**

Maximum speed on land

**Type** float in [0, 100], default 0.0

**land\_stick\_force**

How strong a force must be to start effecting a boid on land

**Type** float in [0, 1000], default 0.0

**pitch**

Amount of rotation around side vector

**Type** float in [0, 2], default 0.0

**range**

Maximum distance from which a boid can attack

**Type** float in [0, 100], default 0.0

**states**

**Type** bpy\_prop\_collection of BoidState, (readonly)

**strength**

Maximum caused damage on attack per second

**Type** float in [0, 100], default 0.0

**use\_climb**

Allow boids to climb goal objects

**Type** boolean, default False

**use\_flight**

Allow boids to move in air

**Type** boolean, default False

**use\_land**

Allow boids to move on land

**Type** boolean, default False



## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `ParticleSystem.boids`

### 2.4.71 BoidState(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.BoidState` (*bpy\_struct*)

Boid state for boid physics

**active\_boid\_rule**

**Type** `BoidRule`, (readonly)

**active\_boid\_rule\_index**

**Type** `int` in `[0, inf]`, default `0`

**falloff**

**Type** `float` in `[0, 10]`, default `0.0`

**name**

Boid state name

**Type** `string`, default `""`

**rule\_fuzzy**

**Type** `float` in `[0, 1]`, default `0.0`

**rules**

**Type** `bpy_prop_collection` of `BoidRule`, (readonly)

**ruleset\_type**

How the rules in the list are evaluated

- FUZZY Fuzzy, Rules are gone through top to bottom. Only the first rule that effect above fuzziness threshold is evaluated.
- RANDOM Random, A random rule is selected for each boid.
- AVERAGE Average, All rules are averaged.

**Type** enum in ['FUZZY', 'RANDOM', 'AVERAGE'], default 'FUZZY'

**volume**

**Type** float in [0, 100], default 0.0

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `BoidSettings.states`

**2.4.72 Bone(`bpy_struct`)**

base class — `bpy_struct`

**class** `bpy.types.Bone` (`bpy_struct`)

Bone in an Armature datablock

**bbone\_in**

Length of first Bezier Handle (for B-Bones only)

**Type** float in [0, 2], default 0.0

**bbone\_out**

Length of second Bezier Handle (for B-Bones only)

**Type** float in [0, 2], default 0.0

**bbone\_segments**

Number of subdivisions of bone (for B-Bones only)

**Type** int in [1, 32], default 0

**bbone\_x**

B-Bone X size

**Type** float in [0, 1000], default 0.0

**bbone\_z**

B-Bone Z size

**Type** float in [0, 1000], default 0.0

**children**

Bones which are children of this bone

**Type** `bpy_prop_collection` of `Bone`, (readonly)

**envelope\_distance**

Bone deformation distance (for Envelope deform only)

**Type** float in [0, 1000], default 0.0

**envelope\_weight**

Bone deformation weight (for Envelope deform only)

**Type** float in [0, 1000], default 0.0

**head**

Location of head end of the bone relative to its parent

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**head\_local**

Location of head end of the bone relative to armature

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**head\_radius**

Radius of head of bone (for Envelope deform only)

**Type** float in [0, inf], default 0.0

**hide**

Bone is not visible when it is not in Edit Mode (i.e. in Object or Pose Modes)

**Type** boolean, default False

**hide\_select**

Bone is able to be selected

**Type** boolean, default False

**layers**

Layers bone exists in

**Type** boolean array of 32 items, default (False, False)

**matrix**

3x3 bone matrix

**Type** float array of 9 items in [-inf, inf], default (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)

**matrix\_local**

4x4 bone matrix relative to armature

**Type** float array of 16 items in [-inf, inf], default (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)

**name**

**Type** string, default ""

**parent**

Parent bone (in same Armature)

**Type** `Bone`, (readonly)

**select**

**Type** boolean, default False

**select\_head**

**Type** boolean, default False

**select\_tail**

**Type** boolean, default False

**show\_wire**

Bone is always drawn as Wireframe regardless of viewport draw mode (useful for non-obstructive custom bone shapes)

**Type** boolean, default False

**tail**

Location of tail end of the bone

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**tail\_local**

Location of tail end of the bone relative to armature

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**tail\_radius**

Radius of tail of bone (for Envelope deform only)

**Type** float in [0, inf], default 0.0

**use\_connect**

When bone has a parent, bone's head is stuck to the parent's tail

**Type** boolean, default False, (readonly)

**use\_cyclic\_offset**

When bone doesn't have a parent, it receives cyclic offset effects

**Type** boolean, default False

**use\_deform**

Bone does not deform any geometry

**Type** boolean, default False

**use\_envelope\_multiply**

When deforming bone, multiply effects of Vertex Group weights with Envelope influence

**Type** boolean, default False

**use\_inherit\_rotation**

Bone inherits rotation or scale from parent bone

**Type** boolean, default False

**use\_inherit\_scale**

Bone inherits scaling from parent bone

**Type** boolean, default False

**use\_local\_location**

Bone location is set in local space

**Type** boolean, default False

**basename**

The name of this bone before any '.' character (readonly)

**center**

The midpoint between the head and the tail. (readonly)

**children**

A list of all the bones children. (readonly)

**children\_recursive**

A list of all children from this bone. (readonly)

**children\_recursive\_basename**

Returns a chain of children with the same base name as this bone. Only direct chains are supported, forks caused by multiple children with matching base names will terminate the function and not be returned. (readonly)

**length**

**The distance from head to tail**, when set the head is moved to fit the length.

**parent\_recursive**

A list of parents, starting with the immediate parent (readonly)

**vector**

**The direction this bone is pointing.** Utility function for (tail - head)

(readonly)

**x\_axis**

Vector pointing down the x-axis of the bone. (readonly)

**y\_axis**

Vector pointing down the x-axis of the bone. (readonly)

**z\_axis**

Vector pointing down the x-axis of the bone. (readonly)

**evaluate\_envelope** (*point*)

Calculate bone envelope at given point

**Parameters** **point** (*float array of 3 items in [-inf, inf]*) – Point, Position in 3d space to evaluate

**Returns** Factor, Envelope factor

**Return type** float in [-inf, inf]

**parent\_index** (*parent\_test*)

The same as ‘bone in other\_bone.parent\_recursive’ but saved generating a list.

**translate** (*vec*)

Utility function to add *vec* to the head and tail of this bone

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Armature.bones`
- `ArmatureBones.active`
- `Bone.children`
- `Bone.parent`
- `PoseBone.bone`

## 2.4.73 BoneGroup(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.BoneGroup` (*bpy\_struct*)  
Groups of Pose Channels (Bones)

**color\_set**

Custom color set to use

**Type** enum in ['DEFAULT', 'THEME01', 'THEME02', 'THEME03', 'THEME04', 'THEME05', 'THEME06', 'THEME07', 'THEME08', 'THEME09', 'THEME10', 'THEME11', 'THEME12', 'THEME13', 'THEME14', 'THEME15', 'THEME16', 'THEME17', 'THEME18', 'THEME19', 'THEME20', 'CUSTOM'], default 'DEFAULT'

**colors**

Copy of the colors associated with the group's color set

**Type** `ThemeBoneColorSet`, (readonly, never None)

**name**

**Type** string, default ""

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `BoneGroups.active`
- `Pose.bone_groups`
- `PoseBone.bone_group`

**2.4.74 BoneGroups(`bpy_struct`)**

base class — `bpy_struct`

**class** `bpy.types.BoneGroups` (*bpy\_struct*)  
Collection of bone groups

**active**

Active bone group for this pose

**Type** `BoneGroup`

**active\_index**

Active index in bone groups array

**Type** int in [-inf, inf], default 0

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `Pose.bone_groups`

## 2.4.75 BooleanModifier(Modifier)

base classes — `bpy_struct`, `Modifier`

**class** `bpy.types.BooleanModifier` (*Modifier*)

Boolean operations modifier

**object**

Mesh object to use for Boolean operation

**Type** `Object`

**operation**

- **INTERSECT** Intersect, Keeps the part of the mesh that intersects with the other selected object.
- **UNION** Union, Combines two meshes in an additive way.
- **DIFFERENCE** Difference, Combines two meshes in a subtractive way.

**Type** enum in ['INTERSECT', 'UNION', 'DIFFERENCE'], default 'INTERSECT'



## Inherited Properties

- `bpy_struct.id_data`
- `Modifier.name`
- `Modifier.use_apply_on_spline`
- `Modifier.show_in_editmode`
- `Modifier.show_expanded`
- `Modifier.show_on_cage`
- `Modifier.show_viewport`
- `Modifier.show_render`
- `Modifier.type`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.76 BooleanProperty(Property)

base classes — `bpy_struct`, `Property`

**class** `bpy.types.BooleanProperty` (*Property*)  
RNA boolean property definition

**array\_length**

Maximum length of the array, 0 means unlimited

**Type** `int` in `[0, inf]`, default 0, (readonly)

**default**

Default value for this number

**Type** `boolean`, default `False`, (readonly)

**default\_array**

Default value for this array

**Type** `boolean array` of 3 items, default `(False, False, False)`, (readonly)

### Inherited Properties

- `bpy_struct.id_data`
- `Property.name`
- `Property.is_animatable`
- `Property.srna`
- `Property.description`
- `Property.is_enum_flag`
- `Property.is_hidden`
- `Property.identifier`
- `Property.is_never_none`
- `Property.is_readonly`
- `Property.is_registered`
- `Property.is_registered_optional`
- `Property.is_required`
- `Property.is_output`
- `Property.is_runtime`
- `Property.is_skip_save`
- `Property.subtype`
- `Property.type`
- `Property.unit`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.77 Brush(ID)

base classes — `bpy_struct`, `ID`

**class** `bpy.types.Brush` (*ID*)

Brush datablock for storing brush settings for painting and sculpting

**auto\_smooth\_factor**

Amount of smoothing to automatically apply to each stroke

**Type** float in [0, 1], default 0.0

**blend**

Brush blending mode

- MIX Mix, Use mix blending mode while painting.
- ADD Add, Use add blending mode while painting.
- SUB Subtract, Use subtract blending mode while painting.
- MUL Multiply, Use multiply blending mode while painting.
- LIGHTEN Lighten, Use lighten blending mode while painting.
- DARKEN Darken, Use darken blending mode while painting.
- ERASE\_ALPHA Erase Alpha, Erase alpha while painting.
- ADD\_ALPHA Add Alpha, Add alpha while painting.

**Type** enum in ['MIX', 'ADD', 'SUB', 'MUL', 'LIGHTEN', 'DARKEN', 'ERASE\_ALPHA', 'ADD\_ALPHA'], default 'MIX'

**clone\_alpha**

Opacity of clone image display

**Type** float in [0, 1], default 0.0

**clone\_image**

Image for clone tool

**Type** [Image](#)

**clone\_offset**

**Type** float array of 2 items in [-inf, inf], default (0.0, 0.0)

**color**

**Type** float array of 3 items in [0, 1], default (0.0, 0.0, 0.0)

**crease\_pinch\_factor**

How much the crease brush pinches

**Type** float in [0, 1], default 0.666667

**cursor\_color\_add**

Color of cursor when adding

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**cursor\_color\_subtract**

Color of cursor when subtracting

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**curve**

Editable falloff curve

**Type** [CurveMapping](#), (readonly, never None)

**direction**

- ADD Add, Add effect of brush.
- SUBTRACT Subtract, Subtract effect of brush.

**Type** enum in ['ADD', 'SUBTRACT'], default 'ADD'

**height**

Affectable height of brush (layer height for layer tool, i.e.)

**Type** float in [0, 1], default 0.5

**icon\_filepath**

File path to brush icon

**Type** string, default ""

**image\_tool**

**Type** enum in ['DRAW', 'SOFTEN', 'SMEAR', 'CLONE'], default 'DRAW'

**jitter**

Jitter the position of the brush while painting

**Type** float in [0, 1], default 0.0

**normal\_weight**

How much grab will pull vertexes out of surface during a grab

**Type** float in [0, 1], default 0.0

**plane\_offset**

Adjusts plane on which the brush acts towards or away from the object surface

**Type** float in [-2, 2], default 0.0

**plane\_trim**

If a vertex is further from offset plane than this then it is not affected

**Type** float in [0, 1], default 0.5

**rate**

Interval between paints for Airbrush

**Type** float in [0.0001, 10000], default 0.0

**sculpt\_plane**

**Type** enum in ['AREA', 'VIEW', 'X', 'Y', 'Z'], default 'AREA'

**sculpt\_tool**

**Type** enum in ['BLOB', 'CLAY', 'CREASE', 'DRAW', 'FILL', 'FLATTEN', 'GRAB', 'INFLATE', 'LAYER', 'NUDGE', 'PINCH', 'ROTATE', 'SCRAPE', 'SMOOTH', 'SNAKE\_HOOK', 'THUMB'], default 'BLOB'

**size**

Radius of the brush in pixels

**Type** int in [1, 2000], default 0

**smooth\_stroke\_factor**

Higher values give a smoother stroke

**Type** float in [0.5, 0.99], default 0.0

**smooth\_stroke\_radius**

Minimum distance from last point before stroke continues

**Type** int in [10, 200], default 0

**spacing**

Spacing between brush daubs as a percentage of brush diameter

**Type** int in [1, 1000], default 0

#### **strength**

How powerful the effect of the brush is when applied

**Type** float in [0, 10], default 0.5

#### **stroke\_method**

- DOTS Dots, Apply paint on each mouse move step.
- DRAG\_DOT Drag Dot, Allows a single dot to be carefully positioned.
- SPACE Space, Limit brush application to the distance specified by spacing.
- ANCHORED Anchored, Keep the brush anchored to the initial location.
- AIRBRUSH Airbrush, Keep applying paint effect while holding mouse (spray).

**Type** enum in ['DOTS', 'DRAG\_DOT', 'SPACE', 'ANCHORED', 'AIRBRUSH'], default 'DOTS'

#### **texture**

**Type** [Texture](#)

#### **texture\_angle\_source\_no\_random**

- USER User, Rotate the brush texture by given angle.
- RAKE Rake, Rotate the brush texture to match the stroke direction.

**Type** enum in ['USER', 'RAKE'], default 'USER'

#### **texture\_angle\_source\_random**

- USER User, Rotate the brush texture by given angle.
- RAKE Rake, Rotate the brush texture to match the stroke direction.
- RANDOM Random, Rotate the brush texture at random.

**Type** enum in ['USER', 'RAKE', 'RANDOM'], default 'USER'

#### **texture\_overlay\_alpha**

**Type** int in [1, 100], default 0

#### **texture\_sample\_bias**

Value added to texture samples

**Type** float in [-1, 1], default 0.0

#### **texture\_slot**

**Type** [BrushTextureSlot](#), (readonly)

#### **unprojected\_radius**

Radius of brush in Blender units

**Type** float in [0.001, inf], default 0.0

#### **use\_accumulate**

Accumulate stroke dabs on top of each other

**Type** boolean, default False

**use\_adaptive\_space**

Space daubs according to surface orientation instead of screen space

**Type** boolean, default False

**use\_airbrush**

Keep applying paint effect while holding mouse (spray)

**Type** boolean, default False

**use\_alpha**

When this is disabled, lock alpha while painting

**Type** boolean, default False

**use\_anchor**

Keep the brush anchored to the initial location

**Type** boolean, default False

**use\_custom\_icon**

Set the brush icon from an image file

**Type** boolean, default False

**use\_edge\_to\_edge**

Drag anchor brush from edge-to-edge

**Type** boolean, default False

**use\_fixed\_texture**

Keep texture origin in fixed position

**Type** boolean, default False

**use\_frontface**

Brush only affects vertexes that face the viewer

**Type** boolean, default False

**use\_inverse\_smooth\_pressure**

Lighter pressure causes more smoothing to be applied

**Type** boolean, default False

**use\_locked\_size**

When locked brush stays same size relative to object; when unlocked brush size is given in pixels

**Type** boolean, default False

**use\_offset\_pressure**

Enable tablet pressure sensitivity for offset

**Type** boolean, default False

**use\_original\_normal**

When locked keep using normal of surface where stroke was initiated

**Type** boolean, default False

**use\_paint\_image**

Use this brush in texture paint mode

**Type** boolean, default False

**use\_paint\_sculpt**

Use this brush in sculpt mode

**Type** boolean, default False

**use\_paint\_vertex**

Use this brush in vertex paint mode

**Type** boolean, default False

**use\_paint\_weight**

Use this brush in weight paint mode

**Type** boolean, default False

**use\_persistent**

Sculpts on a persistent layer of the mesh

**Type** boolean, default False

**use\_plane\_trim**

Enable Plane Trim

**Type** boolean, default False

**use\_pressure\_jitter**

Enable tablet pressure sensitivity for jitter

**Type** boolean, default False

**use\_pressure\_size**

Enable tablet pressure sensitivity for size

**Type** boolean, default False

**use\_pressure\_spacing**

Enable tablet pressure sensitivity for spacing

**Type** boolean, default False

**use\_pressure\_strength**

Enable tablet pressure sensitivity for strength

**Type** boolean, default False

**use\_rake**

Rotate the brush texture to match the stroke direction

**Type** boolean, default False

**use\_random\_rotation**

Rotate the brush texture at random

**Type** boolean, default False

**use\_restore\_mesh**

Allows a single dot to be carefully positioned

**Type** boolean, default False

**use\_smooth\_stroke**

Brush lags behind mouse and follows a smoother path

**Type** boolean, default False

**use\_space**

Limit brush application to the distance specified by spacing

**Type** boolean, default False

**use\_space\_atten**

Automatically adjusts strength to give consistent results for different spacings

**Type** boolean, default False

**use\_texture\_overlay**

Show texture in viewport

**Type** boolean, default False

**use\_wrap**

Enable torus wrapping while painting

**Type** boolean, default False

**vertex\_tool**

- MIX Mix, Use mix blending mode while painting.
- ADD Add, Use add blending mode while painting.
- SUB Subtract, Use subtract blending mode while painting.
- MUL Multiply, Use multiply blending mode while painting.
- BLUR Blur, Blur the color with surrounding values.
- LIGHTEN Lighten, Use lighten blending mode while painting.
- DARKEN Darken, Use darken blending mode while painting.

**Type** enum in ['MIX', 'ADD', 'SUB', 'MUL', 'BLUR', 'LIGHTEN', 'DARKEN'], default 'MIX'

**Inherited Properties**

- bpy\_struct.id\_data
- ID.name
- ID.use\_fake\_user
- ID.is\_updated
- ID.is\_updated\_data
- ID.library
- ID.tag
- ID.users

**Inherited Functions**

- bpy\_struct.as\_pointer
- bpy\_struct.callback\_add
- bpy\_struct.callback\_remove
- bpy\_struct.driver\_add
- bpy\_struct.driver\_remove
- bpy\_struct.get
- bpy\_struct.is\_property\_hidden
- bpy\_struct.is\_property\_set
- bpy\_struct.items
- bpy\_struct.keyframe\_delete
- bpy\_struct.keyframe\_insert
- bpy\_struct.keys



- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`

## References

- `BlendData.brushes`
- `BlendDataBrushes.new`
- `BlendDataBrushes.remove`
- `Paint.brush`

## 2.4.78 BrushTextureSlot(TextureSlot)

base classes — `bpy_struct`, `TextureSlot`

**class** `bpy.types.BrushTextureSlot` (*TextureSlot*)

Texture slot for textures in a Brush datablock

### **angle**

Defines brush texture rotation

**Type** float in [0, 6.28319], default 0.0

### **map\_mode**

**Type** enum in ['FIXED', 'TILED', '3D'], default 'FIXED'

## Inherited Properties

- `bpy_struct.id_data`
- `TextureSlot.name`
- `TextureSlot.blend_type`
- `TextureSlot.color`
- `TextureSlot.default_value`
- `TextureSlot.invert`
- `TextureSlot.offset`
- `TextureSlot.output_node`
- `TextureSlot.use_rgb_to_intensity`
- `TextureSlot.scale`
- `TextureSlot.use_stencil`
- `TextureSlot.texture`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`

- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Brush.texture_slot`

## 2.4.79 BuildModifier(Modifier)

base classes — `bpy_struct`, `Modifier`

**class** `bpy.types.BuildModifier` (*Modifier*)

Build effect modifier

**frame\_duration**

Specify the total time the build effect requires

**Type** float in [1, 300000], default 0.0

**frame\_start**

Specify the start frame of the effect

**Type** float in [-300000, 300000], default 0.0

**seed**

Specify the seed for random if used

**Type** int in [1, 300000], default 0

**use\_random\_order**

Randomize the faces or edges during build

**Type** boolean, default False

## Inherited Properties

- `bpy_struct.id_data`
- `Modifier.name`
- `Modifier.use_apply_on_spline`
- `Modifier.show_in_editmode`
- `Modifier.show_expanded`
- `Modifier.show_on_cage`
- `Modifier.show_viewport`
- `Modifier.show_render`
- `Modifier.type`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.80 Camera(ID)

base classes — `bpy_struct`, `ID`

**class** `bpy.types.Camera` (*ID*)

Camera datablock for storing camera settings

### **angle**

Perspective Camera lens field of view in degrees

**Type** float in [0.00640536, 3.01675], default 0.0

### **animation\_data**

Animation data for this datablock

**Type** `AnimData`, (readonly)

### **clip\_end**

Camera far clipping distance

**Type** float in [1, inf], default 0.0

### **clip\_start**

Camera near clipping distance

**Type** float in [0.001, inf], default 0.0

### **dof\_distance**

Distance to the focus point for depth of field

**Type** float in [0, 5000], default 0.0

### **dof\_object**

Use this object to define the depth of field focal point

**Type** `Object`

### **draw\_size**

Apparent size of the Camera object in the 3D View

**Type** float in [0.01, 1000], default 0.0

**lens**

Perspective Camera lens value in millimeters

**Type** float in [1, 5000], default 0.0

**lens\_unit**

Unit to edit lens in for the user interface

**Type** enum in ['MILLIMETERS', 'DEGREES'], default 'MILLIMETERS'

**ortho\_scale**

Orthographic Camera scale (similar to zoom)

**Type** float in [0.01, 4000], default 0.0

**passepartout\_alpha**

Opacity (alpha) of the darkened overlay in Camera view

**Type** float in [0, 1], default 0.0

**shift\_x**

Perspective Camera horizontal shift

**Type** float in [-10, 10], default 0.0

**shift\_y**

Perspective Camera vertical shift

**Type** float in [-10, 10], default 0.0

**show\_guide**

Draw overlay

**Type** enum set in {'CENTER', 'CENTER\_DIAGONAL', 'THIRDS', 'GOLDEN', 'GOLDEN\_TRIANGLE\_A', 'GOLDEN\_TRIANGLE\_B', 'HARMONY\_TRIANGLE\_A', 'HARMONY\_TRIANGLE\_B'}, default {'CENTER'}

**show\_limits**

Draw the clipping range and focus point on the camera

**Type** boolean, default False

**show\_mist**

Draw a line from the Camera to indicate the mist area

**Type** boolean, default False

**show\_name**

Show the active Camera's name in Camera view

**Type** boolean, default False

**show\_passepartout**

Show a darkened overlay outside the image area in Camera view

**Type** boolean, default False

**show\_title\_safe**

Show indicators for the title safe zone in Camera view

**Type** boolean, default False

**type**

Camera types

**Type** enum in ['PERSP', 'ORTHO'], default 'PERSP'

**use\_panorama**

Render the scene with a cylindrical camera for pseudo-fisheye lens effects

**Type** boolean, default False

**view\_frame** (*scene=None*)

Return 4 points for the cameras frame (before object transformation)

**Parameters** **scene** (*Scene*, (optional)) – Scene to use for aspect calculation, when omitted 1:1 aspect is used

**Return** (**result\_1, result\_2, result\_3, result\_4**) *result\_1*, Result, float array of 3 items in [-inf, inf]

*result\_2*, Result, float array of 3 items in [-inf, inf]

*result\_3*, Result, float array of 3 items in [-inf, inf]

*result\_4*, Result, float array of 3 items in [-inf, inf]

**Inherited Properties**

- bpy\_struct.id\_data
- ID.name
- ID.use\_fake\_user
- ID.is\_updated
- ID.is\_updated\_data
- ID.library
- ID.tag
- ID.users

**Inherited Functions**

- bpy\_struct.as\_pointer
- bpy\_struct.callback\_add
- bpy\_struct.callback\_remove
- bpy\_struct.driver\_add
- bpy\_struct.driver\_remove
- bpy\_struct.get
- bpy\_struct.is\_property\_hidden
- bpy\_struct.is\_property\_set
- bpy\_struct.items
- bpy\_struct.keyframe\_delete
- bpy\_struct.keyframe\_insert
- bpy\_struct.keys
- bpy\_struct.path\_from\_id
- bpy\_struct.path\_resolve
- bpy\_struct.type\_recast
- bpy\_struct.values
- ID.copy
- ID.user\_clear
- ID.animation\_data\_create
- ID.animation\_data\_clear
- ID.update\_tag

## References

- `BlendData.cameras`
- `BlendDataCameras.new`
- `BlendDataCameras.remove`

### 2.4.81 CameraActuator(Actuator)

base classes — `bpy_struct`, `Actuator`

class `bpy.types.CameraActuator` (*Actuator*)

#### **axis**

Specify the axis the Camera will try to get behind

- X X, Camera tries to get behind the X axis.
- Y Y, Camera tries to get behind the Y axis.

**Type** enum in ['X', 'Y'], default 'X'

#### **damping**

Specify the strength of the constraint that drive the camera behind the target

**Type** float in [0, 10], default 0.0

#### **height**

**Type** float in [-inf, inf], default 0.0

#### **max**

**Type** float in [-inf, inf], default 0.0

#### **min**

**Type** float in [-inf, inf], default 0.0

#### **object**

Look at this Object

**Type** Object

## Inherited Properties

- `bpy_struct.id_data`
- `Actuator.name`
- `Actuator.show_expanded`
- `Actuator.pin`
- `Actuator.type`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`

- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Actuator.link`
- `Actuator.unlink`

## 2.4.82 CastModifier(Modifier)

base classes — `bpy_struct`, `Modifier`

**class** `bpy.types.CastModifier` (*Modifier*)

Cast modifier to cast to other shapes

**cast\_type**

Target object shape

**Type** enum in ['SPHERE', 'CYLINDER', 'CUBOID'], default 'SPHERE'

**factor**

**Type** float in [-inf, inf], default 0.0

**object**

Control object: if available, its location determines the center of the effect

**Type** `Object`

**radius**

Only deform vertices within this distance from the center of the effect (leave as 0 for infinite.)

**Type** float in [0, inf], default 0.0

**size**

Size of projection shape (leave as 0 for auto.)

**Type** float in [0, inf], default 0.0

**use\_radius\_as\_size**

Use radius as size of projection shape (0 = auto)

**Type** boolean, default False

**use\_transform**

Use object transform to control projection shape

**Type** boolean, default False

**use\_x**

**Type** boolean, default False

**use\_y**

**Type** boolean, default False

**use\_z**

**Type** boolean, default False

**vertex\_group**

Vertex group name

**Type** string, default ""

### Inherited Properties

- `bpy_struct.id_data`
- `Modifier.name`
- `Modifier.use_apply_on_spline`
- `Modifier.show_in_editmode`
- `Modifier.show_expanded`
- `Modifier.show_on_cage`
- `Modifier.show_viewport`
- `Modifier.show_render`
- `Modifier.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.83 ChannelDriverVariables(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.ChannelDriverVariables` (*bpy\_struct*)  
Collection of channel driver Variables

**new** ()

Add a new variable for the driver

**Returns** Newly created Driver Variable

**Return type** `DriverVariable`



**remove** (*variable*)

Remove an existing variable from the driver

**Parameters** **variable** (`DriverVariable`, (never None)) – Variable to remove from the driver

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Driver.variables`

## 2.4.84 ChildOfConstraint(Constraint)

base classes — `bpy_struct`, `Constraint`

**class** `bpy.types.ChildOfConstraint` (*Constraint*)

Creates constraint-based parent-child relationship

**inverse\_matrix**

Transformation matrix to apply before

**Type** float array of 16 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)`

**subtarget**

**Type** string, default ""

**target**

Target Object

**Type** `Object`

**use\_location\_x**  
Use X Location of Parent  
**Type** boolean, default False

**use\_location\_y**  
Use Y Location of Parent  
**Type** boolean, default False

**use\_location\_z**  
Use Z Location of Parent  
**Type** boolean, default False

**use\_rotation\_x**  
Use X Rotation of Parent  
**Type** boolean, default False

**use\_rotation\_y**  
Use Y Rotation of Parent  
**Type** boolean, default False

**use\_rotation\_z**  
Use Z Rotation of Parent  
**Type** boolean, default False

**use\_scale\_x**  
Use X Scale of Parent  
**Type** boolean, default False

**use\_scale\_y**  
Use Y Scale of Parent  
**Type** boolean, default False

**use\_scale\_z**  
Use Z Scale of Parent  
**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `Constraint.name`
- `Constraint.active`
- `Constraint.mute`
- `Constraint.show_expanded`
- `Constraint.influence`
- `Constraint.error_location`
- `Constraint.owner_space`
- `Constraint.is_proxy_local`
- `Constraint.error_rotation`
- `Constraint.target_space`
- `Constraint.type`
- `Constraint.is_valid`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.85 ChildParticle(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.ChildParticle` (*bpy\_struct*)  
 Child particle interpolated from simulated or edited particles

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `ParticleSystem.child_particles`

## 2.4.86 ClampToConstraint(Constraint)

base classes — `bpy_struct`, `Constraint`

**class** `bpy.types.ClampToConstraint` (*Constraint*)

Constrains an object's location to the nearest point along the target path

**main\_axis**

Main axis of movement

**Type** enum in ['CLAMPTO\_AUTO', 'CLAMPTO\_X', 'CLAMPTO\_Y', 'CLAMPTO\_Z'], default 'CLAMPTO\_AUTO'

**target**

Target Object

**Type** `Object`

**use\_cyclic**

Treat curve as cyclic curve (no clamping to curve bounding box)

**Type** boolean, default False

## Inherited Properties

- `bpy_struct.id_data`
- `Constraint.name`
- `Constraint.active`
- `Constraint.mute`
- `Constraint.show_expanded`
- `Constraint.influence`
- `Constraint.error_location`
- `Constraint.owner_space`
- `Constraint.is_proxy_local`
- `Constraint.error_rotation`
- `Constraint.target_space`
- `Constraint.type`
- `Constraint.is_valid`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`

- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.87 ClothCollisionSettings(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.ClothCollisionSettings` (*bpy\_struct*)

Cloth simulation settings for self collision and collision with other objects

**collision\_quality**

How many collision iterations should be done. (higher is better quality but slower)

**Type** int in [1, 20], default 0

**distance\_min**

Minimum distance between collision objects before collision response takes in

**Type** float in [0.001, 1], default 0.0

**distance\_repel**

Maximum distance to apply repulsion force, must be greater than minimum distance

**Type** float in [0.001, 10], default 0.005

**friction**

Friction force if a collision happened (higher = less movement)

**Type** float in [0, 80], default 0.0

**group**

Limit colliders to this Group

**Type** `Group`

**repel\_force**

Repulsion force to apply on cloth when close to colliding

**Type** float in [0, 20], default 1.0

**self\_collision\_quality**

How many self collision iterations should be done (higher is better quality but slower)

**Type** int in [1, 10], default 0

**self\_distance\_min**

0.5 means no distance at all, 1.0 is maximum distance

**Type** float in [0.5, 1], default 0.0

**self\_friction**

Friction/damping with self contact

**Type** float in [0, 80], default 0.0

**use\_collision**

Enable collisions with other objects

**Type** boolean, default False

**use\_self\_collision**

Enable self collisions

**Type** boolean, default False

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `ClothModifier.collision_settings`

## 2.4.88 ClothModifier(Modifier)

base classes — `bpy_struct`, `Modifier`

**class** `bpy.types.ClothModifier` (*Modifier*)

Cloth simulation modifier

**collision\_settings**

**Type** `ClothCollisionSettings`, (readonly, never None)

**point\_cache**

**Type** `PointCache`, (readonly, never None)

**settings**

**Type** `ClothSettings`, (readonly, never None)

## Inherited Properties

- `bpy_struct.id_data`
- `Modifier.name`
- `Modifier.use_apply_on_spline`
- `Modifier.show_in_editmode`
- `Modifier.show_expanded`
- `Modifier.show_on_cage`
- `Modifier.show_viewport`
- `Modifier.show_render`
- `Modifier.type`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `ParticleSystem.cloth`

## 2.4.89 ClothSettings(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.ClothSettings` (*`bpy_struct`*)

Cloth simulation settings for an object

### **air\_damping**

Air has normally some thickness which slows falling things down

**Type** float in [0, 10], default 0.0

### **bending\_stiffness**

Wrinkle coefficient (higher = less smaller but more big wrinkles)

**Type** float in [0, 10000], default 0.0

### **bending\_stiffness\_max**

Maximum bending stiffness value

**Type** float in [0, 10000], default 0.0

**collider\_friction**

**Type** float in [0, 1], default 0.0

**effector\_weights**

**Type** `EffectorWeights`, (readonly)

**goal\_default**

Default Goal (vertex target position) value, when no Vertex Group used

**Type** float in [0, 1], default 0.0

**goal\_friction**

Goal (vertex target position) friction

**Type** float in [0, 50], default 0.0

**goal\_max**

Goal maximum, vertex group weights are scaled to match this range

**Type** float in [0, 1], default 0.0

**goal\_min**

Goal minimum, vertex group weights are scaled to match this range

**Type** float in [0, 1], default 0.0

**goal\_spring**

Goal (vertex target position) spring stiffness

**Type** float in [0, 0.999], default 0.0

**gravity**

Gravity or external force vector

**Type** float array of 3 items in [-100, 100], default (0.0, 0.0, 0.0)

**internal\_friction**

**Type** float in [0, 1], default 0.0

**mass**

Mass of cloth material

**Type** float in [0, 10], default 0.0

**pin\_stiffness**

Pin (vertex target position) spring stiffness

**Type** float in [0, 50], default 0.0

**pre\_roll**

Simulation starts on this frame

**Type** int in [0, 200], default 0

**quality**

Quality of the simulation in steps per frame (higher is better quality but slower)

**Type** int in [4, 80], default 0

**rest\_shape\_key**

Shape key to use the rest spring lengths from

**Type** `ShapeKey`



**spring\_damping**

Damping of cloth velocity (higher = more smooth, less jiggling)

**Type** float in [0, 50], default 0.0

**structural\_stiffness**

Overall stiffness of structure

**Type** float in [0, 10000], default 0.0

**structural\_stiffness\_max**

Maximum structural stiffness value

**Type** float in [0, 10000], default 0.0

**use\_pin\_cloth**

Enable pinning of cloth vertices to other objects/positions

**Type** boolean, default False

**use\_stiffness\_scale**

If enabled, stiffness can be scaled along a weight painted vertex group

**Type** boolean, default False

**vertex\_group\_bending**

Vertex group for fine control over bending stiffness

**Type** string, default ""

**vertex\_group\_mass**

Vertex Group for pinning of vertices

**Type** string, default ""

**vertex\_group\_structural\_stiffness**

Vertex group for fine control over structural stiffness

**Type** string, default ""

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`

- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `ClothModifier.settings`

## 2.4.90 CloudsTexture(Texture)

base classes — `bpy_struct`, `ID`, `Texture`

**class** `bpy.types.CloudsTexture` (*Texture*)

Procedural noise texture

### **cloud\_type**

Determines whether Noise returns grayscale or RGB values

**Type** enum in ['GREYSCALE', 'COLOR'], default 'GREYSCALE'

### **nabla**

Size of derivative offset used for calculating normal

**Type** float in [0.001, 0.1], default 0.0

### **noise\_basis**

Sets the noise basis used for turbulence

- `BLENDER_ORIGINAL` Blender Original, Noise algorithm - Blender original: Smooth interpolated noise.
- `ORIGINAL_PERLIN` Original Perlin, Noise algorithm - Original Perlin: Smooth interpolated noise.
- `IMPROVED_PERLIN` Improved Perlin, Noise algorithm - Improved Perlin: Smooth interpolated noise.
- `VORONOI_F1` Voronoi F1, Noise algorithm - Voronoi F1: Returns distance to the closest feature point.
- `VORONOI_F2` Voronoi F2, Noise algorithm - Voronoi F2: Returns distance to the 2nd closest feature point.
- `VORONOI_F3` Voronoi F3, Noise algorithm - Voronoi F3: Returns distance to the 3rd closest feature point.
- `VORONOI_F4` Voronoi F4, Noise algorithm - Voronoi F4: Returns distance to the 4th closest feature point.
- `VORONOI_F2_F1` Voronoi F2-F1, Noise algorithm - Voronoi F1-F2.
- `VORONOI_CRACKLE` Voronoi Crackle, Noise algorithm - Voronoi Crackle: Voronoi tessellation with sharp edges.
- `CELL_NOISE` Cell Noise, Noise algorithm - Cell Noise: Square cell tessellation.

**Type** enum in ['BLENDER\_ORIGINAL', 'ORIGINAL\_PERLIN', 'IMPROVED\_PERLIN', 'VORONOI\_F1', 'VORONOI\_F2', 'VORONOI\_F3', 'VORONOI\_F4', 'VORONOI\_F2\_F1', 'VORONOI\_CRACKLE', 'CELL\_NOISE'], default 'BLENDER\_ORIGINAL'

**noise\_depth**

Sets the depth of the cloud calculation

**Type** int in [0, 30], default 0

**noise\_scale**

Sets scaling for noise input

**Type** float in [0.0001, inf], default 0.0

**noise\_type**

- **SOFT\_NOISE** Soft, Generate soft noise (smooth transitions).
- **HARD\_NOISE** Hard, Generate hard noise (sharp transitions).

**Type** enum in ['SOFT\_NOISE', 'HARD\_NOISE'], default 'SOFT\_NOISE'

**users\_material**

Materials that use this texture (readonly)

**users\_object\_modifier**

Object modifiers that use this texture (readonly)

**Inherited Properties**

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`
- `Texture.animation_data`
- `Texture.intensity`
- `Texture.color_ramp`
- `Texture.contrast`
- `Texture.factor_blue`
- `Texture.factor_green`
- `Texture.factor_red`
- `Texture.node_tree`
- `Texture.saturation`
- `Texture.use_preview_alpha`
- `Texture.type`
- `Texture.use_color_ramp`
- `Texture.use_nodes`
- `Texture.users_material`
- `Texture.users_object_modifier`
- `Texture.users_material`
- `Texture.users_object_modifier`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`

- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`
- `Texture.evaluate`

## 2.4.91 CollectionProperty(Property)

base classes — `bpy_struct`, `Property`

**class** `bpy.types.CollectionProperty` (*Property*)  
RNA collection property to define lists, arrays and mappings

**fixed\_type**

Fixed pointer type, empty if variable type

**Type** `Struct`, (readonly)

### Inherited Properties

- `bpy_struct.id_data`
- `Property.name`
- `Property.is_animatable`
- `Property.srna`
- `Property.description`
- `Property.is_enum_flag`
- `Property.is_hidden`
- `Property.identifier`
- `Property.is_never_none`
- `Property.is_readonly`
- `Property.is_registered`
- `Property.is_registered_optional`
- `Property.is_required`
- `Property.is_output`
- `Property.is_runtime`
- `Property.is_skip_save`
- `Property.subtype`
- `Property.type`
- `Property.unit`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.92 CollisionModifier(Modifier)

base classes — `bpy_struct`, `Modifier`

**class** `bpy.types.CollisionModifier` (*Modifier*)  
Collision modifier defining modifier stack position used for collision

### settings

**Type** `CollisionSettings`, (readonly, never None)

## Inherited Properties

- `bpy_struct.id_data`
- `Modifier.name`
- `Modifier.use_apply_on_spline`
- `Modifier.show_in_editmode`
- `Modifier.show_expanded`
- `Modifier.show_on_cage`
- `Modifier.show_viewport`
- `Modifier.show_render`
- `Modifier.type`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`

- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.93 CollisionSensor(Sensor)

base classes — `bpy_struct`, `Sensor`

**class** `bpy.types.CollisionSensor` (*Sensor*)

Sensor to detect objects colliding with the current object, with more settings than the Touch sensor

**material**

Only look for Objects with this material (blank = all objects)

**Type** string, default ""

**property**

Only look for Objects with this property (blank = all objects)

**Type** string, default ""

**use\_material**

Toggle collision on material or property

**Type** boolean, default False

**use\_pulse**

Changes to the set of colliding objects generates pulse

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `Sensor.name`
- `Sensor.show_expanded`
- `Sensor.frequency`
- `Sensor.invert`
- `Sensor.use_level`
- `Sensor.pin`
- `Sensor.use_pulse_false_level`
- `Sensor.use_pulse_true_level`
- `Sensor.use_tap`
- `Sensor.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`

- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Sensor.link`
- `Sensor.unlink`

## 2.4.94 CollisionSettings(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.CollisionSettings` (*bpy\_struct*)  
Collision settings for object in physics simulation

### **absorption**

How much of effector force gets lost during collision with this object (in percent)

**Type** float in [0, 1], default 0.0

### **damping**

Amount of damping during collision

**Type** float in [0, 1], default 0.0

### **damping\_factor**

Amount of damping during particle collision

**Type** float in [0, 1], default 0.0

### **damping\_random**

Random variation of damping

**Type** float in [0, 1], default 0.0

### **friction\_factor**

Amount of friction during particle collision

**Type** float in [0, 1], default 0.0

### **friction\_random**

Random variation of friction

**Type** float in [0, 1], default 0.0

### **permeability**

Chance that the particle will pass through the mesh

**Type** float in [0, 1], default 0.0

### **stickiness**

Amount of stickiness to surface collision

**Type** float in [0, 10], default 0.0

**thickness\_inner**

Inner face thickness

**Type** float in [0.001, 1], default 0.0

**thickness\_outer**

Outer face thickness

**Type** float in [0.001, 1], default 0.0

**use**

Enable this objects as a collider for physics systems

**Type** boolean, default False

**use\_particle\_kill**

Kill collided particles

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `CollisionModifier.settings`
- `Object.collision`

## 2.4.95 ColorRamp(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.ColorRamp` (*bpy\_struct*)

Color ramp mapping a scalar value to a color



**elements**

**Type** `ColorRampElements` `bpy_prop_collection` of `ColorRampElement`, (read-only)

**interpolation**

**Type** enum in ['EASE', 'CARDINAL', 'LINEAR', 'B\_SPLINE', 'CONSTANT'], default 'LINEAR'

**evaluate** (*position*)

Evaluate `ColorRamp`

**Parameters** **position** (*float in [0, 1]*) – Position, Evaluate `ColorRamp` at position

**Returns** Color, Color at given position

**Return type** float array of 4 items in [-inf, inf]

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `CompositorNodeValToRGB.color_ramp`
- `Material.diffuse_ramp`
- `Material.specular_ramp`
- `PointDensity.color_ramp`
- `ShaderNodeValToRGB.color_ramp`
- `Texture.color_ramp`
- `TextureNodeValToRGB.color_ramp`
- `UserPreferencesSystem.weight_color_range`

## 2.4.96 ColorRampElement(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.ColorRampElement` (*bpy\_struct*)  
Element defining a color at a position in the color ramp

**color**

Set color of selected color stop

**Type** float array of 4 items in [-inf, inf], default (0.0, 0.0, 0.0, 0.0)

**position**

Set position of selected color stop

**Type** float in [0, 1], default 0.0

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `ColorRamp.elements`
- `ColorRampElements.new`
- `ColorRampElements.remove`

## 2.4.97 ColorRampElements(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.ColorRampElements` (*bpy\_struct*)  
Collection of Color Ramp Elements

**new** (*position*)

Add element to ColorRamp

**Parameters** **position** (*float in [0, 1]*) – Position, Position to add element

**Returns** New element

**Return type** `ColorRampElement`

**remove** (*element*)

Delete element from ColorRamp

**Parameters** **element** (`ColorRampElement`, (never `None`)) – Element to remove

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `ColorRamp.elements`

## 2.4.98 ColorSequence(EffectSequence)

base classes — `bpy_struct`, `Sequence`, `EffectSequence`

**class** `bpy.types.ColorSequence` (*EffectSequence*)

Sequence strip creating an image filled with a single g

**color**

**Type** float array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`

## Inherited Properties

- `bpy_struct.id_data`
- `Sequence.name`
- `Sequence.blend_type`
- `Sequence.blend_alpha`
- `Sequence.channel`
- `Sequence.waveform`
- `Sequence.effect_fader`
- `Sequence.frame_final_end`
- `Sequence.frame_offset_end`
- `Sequence.frame_still_end`
- `Sequence.input_1`
- `Sequence.input_2`
- `Sequence.input_3`
- `Sequence.select_left_handle`
- `Sequence.frame_final_duration`
- `Sequence.frame_duration`
- `Sequence.lock`
- `Sequence.mute`
- `Sequence.select_right_handle`
- `Sequence.select`
- `Sequence.speed_factor`
- `Sequence.frame_start`
- `Sequence.frame_final_start`
- `Sequence.frame_offset_start`
- `Sequence.frame_still_start`
- `Sequence.type`
- `Sequence.use_default_fade`
- `Sequence.input_count`
- `EffectSequence.color_balance`
- `EffectSequence.use_float`
- `EffectSequence.crop`
- `EffectSequence.use_deinterlace`
- `EffectSequence.use_reverse_frames`
- `EffectSequence.use_flip_x`
- `EffectSequence.use_flip_y`
- `EffectSequence.color_multiply`
- `EffectSequence.use_premultiply`
- `EffectSequence.proxy`
- `EffectSequence.use_proxy_custom_directory`
- `EffectSequence.use_proxy_custom_file`
- `EffectSequence.color_saturation`
- `EffectSequence.strobe`
- `EffectSequence.transform`
- `EffectSequence.use_color_balance`
- `EffectSequence.use_crop`
- `EffectSequence.use_proxy`
- `EffectSequence.use_translation`

## Inherited Functions

- `bpy_struct.as_pointer`

- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Sequence.getStripElem`
- `Sequence.swap`

## 2.4.99 CompositorNode(Node)

base classes — `bpy_struct`, `Node`

subclasses — `CompositorNodeSepHSVA`, `CompositorNodeValToRGB`, `CompositorNodeSepYCCA`, `CompositorNodeSetAlpha`, `CompositorNodeCrop`, `CompositorNodeMath`, `CompositorNodeCurveRGB`, `CompositorNodeDilateErode`, `CompositorNodeGamma`, `CompositorNodeRotate`, `CompositorNodeLensdist`, `CompositorNodeLumaMatte`, `CompositorNodeTranslate`, `CompositorNodeCombyCCA`, `CompositorNodeBilateralblur`, `CompositorNodeZcombine`, `CompositorNodeFilter`, `CompositorNodePremulKey`, `CompositorNodeMixRGB`, `CompositorNodeTime`, `CompositorNodeValue`, `CompositorNodeHueCorrect`, `CompositorNodeNormal`, `CompositorNodeIDMask`, `CompositorNodeVecBlur`, `CompositorNodeRLayers`, `CompositorNodeComposite`, `CompositorNodeCombrGBA`, `CompositorNodeViewer`, `CompositorNodeCombHSVA`, `CompositorNodeOutputFile`, `CompositorNodeInvert`, `CompositorNodeFlip`, `CompositorNodeDistanceMatte`, `CompositorNodeRGBToBW`, `CompositorNodeCombYUVA`, `CompositorNodeBlur`, `CompositorNodeDefocus`, `CompositorNodeGlare`, `CompositorNodeDiffMatte`, `CompositorNodeImage`, `CompositorNodeDBLur`, `CompositorNodeMapValue`, `CompositorNodeBrightContrast`, `CompositorNodeTexture`, `CompositorNodeLevels`, `CompositorNodeNormalize`, `CompositorNodeChannelMatte`, `CompositorNodeRGB`, `CompositorNodeColorSpill`, `CompositorNodeSepRGBA`, `CompositorNodeScale`, `CompositorNodeMapUV`, `CompositorNodeChromaMatte`, `CompositorNodeDisplace`, `CompositorNodeColorBalance`, `CompositorNodeTonemap`, `CompositorNodeAlphaOver`, `CompositorNodeColorMatte`, `CompositorNodeHueSat`, `CompositorNodeSepYUVA`, `CompositorNodeCurveVec`, `CompositorNodeSplitViewer`

class `bpy.types.CompositorNode` (*Node*)

**type**

**Type** enum in [`'VIEWER'`, `'RGB'`, `'VALUE'`, `'MIX_RGB'`, `'VALTORGB'`, `'RGBTOBW'`, `'NORMAL'`, `'CURVE_VEC'`, `'CURVE_RGB'`, `'ALPHAOVER'`, `'BLUR'`, `'FILTER'`, `'MAP_VALUE'`, `'TIME'`, `'VECBLUR'`, `'SEPRGBA'`, `'SEPHSVA'`, `'SE-TALPHA'`, `'HUE_SAT'`, `'IMAGE'`, `'R_LAYERS'`, `'COMPOSITE'`, `'OUTPUT_FILE'`, `'TEXTURE'`, `'TRANSLATE'`, `'ZCOMBINE'`, `'COMBRGBA'`, `'DILATEERODE'`, `'ROTATE'`, `'SCALE'`, `'SEPYCCA'`, `'COMBYCCA'`, `'SEPYUVA'`, `'COMBYUVA'`,

‘DIFF\_MATTE’, ‘COLOR\_SPILL’, ‘CHROMA\_MATTE’, ‘CHANNEL\_MATTE’, ‘FLIP’, ‘SPLITVIEWER’, ‘MAP\_UV’, ‘ID\_MASK’, ‘DEFOCUS’, ‘DISPLACE’, ‘COMBHSVA’, ‘MATH’, ‘LUMA\_MATTE’, ‘BRIGHTCONTRAST’, ‘GAMMA’, ‘INVERT’, ‘NORMALIZE’, ‘CROP’, ‘DBLUR’, ‘BILATERALBLUR’, ‘PREMULKEY’, ‘DISTANCE\_MATTE’, ‘LEVELS’, ‘COLOR\_MATTE’, ‘COLORBALANCE’, ‘HUECORRECT’, ‘GLARE’, ‘TONEMAP’, ‘LENSDIST’, ‘SCRIPT’, ‘GROUP’], default ‘VIEWER’, (readonly)

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.100 CompositorNodeAlphaOver(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeAlphaOver` (*CompositorNode*)

**premul**

Mix Factor

**Type** float in [0, 1], default 0.0

**use\_multiply**

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.101 CompositorNodeBilateralblur(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeBilateralblur` (*CompositorNode*)

#### **iterations**

**Type** int in [1, 128], default 0

#### **sigma\_color**

**Type** float in [0.01, 3], default 0.0

#### **sigma\_space**

**Type** float in [0.01, 30], default 0.0

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`

- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.102 CompositorNodeBlur(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeBlur` (*CompositorNode*)

### **aspect\_correction**

Type of aspect correction to use

**Type** enum in ['NONE', 'Y', 'X'], default 'NONE'

### **factor**

**Type** float in [0, 2], default 0.0

### **factor\_x**

**Type** float in [0, 100], default 0.0

### **factor\_y**

**Type** float in [0, 100], default 0.0

### **filter\_type**

**Type** enum in ['FLAT', 'TENT', 'QUAD', 'CUBIC', 'GAUSS', 'FAST\_GAUSS', 'CATROM', 'MITCH'], default 'FLAT'

### **size\_x**

**Type** int in [0, 2048], default 0

### **size\_y**



**Type** int in [0, 2048], default 0

**use\_bokeh**

Uses circular filter (slower)

**Type** boolean, default False

**use\_gamma\_correction**

Applies filter on gamma corrected values

**Type** boolean, default False

**use\_relative**

Use relative (percent) values to define blur radius

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.103 CompositorNodeBrightContrast(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

class `bpy.types.CompositorNodeBrightContrast` (*CompositorNode*)

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.104 CompositorNodeChannelMatte(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeChannelMatte` (*CompositorNode*)

#### **color\_space**

- RGB RGB, RGB Color Space.
- HSV HSV, HSV Color Space.
- YUV YUV, YUV Color Space.
- YCC YCbCr, YCbCr Color Space.

**Type** enum in ['RGB', 'HSV', 'YUV', 'YCC'], default 'RGB'

#### **limit\_channel**

Limit by this channels value

**Type** enum in ['R', 'G', 'B'], default 'R'

#### **limit\_max**

Values higher than this setting are 100% opaque

**Type** float in [0, 1], default 0.0

**limit\_method**

Algorithm to use to limit channel

- SINGLE Single, Limit by single channel.
- MAX Max, Limit by max of other channels .

**Type** enum in ['SINGLE', 'MAX'], default 'SINGLE'

**limit\_min**

Values lower than this setting are 100% keyed

**Type** float in [0, 1], default 0.0

**matte\_channel**

Channel used to determine matte

**Type** enum in ['R', 'G', 'B'], default 'R'

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.105 CompositorNodeChromaMatte(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

`class bpy.types.CompositorNodeChromaMatte` (*CompositorNode*)

**gain**

Alpha gain

**Type** float in [0, 1], default 0.0

**lift**

Alpha lift

**Type** float in [0, 1], default 0.0

**shadow\_adjust**

Adjusts the brightness of any shadows captured

**Type** float in [0, 1], default 0.0

**threshold**

Tolerance below which colors will be considered as exact matches

**Type** float in [0, 30], default 0.0

**tolerance**

Tolerance for a color to be considered a keying color

**Type** float in [1, 80], default 0.0

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.106 CompositorNodeColorBalance(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

class `bpy.types.CompositorNodeColorBalance` (*CompositorNode*)

### correction\_method

- `LIFT_GAMMA_GAIN` Lift/Gamma/Gain.
- `OFFSET_POWER_SLOPE` Offset/Power/Slope (ASC-CDL), ASC-CDL standard color correction.

**Type** enum in [`'LIFT_GAMMA_GAIN'`, `'OFFSET_POWER_SLOPE'`], default `'LIFT_GAMMA_GAIN'`

### gain

Correction for Highlights

**Type** float array of 3 items in `[-inf, inf]`, default `(1.0, 1.0, 1.0)`

### gamma

Correction for Midtones

**Type** float array of 3 items in `[-inf, inf]`, default `(1.0, 1.0, 1.0)`

### lift

Correction for Shadows

**Type** float array of 3 items in `[-inf, inf]`, default `(1.0, 1.0, 1.0)`

### offset

Correction for Shadows

**Type** float array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`

### power

Correction for Midtones

**Type** float array of 3 items in `[0, inf]`, default `(1.0, 1.0, 1.0)`

### slope

Correction for Highlights

**Type** float array of 3 items in `[0, inf]`, default `(1.0, 1.0, 1.0)`

## Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.107 CompositorNodeColorMatte(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeColorMatte` (*CompositorNode*)

**color\_hue**

Hue tolerance for colors to be considered a keying color

**Type** float in [0, 1], default 0.0

**color\_saturation**

Saturation Tolerance for the color

**Type** float in [0, 1], default 0.0

**color\_value**

Value Tolerance for the color

**Type** float in [0, 1], default 0.0

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`

- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.108 CompositorNodeColorSpill(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeColorSpill` (*CompositorNode*)

#### **channel**

- **R** R, Red Spill Suppression.
- **G** G, Green Spill Suppression.
- **B** B, Blue Spill Suppression.

**Type** enum in ['R', 'G', 'B'], default 'R'

#### **limit\_channel**

- **R** R, Limit by Red.
- **G** G, Limit by Green.
- **B** B, Limit by Blue.

**Type** enum in ['R', 'G', 'B'], default 'R'

#### **limit\_method**

- **SIMPLE** Simple, Simple Limit Algorithm.
- **AVERAGE** Average, Average Limit Algorithm.

**Type** enum in ['SIMPLE', 'AVERAGE'], default 'SIMPLE'

#### **ratio**

Scale limit by value

**Type** float in [0.5, 1.5], default 0.0

#### **unspill\_blue**

Blue spillmap scale

**Type** float in [0, 1.5], default 0.0

**unspill\_green**

Green spillmap scale

**Type** float in [0, 1.5], default 0.0

**unspill\_red**

Red spillmap scale

**Type** float in [0, 1.5], default 0.0

**use\_unspill**

Compensate all channels (differently) by hand

**Type** boolean, default False

**Inherited Properties**

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**2.4.109 CompositorNodeCombHSVA(CompositorNode)**

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeCombHSVA` (*CompositorNode*)



### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.110 CompositorNodeCombRGBA(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeCombRGBA` (*CompositorNode*)

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`

- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.111 CompositorNodeCombYCCA(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeCombYCCA` (*CompositorNode*)

**mode**

**Type** enum in ['ITUBT601', 'ITUBT709', 'JFIF'], default 'ITUBT601'

#### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`

- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.112 CompositorNodeCombYUVA(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeCombYUVA` (*CompositorNode*)

#### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.113 CompositorNodeComposite(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeComposite` (*CompositorNode*)

#### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`

- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.114 CompositorNodeCrop(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

class `bpy.types.CompositorNodeCrop` (*CompositorNode*)

```
max_x
    Type int in [0, 10000], default 0
max_y
    Type int in [0, 10000], default 0
min_x
    Type int in [0, 10000], default 0
min_y
    Type int in [0, 10000], default 0
rel_max_x
    Type float in [0, 1], default 0.0
rel_max_y
    Type float in [0, 1], default 0.0
rel_min_x
    Type float in [0, 1], default 0.0
```

**rel\_min\_y****Type** float in [0, 1], default 0.0**relative**

Use relative values to crop image

**Type** boolean, default False**use\_crop\_size**

Whether to crop the size of the input image

**Type** boolean, default False**Inherited Properties**

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**2.4.115 CompositorNodeCurveRGB(CompositorNode)**base classes — `bpy_struct`, `Node`, `CompositorNode`**class** `bpy.types.CompositorNodeCurveRGB` (*CompositorNode*)**mapping****Type** `CurveMapping`, (readonly)

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.116 CompositorNodeCurveVec(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeCurveVec` (*CompositorNode*)

#### mapping

**Type** `CurveMapping`, (readonly)

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.117 CompositorNodeDBlur(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeDBlur` (*CompositorNode*)

#### **angle**

**Type** float in [0, 360], default 0.0

#### **center\_x**

**Type** float in [0, 1], default 0.0

#### **center\_y**

**Type** float in [0, 1], default 0.0

#### **distance**

**Type** float in [-1, 1], default 0.0

#### **iterations**

**Type** int in [1, 32], default 0

#### **spin**

**Type** float in [-360, 360], default 0.0

#### **use\_wrap**

**Type** boolean, default False

#### **zoom**

**Type** float in [0, 100], default 0.0

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.118 CompositorNodeDefocus(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeDefocus` (*CompositorNode*)

#### **angle**

Bokeh shape rotation offset in degrees

**Type** int in [0, 90], default 0

#### **blur\_max**

blur limit, maximum CoC radius, 0=no limit

**Type** float in [0, 10000], default 0.0

#### **bokeh**

- **OCTAGON** Octagonal, 8 sides.
- **HEPTAGON** Heptagonal, 7 sides.
- **HEXAGON** Hexagonal, 6 sides.
- **PENTAGON** Pentagonal, 5 sides.



- SQUARE Square, 4 sides.
- TRIANGLE Triangular, 3 sides.
- CIRCLE Circular.

**Type** enum in ['OCTAGON', 'HEPTAGON', 'HEXAGON', 'PENTAGON', 'SQUARE', 'TRIANGLE', 'CIRCLE'], default 'CIRCLE'

#### **f\_stop**

Amount of focal blur, 128=infinity=perfect focus, half the value doubles the blur radius

**Type** float in [0, 128], default 0.0

#### **samples**

Number of samples (16=grainy, higher=less noise)

**Type** int in [16, 256], default 0

#### **threshold**

CoC radius threshold, prevents background bleed on in-focus midground, 0=off

**Type** float in [0, 100], default 0.0

#### **use\_gamma\_correction**

Enable gamma correction before and after main process

**Type** boolean, default False

#### **use\_preview**

Enable sampling mode, useful for preview when using low samplecounts

**Type** boolean, default False

#### **use\_zbuffer**

Disable when using an image as input instead of actual z-buffer (auto enabled if node not image based, eg. time node)

**Type** boolean, default False

#### **z\_scale**

Scales the Z input when not using a z-buffer, controls maximum blur designated by the color white or input value 1

**Type** float in [0, 1000], default 0.0

### **Inherited Properties**

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.119 CompositorNodeDiffMatte(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeDiffMatte` (*CompositorNode*)

#### **falloff**

Color distances below this additional threshold are partially keyed

**Type** float in [0, 1], default 0.0

#### **tolerance**

Color distances below this threshold are keyed

**Type** float in [0, 1], default 0.0

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`

- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.120 CompositorNodeDilateErode(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeDilateErode` (*CompositorNode*)

#### **distance**

Distance to grow/shrink (number of iterations)

**Type** `int` in `[-100, 100]`, default `0`

#### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`

- `bpy_struct.values`

### 2.4.121 CompositorNodeDisplace(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeDisplace` (*CompositorNode*)

#### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.122 CompositorNodeDistanceMatte(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeDistanceMatte` (*CompositorNode*)

#### **falloff**

Color distances below this additional threshold are partially keyed

**Type** float in [0, 1], default 0.0

#### **tolerance**

Color distances below this threshold are keyed

**Type** float in [0, 1], default 0.0

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.123 CompositorNodeFilter(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeFilter` (*CompositorNode*)

### **filter\_type**

**Type** enum in ['SOFTEN', 'SHARPEN', 'LAPLACE', 'SOBEL', 'PREWITT', 'KIRSCH', 'SHADOW'], default 'SOFTEN'

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`

- `Node.parent`
- `CompositorNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.124 CompositorNodeFlip(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

class `bpy.types.CompositorNodeFlip` (*CompositorNode*)

### **axis**

**Type** enum in ['X', 'Y', 'XY'], default 'X'

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`

- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.125 CompositorNodeGamma(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeGamma` (*CompositorNode*)

#### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.126 CompositorNodeGlare(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

class `bpy.types.CompositorNodeGlare` (*CompositorNode*)

**angle\_offset**

Streak angle offset in degrees

**Type** int in [0, 180], default 0

**color\_modulation**

Amount of Color Modulation, modulates colors of streaks and ghosts for a spectral dispersion effect

**Type** float in [0, 1], default 0.0

**fade**

Streak fade-out factor

**Type** float in [0.75, 1], default 0.0

**glare\_type**

**Type** enum in ['GHOSTS', 'STREAKS', 'FOG\_GLOW', 'SIMPLE\_STAR'], default 'SIMPLE\_STAR'

**iterations**

**Type** int in [2, 5], default 0

**mix**

-1 is original image only, 0 is exact 50/50 mix, 1 is processed image only

**Type** float in [-1, 1], default 0.0

**quality**

If not set to high quality, the effect will be applied to a low-res copy of the source image

**Type** enum in ['HIGH', 'MEDIUM', 'LOW'], default 'HIGH'

**size**

Glow/glare size (not actual size; relative to initial size of bright area of pixels)

**Type** int in [6, 9], default 0

**streaks**

Total number of streaks

**Type** int in [2, 16], default 0

**threshold**

The glare filter will only be applied to pixels brighter than this value

**Type** float in [0, 1000], default 0.0

**use\_rotate\_45**

Simple star filter: add 45 degree rotation offset

**Type** boolean, default False

**Inherited Properties**

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`



- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.127 CompositorNodeHueCorrect(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

class `bpy.types.CompositorNodeHueCorrect` (*CompositorNode*)

### mapping

**Type** `CurveMapping`, (readonly)

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`

- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.128 CompositorNodeHueSat(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeHueSat` (*CompositorNode*)

#### **color\_hue**

**Type** float in [0, 1], default 0.0

#### **color\_saturation**

**Type** float in [0, 2], default 0.0

#### **color\_value**

**Type** float in [0, 2], default 0.0

#### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`

- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.129 CompositorNodeIDMask(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeIDMask` (*CompositorNode*)

#### **index**

Pass index number to convert to alpha

**Type** int in [0, 32767], default 0

#### **use\_smooth\_mask**

Apply an anti-aliasing filter to the mask

**Type** boolean, default False

#### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.130 CompositorNodeImage(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

class `bpy.types.CompositorNodeImage` (*CompositorNode*)

**frame\_duration**

Number of images of a movie to use

**Type** int in [0, 300000], default 0

**frame\_offset**

Offset the number of the frame to use in the animation

**Type** int in [-300000, 300000], default 0

**frame\_start**

Global starting frame of the movie/sequence, assuming first picture has a #1

**Type** int in [-300000, 300000], default 0

**image**

**Type** `Image`

**layer**

**Type** enum in ['PLACEHOLDER'], default 'PLACEHOLDER'

**use\_auto\_refresh**

Always refresh image on frame changes

**Type** boolean, default False

**use\_cyclic**

Cycle the images in the movie

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`

- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.131 CompositorNodeInvert(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeInvert` (*CompositorNode*)

#### **invert\_alpha**

**Type** boolean, default False

#### **invert\_rgb**

**Type** boolean, default False

#### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.132 CompositorNodeLensdist(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

class `bpy.types.CompositorNodeLensdist` (*CompositorNode*)

**use\_fit**

For positive distortion factor only: scale image such that black areas are not visible

**Type** boolean, default False

**use\_jitter**

Enable/disable jittering; faster, but also noisier

**Type** boolean, default False

**use\_projector**

Enable/disable projector mode. Effect is applied in horizontal direction only

**Type** boolean, default False

#### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.133 CompositorNodeLevels(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeLevels` (*CompositorNode*)

#### **channel**

- COMBINED\_RGB C, Combined RGB.
- RED R, Red Channel.
- GREEN G, Green Channel.
- BLUE B, Blue Channel.
- LUMINANCE L, Luminance Channel.

**Type** enum in ['COMBINED\_RGB', 'RED', 'GREEN', 'BLUE', 'LUMINANCE'], default 'COMBINED\_RGB'

#### **Inherited Properties**

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

#### **Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### **2.4.134 CompositorNodeLumaMatte(CompositorNode)**

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeLumaMatte` (*CompositorNode*)

**limit\_max**

Values higher than this setting are 100% opaque

**Type** float in [0, 1], default 0.0

**limit\_min**

Values lower than this setting are 100% keyed

**Type** float in [0, 1], default 0.0

**Inherited Properties**

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**2.4.135 CompositorNodeMapUV(CompositorNode)**

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeMapUV` (*CompositorNode*)

**alpha**

**Type** int in [0, 100], default 0



## Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.136 CompositorNodeMapValue(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeMapValue` (*CompositorNode*)

**max**

**Type** float array of 1 items in [-1000, 1000], default (0.0)

**min**

**Type** float array of 1 items in [-1000, 1000], default (0.0)

**offset**

**Type** float array of 1 items in [-1000, 1000], default (0.0)

**size**

**Type** float array of 1 items in [-1000, 1000], default (0.0)

**use\_max**

**Type** boolean, default False

**use\_min**

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.137 CompositorNodeMath(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeMath` (*CompositorNode*)

### operation

**Type** enum in ['ADD', 'SUBTRACT', 'MULTIPLY', 'DIVIDE', 'SINE', 'COSINE', 'TANGENT', 'ARCSINE', 'ARCCOSINE', 'ARCTANGENT', 'POWER', 'LOGARITHM', 'MINIMUM', 'MAXIMUM', 'ROUND', 'LESS\_THAN', 'GREATER\_THAN'], default 'ADD'

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`

- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.138 CompositorNodeMixRGB(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

class `bpy.types.CompositorNodeMixRGB` (*CompositorNode*)

### **blend\_type**

**Type** enum in ['MIX', 'ADD', 'MULTIPLY', 'SUBTRACT', 'SCREEN', 'DIVIDE', 'DIFFERENCE', 'DARKEN', 'LIGHTEN', 'OVERLAY', 'DODGE', 'BURN', 'HUE', 'SATURATION', 'VALUE', 'COLOR', 'SOFT\_LIGHT', 'LINEAR\_LIGHT'], default 'MIX'

### **use\_alpha**

Include alpha of second input in this operation

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.139 CompositorNodeNormal(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeNormal` (*CompositorNode*)

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`

- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.140 CompositorNodeNormalize(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeNormalize` (*CompositorNode*)

#### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.141 CompositorNodeOutputFile(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeOutputFile` (*CompositorNode*)

**exr\_codec**

**Type** enum in ['NONE', 'PXR24', 'ZIP', 'PIZ', 'RLE'], default 'NONE'

**filepath**

Output path for the image, same functionality as render output

**Type** string, default ""

**frame\_end**

**Type** int in [0, 300000], default 0

**frame\_start**

**Type** int in [0, 300000], default 0

**image\_type**

**Type** enum in ['TARGA', 'RAW\_TARGA', 'PNG', 'BMP', 'JPEG', 'IRIS', 'RADIANCE\_HDR', 'CINEON', 'DPX', 'OPENEXR'], default 'TARGA'

**quality**

**Type** int in [1, 100], default 0

**use\_exr\_half**

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.142 CompositorNodePremulKey(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodePremulKey` (*CompositorNode*)

**mapping**

Conversion between premultiplied alpha and key alpha

**Type** enum in ['KEY\_TO\_PREMUL', 'PREMUL\_TO\_KEY'], default 'KEY\_TO\_PREMUL'

**Inherited Properties**

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.143 CompositorNodeRGB(*CompositorNode*)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeRGB` (*CompositorNode*)

**Inherited Properties**

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`

- `Node.parent`
- `CompositorNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.144 CompositorNodeRGBToBW(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeRGBToBW` (*CompositorNode*)

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`



- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.145 CompositorNodeRLayers(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeRLayers` (*CompositorNode*)

**layer**

**Type** enum in ['PLACEHOLDER'], default 'PLACEHOLDER'

**scene**

**Type** `Scene`

#### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.146 CompositorNodeRotate(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeRotate` (*CompositorNode*)

### **filter\_type**

Method to use to filter rotation

**Type** enum in ['NEAREST', 'BILINEAR', 'BICUBIC'], default 'NEAREST'

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.147 CompositorNodeScale(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeScale` (*CompositorNode*)

### **space**

Coordinate space to scale relative to

**Type** enum in ['RELATIVE', 'ABSOLUTE', 'SCENE\_SIZE', 'RENDER\_SIZE'], default 'RELATIVE'

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.148 CompositorNodeSepHSVA(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeSepHSVA` (*CompositorNode*)

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`

- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.149 CompositorNodeSepRGBA(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeSepRGBA` (*CompositorNode*)

#### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.150 CompositorNodeSepYCCA(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeSepYCCA` (*CompositorNode*)

**mode**

**Type** enum in ['ITUBT601', 'ITUBT709', 'JFIF'], default 'ITUBT601'

#### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.151 CompositorNodeSepYUVA(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeSepYUVA` (*CompositorNode*)

#### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`

- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.152 CompositorNodeSetAlpha(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

class `bpy.types.CompositorNodeSetAlpha` (*CompositorNode*)

#### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`

- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.153 CompositorNodeSplitViewer(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeSplitViewer` (*CompositorNode*)

#### **axis**

**Type** enum in ['X', 'Y'], default 'X'

#### **factor**

**Type** int in [0, 100], default 0

#### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.154 CompositorNodeTexture(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeTexture` (*CompositorNode*)

**node\_output**

For node-based textures, which output node to use

**Type** `int` in `[-32768, 32767]`, default `0`

**texture**

**Type** `Texture`

#### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.155 CompositorNodeTime(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeTime` (*CompositorNode*)

**curve**



**Type** `CurveMapping`, (readonly)

**frame\_end**

**Type** `int` in `[-32768, 32767]`, default `0`

**frame\_start**

**Type** `int` in `[-32768, 32767]`, default `0`

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.156 CompositorNodeTonemap(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeTonemap` (*CompositorNode*)

**adaptation**

If 0, global; if 1, based on pixel intensity

**Type** `float` in `[0, 1]`, default `0.0`

**contrast**

Set to 0 to use estimate from input image

**Type** `float` in `[0, 1]`, default `0.0`

**correction**

If 0, same for all channels; if 1, each independent

**Type** float in [0, 1], default 0.0

**gamma**

If not used, set to 1

**Type** float in [0.001, 3], default 0.0

**intensity**

If less than zero, darkens image; otherwise, makes it brighter

**Type** float in [-8, 8], default 0.0

**key**

The value the average luminance is mapped to

**Type** float in [0, 1], default 0.0

**offset**

Normally always 1, but can be used as an extra control to alter the brightness curve

**Type** float in [0.001, 10], default 0.0

**tonemap\_type**

**Type** enum in ['RD\_PHOTORECEPTOR', 'RH\_SIMPLE'], default 'RH\_SIMPLE'

**Inherited Properties**

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.157 CompositorNodeTranslate(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeTranslate` (*CompositorNode*)

#### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.158 CompositorNodeTree(NodeTree)

base classes — `bpy_struct`, `ID`, `NodeTree`

**class** `bpy.types.CompositorNodeTree` (*NodeTree*)

Node tree consisting of linked nodes used for compositing

**nodes**

**Type** `CompositorNodes` `bpy_prop_collection` of `Node`, (readonly)

#### Inherited Properties

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`

- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`
- `NodeTree.animation_data`
- `NodeTree.grease_pencil`
- `NodeTree.inputs`
- `NodeTree.links`
- `NodeTree.outputs`
- `NodeTree.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`

### 2.4.159 CompositorNodeValToRGB(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

class `bpy.types.CompositorNodeValToRGB` (*CompositorNode*)

**color\_ramp**

**Type** `ColorRamp`, (readonly)

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`

- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.160 CompositorNodeValue(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

class `bpy.types.CompositorNodeValue` (*CompositorNode*)

#### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`

- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.161 CompositorNodeVecBlur(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeVecBlur` (*CompositorNode*)

### **factor**

Scaling factor for motion vectors; actually ‘shutter speed’ in frames

**Type** float in [0, 2], default 0.0

### **samples**

**Type** int in [1, 256], default 0

### **speed\_max**

Maximum speed, or zero for none

**Type** int in [0, 1024], default 0

### **speed\_min**

Minimum speed for a pixel to be blurred; used to separate background from foreground

**Type** int in [0, 1024], default 0

### **use\_curved**

Interpolate between frames in a Bezier curve, rather than linearly

**Type** boolean, default False

## Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`

- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.162 CompositorNodeViewer(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeViewer` (*CompositorNode*)

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.163 CompositorNodeZcombine(CompositorNode)

base classes — `bpy_struct`, `Node`, `CompositorNode`

**class** `bpy.types.CompositorNodeZcombine` (*CompositorNode*)

**use\_alpha**

Takes Alpha channel into account when doing the Z operation

**Type** boolean, default False

#### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `CompositorNode.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.164 CompositorNodes(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.CompositorNodes` (*bpy\_struct*)

Collection of Compositor Nodes

**new** (*type*, *group=None*)

Add a node to this node tree

**Parameters**



- **type** (*enum in* [`'VIEWER'`, `'RGB'`, `'VALUE'`, `'MIX_RGB'`, `'VALTORGB'`, `'RGBTOBW'`, `'NORMAL'`, `'CURVE_VEC'`, `'CURVE_RGB'`, `'ALPHAOVER'`, `'BLUR'`, `'FILTER'`, `'MAP_VALUE'`, `'TIME'`, `'VECBLUR'`, `'SEPRGBA'`, `'SEPHSVA'`, `'SE-TALPHA'`, `'HUE_SAT'`, `'IMAGE'`, `'R_LAYERS'`, `'COMPOSITE'`, `'OUTPUT_FILE'`, `'TEXTURE'`, `'TRANSLATE'`, `'ZCOMBINE'`, `'COMBRGBA'`, `'DILATEERODE'`, `'ROTATE'`, `'SCALE'`, `'SEPYCCA'`, `'COMBYCCA'`, `'SEPYUVA'`, `'COMBYUVA'`, `'DIFF_MATTE'`, `'COLOR_SPILL'`, `'CHROMA_MATTE'`, `'CHANNEL_MATTE'`, `'FLIP'`, `'SPLITVIEWER'`, `'MAP_UV'`, `'ID_MASK'`, `'DEFOCUS'`, `'DISPLACE'`, `'COMBHSVA'`, `'MATH'`, `'LUMA_MATTE'`, `'BRIGHTCONTRAST'`, `'GAMMA'`, `'INVERT'`, `'NORMALIZE'`, `'CROP'`, `'DBLUR'`, `'BILATERALBLUR'`, `'PREMULKEY'`, `'DISTANCE_MATTE'`, `'LEVELS'`, `'COLOR_MATTE'`, `'COLORBALANCE'`, `'HUECORRECT'`, `'GLARE'`, `'TONEMAP'`, `'LENSDIST'`, `'SCRIPT'`, `'GROUP'`]) – Type, Type of node to add
- **group** (`NodeTree`, (optional)) – The group tree

**Returns** New node

**Return type** `Node`

**remove** (*node*)

Remove a node from this node tree

**Parameters** **node** (`Node`) – The node to remove

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `CompositorNodeTree.nodes`

### 2.4.165 ConsoleLine(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.ConsoleLine` (*bpy\_struct*)

Input line for the interactive console

**body**

Text in the line

**Type** string, default ""

**current\_character**

**Type** int in [-inf, inf], default 0

#### Inherited Properties

- `bpy_struct.id_data`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

#### References

- `SpaceConsole.history`
- `SpaceConsole.scrollback`

### 2.4.166 Constraint(bpy\_struct)

base class — `bpy_struct`

subclasses — `KinematicConstraint`, `CopyScaleConstraint`, `ShrinkwrapConstraint`,  
`FollowPathConstraint`, `LockedTrackConstraint`, `CopyTransformsConstraint`,  
`PythonConstraint`, `LimitDistanceConstraint`, `StretchToConstraint`, `PivotConstraint`,  
`ClampToConstraint`, `TransformConstraint`, `LimitRotationConstraint`,  
`LimitLocationConstraint`, `ChildOfConstraint`, `CopyRotationConstraint`,

DampedTrackConstraint, TrackToConstraint, SplineIKConstraint,  
 MaintainVolumeConstraint, LimitScaleConstraint, RigidBodyJointConstraint,  
 FloorConstraint, ActionConstraint, CopyLocationConstraint

**class** `bpy.types.Constraint` (*bpy\_struct*)

Constraint modifying the transformation of objects and bones

**active**

Constraint is the one being edited

**Type** boolean, default False

**error\_location**

Amount of residual error in Blender space unit for constraints that work on position

**Type** float in [-inf, inf], default 0.0, (readonly)

**error\_rotation**

Amount of residual error in radiant for constraints that work on orientation

**Type** float in [-inf, inf], default 0.0, (readonly)

**influence**

Amount of influence constraint will have on the final solution

**Type** float in [0, 1], default 0.0

**is\_proxy\_local**

Constraint was added in this proxy instance (i.e. did not belong to source Armature)

**Type** boolean, default False

**is\_valid**

Constraint has valid settings and can be evaluated

**Type** boolean, default False, (readonly)

**mute**

Enable/Disable Constraint

**Type** boolean, default False

**name**

Constraint name

**Type** string, default ""

**owner\_space**

Space that owner is evaluated in

- **WORLD** World Space, The constraint is applied relative to the world coordinate system.
- **POSE** Pose Space, The constraint is applied in Pose Space, the object transformation is ignored.
- **LOCAL\_WITH\_PARENT** Local With Parent, The constraint is applied relative to the local coordinate system of the object, with the parent transformation added.
- **LOCAL** Local Space, The constraint is applied relative to the local coordinate system of the object.

**Type** enum in ['WORLD', 'POSE', 'LOCAL\_WITH\_PARENT', 'LOCAL'], default 'WORLD'

**show\_expanded**

Constraint's panel is expanded in UI

**Type** boolean, default False

**target\_space**

Space that target is evaluated in

- WORLD** World Space, The transformation of the target is evaluated relative to the world coordinate system.
- POSE** Pose Space, The transformation of the target is only evaluated in the Pose Space, the target armature object transformation is ignored.
- LOCAL\_WITH\_PARENT** Local With Parent, The transformation of the target bone is evaluated relative its local coordinate system, with the parent transformation added.
- LOCAL** Local Space, The transformation of the target is evaluated relative to its local coordinate system.

**Type** enum in ['WORLD', 'POSE', 'LOCAL\_WITH\_PARENT', 'LOCAL'], default 'WORLD'

**type**

- COPY\_LOCATION** Copy Location.
- COPY\_ROTATION** Copy Rotation.
- COPY\_SCALE** Copy Scale.
- COPY\_TRANSFORMS** Copy Transforms.
- LIMIT\_DISTANCE** Limit Distance.
- LIMIT\_LOCATION** Limit Location.
- LIMIT\_ROTATION** Limit Rotation.
- LIMIT\_SCALE** Limit Scale.
- MAINTAIN\_VOLUME** Maintain Volume.
- TRANSFORM** Transformation.
- CLAMP\_TO** Clamp To.
- DAMPED\_TRACK** Damped Track, Tracking by taking the shortest path.
- IK** Inverse Kinematics.
- LOCKED\_TRACK** Locked Track, Tracking along a single axis.
- SPLINE\_IK** Spline IK.
- STRETCH\_TO** Stretch To.
- TRACK\_TO** Track To, Legacy tracking constraint prone to twisting artifacts.
- ACTION** Action.
- CHILD\_OF** Child Of.
- FLOOR** Floor.
- FOLLOW\_PATH** Follow Path.
- PIVOT** Pivot.
- RIGID\_BODY\_JOINT** Rigid Body Joint.

- SCRIPT Script.
- SHRINKWRAP Shrinkwrap.

**Type** enum in ['COPY\_LOCATION', 'COPY\_ROTATION', 'COPY\_SCALE', 'COPY\_TRANSFORMS', 'LIMIT\_DISTANCE', 'LIMIT\_LOCATION', 'LIMIT\_ROTATION', 'LIMIT\_SCALE', 'MAINTAIN\_VOLUME', 'TRANSFORM', 'CLAMP\_TO', 'DAMPED\_TRACK', 'IK', 'LOCKED\_TRACK', 'SPLINE\_IK', 'STRETCH\_TO', 'TRACK\_TO', 'ACTION', 'CHILD\_OF', 'FLOOR', 'FOLLOW\_PATH', 'PIVOT', 'RIGID\_BODY\_JOINT', 'SCRIPT', 'SHRINKWRAP'], default 'COPY\_LOCATION', (readonly)

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Object.constraints`
- `ObjectConstraints.active`
- `ObjectConstraints.new`
- `ObjectConstraints.remove`
- `PoseBone.constraints`
- `PoseBoneConstraints.active`
- `PoseBoneConstraints.new`
- `PoseBoneConstraints.remove`
- `UILayout.template_constraint`

## 2.4.167 ConstraintActuator(Actuator)

base classes — `bpy_struct`, `Actuator`

**class** `bpy.types.ConstraintActuator` (*Actuator*)

Actuator to handle Constraints

**angle\_max**

Maximum angle (in degree) allowed with target direction. No correction is done if angle with target direction is between min and max

**Type** float in [0, 180], default 0.0

**angle\_min**

Minimum angle (in degree) to maintain with target direction. No correction is done if angle with target direction is between min and max

**Type** float in [0, 180], default 0.0

**damping**

Damping factor: time constant (in frame) of low pass filter

**Type** int in [-32768, 32767], default 0

**damping\_rotation**

Use a different damping for orientation

**Type** int in [-32768, 32767], default 0

**direction**

Set the direction of the ray

**Type** enum in ['NONE', 'DIRPX', 'DIRPY', 'DIRPZ', 'DIRNX', 'DIRNY', 'DIRNZ'], default 'NONE'

**direction\_axis**

Select the axis to be aligned along the reference direction

**Type** enum in ['NONE', 'DIRPX', 'DIRPY', 'DIRPZ', 'DIRNX', 'DIRNY', 'DIRNZ'], default 'NONE'

**direction\_axis\_pos**

Select the axis to be aligned along the reference direction

**Type** enum in ['NONE', 'DIRPX', 'DIRPY', 'DIRPZ'], default 'NONE'

**distance**

Keep this distance to target

**Type** float in [-inf, inf], default 0.0

**fh\_damping**

Damping factor of the force field spring

**Type** float in [-inf, inf], default 0.0

**fh\_force**

Spring force within the force field area

**Type** float in [-inf, inf], default 0.0

**fh\_height**

Height of the force field area

**Type** float in [-inf, inf], default 0.0

**limit**

**Type** enum in ['NONE', 'LOCX', 'LOCY', 'LOCZ'], default 'NONE'

**limit\_max**

**Type** float in [-inf, inf], default 0.0

**limit\_min**

**Type** float in [-inf, inf], default 0.0

**material**

Ray detects only Objects with this material

**Type** string, default ""

**mode**

The type of the constraint

**Type** enum in ['LOC', 'DIST', 'ORI', 'FH'], default 'LOC'

**property**

Ray detect only Objects with this property

**Type** string, default ""

**range**

Set the maximum length of ray

**Type** float in [-inf, inf], default 0.0

**rotation\_max**

Reference Direction

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**time**

Maximum activation time in frame, 0 for unlimited

**Type** int in [-32768, 32767], default 0

**use\_fh\_normal**

Add a horizontal spring force on slopes

**Type** boolean, default False

**use\_fh\_paralel\_axis**

Keep object axis parallel to normal

**Type** boolean, default False

**use\_force\_distance**

Force distance of object to point of impact of ray

**Type** boolean, default False

**use\_local**

Set ray along object's axis or global axis

**Type** boolean, default False

**use\_material\_detect**

Detect material instead of property

**Type** boolean, default False

**use\_normal**

Set object axis along (local axis) or parallel (global axis) to the normal at hit position

**Type** boolean, default False

**use\_persistent**

Persistent actuator: stays active even if ray does not reach target

**Type** boolean, default False

**Inherited Properties**

- `bpy_struct.id_data`
- `Actuator.name`
- `Actuator.show_expanded`
- `Actuator.pin`
- `Actuator.type`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Actuator.link`
- `Actuator.unlink`

**2.4.168 ConstraintTarget(`bpy_struct`)**

base class — `bpy_struct`

**class** `bpy.types.ConstraintTarget` (*`bpy_struct`*)

Target object for multi-target constraints

**subtarget**

**Type** string, default ""

**target**

Target Object

**Type** Object

**Inherited Properties**

- `bpy_struct.id_data`



## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `PythonConstraint.targets`

### 2.4.169 Context(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.Context` (*bpy\_struct*)  
Current windowmanager and data context

**area**

**Type** `Area`, (readonly)

**blend\_data**

**Type** `BlendData`, (readonly)

**mode**

**Type** enum in ['EDIT\_MESH', 'EDIT\_CURVE', 'EDIT\_SURFACE', 'EDIT\_TEXT', 'EDIT\_ARMATURE', 'EDIT\_METABALL', 'EDIT\_LATTICE', 'POSE', 'SCULPT', 'PAINT\_WEIGHT', 'PAINT\_VERTEX', 'PAINT\_TEXTURE', 'PARTICLE', 'OBJECT'], default 'EDIT\_MESH', (readonly)

**region**

**Type** `Region`, (readonly)

**region\_data**

**Type** `RegionView3D`, (readonly)

**scene**

**Type** `Scene`, (readonly)

**screen**

**Type** `Screen`, (readonly)

**space\_data**

**Type** `Space`, (readonly)

**tool\_settings**

**Type** `ToolSettings`, (readonly)

**user\_preferences**

**Type** `UserPreferences`, (readonly)

**window**

**Type** `Window`, (readonly)

**window\_manager**

**Type** `WindowManager`, (readonly)

**static copy** (*self*)

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Header.draw`
- `KeyingSetInfo.generate`
- `KeyingSetInfo.iterator`
- `KeyingSetInfo.poll`
- `Macro.draw`
- `Macro.poll`
- `Menu.draw`
- `Menu.poll`

- `Operator.cancel`
- `Operator.check`
- `Operator.draw`
- `Operator.execute`
- `Operator.invoke`
- `Operator.modal`
- `Operator.poll`
- `Panel.draw`
- `Panel.draw_header`
- `Panel.poll`
- `RenderEngine.view_draw`
- `RenderEngine.view_update`

### 2.4.170 ControlFluidSettings(FluidSettings)

base classes — `bpy_struct`, `FluidSettings`

**class** `bpy.types.ControlFluidSettings` (*FluidSettings*)

Fluid simulation settings for objects controlling the motion of fluid in the simulation

**attraction\_radius**

Specifies the force field radius around the control object

**Type** float in [0, 10], default 0.0

**attraction\_strength**

Force strength for directional attraction towards the control object

**Type** float in [-10, 10], default 0.0

**end\_time**

Specifies time when the control particles are deactivated

**Type** float in [0, 100], default 0.0

**quality**

Specifies the quality which is used for object sampling. (higher = better but slower)

**Type** float in [5, 100], default 0.0

**start\_time**

Specifies time when the control particles are activated

**Type** float in [0, 100], default 0.0

**use**

Object contributes to the fluid simulation

**Type** boolean, default False

**use\_reverse\_frames**

Reverse control object movement

**Type** boolean, default False

**velocity\_radius**

Specifies the force field radius around the control object

**Type** float in [0, 10], default 0.0

**velocity\_strength**

Force strength of how much of the control object's velocity is influencing the fluid velocity

**Type** float in [0, 10], default 0.0

### Inherited Properties

- `bpy_struct.id_data`
- `FluidSettings.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.171 Controller(`bpy_struct`)

base class — `bpy_struct`

subclasses — `ExpressionController`, `XnorController`, `PythonController`, `AndController`, `NorController`, `OrController`, `XorController`, `NandController`

**class** `bpy.types.Controller` (*`bpy_struct`*)

Game engine logic brick to process events, connecting sensors to actuators

#### **name**

**Type** string, default ""

#### **show\_expanded**

Set controller expanded in the user interface

**Type** boolean, default False

#### **states**

Set Controller state index (1 to 30)

**Type** int in [1, 30], default 0

#### **type**

- `LOGIC_AND` And, Logic And.
- `LOGIC_OR` Or, Logic Or.
- `LOGIC_NAND` Nand, Logic Nand.

- LOGIC\_NOR Nor, Logic Nor.
- LOGIC\_XOR Xor, Logic Xor.
- LOGIC\_XNOR Xnor, Logic Xnor.
- EXPRESSION Expression.
- PYTHON Python.

**Type** enum in ['LOGIC\_AND', 'LOGIC\_OR', 'LOGIC\_NAND', 'LOGIC\_NOR', 'LOGIC\_XOR', 'LOGIC\_XNOR', 'EXPRESSION', 'PYTHON'], default 'LOGIC\_AND'

#### **use\_priority**

Mark controller for execution before all non-marked controllers (good for startup scripts)

**Type** boolean, default False

**link** (*sensor=None, actuator=None*)

Link the controller with a sensor/actuator

#### **Parameters**

- **sensor** (*Sensor*, (optional)) – Sensor to link the controller to
- **actuator** (*Actuator*, (optional)) – Actuator to link the controller to

**unlink** (*sensor=None, actuator=None*)

Unlink the controller from a sensor/actuator

#### **Parameters**

- **sensor** (*Sensor*, (optional)) – Sensor to unlink the controller from
- **actuator** (*Actuator*, (optional)) – Actuator to unlink the controller from

### **Inherited Properties**

- `bpy_struct.id_data`

### **Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Actuator.link`
- `Actuator.unlink`
- `GameObjectSettings.controllers`
- `Sensor.link`
- `Sensor.unlink`

### 2.4.172 CopyLocationConstraint(Constraint)

base classes — `bpy_struct`, `Constraint`

**class** `bpy.types.CopyLocationConstraint` (*Constraint*)  
Copies the location of the target

**head\_tail**

Target along length of bone: Head=0, Tail=1

**Type** float in [0, 1], default 0.0

**invert\_x**

Invert the X location

**Type** boolean, default False

**invert\_y**

Invert the Y location

**Type** boolean, default False

**invert\_z**

Invert the Z location

**Type** boolean, default False

**subtarget**

**Type** string, default ""

**target**

Target Object

**Type** `Object`

**use\_offset**

Add original location into copied location

**Type** boolean, default False

**use\_x**

Copy the target's X location

**Type** boolean, default False

**use\_y**

Copy the target's Y location

**Type** boolean, default False

**use\_z**

Copy the target's Z location

**Type** boolean, default False

## Inherited Properties

- `bpy_struct.id_data`
- `Constraint.name`
- `Constraint.active`
- `Constraint.mute`
- `Constraint.show_expanded`
- `Constraint.influence`
- `Constraint.error_location`
- `Constraint.owner_space`
- `Constraint.is_proxy_local`
- `Constraint.error_rotation`
- `Constraint.target_space`
- `Constraint.type`
- `Constraint.is_valid`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.173 CopyRotationConstraint(Constraint)

base classes — `bpy_struct`, `Constraint`

**class** `bpy.types.CopyRotationConstraint` (*Constraint*)  
Copies the rotation of the target

**invert\_x**

Invert the X rotation

**Type** boolean, default False

**invert\_y**

Invert the Y rotation

**Type** boolean, default False

**invert\_z**

Invert the Z rotation

**Type** boolean, default False

**subtarget**

**Type** string, default ""

**target**

Target Object

**Type** Object

**use\_offset**

Add original rotation into copied rotation

**Type** boolean, default False

**use\_x**

Copy the target's X rotation

**Type** boolean, default False

**use\_y**

Copy the target's Y rotation

**Type** boolean, default False

**use\_z**

Copy the target's Z rotation

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `Constraint.name`
- `Constraint.active`
- `Constraint.mute`
- `Constraint.show_expanded`
- `Constraint.influence`
- `Constraint.error_location`
- `Constraint.owner_space`
- `Constraint.is_proxy_local`
- `Constraint.error_rotation`
- `Constraint.target_space`
- `Constraint.type`
- `Constraint.is_valid`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`



- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.174 CopyScaleConstraint(Constraint)

base classes — `bpy_struct`, `Constraint`

**class** `bpy.types.CopyScaleConstraint` (*Constraint*)

Copies the scale of the target

**subtarget**

**Type** string, default ""

**target**

Target Object

**Type** `Object`

**use\_offset**

Add original scale into copied scale

**Type** boolean, default False

**use\_x**

Copy the target's X scale

**Type** boolean, default False

**use\_y**

Copy the target's Y scale

**Type** boolean, default False

**use\_z**

Copy the target's Z scale

**Type** boolean, default False

#### Inherited Properties

- `bpy_struct.id_data`
- `Constraint.name`
- `Constraint.active`
- `Constraint.mute`
- `Constraint.show_expanded`
- `Constraint.influence`
- `Constraint.error_location`
- `Constraint.owner_space`
- `Constraint.is_proxy_local`
- `Constraint.error_rotation`
- `Constraint.target_space`
- `Constraint.type`
- `Constraint.is_valid`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.175 CopyTransformsConstraint(Constraint)

base classes — `bpy_struct`, `Constraint`

**class** `bpy.types.CopyTransformsConstraint` (*Constraint*)

Copies all the transforms of the target

**head\_tail**

Target along length of bone: Head=0, Tail=1

**Type** float in [0, 1], default 0.0

**subtarget**

**Type** string, default ""

**target**

Target Object

**Type** Object

## Inherited Properties

- `bpy_struct.id_data`
- `Constraint.name`
- `Constraint.active`
- `Constraint.mute`
- `Constraint.show_expanded`
- `Constraint.influence`
- `Constraint.error_location`
- `Constraint.owner_space`
- `Constraint.is_proxy_local`
- `Constraint.error_rotation`
- `Constraint.target_space`
- `Constraint.type`
- `Constraint.is_valid`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.176 Curve(ID)

base classes — `bpy_struct`, `ID`

subclasses — `SurfaceCurve`, `TextCurve`

**class** `bpy.types.Curve` (*ID*)

Curve datablock storing curves, splines and NURBS

**animation\_data**

Animation data for this datablock

**Type** `AnimData`, (readonly)

**bevel\_depth**

Bevel depth when not using a bevel object

**Type** float in `[-inf, inf]`, default 0.0

**bevel\_object**

Curve object name that defines the bevel shape

**Type** `Object`

**bevel\_resolution**

Bevel resolution when depth is non-zero and no specific bevel object has been defined

**Type** int in `[0, 32]`, default 0

**dimensions**

Select 2D or 3D curve type

- 2D 2D, Clamp the Z axis of of the curve.
- 3D 3D, Allow editing on the Z axis of this curve, also allows tilt and curve radius to be used.

**Type** enum in `['2D', '3D']`, default '2D'

**eval\_time**

Parametric position along the length of the curve that Objects ‘following’ it should be at. Position is evaluated by dividing by the ‘Path Length’ value

**Type** float in [-inf, inf], default 0.0

**extrude**

Amount of curve extrusion when not using a bevel object

**Type** float in [0, inf], default 0.0

**fill\_mode**

Mode of filling curve

**Type** enum in [‘FULL’, ‘BACK’, ‘FRONT’, ‘HALF’], default ‘FULL’

**materials**

**Type** `IDMaterials bpy_prop_collection` of `Material`, (readonly)

**offset**

Offset the curve to adjust the width of a text

**Type** float in [-inf, inf], default 0.0

**path\_duration**

The number of frames that are needed to traverse the path, defining the maximum value for the ‘Evaluation Time’ setting

**Type** int in [1, 300000], default 0

**render\_resolution\_u**

Surface resolution in U direction used while rendering. Zero skips this property

**Type** int in [0, 32767], default 0

**render\_resolution\_v**

Surface resolution in V direction used while rendering. Zero skips this property

**Type** int in [0, 32767], default 0

**resolution\_u**

Surface resolution in U direction

**Type** int in [1, 32767], default 0

**resolution\_v**

Surface resolution in V direction

**Type** int in [1, 32767], default 0

**shape\_keys**

**Type** `Key`, (readonly)

**show\_handles**

Display Bezier handles in editmode

**Type** boolean, default False

**show\_normal\_face**

Display 3D curve normals in editmode

**Type** boolean, default False

**splines**

Collection of splines in this curve data object

**Type** `CurveSplines bpy_prop_collection` of `Spline`, (readonly)

**taper\_object**

Curve object name that defines the taper (width)

**Type** `Object`

**texspace\_location**

Texture space location

**Type** float array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`

**texspace\_size**

Texture space size

**Type** float array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`

**twist\_mode**

The type of tilt calculation for 3D Curves

- `Z_UP` Z-Up, Use Z-Up axis to calculate the curve twist at each point.
- `MINIMUM` Minimum, Use the least twist over the entire curve.
- `TANGENT` Tangent, Use the tangent to calculate twist.

**Type** enum in `['Z_UP', 'MINIMUM', 'TANGENT']`, default `'Z_UP'`

**twist\_smooth**

Smoothing iteration for tangents

**Type** float in `[-inf, inf]`, default `0.0`

**use\_auto\_texspace**

Adjusts active object's texture space automatically when transforming object

**Type** boolean, default `False`

**use\_deform\_bounds**

Use the mesh bounds to clamp the deformation

**Type** boolean, default `False`

**use\_fill\_deform**

Fill curve after applying shape keys and all modifiers

**Type** boolean, default `False`

**use\_path**

Enable the curve to become a translation path

**Type** boolean, default `False`

**use\_path\_follow**

Make curve path children to rotate along the path

**Type** boolean, default `False`

**use\_radius**

Option for paths: apply the curve radius with path following it and deforming

**Type** boolean, default `False`

**use\_stretch**

Option for curve-deform: makes deformed child to stretch along entire path

**Type** boolean, default False

**use\_time\_offset**

Children will use TimeOfs value as path distance offset

**Type** boolean, default False

**use\_uv\_as\_generated**

Uses the UV values as Generated textured coordinates

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`

### References

- `BlendData.curves`
- `BlendDataCurves.new`
- `BlendDataCurves.remove`

### 2.4.177 CurveMap(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.CurveMap` (*bpy\_struct*)

Curve in a curve mapping

**extend**

Extrapolate the curve or extend it horizontally

**Type** enum in ['HORIZONTAL', 'EXTRAPOLATED'], default 'HORIZONTAL', (readonly)

**points**

**Type** `bpy_prop_collection` of `CurveMapPoint`, (readonly)

#### Inherited Properties

- `bpy_struct.id_data`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

#### References

- `CurveMapping.curves`

### 2.4.178 CurveMapPoint(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.CurveMapPoint` (*bpy\_struct*)

Point of a curve used for a curve mapping

**handle\_type**

Curve interpolation at this point: Bezier or vector

**Type** enum in ['AUTO', 'VECTOR'], default 'AUTO', (readonly)

**location**

X/Y coordinates of the curve point

**Type** float array of 2 items in  $[-\text{inf}, \text{inf}]$ , default (0.0, 0.0), (readonly)

**select**

Selection state of the curve point

**Type** boolean, default False

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `CurveMap.points`

**2.4.179 CurveMapping(`bpy_struct`)**

base class — `bpy_struct`

**class** `bpy.types.CurveMapping` (*bpy\_struct*)

Curve mapping to map color, vector and scalar values to other values using a user defined curve

**black\_level**

For RGB curves, the color that black is mapped to

**Type** float array of 3 items in  $[-1000, 1000]$ , default (0.0, 0.0, 0.0)

**clip\_max\_x**

**Type** float in  $[-100, 100]$ , default 0.0

**clip\_max\_y**



**Type** float in [-100, 100], default 0.0

**clip\_min\_x**

**Type** float in [-100, 100], default 0.0

**clip\_min\_y**

**Type** float in [-100, 100], default 0.0

**curves**

**Type** `bpy_prop_collection` of `CurveMap`, (readonly)

**use\_clip**

Force the curve view to fit a defined boundary

**Type** boolean, default False

**white\_level**

For RGB curves, the color that white is mapped to

**Type** float array of 3 items in [-1000, 1000], default (0.0, 0.0, 0.0)

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Brush.curve`
- `CompositorNodeCurveRGB.mapping`
- `CompositorNodeCurveVec.mapping`
- `CompositorNodeHueCorrect.mapping`
- `CompositorNodeTime.curve`
- `ParticleBrush.curve`
- `PointDensity.falloff_curve`

- `PointLamp.falloff_curve`
- `ShaderNodeRGBCurve.mapping`
- `ShaderNodeVectorCurve.mapping`
- `SpaceImageEditor.curve`
- `SpotLamp.falloff_curve`
- `TextureNodeCurveRGB.mapping`
- `TextureNodeCurveTime.curve`
- `VertexWeightEditModifier.map_curve`
- `WarpModifier.falloff_curve`

## 2.4.180 CurveModifier(Modifier)

base classes — `bpy_struct`, `Modifier`

**class** `bpy.types.CurveModifier` (*Modifier*)

Curve deformation modifier

**deform\_axis**

The axis that the curve deforms along

**Type** enum in ['POS\_X', 'POS\_Y', 'POS\_Z', 'NEG\_X', 'NEG\_Y', 'NEG\_Z'], default 'POS\_X'

**object**

Curve object to deform with

**Type** `Object`

**vertex\_group**

Name of Vertex Group which determines influence of modifier per point

**Type** string, default ""

### Inherited Properties

- `bpy_struct.id_data`
- `Modifier.name`
- `Modifier.use_apply_on_spline`
- `Modifier.show_in_editmode`
- `Modifier.show_expanded`
- `Modifier.show_on_cage`
- `Modifier.show_viewport`
- `Modifier.show_render`
- `Modifier.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`

- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.181 CurveSplines(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.CurveSplines` (*bpy\_struct*)

Collection of curve splines

**active**

Active curve spline

**Type** `Object`

**new** (*type*)

Add a new spline to the curve

**Parameters** `type` (*enum in ['POLY', 'BEZIER', 'BSPLINE', 'CARDINAL', 'NURBS']*) – type for the new spline

**Returns** The newly created spline

**Return type** `Spline`

**remove** (*spline*)

Remove a spline from a curve

**Parameters** `spline` (`Spline`, (never `None`)) – The spline to remove

#### Inherited Properties

- `bpy_struct.id_data`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`

- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Curve.splines`

## 2.4.182 DampedTrackConstraint(Constraint)

base classes — `bpy_struct`, `Constraint`

**class** `bpy.types.DampedTrackConstraint` (*Constraint*)

Points toward target by taking the shortest rotation path

### **head\_tail**

Target along length of bone: Head=0, Tail=1

**Type** float in [0, 1], default 0.0

### **subtarget**

**Type** string, default ""

### **target**

Target Object

**Type** `Object`

### **track\_axis**

Axis that points to the target object

**Type** enum in ['TRACK\_X', 'TRACK\_Y', 'TRACK\_Z', 'TRACK\_NEGATIVE\_X', 'TRACK\_NEGATIVE\_Y', 'TRACK\_NEGATIVE\_Z'], default 'TRACK\_X'

## Inherited Properties

- `bpy_struct.id_data`
- `Constraint.name`
- `Constraint.active`
- `Constraint.mute`
- `Constraint.show_expanded`
- `Constraint.influence`
- `Constraint.error_location`
- `Constraint.owner_space`
- `Constraint.is_proxy_local`
- `Constraint.error_rotation`
- `Constraint.target_space`
- `Constraint.type`
- `Constraint.is_valid`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`

- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.183 DecimateModifier(Modifier)

base classes — `bpy_struct`, `Modifier`

**class** `bpy.types.DecimateModifier` (*Modifier*)

Decimation modifier

**face\_count**

The current number of faces in the decimated mesh

**Type** `int` in `[-inf, inf]`, default 0, (readonly)

**ratio**

Defines the ratio of triangles to reduce to

**Type** `float` in `[0, 1]`, default 0.0

#### Inherited Properties

- `bpy_struct.id_data`
- `Modifier.name`
- `Modifier.use_apply_on_spline`
- `Modifier.show_in_editmode`
- `Modifier.show_expanded`
- `Modifier.show_on_cage`
- `Modifier.show_viewport`
- `Modifier.show_render`
- `Modifier.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`

- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.184 DelaySensor(Sensor)

base classes — `bpy_struct`, `Sensor`

**class** `bpy.types.DelaySensor` (*Sensor*)

Sensor to send delayed events

**delay**

Delay in number of logic tics before the positive trigger (default 60 per second)

**Type** `int` in `[0, 5000]`, default 0

**duration**

If >0, delay in number of logic tics before the negative trigger following the positive trigger

**Type** `int` in `[0, 5000]`, default 0

**use\_repeat**

Toggle repeat option. If selected, the sensor restarts after Delay+Dur logic tics

**Type** `boolean`, default `False`

#### Inherited Properties

- `bpy_struct.id_data`
- `Sensor.name`
- `Sensor.show_expanded`
- `Sensor.frequency`
- `Sensor.invert`
- `Sensor.use_level`
- `Sensor.pin`
- `Sensor.use_pulse_false_level`
- `Sensor.use_pulse_true_level`
- `Sensor.use_tap`
- `Sensor.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`

- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Sensor.link`
- `Sensor.unlink`

### 2.4.185 DisplaceModifier(Modifier)

base classes — `bpy_struct`, `Modifier`

**class** `bpy.types.DisplaceModifier` (*Modifier*)

Displacement modifier

#### **direction**

- **X** X, Uses the texture's intensity value to displace in the X direction.
- **Y** Y, Uses the texture's intensity value to displace in the Y direction.
- **Z** Z, Uses the texture's intensity value to displace in the Z direction.
- **NORMAL** Normal, Uses the texture's intensity value to displace in the normal direction.
- **RGB\_TO\_XYZ** RGB to XYZ, Uses the texture's RGB values to displace the mesh in the XYZ direction.

**Type** enum in ['X', 'Y', 'Z', 'NORMAL', 'RGB\_TO\_XYZ'], default 'X'

#### **mid\_level**

Material value that gives no displacement

**Type** float in [-inf, inf], default 0.0

#### **strength**

Amount to displace geometry

**Type** float in [-inf, inf], default 0.0

#### **texture**

**Type** `Texture`

#### **texture\_coords**

- **LOCAL** Map, Uses the local coordinate system for the texture coordinates.
- **GLOBAL** Global, Uses the global coordinate system for the texture coordinates.
- **OBJECT** Object, Uses the linked object's local coordinate system for the texture coordinates.
- **UV** UV, Uses UV coordinates for the texture coordinates.

**Type** enum in ['LOCAL', 'GLOBAL', 'OBJECT', 'UV'], default 'LOCAL'

#### **texture\_coords\_object**

Object to set the texture coordinates

**Type** `Object`

**uv\_layer**

UV layer name

**Type** string, default ""

**vertex\_group**

Name of Vertex Group which determines influence of modifier per point

**Type** string, default ""

**Inherited Properties**

- `bpy_struct.id_data`
- `Modifier.name`
- `Modifier.use_apply_on_spline`
- `Modifier.show_in_editmode`
- `Modifier.show_expanded`
- `Modifier.show_on_cage`
- `Modifier.show_viewport`
- `Modifier.show_render`
- `Modifier.type`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**2.4.186 DistortedNoiseTexture(Texture)**

base classes — `bpy_struct`, `ID`, `Texture`

**class** `bpy.types.DistortedNoiseTexture` (*Texture*)

Procedural distorted noise texture

**distortion**

Amount of distortion

**Type** float in [0, 10], default 0.0

**nabla**

Size of derivative offset used for calculating normal



**Type** float in [0.001, 0.1], default 0.0

### **noise\_basis**

Sets the noise basis used for turbulence

- BLENDER\_ORIGINAL** Blender Original, Noise algorithm - Blender original: Smooth interpolated noise.
- ORIGINAL\_PERLIN** Original Perlin, Noise algorithm - Original Perlin: Smooth interpolated noise.
- IMPROVED\_PERLIN** Improved Perlin, Noise algorithm - Improved Perlin: Smooth interpolated noise.
- VORONOI\_F1** Voronoi F1, Noise algorithm - Voronoi F1: Returns distance to the closest feature point.
- VORONOI\_F2** Voronoi F2, Noise algorithm - Voronoi F2: Returns distance to the 2nd closest feature point.
- VORONOI\_F3** Voronoi F3, Noise algorithm - Voronoi F3: Returns distance to the 3rd closest feature point.
- VORONOI\_F4** Voronoi F4, Noise algorithm - Voronoi F4: Returns distance to the 4th closest feature point.
- VORONOI\_F2\_F1** Voronoi F2-F1, Noise algorithm - Voronoi F1-F2.
- VORONOI\_CRACKLE** Voronoi Crackle, Noise algorithm - Voronoi Crackle: Voronoi tessellation with sharp edges.
- CELL\_NOISE** Cell Noise, Noise algorithm - Cell Noise: Square cell tessellation.

**Type** enum in ['BLENDER\_ORIGINAL', 'ORIGINAL\_PERLIN', 'IMPROVED\_PERLIN', 'VORONOI\_F1', 'VORONOI\_F2', 'VORONOI\_F3', 'VORONOI\_F4', 'VORONOI\_F2\_F1', 'VORONOI\_CRACKLE', 'CELL\_NOISE'], default 'BLENDER\_ORIGINAL'

### **noise\_distortion**

Sets the noise basis for the distortion

- BLENDER\_ORIGINAL** Blender Original, Noise algorithm - Blender original: Smooth interpolated noise.
- ORIGINAL\_PERLIN** Original Perlin, Noise algorithm - Original Perlin: Smooth interpolated noise.
- IMPROVED\_PERLIN** Improved Perlin, Noise algorithm - Improved Perlin: Smooth interpolated noise.
- VORONOI\_F1** Voronoi F1, Noise algorithm - Voronoi F1: Returns distance to the closest feature point.
- VORONOI\_F2** Voronoi F2, Noise algorithm - Voronoi F2: Returns distance to the 2nd closest feature point.
- VORONOI\_F3** Voronoi F3, Noise algorithm - Voronoi F3: Returns distance to the 3rd closest feature point.
- VORONOI\_F4** Voronoi F4, Noise algorithm - Voronoi F4: Returns distance to the 4th closest feature point.
- VORONOI\_F2\_F1** Voronoi F2-F1, Noise algorithm - Voronoi F1-F2.
- VORONOI\_CRACKLE** Voronoi Crackle, Noise algorithm - Voronoi Crackle: Voronoi tessellation with sharp edges.

- CELL\_NOISE Cell Noise, Noise algorithm - Cell Noise: Square cell tessellation.

**Type** enum in ['BLENDER\_ORIGINAL', 'ORIGINAL\_PERLIN', 'IMPROVED\_PERLIN', 'VORONOI\_F1', 'VORONOI\_F2', 'VORONOI\_F3', 'VORONOI\_F4', 'VORONOI\_F2\_F1', 'VORONOI\_CRACKLE', 'CELL\_NOISE'], default 'BLENDER\_ORIGINAL'

**noise\_scale**

Sets scaling for noise input

**Type** float in [0.0001, inf], default 0.0

**users\_material**

Materials that use this texture (readonly)

**users\_object\_modifier**

Object modifiers that use this texture (readonly)

### Inherited Properties

- bpy\_struct.id\_data
- ID.name
- ID.use\_fake\_user
- ID.is\_updated
- ID.is\_updated\_data
- ID.library
- ID.tag
- ID.users
- Texture.animation\_data
- Texture.intensity
- Texture.color\_ramp
- Texture.contrast
- Texture.factor\_blue
- Texture.factor\_green
- Texture.factor\_red
- Texture.node\_tree
- Texture.saturation
- Texture.use\_preview\_alpha
- Texture.type
- Texture.use\_color\_ramp
- Texture.use\_nodes
- Texture.users\_material
- Texture.users\_object\_modifier
- Texture.users\_material
- Texture.users\_object\_modifier

### Inherited Functions

- bpy\_struct.as\_pointer
- bpy\_struct.callback\_add
- bpy\_struct.callback\_remove
- bpy\_struct.driver\_add
- bpy\_struct.driver\_remove
- bpy\_struct.get

- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`
- `Texture.evaluate`

### 2.4.187 DomainFluidSettings(FluidSettings)

base classes — `bpy_struct`, `FluidSettings`

**class** `bpy.types.DomainFluidSettings` (*FluidSettings*)  
Fluid simulation settings for the domain of a fluid simulation

**compressibility**

Allowed compressibility due to gravitational force for standing fluid. (directly affects simulation step size)

**Type** float in [0.001, 0.1], default 0.0

**end\_time**

Simulation time of the last blender frame (in seconds)

**Type** float in [0, 100], default 0.0

**filepath**

Directory (and/or filename prefix) to store baked fluid simulation files in

**Type** string, default ""

**fluid\_mesh\_vertices**

Vertices of the fluid mesh generated by simulation

**Type** `bpy_prop_collection` of `FluidMeshVertex`, (readonly)

**generate\_particles**

Amount of particles to generate (0=off, 1=normal, >1=more)

**Type** float in [0, 10], default 0.0

**gravity**

Gravity in X, Y and Z direction

**Type** float array of 3 items in [-1000.1, 1000.1], default (0.0, 0.0, 0.0)

**grid\_levels**

Number of coarsened grids to use (-1 for automatic)

**Type** int in [-1, 4], default 0

**memory\_estimate**

Estimated amount of memory needed for baking the domain

**Type** string, default "", (readonly)

**partial\_slip\_factor**

Amount of mixing between no- and free-slip, 0 is no slip and 1 is free slip

**Type** float in [0, 1], default 0.0

**preview\_resolution**

Preview resolution in X,Y and Z direction

**Type** int in [1, 100], default 0

**render\_display\_mode**

How to display the mesh for rendering

- GEOMETRY Geometry, Display geometry.
- PREVIEW Preview, Display preview quality results.
- FINAL Final, Display final quality results.

**Type** enum in ['GEOMETRY', 'PREVIEW', 'FINAL'], default 'GEOMETRY'

**resolution**

Domain resolution in X,Y and Z direction

**Type** int in [1, 1024], default 0

**simulation\_scale**

Size of the simulation domain in metres

**Type** float in [0.001, 10], default 0.0

**slip\_type**

- NOSLIP No Slip, Obstacle causes zero normal and tangential velocity (=sticky). Default for all. Only option for moving objects.
- PARTIALSLIP Partial Slip, Mix between no-slip and free-slip. Non moving objects only!.
- FREESLIP Free Slip, Obstacle only causes zero normal velocity (=not sticky). Non moving objects only!.

**Type** enum in ['NOSLIP', 'PARTIALSLIP', 'FREESLIP'], default 'NOSLIP'

**start\_time**

Simulation time of the first blender frame (in seconds)

**Type** float in [0, 100], default 0.0

**surface\_noobs**

**Type** boolean, default False

**surface\_smooth**

Amount of surface smoothing. A value of 0 is off, 1 is normal smoothing and more than 1 is extra smoothing

**Type** float in [0, 5], default 0.0

**surface\_subdivisions**

Number of isosurface subdivisions. This is necessary for the inclusion of particles into the surface generation. Warning - can lead to longer computation times!

**Type** int in [0, 5], default 0

**tracer\_particles**

Number of tracer particles to generate

**Type** int in [0, 10000], default 0

**use\_reverse\_frames**

Reverse fluid frames

**Type** boolean, default False

**use\_speed\_vectors**

Generate speed vectors for vector blur

**Type** boolean, default False

**use\_time\_override**

Use a custom start and end time (in seconds) instead of the scene's timeline

**Type** boolean, default False

**viewport\_display\_mode**

How to display the mesh in the viewport

- GEOMETRY Geometry, Display geometry.
- PREVIEW Preview, Display preview quality results.
- FINAL Final, Display final quality results.

**Type** enum in ['GEOMETRY', 'PREVIEW', 'FINAL'], default 'GEOMETRY'

**viscosity\_base**

Viscosity setting: value that is multiplied by 10 to the power of (exponent\*-1)

**Type** float in [0, 10], default 0.0

**viscosity\_exponent**

Negative exponent for the viscosity value (to simplify entering small values e.g.  $5 \cdot 10^{-6}$ .)

**Type** int in [0, 10], default 0

**viscosity\_preset**

Set viscosity of the fluid to a preset value, or use manual input

- MANUAL Manual, Manual viscosity settings.
- WATER Water, Viscosity of  $1.0 \cdot 10^{-6}$ .
- OIL Oil, Viscosity of  $5.0 \cdot 10^{-5}$ .
- HONEY Honey, Viscosity of  $2.0 \cdot 10^{-3}$ .

**Type** enum in ['MANUAL', 'WATER', 'OIL', 'HONEY'], default 'MANUAL'

**Inherited Properties**

- `bpy_struct.id_data`
- `FluidSettings.type`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.188 DopeSheet(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.DopeSheet` (*`bpy_struct`*)

Settings for filtering the channels shown in Animation Editors

**filter\_fcurve\_name**

F-Curve live filtering string

**Type** string, default ""

**filter\_group**

Group that included Object should be a member of

**Type** Group

**show\_armatures**

Include visualization of Armature related Animation data

**Type** boolean, default False

**show\_cameras**

Include visualization of Camera related Animation data

**Type** boolean, default False

**show\_curves**

Include visualization of Curve related Animation data

**Type** boolean, default False

**show\_datablock\_filters**

Show options for whether channels related to certain types of data are included

**Type** boolean, default False

**show\_expanded\_summary**

Collapse summary when shown, so all other channels get hidden (DopeSheet Editors Only)

**Type** boolean, default False

**show\_hidden**

Include channels from objects/bone that aren't visible

**Type** boolean, default False

**show\_lamps**

Include visualization of Lamp related Animation data

**Type** boolean, default False

**show\_lattices**

Include visualization of Lattice related Animation data

**Type** boolean, default False

**show\_materials**

Include visualization of Material related Animation data

**Type** boolean, default False

**show\_meshes**

Include visualization of Mesh related Animation data

**Type** boolean, default False

**show\_metaballs**

Include visualization of Metaball related Animation data

**Type** boolean, default False

**show\_missing\_nla**

Include Animation Data blocks with no NLA data (NLA Editor only)

**Type** boolean, default False

**show\_nodes**

Include visualization of Node related Animation data

**Type** boolean, default False

**show\_only\_group\_objects**

Only include channels from Objects in the specified Group

**Type** boolean, default False

**show\_only\_matching\_fcurves**

Only include F-Curves with names containing search text

**Type** boolean, default False

**show\_only\_selected**

Only include channels relating to selected objects and data

**Type** boolean, default False

**show\_particles**

Include visualization of Particle related Animation data

**Type** boolean, default False

**show\_scenes**

Include visualization of Scene related Animation data

**Type** boolean, default False

**show\_shapekeys**

Include visualization of ShapeKey related Animation data

**Type** boolean, default False

**show\_speakers**

Include visualization of Speaker related Animation data

**Type** boolean, default False

**show\_summary**

Display an additional 'summary' line (DopeSheet Editors only)

**Type** boolean, default False

**show\_textures**

Include visualization of Texture related Animation data

**Type** boolean, default False

**show\_transforms**

Include visualization of Object-level Animation data (mostly Transforms)

**Type** boolean, default False

**show\_worlds**

Include visualization of World related Animation data

**Type** boolean, default False

**source**

ID-Block representing source data, currently ID\_SCE (for Dopesheet), and ID\_SC (for Grease Pencil)

**Type** ID, (readonly)

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`



## References

- `SpaceDopeSheetEditor.dopesheet`
- `SpaceGraphEditor.dopesheet`
- `SpaceNLA.dopesheet`

### 2.4.189 Driver(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.Driver` (*bpy\_struct*)

Driver for the value of a setting based on an external value

**expression**

Expression to use for Scripted Expression

**Type** string, default ""

**is\_valid**

Driver could not be evaluated in past, so should be skipped

**Type** boolean, default False

**show\_debug\_info**

Show intermediate values for the driver calculations to allow debugging of drivers

**Type** boolean, default False

**type**

Driver type

**Type** enum in ['AVERAGE', 'SUM', 'SCRIPTED', 'MIN', 'MAX'], default 'AVERAGE'

**variables**

Properties acting as inputs for this driver

**Type** `ChannelDriverVariables` `bpy_prop_collection` of `DriverVariable`, (readonly)

#### Inherited Properties

- `bpy_struct.id_data`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`

- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `FCurve.driver`

### 2.4.190 DriverTarget(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.DriverTarget` (*bpy\_struct*)

Source of input values for driver variables

**bone\_target**

Name of PoseBone to use as target

**Type** string, default ""

**data\_path**

RNA Path (from ID-block) to property used

**Type** string, default ""

**id**

ID-block that the specific property used can be found from (id\_type property must be set first)

**Type** ID

**id\_type**

Type of ID-block that can be used

**Type** enum in ['ACTION', 'ARMATURE', 'BRUSH', 'CAMERA', 'CURVE', 'FONT', 'GREASEPENCIL', 'GROUP', 'IMAGE', 'KEY', 'LAMP', 'LIBRARY', 'LATTICE', 'MATERIAL', 'META', 'MESH', 'NODETREE', 'OBJECT', 'PARTICLE', 'SCENE', 'SCREEN', 'SPEAKER', 'SOUND', 'TEXT', 'TEXTURE', 'WORLD', 'WINDOWMANAGER'], default 'OBJECT'

**transform\_space**

Space in which transforms are used

- `WORLD_SPACE` World Space, Transforms include effects of parenting/restpose and constraints.
- `TRANSFORM_SPACE` Transform Space, Transforms don't include parenting/restpose or constraints.
- `LOCAL_SPACE` Local Space, Transforms include effects of constraints but not parenting/restpose.

**Type** enum in ['WORLD\_SPACE', 'TRANSFORM\_SPACE', 'LOCAL\_SPACE'], default 'WORLD\_SPACE'

**transform\_type**

Driver variable type

**Type** enum in ['LOC\_X', 'LOC\_Y', 'LOC\_Z', 'ROT\_X', 'ROT\_Y', 'ROT\_Z', 'SCALE\_X', 'SCALE\_Y', 'SCALE\_Z'], default 'LOC\_X'

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `DriverVariable.targets`

### 2.4.191 DriverVariable(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.DriverVariable` (*bpy\_struct*)

Variable from some source/target for driver relationship

**name**

Name to use in scripted expressions/functions (no spaces or dots are allowed, and must start with a letter)

**Type** string, default ""

**targets**

Sources of input data for evaluating this variable

**Type** `bpy_prop_collection` of `DriverTarget`, (readonly)

**type**

Driver variable type

- `SINGLE_PROP` Single Property, Use the value from some RNA property (Default).
- `TRANSFORMS` Transform Channel, Final transformation value of object or bone.
- `ROTATION_DIFF` Rotational Difference, Use the angle between two bones.
- `LOC_DIFF` Distance, Distance between two bones or objects.

**Type** enum in ['SINGLE\_PROP', 'TRANSFORMS', 'ROTATION\_DIFF', 'LOC\_DIFF'], default 'SINGLE\_PROP'

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `ChannelDriverVariables.new`
- `ChannelDriverVariables.remove`
- `Driver.variables`

## 2.4.192 DupliObject(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.DupliObject` (*bpy\_struct*)

An object duplicate

#### **matrix**

Object duplicate transformation matrix

**Type** float array of 16 items in [-inf, inf], default (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)

#### **matrix\_original**

The original matrix of this object before it was duplicated

**Type** float array of 16 items in [-inf, inf], default (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)

#### **object**

Object being duplicated

**Type** `Object`, (readonly)

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Object.dupli_list`

## 2.4.193 EdgeSplitModifier(Modifier)

base classes — `bpy_struct`, `Modifier`

**class** `bpy.types.EdgeSplitModifier` (*Modifier*)  
Edge splitting modifier to create sharp edges

#### **split\_angle**

Angle above which to split edges

**Type** float in [0, 3.14159], default 0.0

#### **use\_edge\_angle**

Split edges with high angle between faces

**Type** boolean, default False

#### **use\_edge\_sharp**

Split edges that are marked as sharp

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `Modifier.name`
- `Modifier.use_apply_on_spline`
- `Modifier.show_in_editmode`
- `Modifier.show_expanded`
- `Modifier.show_on_cage`
- `Modifier.show_viewport`
- `Modifier.show_render`
- `Modifier.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.194 EditBone(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.EditBone` (*bpy\_struct*)  
Editmode bone in an Armature datablock

**bbone\_in**

Length of first Bezier Handle (for B-Bones only)

**Type** float in [0, 2], default 0.0

**bbone\_out**

Length of second Bezier Handle (for B-Bones only)

**Type** float in [0, 2], default 0.0

**bbone\_segments**

Number of subdivisions of bone (for B-Bones only)

**Type** int in [1, 32], default 0

**bbone\_x**

B-Bone X size

**Type** float in [0, 1000], default 0.0

#### **bbone\_z**

B-Bone Z size

**Type** float in [0, 1000], default 0.0

#### **envelope\_distance**

Bone deformation distance (for Envelope deform only)

**Type** float in [0, 1000], default 0.0

#### **envelope\_weight**

Bone deformation weight (for Envelope deform only)

**Type** float in [0, 1000], default 0.0

#### **head**

Location of head end of the bone

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

#### **head\_radius**

Radius of head of bone (for Envelope deform only)

**Type** float in [0, inf], default 0.0

#### **hide**

Bone is not visible when in Edit Mode

**Type** boolean, default False

#### **hide\_select**

Bone is able to be selected

**Type** boolean, default False

#### **layers**

Layers bone exists in

**Type** boolean array of 32 items, default (False, False)

#### **lock**

Bone is not able to be transformed when in Edit Mode

**Type** boolean, default False

#### **matrix**

Read-only matrix calculated from the roll (armature space)

**Type** float array of 16 items in [-inf, inf], default (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0), (readonly)

#### **name**

**Type** string, default ""

#### **parent**

Parent edit bone (in same Armature)

**Type** [EditBone](#)

#### **roll**

Bone rotation around head-tail axis

**Type** float in [-inf, inf], default 0.0

**select**

**Type** boolean, default False

**select\_head**

**Type** boolean, default False

**select\_tail**

**Type** boolean, default False

**show\_wire**

Bone is always drawn as Wireframe regardless of viewport draw mode (useful for non-obstructive custom bone shapes)

**Type** boolean, default False

**tail**

Location of tail end of the bone

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**tail\_radius**

Radius of tail of bone (for Envelope deform only)

**Type** float in [0, inf], default 0.0

**use\_connect**

When bone has a parent, bone's head is stuck to the parent's tail

**Type** boolean, default False

**use\_cyclic\_offset**

When bone doesn't have a parent, it receives cyclic offset effects

**Type** boolean, default False

**use\_deform**

Bone does not deform any geometry

**Type** boolean, default False

**use\_envelope\_multiply**

When deforming bone, multiply effects of Vertex Group weights with Envelope influence

**Type** boolean, default False

**use\_inherit\_rotation**

Bone inherits rotation or scale from parent bone

**Type** boolean, default False

**use\_inherit\_scale**

Bone inherits scaling from parent bone

**Type** boolean, default False

**use\_local\_location**

Bone location is set in local space

**Type** boolean, default False

**basename**

The name of this bone before any '.' character (readonly)



**center**

The midpoint between the head and the tail. (readonly)

**children**

A list of all the bones children. (readonly)

**children\_recursive**

A list of all children from this bone. (readonly)

**children\_recursive\_basename**

Returns a chain of children with the same base name as this bone. Only direct chains are supported, forks caused by multiple children with matching base names will terminate the function and not be returned. (readonly)

**length**

**The distance from head to tail**, when set the head is moved to fit the length.

**parent\_recursive**

A list of parents, starting with the immediate parent (readonly)

**vector**

**The direction this bone is pointing.** Utility function for (tail - head)

(readonly)

**x\_axis**

Vector pointing down the x-axis of the bone. (readonly)

**y\_axis**

Vector pointing down the x-axis of the bone. (readonly)

**z\_axis**

Vector pointing down the x-axis of the bone. (readonly)

**align\_roll** (*vector*)

Align the bone to a localspace roll so the Z axis points in the direction of the vector given

**Parameters** **vector** (*float array of 3 items in [-inf, inf]*) – Vector

**align\_orientation** (*other*)

Align this bone to another by moving its tail and settings its roll the length of the other bone is not used.

**parent\_index** (*parent\_test*)

The same as ‘bone in other\_bone.parent\_recursive’ but saved generating a list.

**ttransform** (*matrix, scale=True, roll=True*)

Transform the the bones head, tail, roll and envelope (when the matrix has a scale component).

**Parameters**

- **matrix** (*mathutils.Matrix*) – 3x3 or 4x4 transformation matrix.
- **scale** (*bool*) – Scale the bone envelope by the matrix.
- **roll** (*bool*) – Correct the roll to point in the same relative direction to the head and tail.

**translate** (*vec*)

Utility function to add *vec* to the head and tail of this bone

**Inherited Properties**

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Armature.edit_bones`
- `ArmatureEditBones.active`
- `ArmatureEditBones.new`
- `ArmatureEditBones.remove`
- `EditBone.parent`

### 2.4.195 EditObjectActuator(Actuator)

base classes — `bpy_struct`, `Actuator`

**class** `bpy.types.EditObjectActuator` (*Actuator*)  
Actuator used to edit objects

**angular\_velocity**

Angular velocity upon creation

**Type** float array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`

**dynamic\_operation**

**Type** enum in `['RESTOREDYN', 'SUSPENDDYN', 'ENABLERIGIDBODY', 'DISABLE-RIGIDBODY', 'SETMASS']`, default `'RESTOREDYN'`

**linear\_velocity**

Velocity upon creation

**Type** float array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`

**mass**

The mass of the object

**Type** float in `[-inf, inf]`, default `0.0`

**mesh**

Replace the existing, when left blank `'Phys'` will remake the existing physics mesh

**Type** `Mesh`

**mode**

The mode of the actuator

**Type** enum in ['ADDOBJECT', 'ENDOBJECT', 'REPLACEMESH', 'TRACKTO', 'DYNAMICS'], default 'ADDOBJECT'

**object**

Add this Object and all its children (cant be on an visible layer)

**Type** `Object`

**time**

Duration the new Object lives or the track takes

**Type** int in [-inf, inf], default 0

**track\_object**

Track to this Object

**Type** `Object`

**use\_3d\_tracking**

Enable 3D tracking

**Type** boolean, default False

**use\_local\_angular\_velocity**

Apply the rotation locally

**Type** boolean, default False

**use\_local\_linear\_velocity**

Apply the transformation locally

**Type** boolean, default False

**use\_replace\_display\_mesh**

Replace the display mesh

**Type** boolean, default False

**use\_replace\_physics\_mesh**

Replace the physics mesh (triangle bounds only - compound shapes not supported)

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `Actuator.name`
- `Actuator.show_expanded`
- `Actuator.pin`
- `Actuator.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`

- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Actuator.link`
- `Actuator.unlink`

### 2.4.196 EffectSequence(Sequence)

base classes — `bpy_struct`, `Sequence`

subclasses — `GlowSequence`, `PluginSequence`, `TransformSequence`, `WipeSequence`, `ColorSequence`, `SpeedControlSequence`

**class** `bpy.types.EffectSequence` (*Sequence*)

Sequence strip applying an effect on the images created by other strips

**color\_balance**

**Type** `SequenceColorBalance`, (readonly)

**color\_multiply**

**Type** float in [0, 20], default 0.0

**color\_saturation**

**Type** float in [0, 20], default 0.0

**crop**

**Type** `SequenceCrop`, (readonly)

**proxy**

**Type** `SequenceProxy`, (readonly)

**strobe**

Only display every nth frame

**Type** float in [1, 30], default 0.0

**transform**

**Type** `SequenceTransform`, (readonly)

**use\_color\_balance**

(3-Way color correction) on input

**Type** boolean, default False

**use\_crop**

Crop image before processing

**Type** boolean, default False

**use\_deinterlace**

For video movies to remove fields

**Type** boolean, default False

**use\_flip\_x**

Flip on the X axis

**Type** boolean, default False

**use\_flip\_y**

Flip on the Y axis

**Type** boolean, default False

**use\_float**

Convert input to float data

**Type** boolean, default False

**use\_premultiply**

Convert RGB from key alpha to premultiplied alpha

**Type** boolean, default False

**use\_proxy**

Use a preview proxy and/or timecode index for this strip

**Type** boolean, default False

**use\_proxy\_custom\_directory**

Use a custom directory to store data

**Type** boolean, default False

**use\_proxy\_custom\_file**

Use a custom file to read proxy data from

**Type** boolean, default False

**use\_reverse\_frames**

Reverse frame order

**Type** boolean, default False

**use\_translation**

Translate image before processing

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `Sequence.name`
- `Sequence.blend_type`
- `Sequence.blend_alpha`
- `Sequence.channel`
- `Sequence.waveform`
- `Sequence.effect_fader`
- `Sequence.frame_final_end`
- `Sequence.frame_offset_end`

- `Sequence.frame_still_end`
- `Sequence.input_1`
- `Sequence.input_2`
- `Sequence.input_3`
- `Sequence.select_left_handle`
- `Sequence.frame_final_duration`
- `Sequence.frame_duration`
- `Sequence.lock`
- `Sequence.mute`
- `Sequence.select_right_handle`
- `Sequence.select`
- `Sequence.speed_factor`
- `Sequence.frame_start`
- `Sequence.frame_final_start`
- `Sequence.frame_offset_start`
- `Sequence.frame_still_start`
- `Sequence.type`
- `Sequence.use_default_fade`
- `Sequence.input_count`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Sequence.getStripElem`
- `Sequence.swap`

### 2.4.197 EffectorWeights(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.EffectorWeights` (*bpy\_struct*)  
Effector weights for physics simulation

**all**

All effector's weight

**Type** float in [-200, 200], default 0.0

**apply\_to\_hair\_growing**

Use force fields when growing hair

**Type** boolean, default False

**boid**

Boid effector weight

**Type** float in [-200, 200], default 0.0

**charge**

Charge effector weight

**Type** float in [-200, 200], default 0.0

**curve\_guide**

Curve guide effector weight

**Type** float in [-200, 200], default 0.0

**drag**

Drag effector weight

**Type** float in [-200, 200], default 0.0

**force**

Force effector weight

**Type** float in [-200, 200], default 0.0

**gravity**

Global gravity weight

**Type** float in [-200, 200], default 0.0

**group**

Limit effectors to this Group

**Type** [Group](#)

**harmonic**

Harmonic effector weight

**Type** float in [-200, 200], default 0.0

**lennardjones**

Lennard-Jones effector weight

**Type** float in [-200, 200], default 0.0

**magnetic**

Magnetic effector weight

**Type** float in [-200, 200], default 0.0

**texture**

Texture effector weight

**Type** float in [-200, 200], default 0.0

**turbulence**

Turbulence effector weight

**Type** float in [-200, 200], default 0.0

**vortex**

Vortex effector weight

**Type** float in [-200, 200], default 0.0

**wind**

Wind effector weight

**Type** float in [-200, 200], default 0.0

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `ClothSettings.effector_weights`
- `ParticleSettings.effector_weights`
- `SmokeDomainSettings.effector_weights`
- `SoftBodySettings.effector_weights`

## 2.4.198 EnumProperty(Property)

base classes — `bpy_struct`, `Property`

**class** `bpy.types.EnumProperty` (*Property*)

RNA enumeration property definition, to choose from a number of predefined options

**default**

Default value for this enum

**Type** enum in ['DUMMY'], default 'DUMMY', (readonly)

**default\_flag**

Default value for this enum

**Type** enum set in {'DUMMY'}, default set(), (readonly)



**enum\_items**

Possible values for the property

**Type** bpy\_prop\_collection of EnumPropertyItem, (readonly)**Inherited Properties**

- bpy\_struct.id\_data
- Property.name
- Property.is\_animatable
- Property.srna
- Property.description
- Property.is\_enum\_flag
- Property.is\_hidden
- Property.identifier
- Property.is\_never\_none
- Property.is\_readonly
- Property.is\_registered
- Property.is\_registered\_optional
- Property.is\_required
- Property.is\_output
- Property.is\_runtime
- Property.is\_skip\_save
- Property.subtype
- Property.type
- Property.unit

**Inherited Functions**

- bpy\_struct.as\_pointer
- bpy\_struct.callback\_add
- bpy\_struct.callback\_remove
- bpy\_struct.driver\_add
- bpy\_struct.driver\_remove
- bpy\_struct.get
- bpy\_struct.is\_property\_hidden
- bpy\_struct.is\_property\_set
- bpy\_struct.items
- bpy\_struct.keyframe\_delete
- bpy\_struct.keyframe\_insert
- bpy\_struct.keys
- bpy\_struct.path\_from\_id
- bpy\_struct.path\_resolve
- bpy\_struct.type\_recast
- bpy\_struct.values

**2.4.199 EnumPropertyItem(bpy\_struct)**

base class — bpy\_struct

**class** bpy.types.**EnumPropertyItem** (*bpy\_struct*)

Definition of a choice in an RNA enum property

**description**

Description of the item's purpose

**Type** string, default "", (readonly)

**identifier**

Unique name used in the code and scripting

**Type** string, default "", (readonly)

**name**

Human readable name

**Type** string, default "", (readonly)

**value**

Value of the item

**Type** int in [0, inf], default 0, (readonly)

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `EnumProperty.enum_items`

**2.4.200 EnvironmentMap(`bpy_struct`)**

base class — `bpy_struct`

**class** `bpy.types.EnvironmentMap` (*bpy\_struct*)

Environment map created by the renderer and cached for subsequent renders

**clip\_end**

Objects further than this are not visible to map

**Type** float in [0.01, inf], default 0.0

**clip\_start**

Objects nearer than this are not visible to map

**Type** float in [0.001, inf], default 0.0

**depth**

Number of times a map will be rendered recursively (mirror effects)

**Type** int in [0, 5], default 0

**is\_valid**

True if this map is ready for use, False if it needs rendering

**Type** boolean, default False, (readonly)

**layers\_ignore**

Hide objects on these layers when generating the Environment Map

**Type** boolean array of 20 items, default (False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False)

**mapping**

- CUBE Cube, Use environment map with six cube sides.
- PLANE Plane, Only one side is rendered, with Z axis pointing in direction of image.

**Type** enum in ['CUBE', 'PLANE'], default 'CUBE'

**resolution**

Pixel resolution of the rendered environment map

**Type** int in [50, 4096], default 0

**source**

- STATIC Static, Calculate environment map only once.
- ANIMATED Animated, Calculate environment map at each rendering.
- IMAGE\_FILE Image File, Load a saved environment map image from disk.

**Type** enum in ['STATIC', 'ANIMATED', 'IMAGE\_FILE'], default 'STATIC'

**viewpoint\_object**

Object to use as the environment map's viewpoint location

**Type** Object

**zoom**

**Type** float in [0.1, 5], default 0.0

**clear()**

Discard the environment map and free it from memory

**save** (*filepath*, *scene=None*, *layout=(0.0, 0.0, 1.0, 0.0, 2.0, 0.0, 0.0, 1.0, 1.0, 1.0, 2.0, 1.0)*)

Save the environment map to disc using the scene render settings

**Parameters**

- **filepath** (*string*) – File path, Location of the output file
- **scene** (*Scene*, (optional)) – Overrides the scene from which image parameters are taken
- **layout** (*float array of 12 items in [-inf, inf]*, (optional)) – File layout, Flat array describing the X,Y position of each cube face in the output image, where 1 is the size of a face - order is [+Z -Z +Y -X -Y +X] (use -1 to skip a face)

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `EnvironmentMapTexture.environment_map`

## 2.4.201 EnvironmentMapTexture(Texture)

base classes — `bpy_struct`, `ID`, `Texture`

**class** `bpy.types.EnvironmentMapTexture` (*Texture*)  
Environment map texture

**environment\_map**  
Gets the environment map associated with this texture

**Type** `EnvironmentMap`, (readonly)

**filter\_eccentricity**  
Maximum eccentricity. Higher gives less blur at distant/oblique angles, but is also slower

**Type** `int` in [1, 256], default 0

**filter\_probes**  
Maximum number of samples. Higher gives less blur at distant/oblique angles, but is also slower

**Type** int in [1, 256], default 0

**filter\_size**

Multiply the filter size used by MIP Map and Interpolation

**Type** float in [0.1, 50], default 0.0

**filter\_type**

Texture filter to use for sampling image

**Type** enum in ['BOX', 'EWA', 'FELINE', 'AREA'], default 'BOX'

**image**

Source image file to read the environment map from

**Type** Image

**image\_user**

Parameters defining which layer, pass and frame of the image is displayed

**Type** ImageUser, (readonly)

**use\_filter\_size\_min**

Use Filter Size as a minimal filter value in pixels

**Type** boolean, default False

**use\_mipmap**

Uses auto-generated MIP maps for the image

**Type** boolean, default False

**use\_mipmap\_gauss**

Use Gauss filter to sample down MIP maps

**Type** boolean, default False

**users\_material**

Materials that use this texture (readonly)

**users\_object\_modifier**

Object modifiers that use this texture (readonly)

## Inherited Properties

- bpy\_struct.id\_data
- ID.name
- ID.use\_fake\_user
- ID.is\_updated
- ID.is\_updated\_data
- ID.library
- ID.tag
- ID.users
- Texture.animation\_data
- Texture.intensity
- Texture.color\_ramp
- Texture.contrast
- Texture.factor\_blue
- Texture.factor\_green
- Texture.factor\_red
- Texture.node\_tree

- `Texture.saturation`
- `Texture.use_preview_alpha`
- `Texture.type`
- `Texture.use_color_ramp`
- `Texture.use_nodes`
- `Texture.users_material`
- `Texture.users_object_modifier`
- `Texture.users_material`
- `Texture.users_object_modifier`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`
- `Texture.evaluate`

### 2.4.202 Event(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.Event` (*bpy\_struct*)

Window Manager Event

**alt**

True when the Alt/Option key is held

**Type** boolean, default False, (readonly)

**ascii**

Single ASCII character for this event

**Type** string, default "", (readonly)

**ctrl**

True when the Ctrl key is held

**Type** boolean, default False, (readonly)

**mouse\_prev\_x**

The window relative vertical location of the mouse

**Type** int in [-inf, inf], default 0, (readonly)

**mouse\_prev\_y**

The window relative horizontal location of the mouse

**Type** int in [-inf, inf], default 0, (readonly)

**mouse\_region\_x**

The region relative vertical location of the mouse

**Type** int in [-inf, inf], default 0, (readonly)

**mouse\_region\_y**

The region relative horizontal location of the mouse

**Type** int in [-inf, inf], default 0, (readonly)

**mouse\_x**

The window relative vertical location of the mouse

**Type** int in [-inf, inf], default 0, (readonly)

**mouse\_y**

The window relative horizontal location of the mouse

**Type** int in [-inf, inf], default 0, (readonly)

**oskey**

True when the Cmd key is held

**Type** boolean, default False, (readonly)

**shift**

True when the Shift key is held

**Type** boolean, default False, (readonly)

**type**

**Type** enum in ['NONE', 'LEFTMOUSE', 'MIDDLEMOUSE', 'RIGHTMOUSE', 'BUTTON4MOUSE', 'BUTTON5MOUSE', 'ACTIONMOUSE', 'SELECTMOUSE', 'MOUSEMOVE', 'INBETWEEN\_MOUSEMOVE', 'TRACKPADPAN', 'TRACKPADZOOM', 'MOUSEROTATE', 'WHEELUPMOUSE', 'WHEELDOWNMOUSE', 'WHEELINMOUSE', 'WHEELOUTMOUSE', 'EVT\_TWEAK\_L', 'EVT\_TWEAK\_M', 'EVT\_TWEAK\_R', 'EVT\_TWEAK\_A', 'EVT\_TWEAK\_S', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z', 'ZERO', 'ONE', 'TWO', 'THREE', 'FOUR', 'FIVE', 'SIX', 'SEVEN', 'EIGHT', 'NINE', 'LEFT\_CTRL', 'LEFT\_ALT', 'LEFT\_SHIFT', 'RIGHT\_ALT', 'RIGHT\_CTRL', 'RIGHT\_SHIFT', 'OSKEY', 'GRLESS', 'ESC', 'TAB', 'RET', 'SPACE', 'LINE\_FEED', 'BACK\_SPACE', 'DEL', 'SEMI\_COLON', 'PERIOD', 'COMMA', 'QUOTE', 'ACCENT\_GRAVE', 'MINUS', 'SLASH', 'BACK\_SLASH', 'EQUAL', 'LEFT\_BRACKET', 'RIGHT\_BRACKET', 'LEFT\_ARROW', 'DOWN\_ARROW', 'RIGHT\_ARROW', 'UP\_ARROW', 'NUMPAD\_2', 'NUMPAD\_4', 'NUMPAD\_6', 'NUMPAD\_8', 'NUMPAD\_1', 'NUMPAD\_3', 'NUMPAD\_5', 'NUMPAD\_7', 'NUMPAD\_9', 'NUMPAD\_PERIOD', 'NUMPAD\_SLASH', 'NUMPAD\_ASTERIX', 'NUMPAD\_0', 'NUMPAD\_MINUS', 'NUMPAD\_ENTER', 'NUMPAD\_PLUS', 'F1', 'F2', 'F3', 'F4', 'F5', 'F6', 'F7', 'F8', 'F9', 'F10', 'F11', 'F12', 'F13', 'F14', 'F15', 'F16', 'F17', 'F18', 'F19', 'PAUSE', 'INSERT', 'HOME',

'PAGE\_UP', 'PAGE\_DOWN', 'END', 'MEDIA\_PLAY', 'MEDIA\_STOP', 'MEDIA\_FIRST', 'MEDIA\_LAST', 'WINDOW\_DEACTIVATE', 'TIMER', 'TIMER0', 'TIMER1', 'TIMER2', 'NDOF\_BUTTON\_MENU', 'NDOF\_BUTTON\_FIT', 'NDOF\_BUTTON\_TOP', 'NDOF\_BUTTON\_BOTTOM', 'NDOF\_BUTTON\_LEFT', 'NDOF\_BUTTON\_RIGHT', 'NDOF\_BUTTON\_FRONT', 'NDOF\_BUTTON\_BACK', 'NDOF\_BUTTON\_ISO1', 'NDOF\_BUTTON\_ISO2', 'NDOF\_BUTTON\_ROLL\_CW', 'NDOF\_BUTTON\_ROLL\_CCW', 'NDOF\_BUTTON\_SPIN\_CW', 'NDOF\_BUTTON\_SPIN\_CCW', 'NDOF\_BUTTON\_TILT\_CW', 'NDOF\_BUTTON\_TILT\_CCW', 'NDOF\_BUTTON\_ROTATE', 'NDOF\_BUTTON\_PANZOOM', 'NDOF\_BUTTON\_DOMINANT', 'NDOF\_BUTTON\_PLUS', 'NDOF\_BUTTON\_MINUS', 'NDOF\_BUTTON\_1', 'NDOF\_BUTTON\_2', 'NDOF\_BUTTON\_3', 'NDOF\_BUTTON\_4', 'NDOF\_BUTTON\_5', 'NDOF\_BUTTON\_6', 'NDOF\_BUTTON\_7', 'NDOF\_BUTTON\_8', 'NDOF\_BUTTON\_9', 'NDOF\_BUTTON\_10'], default 'NONE', (readonly)

**unicode**

Single unicode character for this event

**Type** string, default "", (readonly)

**value**

The type of event, only applies to some

**Type** enum in ['ANY', 'NOTHING', 'PRESS', 'RELEASE', 'CLICK', 'DOUBLE\_CLICK'], default 'NOTHING', (readonly)

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Operator.invoke`



- `Operator.modal`
- `WindowManager.invoke_confirm`
- `WindowManager.invoke_props_popup`

### 2.4.203 ExplodeModifier(Modifier)

base classes — `bpy_struct`, `Modifier`

**class** `bpy.types.ExplodeModifier` (*Modifier*)  
Explosion effect modifier based on a particle system

**particle\_uv**

UV Layer to change with particle age

**Type** string, default ""

**protect**

Clean vertex group edges

**Type** float in [0, 1], default 0.0

**show\_alive**

Show mesh when particles are alive

**Type** boolean, default False

**show\_dead**

Show mesh when particles are dead

**Type** boolean, default False

**show\_unborn**

Show mesh when particles are unborn

**Type** boolean, default False

**use\_edge\_cut**

Cut face edges for nicer shrapnel

**Type** boolean, default False

**use\_size**

Use particle size for the shrapnel

**Type** boolean, default False

**vertex\_group**

**Type** string, default ""

#### Inherited Properties

- `bpy_struct.id_data`
- `Modifier.name`
- `Modifier.use_apply_on_spline`
- `Modifier.show_in_editmode`
- `Modifier.show_expanded`
- `Modifier.show_on_cage`
- `Modifier.show_viewport`
- `Modifier.show_render`
- `Modifier.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.204 ExpressionController(Controller)

base classes — `bpy_struct`, `Controller`

**class** `bpy.types.ExpressionController` (*Controller*)  
Controller passing on events based on the evaluation of an expression

**expression**

**Type** string, default ""

### Inherited Properties

- `bpy_struct.id_data`
- `Controller.name`
- `Controller.states`
- `Controller.show_expanded`
- `Controller.use_priority`
- `Controller.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`

- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Controller.link`
- `Controller.unlink`

### 2.4.205 FCurve(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.FCurve` (*bpy\_struct*)

F-Curve defining values of a period of time

**array\_index**

Index to the specific property affected by F-Curve if applicable

**Type** `int` in `[-inf, inf]`, default `0`

**color**

Color of the F-Curve in the Graph Editor

**Type** `float` array of 3 items in `[0, 1]`, default `(0.0, 0.0, 0.0)`

**color\_mode**

Method used to determine color of F-Curve in Graph Editor

**Type** `enum` in `['AUTO_RAINBOW', 'AUTO_RGB', 'CUSTOM']`, default `'AUTO_RAINBOW'`

**data\_path**

RNA Path to property affected by F-Curve

**Type** `string`, default `""`

**driver**

Channel Driver (only set for Driver F-Curves)

**Type** `Driver`, (readonly)

**extrapolation**

**Type** `enum` in `['CONSTANT', 'LINEAR']`, default `'CONSTANT'`

**group**

Action Group that this F-Curve belongs to

**Type** `ActionGroup`

**hide**

F-Curve and its keyframes are hidden in the Graph Editor graphs

**Type** `boolean`, default `False`

**is\_valid**

False when F-Curve could not be evaluated in past, so should be skipped when evaluating

**Type** `boolean`, default `False`

**keyframe\_points**

User-editable keyframes

**Type** `FCurveKeyframePoints` `bpy_prop_collection` of `Keyframe`, (readonly)

**lock**

F-Curve's settings cannot be edited

**Type** boolean, default False

**modifiers**

Modifiers affecting the shape of the F-Curve

**Type** `FCurveModifiers bpy_prop_collection of FModifier`, (readonly)

**mute**

F-Curve is not evaluated

**Type** boolean, default False

**sampled\_points**

Sampled animation data

**Type** `bpy_prop_collection of FCurveSample`, (readonly)

**select**

F-Curve is selected for editing

**Type** boolean, default False

**evaluate** (*frame*)

Evaluate F-Curve

**Parameters** **frame** (*float in [-inf, inf]*) – Frame, Evaluate F-Curve at given frame

**Returns** Position, F-Curve position

**Return type** float in [-inf, inf]

**range** ()

Get the time extents for F-Curve

**Returns** Range, Min/Max values

**Return type** float array of 2 items in [-inf, inf]

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`

- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Action.fcurves`
- `ActionFCurves.new`
- `ActionFCurves.remove`
- `ActionGroup.channels`
- `AnimData.drivers`
- `AnimDataDrivers.from_existing`
- `AnimDataDrivers.from_existing`
- `NlaStrip.fcurves`

### 2.4.206 FCurveKeyframePoints(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.FCurveKeyframePoints` (*bpy\_struct*)

Collection of keyframe points

**insert** (*frame, value, options=set()*)

Add a keyframe point to a F-Curve

#### Parameters

- **frame** (*float in [-inf, inf]*) – X Value of this keyframe point
- **value** (*float in [-inf, inf]*) – Y Value of this keyframe point
- **options** (*enum set in {'REPLACE', 'NEEDED', 'FAST'}, (optional)*) – Keyframe options
  - REPLACE Replace, Don't add any new keyframes, but just replace existing ones.
  - NEEDED Needed, Only adds keyframes that are needed.
  - FAST Fast, Fast keyframe insertion to avoid recalculating the curve each time.

**Returns** Newly created keyframe

**Return type** `Keyframe`

**add** (*count=1*)

Add a keyframe point to a F-Curve

**Parameters** **count** (*int in [1, inf], (optional)*) – Number, Number of points to add to the spline

**remove** (*keyframe, fast=False*)

Remove keyframe from an F-Curve

#### Parameters

- **keyframe** (`Keyframe`, (never `None`)) – Keyframe to remove
- **fast** (*boolean, (optional)*) – Fast, Fast keyframe removal to avoid recalculating the curve each time

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `FCurve.keyframe_points`

## 2.4.207 FCurveModifiers(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.FCurveModifiers` (*bpy\_struct*)

Collection of F-Curve Modifiers

**active**

Active F-Curve Modifier

**Type** `FModifier`

**new** (*type*)

Add a constraint to this object

**Parameters** `type` (*enum in ['NULL', 'GENERATOR', 'FNGENERATOR', 'ENVELOPE', 'CYCLES', 'NOISE', 'FILTER', 'LIMITS', 'STEPPED']*) – Constraint type to add

**Returns** New `fmodifier`

**Return type** `FModifier`

**remove** (*modifier*)

Remove a modifier from this F-Curve

**Parameters** `modifier` (`FModifier`, (never `None`)) – Removed modifier

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `FCurve.modifiers`

## 2.4.208 FCurveSample(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.FCurveSample` (*bpy\_struct*)

Sample point for F-Curve

**co**

Point coordinates

**Type** float array of 2 items in  $[-\text{inf}, \text{inf}]$ , default (0.0, 0.0)

**select**

Selection status

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`

- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `FCurve.sampled_points`

### 2.4.209 FModifier(`bpy_struct`)

base class — `bpy_struct`

subclasses — `FModifierCycles`, `FModifierPython`, `FModifierFunctionGenerator`, `FModifierLimits`, `FModifierEnvelope`, `FModifierNoise`, `FModifierStepped`, `FModifierGenerator`

**class** `bpy.types.FModifier` (*`bpy_struct`*)

Modifier for values of F-Curve

**active**

F-Curve Modifier is the one being edited

**Type** boolean, default False

**blend\_in**

Number of frames from start frame for influence to take effect

**Type** float in [-inf, inf], default 0.0

**blend\_out**

Number of frames from start frame for influence to fade out

**Type** float in [-inf, inf], default 0.0

**frame\_end**

Frame that modifier's influence ends (if Restrict Frame Range is in use)

**Type** float in [-inf, inf], default 0.0

**frame\_start**

Frame that modifier's influence starts (if Restrict Frame Range is in use)

**Type** float in [-inf, inf], default 0.0

**influence**

Amount of influence F-Curve Modifier will have when not fading in/out

**Type** float in [0, 1], default 1.0



**is\_valid**

F-Curve Modifier has invalid settings and will not be evaluated

**Type** boolean, default False, (readonly)

**mute**

F-Curve Modifier will not be evaluated

**Type** boolean, default False

**show\_expanded**

F-Curve Modifier's panel is expanded in UI

**Type** boolean, default False

**type**

F-Curve Modifier Type

**Type** enum in ['NULL', 'GENERATOR', 'FNGENERATOR', 'ENVELOPE', 'CYCLES', 'NOISE', 'FILTER', 'LIMITS', 'STEPPED'], default 'NULL', (readonly)

**use\_influence**

F-Curve Modifier's effects will be tempered by a default factor

**Type** boolean, default False

**use\_restricted\_range**

F-Curve Modifier is only applied for the specified frame range to help mask off effects in order to chain them

**Type** boolean, default False

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `FCurve.modifiers`
- `FCurveModifiers.active`
- `FCurveModifiers.new`
- `FCurveModifiers.remove`
- `NlaStrip.modifiers`

### 2.4.210 FModifierCycles(FModifier)

base classes — `bpy_struct`, `FModifier`

class `bpy.types.FModifierCycles` (*FModifier*)

Repeats the values of the modified F-Curve

#### **`cycles_after`**

Maximum number of cycles to allow after last keyframe. (0 = infinite)

**Type** int in [-32768, 32767], default 0

#### **`cycles_before`**

Maximum number of cycles to allow before first keyframe. (0 = infinite)

**Type** int in [-32768, 32767], default 0

#### **`mode_after`**

Cycling mode to use after last keyframe

- `NONE` No Cycles, Don't do anything.
- `REPEAT` Repeat Motion, Repeat keyframe range as-is.
- `REPEAT_OFFSET` Repeat with Offset, Repeat keyframe range, but with offset based on gradient between values.
- `MIRROR` Repeat Mirrored, Alternate between forward and reverse playback of keyframe range.

**Type** enum in ['NONE', 'REPEAT', 'REPEAT\_OFFSET', 'MIRROR'], default 'NONE'

#### **`mode_before`**

Cycling mode to use before first keyframe

- `NONE` No Cycles, Don't do anything.
- `REPEAT` Repeat Motion, Repeat keyframe range as-is.
- `REPEAT_OFFSET` Repeat with Offset, Repeat keyframe range, but with offset based on gradient between values.
- `MIRROR` Repeat Mirrored, Alternate between forward and reverse playback of keyframe range.

**Type** enum in ['NONE', 'REPEAT', 'REPEAT\_OFFSET', 'MIRROR'], default 'NONE'

## Inherited Properties

- `bpy_struct.id_data`
- `FModifier.active`
- `FModifier.blend_in`
- `FModifier.blend_out`

- `FModifier.is_valid`
- `FModifier.frame_end`
- `FModifier.show_expanded`
- `FModifier.influence`
- `FModifier.mute`
- `FModifier.use_restricted_range`
- `FModifier.frame_start`
- `FModifier.type`
- `FModifier.use_influence`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.211 FModifierEnvelope(FModifier)

base classes — `bpy_struct`, `FModifier`

**class** `bpy.types.FModifierEnvelope` (*FModifier*)

Scales the values of the modified F-Curve

### **control\_points**

Control points defining the shape of the envelope

**Type** `bpy_prop_collection` of `FModifierEnvelopeControlPoint`, (readonly)

### **default\_max**

Upper distance from Reference Value for 1:1 default influence

**Type** float in `[-inf, inf]`, default 0.0

### **default\_min**

Lower distance from Reference Value for 1:1 default influence

**Type** float in `[-inf, inf]`, default 0.0

### **reference\_value**

Value that envelope's influence is centered around / based on

**Type** float in `[-inf, inf]`, default 0.0

### Inherited Properties

- `bpy_struct.id_data`
- `FModifier.active`
- `FModifier.blend_in`
- `FModifier.blend_out`
- `FModifier.is_valid`
- `FModifier.frame_end`
- `FModifier.show_expanded`
- `FModifier.influence`
- `FModifier.mute`
- `FModifier.use_restricted_range`
- `FModifier.frame_start`
- `FModifier.type`
- `FModifier.use_influence`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.212 FModifierEnvelopeControlPoint(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.FModifierEnvelopeControlPoint` (*bpy\_struct*)  
Control point for envelope F-Modifier

**frame**

Frame this control-point occurs on

**Type** float in [-inf, inf], default 0.0

**max**

Upper bound of envelope at this control-point

**Type** float in [-inf, inf], default 0.0

**min**

Lower bound of envelope at this control-point

**Type** float in [-inf, inf], default 0.0

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `FModifierEnvelope.control_points`

## 2.4.213 FModifierFunctionGenerator(FModifier)

base classes — `bpy_struct`, `FModifier`

**class** `bpy.types.FModifierFunctionGenerator` (*FModifier*)  
Generates values using a Built-In Function

#### **amplitude**

Scale factor determining the maximum/minimum values

**Type** float in [-inf, inf], default 0.0

#### **function\_type**

Type of built-in function to use

- `SIN` Sine.
- `COS` Cosine.
- `TAN` Tangent.
- `SQRT` Square Root.
- `LN` Natural Logarithm.
- `SINC` Normalised Sine,  $\sin(x) / x$ .

**Type** enum in ['SIN', 'COS', 'TAN', 'SQRT', 'LN', 'SINC'], default 'SIN'

**phase\_multiplier**

Scale factor determining the 'speed' of the function

**Type** float in [-inf, inf], default 0.0

**phase\_offset**

Constant factor to offset time by for function

**Type** float in [-inf, inf], default 0.0

**use\_additive**

Values generated by this modifier are applied on top of the existing values instead of overwriting them

**Type** boolean, default False

**value\_offset**

Constant factor to offset values by

**Type** float in [-inf, inf], default 0.0

### Inherited Properties

- `bpy_struct.id_data`
- `FModifier.active`
- `FModifier.blend_in`
- `FModifier.blend_out`
- `FModifier.is_valid`
- `FModifier.frame_end`
- `FModifier.show_expanded`
- `FModifier.influence`
- `FModifier.mute`
- `FModifier.use_restricted_range`
- `FModifier.frame_start`
- `FModifier.type`
- `FModifier.use_influence`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.214 FModifierGenerator(FModifier)

base classes — `bpy_struct`, `FModifier`

**class** `bpy.types.FModifierGenerator` (*FModifier*)

Deterministically generates values for the modified F-Curve

**coefficients**

Coefficients for 'x' (starting from lowest power of  $x^0$ )

**Type** float array of 32 items in `[-inf, inf]`, default `(0.0, 0.0)`

**mode**

Type of generator to use

**Type** enum in `['POLYNOMIAL', 'POLYNOMIAL_FACTORISED']`, default `'POLYNOMIAL'`

**poly\_order**

The highest power of 'x' for this polynomial. (number of coefficients - 1)

**Type** int in `[-inf, inf]`, default 0

**use\_additive**

Values generated by this modifier are applied on top of the existing values instead of overwriting them

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `FModifier.active`
- `FModifier.blend_in`
- `FModifier.blend_out`
- `FModifier.is_valid`
- `FModifier.frame_end`
- `FModifier.show_expanded`
- `FModifier.influence`
- `FModifier.mute`
- `FModifier.use_restricted_range`
- `FModifier.frame_start`
- `FModifier.type`
- `FModifier.use_influence`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`

- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.215 FModifierLimits(FModifier)

base classes — `bpy_struct`, `FModifier`

**class** `bpy.types.FModifierLimits` (*FModifier*)  
Limits the time/value ranges of the modified F-Curve

**max\_x**  
Highest X value to allow  
**Type** float in [-inf, inf], default 0.0

**max\_y**  
Highest Y value to allow  
**Type** float in [-inf, inf], default 0.0

**min\_x**  
Lowest X value to allow  
**Type** float in [-inf, inf], default 0.0

**min\_y**  
Lowest Y value to allow  
**Type** float in [-inf, inf], default 0.0

**use\_max\_x**  
Use the maximum X value  
**Type** boolean, default False

**use\_max\_y**  
Use the maximum Y value  
**Type** boolean, default False

**use\_min\_x**  
Use the minimum X value  
**Type** boolean, default False

**use\_min\_y**  
Use the minimum Y value  
**Type** boolean, default False

#### Inherited Properties

- `bpy_struct.id_data`
- `FModifier.active`
- `FModifier.blend_in`
- `FModifier.blend_out`



- `FModifier.is_valid`
- `FModifier.frame_end`
- `FModifier.show_expanded`
- `FModifier.influence`
- `FModifier.mute`
- `FModifier.use_restricted_range`
- `FModifier.frame_start`
- `FModifier.type`
- `FModifier.use_influence`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.216 FModifierNoise(FModifier)

base classes — `bpy_struct`, `FModifier`

**class** `bpy.types.FModifierNoise` (*FModifier*)

Gives randomness to the modified F-Curve

### **blend\_type**

Method of modifying the existing F-Curve

**Type** enum in ['REPLACE', 'ADD', 'SUBTRACT', 'MULTIPLY'], default 'REPLACE'

### **depth**

Amount of fine level detail present in the noise

**Type** int in [0, 32767], default 0

### **phase**

A random seed for the noise effect

**Type** float in [-inf, inf], default 0.0

### **scale**

Scaling (in time) of the noise

**Type** float in [-inf, inf], default 0.0

**strength**

Amplitude of the noise - the amount that it modifies the underlying curve

**Type** float in [-inf, inf], default 0.0

**Inherited Properties**

- `bpy_struct.id_data`
- `FModifier.active`
- `FModifier.blend_in`
- `FModifier.blend_out`
- `FModifier.is_valid`
- `FModifier.frame_end`
- `FModifier.show_expanded`
- `FModifier.influence`
- `FModifier.mute`
- `FModifier.use_restricted_range`
- `FModifier.frame_start`
- `FModifier.type`
- `FModifier.use_influence`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**2.4.217 FModifierPython(FModifier)**

base classes — `bpy_struct`, `FModifier`

**class** `bpy.types.FModifierPython` (*FModifier*)  
Performs user-defined operation on the modified F-Curve

**Inherited Properties**

- `bpy_struct.id_data`
- `FModifier.active`

- `FModifier.blend_in`
- `FModifier.blend_out`
- `FModifier.is_valid`
- `FModifier.frame_end`
- `FModifier.show_expanded`
- `FModifier.influence`
- `FModifier.mute`
- `FModifier.use_restricted_range`
- `FModifier.frame_start`
- `FModifier.type`
- `FModifier.use_influence`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.218 FModifierStepped(FModifier)

base classes — `bpy_struct`, `FModifier`

**class** `bpy.types.FModifierStepped` (*FModifier*)

Hold each interpolated value from the F-Curve for several frames without changing the timing

#### **frame\_end**

Frame that modifier's influence ends (if applicable)

**Type** float in [-inf, inf], default 0.0

#### **frame\_offset**

Reference number of frames before frames get held (use to get hold for '1-3' vs '5-7' holding patterns)

**Type** float in [-inf, inf], default 0.0

#### **frame\_start**

Frame that modifier's influence starts (if applicable)

**Type** float in [-inf, inf], default 0.0

#### **frame\_step**

Number of frames to hold each value

**Type** float in [-inf, inf], default 0.0

**use\_frame\_end**

Restrict modifier to only act before its ‘end’ frame

**Type** boolean, default False

**use\_frame\_start**

Restrict modifier to only act after its ‘start’ frame

**Type** boolean, default False

**Inherited Properties**

- `bpy_struct.id_data`
- `FModifier.active`
- `FModifier.blend_in`
- `FModifier.blend_out`
- `FModifier.is_valid`
- `FModifier.frame_end`
- `FModifier.show_expanded`
- `FModifier.influence`
- `FModifier.mute`
- `FModifier.use_restricted_range`
- `FModifier.frame_start`
- `FModifier.type`
- `FModifier.use_influence`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**2.4.219 FieldSettings(bpy\_struct)**

base class — `bpy_struct`

**class** `bpy.types.FieldSettings` (*bpy\_struct*)  
Field settings for an object in physics simulation

**apply\_to\_location**

Effect particles’ location

**Type** boolean, default False

**apply\_to\_rotation**

Effect particles' dynamic rotation

**Type** boolean, default False

**distance\_max**

Maximum distance for the field to work

**Type** float in [0, 1000], default 0.0

**distance\_min**

Minimum distance for the field's fall-off

**Type** float in [0, 1000], default 0.0

**falloff\_power**

Falloff power (real gravitational falloff = 2)

**Type** float in [0, 10], default 0.0

**falloff\_type**

**Type** enum in ['SPHERE', 'TUBE', 'CONE'], default 'SPHERE'

**flow**

Convert effector force into air flow velocity

**Type** float in [0, 10], default 0.0

**guide\_clump\_amount**

Amount of clumping

**Type** float in [-1, 1], default 0.0

**guide\_clump\_shape**

Shape of clumping

**Type** float in [-0.999, 0.999], default 0.0

**guide\_free**

Guide-free time from particle life's end

**Type** float in [0, 0.99], default 0.0

**guide\_kink\_amplitude**

The amplitude of the offset

**Type** float in [0, 10], default 0.0

**guide\_kink\_axis**

Which axis to use for offset

**Type** enum in ['X', 'Y', 'Z'], default 'X'

**guide\_kink\_frequency**

The frequency of the offset (1/total length)

**Type** float in [0, 10], default 0.0

**guide\_kink\_shape**

Adjust the offset to the beginning/end

**Type** float in [-0.999, 0.999], default 0.0

**guide\_kink\_type**

Type of periodic offset on the curve

**Type** enum in ['NONE', 'CURL', 'RADIAL', 'WAVE', 'BRAID', 'ROTATION', 'ROLL'], default 'NONE'

**guide\_minimum**

The distance from which particles are affected fully

**Type** float in [0, 1000], default 0.0

**harmonic\_damping**

Damping of the harmonic force

**Type** float in [0, 10], default 0.0

**inflow**

Inwards component of the vortex force

**Type** float in [-10, 10], default 0.0

**linear\_drag**

Drag component proportional to velocity

**Type** float in [-2, 2], default 0.0

**noise**

Amount of noise for the force strength

**Type** float in [0, 10], default 0.0

**quadratic\_drag**

Drag component proportional to the square of velocity

**Type** float in [-2, 2], default 0.0

**radial\_falloff**

Radial falloff power (real gravitational falloff = 2)

**Type** float in [0, 10], default 0.0

**radial\_max**

Maximum radial distance for the field to work

**Type** float in [0, 1000], default 0.0

**radial\_min**

Minimum radial distance for the field's fall-off

**Type** float in [0, 1000], default 0.0

**rest\_length**

Rest length of the harmonic force

**Type** float in [0, 1000], default 0.0

**seed**

Seed of the noise

**Type** int in [1, 128], default 0

**shape**

Which direction is used to calculate the effector force

**Type** enum in ['POINT', 'PLANE', 'SURFACE', 'POINTS'], default 'POINT'

- size**  
Size of the turbulence  
**Type** float in [0, 10], default 0.0
- strength**  
Strength of force field  
**Type** float in [-1000, 1000], default 0.0
- texture**  
Texture to use as force  
**Type** [Texture](#)
- texture\_mode**  
How the texture effect is calculated (RGB & Curl need a RGB texture, else Gradient will be used instead)  
**Type** enum in ['RGB', 'GRADIENT', 'CURL'], default 'RGB'
- texture\_nabla**  
Defines size of derivative offset used for calculating gradient and curl  
**Type** float in [0.0001, 1], default 0.0
- type**  
Type of field
- NONE None.
  - FORCE Force, Radial field toward the center of object.
  - WIND Wind, Constant force along the force object's local Z axis.
  - VORTEX Vortex, Spiraling force that twists the force object's local Z axis.
  - MAGNET Magnetic, Forcefield depends on the speed of the particles.
  - HARMONIC Harmonic, The source of this force field is the zero point of a harmonic oscillator.
  - CHARGE Charge, Spherical forcefield based on the charge of particles, only influences other charge force fields.
  - LENNARDJ Lennard-Jones, Forcefield based on the Lennard-Jones potential.
  - TEXTURE Texture, Forcefield based on a texture.
  - GUIDE Curve Guide, Create a force along a curve object.
  - BOID Boid.
  - TURBULENCE Turbulence, Create turbulence with a noise field.
  - DRAG Drag, Create a force that dampens motion.
- Type** enum in ['NONE', 'FORCE', 'WIND', 'VORTEX', 'MAGNET', 'HARMONIC', 'CHARGE', 'LENNARDJ', 'TEXTURE', 'GUIDE', 'BOID', 'TURBULENCE', 'DRAG'], default 'NONE'
- use\_2d\_force**  
Apply force only in 2d  
**Type** boolean, default False
- use\_absorption**  
Force gets absorbed by collision objects

**Type** boolean, default False

**use\_global\_coords**

Use effector/global coordinates for turbulence

**Type** boolean, default False

**use\_guide\_path\_add**

Based on distance/falloff it adds a portion of the entire path

**Type** boolean, default False

**use\_guide\_path\_weight**

Use curve weights to influence the particle influence along the curve

**Type** boolean, default False

**use\_max\_distance**

Use a maximum distance for the field to work

**Type** boolean, default False

**use\_min\_distance**

Use a minimum distance for the field's fall-off

**Type** boolean, default False

**use\_multiple\_springs**

Every point is effected by multiple springs

**Type** boolean, default False

**use\_object\_coords**

Use object/global coordinates for texture

**Type** boolean, default False

**use\_radial\_max**

Use a maximum radial distance for the field to work

**Type** boolean, default False

**use\_radial\_min**

Use a minimum radial distance for the field's fall-off

**Type** boolean, default False

**use\_root\_coords**

Texture coordinates from root particle locations

**Type** boolean, default False

**z\_direction**

Effect in full or only positive/negative Z direction

**Type** enum in ['BOTH', 'POSITIVE', 'NEGATIVE'], default 'BOTH'

### Inherited Properties

- `bpy_struct.id_data`



## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Object.field`
- `ParticleSettings.force_field_1`
- `ParticleSettings.force_field_2`

## 2.4.220 FileSelectParams(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.FileSelectParams` (*bpy\_struct*)

File Select Parameters

### **directory**

Directory displayed in the file browser

**Type** string, default ""

### **display\_type**

Display mode for the file list

- `FILE_SHORTDISPLAY` Short List, Display files as short list.
- `FILE_LONGDISPLAY` Long List, Display files as a detailed list.
- `FILE_IMGDISPLAY` Thumbnails, Display files as thumbnails.

**Type** enum in [`'FILE_SHORTDISPLAY'`, `'FILE_LONGDISPLAY'`, `'FILE_IMGDISPLAY'`], default `'FILE_SHORTDISPLAY'`

### **filename**

Active file in the file browser

**Type** string, default ""

### **filter\_glob**

**Type** string, default ""

**show\_hidden**

Show hidden dot files

**Type** boolean, default False

**sort\_method**

- **FILE\_SORT\_ALPHA** Sort alphabetically, Sort the file list alphabetically.
- **FILE\_SORT\_EXTENSION** Sort by extension, Sort the file list by extension.
- **FILE\_SORT\_TIME** Sort by time, Sort files by modification time.
- **FILE\_SORT\_SIZE** Sort by size, Sort files by size.

**Type** enum in ['FILE\_SORT\_ALPHA', 'FILE\_SORT\_EXTENSION', 'FILE\_SORT\_TIME', 'FILE\_SORT\_SIZE'], default 'FILE\_SORT\_ALPHA'

**title**

Title for the file browser

**Type** string, default "", (readonly)

**use\_filter**

Enable filtering of files

**Type** boolean, default False

**use\_filter\_blender**

Show .blend files

**Type** boolean, default False

**use\_filter\_folder**

Show folders

**Type** boolean, default False

**use\_filter\_font**

Show font files

**Type** boolean, default False

**use\_filter\_image**

Show image files

**Type** boolean, default False

**use\_filter\_movie**

Show movie files

**Type** boolean, default False

**use\_filter\_script**

Show script files

**Type** boolean, default False

**use\_filter\_sound**

Show sound files

**Type** boolean, default False

**use\_filter\_text**

Show text files

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `SpaceFileBrowser.params`

## 2.4.221 Filter2DActuator(Actuator)

base classes — `bpy_struct`, `Actuator`

**class** `bpy.types.Filter2DActuator` (*Actuator*)

Actuator to apply screen graphic effects

**filter\_pass**

Set filter order

**Type** int in [0, 99], default 0

**gls1\_shader**

**Type** Text

**mode**

**Type** enum in ['ENABLE', 'DISABLE', 'REMOVE', 'MOTIONBLUR', 'BLUR', 'SHARPEN', 'DILATION', 'EROSION', 'LAPLACIAN', 'SOBEL', 'PREWITT', 'GRAYSCALE', 'SEPIA', 'INVERT', 'CUSTOMFILTER'], default 'REMOVE'

**motion\_blur\_factor**

Set motion blur factor

**Type** float in [0, 1], default 0.0

**use\_motion\_blur**

Enable/Disable Motion Blur

**Type** boolean, default False

**Inherited Properties**

- bpy\_struct.id\_data
- Actuator.name
- Actuator.show\_expanded
- Actuator.pin
- Actuator.type

**Inherited Functions**

- bpy\_struct.as\_pointer
- bpy\_struct.callback\_add
- bpy\_struct.callback\_remove
- bpy\_struct.driver\_add
- bpy\_struct.driver\_remove
- bpy\_struct.get
- bpy\_struct.is\_property\_hidden
- bpy\_struct.is\_property\_set
- bpy\_struct.items
- bpy\_struct.keyframe\_delete
- bpy\_struct.keyframe\_insert
- bpy\_struct.keys
- bpy\_struct.path\_from\_id
- bpy\_struct.path\_resolve
- bpy\_struct.type\_recast
- bpy\_struct.values
- Actuator.link
- Actuator.unlink

## 2.4.222 FloatProperty(Property)

base classes — bpy\_struct, Property

**class** bpy.types.**FloatProperty** (*Property*)  
RNA floating pointer number property definition

**array\_length**

Maximum length of the array, 0 means unlimited

**Type** int in [0, inf], default 0, (readonly)

**default**

Default value for this number

**Type** float in [-inf, inf], default 0.0, (readonly)

**default\_array**

Default value for this array

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0), (readonly)

**hard\_max**

Maximum value used by buttons

**Type** float in [-inf, inf], default 0.0, (readonly)

**hard\_min**

Minimum value used by buttons

**Type** float in [-inf, inf], default 0.0, (readonly)

**precision**

Number of digits after the dot used by buttons

**Type** int in [0, inf], default 0, (readonly)

**soft\_max**

Maximum value used by buttons

**Type** float in [-inf, inf], default 0.0, (readonly)

**soft\_min**

Minimum value used by buttons

**Type** float in [-inf, inf], default 0.0, (readonly)

**step**

Step size used by number buttons, for floats 1/100th of the step size

**Type** float in [0, inf], default 0.0, (readonly)

**Inherited Properties**

- `bpy_struct.id_data`
- `Property.name`
- `Property.is_animatable`
- `Property.srna`
- `Property.description`
- `Property.is_enum_flag`
- `Property.is_hidden`
- `Property.identifier`
- `Property.is_never_none`
- `Property.is_readonly`
- `Property.is_registered`
- `Property.is_registered_optional`
- `Property.is_required`
- `Property.is_output`
- `Property.is_runtime`
- `Property.is_skip_save`
- `Property.subtype`
- `Property.type`
- `Property.unit`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`

- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.223 FloorConstraint(Constraint)

base classes — `bpy_struct`, `Constraint`

**class** `bpy.types.FloorConstraint` (*Constraint*)

Uses the target object for location limitation

**floor\_location**

Location of target that object will not pass through

**Type** enum in ['FLOOR\_X', 'FLOOR\_Y', 'FLOOR\_Z', 'FLOOR\_NEGATIVE\_X', 'FLOOR\_NEGATIVE\_Y', 'FLOOR\_NEGATIVE\_Z'], default 'FLOOR\_X'

**offset**

Offset of floor from object origin

**Type** float in [0, 100], default 0.0

**subtarget**

**Type** string, default ""

**target**

Target Object

**Type** `Object`

**use\_rotation**

Use the target's rotation to determine floor

**Type** boolean, default False

**use\_sticky**

Immobilize object while constrained

**Type** boolean, default False

#### Inherited Properties

- `bpy_struct.id_data`
- `Constraint.name`
- `Constraint.active`
- `Constraint.mute`
- `Constraint.show_expanded`
- `Constraint.influence`
- `Constraint.error_location`

- `Constraint.owner_space`
- `Constraint.is_proxy_local`
- `Constraint.error_rotation`
- `Constraint.target_space`
- `Constraint.type`
- `Constraint.is_valid`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.224 FluidFluidSettings(FluidSettings)

base classes — `bpy_struct`, `FluidSettings`

**class** `bpy.types.FluidFluidSettings` (*FluidSettings*)

Fluid simulation settings for the fluid in the simulation

### **initial\_velocity**

Initial velocity of fluid

**Type** float array of 3 items in [-1000.1, 1000.1], default (0.0, 0.0, 0.0)

### **use**

Object contributes to the fluid simulation

**Type** boolean, default False

### **use\_animated\_mesh**

Export this mesh as an animated one. Slower, only use if really necessary (e.g. armatures or parented objects), animated pos/rot/scale IPOs do not require it

**Type** boolean, default False

### **volume\_initialization**

Volume initialization type

- **VOLUME** Volume, Use only the inner volume of the mesh.
- **SHELL** Shell, Use only the outer shell of the mesh.
- **BOTH** Both, Use both the inner volume and the outer shell of the mesh.

**Type** enum in ['VOLUME', 'SHELL', 'BOTH'], default 'VOLUME'

#### Inherited Properties

- `bpy_struct.id_data`
- `FluidSettings.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.225 FluidMeshVertex(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.FluidMeshVertex` (*bpy\_struct*)

Vertex of a simulated fluid mesh

**velocity**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0), (readonly)

#### Inherited Properties

- `bpy_struct.id_data`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`



- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `DomainFluidSettings.fluid_mesh_vertices`

### 2.4.226 FluidSettings(bpy\_struct)

base class — `bpy_struct`

subclasses — `FluidFluidSettings`, `DomainFluidSettings`, `ControlFluidSettings`,  
`InflowFluidSettings`, `ObstacleFluidSettings`, `ParticleFluidSettings`,  
`OutflowFluidSettings`

**class** `bpy.types.FluidSettings` (*bpy\_struct*)

Fluid simulation settings for an object taking part in the simulation

#### type

Type of participation in the fluid simulation

- **NONE** None.
- **DOMAIN** Domain, Bounding box of this object represents the computational domain of the fluid simulation.
- **FLUID** Fluid, Object represents a volume of fluid in the simulation.
- **OBSTACLE** Obstacle, Object is a fixed obstacle.
- **INFLOW** Inflow, Object adds fluid to the simulation.
- **OUTFLOW** Outflow, Object removes fluid from the simulation.
- **PARTICLE** Particle, Object is made a particle system to display particles generated by a fluidsim domain object.
- **CONTROL** Control, Object is made a fluid control mesh, which influences the fluid.

**Type** enum in ['NONE', 'DOMAIN', 'FLUID', 'OBSTACLE', 'INFLOW', 'OUTFLOW', 'PARTICLE', 'CONTROL'], default 'NONE'

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`

- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `FluidSimulationModifier.settings`

## 2.4.227 FluidSimulationModifier(Modifier)

base classes — `bpy_struct`, `Modifier`

**class** `bpy.types.FluidSimulationModifier` (*Modifier*)  
Fluid simulation modifier

### **settings**

Settings for how this object is used in the fluid simulation

**Type** `FluidSettings`, (readonly, never None)

## Inherited Properties

- `bpy_struct.id_data`
- `Modifier.name`
- `Modifier.use_apply_on_spline`
- `Modifier.show_in_editmode`
- `Modifier.show_expanded`
- `Modifier.show_on_cage`
- `Modifier.show_viewport`
- `Modifier.show_render`
- `Modifier.type`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`

- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.228 FollowPathConstraint(Constraint)

base classes — `bpy_struct`, `Constraint`

**class** `bpy.types.FollowPathConstraint` (*Constraint*)

Locks motion to the target path

**forward\_axis**

Axis that points forward along the path

**Type** enum in ['FORWARD\_X', 'FORWARD\_Y', 'FORWARD\_Z', 'TRACK\_NEGATIVE\_X', 'TRACK\_NEGATIVE\_Y', 'TRACK\_NEGATIVE\_Z'], default 'FORWARD\_X'

**offset**

Offset from the position corresponding to the time frame

**Type** float in [-300000, 300000], default 0.0

**offset\_factor**

Percentage value defining target position along length of bone

**Type** float in [0, 1], default 0.0

**target**

Target Object

**Type** `Object`

**up\_axis**

Axis that points upward

**Type** enum in ['UP\_X', 'UP\_Y', 'UP\_Z'], default 'UP\_X'

**use\_curve\_follow**

Object will follow the heading and banking of the curve

**Type** boolean, default False

**use\_curve\_radius**

Objects scale by the curve radius

**Type** boolean, default False

**use\_fixed\_location**

Object will stay locked to a single point somewhere along the length of the curve regardless of time

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `Constraint.name`
- `Constraint.active`
- `Constraint.mute`
- `Constraint.show_expanded`
- `Constraint.influence`
- `Constraint.error_location`
- `Constraint.owner_space`
- `Constraint.is_proxy_local`
- `Constraint.error_rotation`
- `Constraint.target_space`
- `Constraint.type`
- `Constraint.is_valid`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.229 Function(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.Function` (*bpy\_struct*)  
RNA function definition

**description**

Description of the Function's purpose

**Type** string, default "", (readonly)

**identifier**

Unique name used in the code and scripting

**Type** string, default "", (readonly)

**is\_registered**

Function is registered as callback as part of type registration

**Type** boolean, default False, (readonly)

**is\_registered\_optional**

Function is optionally registered as callback part of type registration

**Type** boolean, default False, (readonly)

**parameters**

Parameters for the function

**Type** `bpy_prop_collection` of `Property`, (readonly)

**use\_self**

Function does not pass its self as an argument (becomes a class method in python)

**Type** boolean, default False, (readonly)

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Struct.functions`

## 2.4.230 GPencilFrame(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.GPencilFrame` (*bpy\_struct*)

Collection of related sketches on a particular frame

**frame\_number**

The frame on which this sketch appears

**Type** int in [0, 300000], default 0

**is\_edited**

Frame is being edited (painted on)

**Type** boolean, default False

**select**

Frame is selected for editing in the DopeSheet

**Type** boolean, default False

**strokes**

Freehand curves defining the sketch on this frame

**Type** `bpy_prop_collection` of `GPencilStroke`, (readonly)

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `GPencilLayer.active_frame`
- `GPencilLayer.frames`

**2.4.231 GPencilLayer(bpy\_struct)**

base class — `bpy_struct`

**class** `bpy.types.GPencilLayer` (*bpy\_struct*)

Collection of related sketches

**active\_frame**

Frame currently being displayed for this layer

**Type** `GPencilFrame`, (readonly)

**alpha**

Layer Opacity

**Type** float in [0.3, 1], default 0.0**color**

Color for all strokes in this layer

**Type** float array of 3 items in [0, 1], default (0.0, 0.0, 0.0)**frames**

Sketches for this layer on different frames

**Type** `bpy_prop_collection` of `GPencilFrame`, (readonly)**ghost\_range\_max**

Maximum number of frames on either side of the active frame to show (0 = show the ‘first’ available sketch on either side)

**Type** int in [0, 120], default 0**hide**

Set layer Visibility

**Type** boolean, default False**info**

Layer name

**Type** string, default “”**line\_width**

Thickness of strokes (in pixels)

**Type** int in [1, 10], default 0**lock**

Protect layer from further editing and/or frame changes

**Type** boolean, default False**lock\_frame**

Lock current frame displayed by layer

**Type** boolean, default False**select**

Layer is selected for editing in the DopeSheet

**Type** boolean, default False**show\_points**

Draw the points which make up the strokes (for debugging purposes)

**Type** boolean, default False**show\_x\_ray****Type** boolean, default False**use\_onion\_skinning**

Ghost frames on either side of frame

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `GreasePencil.layers`

## 2.4.232 GPencilStroke(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.GPencilStroke` (*bpy\_struct*)  
Freehand curve defining part of a sketch

**points**

Stroke data points

**Type** `bpy_prop_collection` of `GPencilStrokePoint`, (readonly)

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`



- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `GPencilFrame.strokes`

### 2.4.233 GPencilStrokePoint(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.GPencilStrokePoint` (*bpy\_struct*)

Data point for freehand stroke curve

**co**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**pressure**

Pressure of tablet at point when drawing it

**Type** float in [0, 1], default 0.0

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `GPencilStroke.points`

### 2.4.234 GameActuator(Actuator)

base classes — `bpy_struct`, `Actuator`

class `bpy.types.GameActuator` (*Actuator*)

#### **filename**

Load this blend file, use the “/” prefix for a path relative to the current blend file

**Type** string, default “”

#### **mode**

**Type** enum in ['START', 'RESTART', 'QUIT', 'SAVECFG', 'LOADCFG'], default 'START'

## Inherited Properties

- `bpy_struct.id_data`
- `Actuator.name`
- `Actuator.show_expanded`
- `Actuator.pin`
- `Actuator.type`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Actuator.link`
- `Actuator.unlink`

### 2.4.235 GameBooleanProperty(GameProperty)

base classes — `bpy_struct`, `GameProperty`

**class** `bpy.types.GameBooleanProperty` (*GameProperty*)  
 Game engine user defined Boolean property

**value**  
 Property value

**Type** boolean, default False

#### Inherited Properties

- `bpy_struct.id_data`
- `GameProperty.name`
- `GameProperty.show_debug`
- `GameProperty.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.236 GameFloatProperty(GameProperty)

base classes — `bpy_struct`, `GameProperty`

**class** `bpy.types.GameFloatProperty` (*GameProperty*)  
 Game engine user defined floating pointer number property

**value**  
 Property value

**Type** float in [-10000, 10000], default 0.0

#### Inherited Properties

- `bpy_struct.id_data`
- `GameProperty.name`
- `GameProperty.show_debug`
- `GameProperty.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.237 GameIntProperty(GameProperty)

base classes — `bpy_struct`, `GameProperty`

**class** `bpy.types.GameIntProperty` (*GameProperty*)

Game engine user defined integer number property

**value**

Property value

**Type** `int` in `[-10000, 10000]`, default `0`

### Inherited Properties

- `bpy_struct.id_data`
- `GameProperty.name`
- `GameProperty.show_debug`
- `GameProperty.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`

- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.238 `GameObjectSettings(bpy_struct)`

base class — `bpy_struct`

**class** `bpy.types.GameObjectSettings` (*bpy\_struct*)

Game engine related settings for the object

**actuators**

Game engine actuators to act on events

**Type** `bpy_prop_collection` of `Actuator`, (readonly)

**collision\_bounds\_type**

Select the collision type

**Type** enum in ['BOX', 'SPHERE', 'CYLINDER', 'CONE', 'CONVEX\_HULL', 'TRIANGLE\_MESH', 'CAPSULE'], default 'BOX'

**collision\_margin**

Extra margin around object for collision detection, small amount required for stability

**Type** float in [0, 1], default 0.0

**controllers**

Game engine controllers to process events, connecting sensors to actuators

**Type** `bpy_prop_collection` of `Controller`, (readonly)

**damping**

General movement damping

**Type** float in [0, 1], default 0.0

**form\_factor**

Form factor scales the inertia tensor

**Type** float in [0, 1], default 0.0

**friction\_coefficients**

Relative friction coefficients in the in the X, Y and Z directions, when anisotropic friction is enabled

**Type** float array of 3 items in [0, 1], default (0.0, 0.0, 0.0)

**lock\_location\_x**

Disable simulation of linear motion along the X axis

**Type** boolean, default False

**lock\_location\_y**

Disable simulation of linear motion along the Y axis

**Type** boolean, default False

**lock\_location\_z**

Disable simulation of linear motion along the Z axis

**Type** boolean, default False

**lock\_rotation\_x**

Disable simulation of angular motion along the X axis

**Type** boolean, default False

**lock\_rotation\_y**

Disable simulation of angular motion along the Y axis

**Type** boolean, default False

**lock\_rotation\_z**

Disable simulation of angular motion along the Z axis

**Type** boolean, default False

**mass**

Mass of the object

**Type** float in [0.01, 10000], default 0.0

**obstacle\_radius**

Radius of object representation in obstacle simulation

**Type** float in [0, 1000], default 0.0

**physics\_type**

Select the type of physical representation

- NO\_COLLISION No Collision, Disable collision for this object.
- STATIC Static, Stationary object.
- DYNAMIC Dynamic, Linear physics.
- RIGID\_BODY Rigid Body, Linear and angular physics.
- SOFT\_BODY Soft Body, Soft body.
- OCCLUDE Occlude, Occluder for optimizing scene rendering.
- SENSOR Sensor, Collision Sensor, detects static and dynamic objects but not the other collision sensor objects.
- NAVMESH Navigation Mesh, Navigation mesh.

**Type** enum in ['NO\_COLLISION', 'STATIC', 'DYNAMIC', 'RIGID\_BODY', 'SOFT\_BODY', 'OCCLUDE', 'SENSOR', 'NAVMESH'], default 'NO\_COLLISION'

**properties**

Game engine properties

**Type** `bpy_prop_collection` of `GameProperty`, (readonly)

**radius**

Radius of bounding sphere and material physics

**Type** float in [0.01, 10], default 0.0

**rotation\_damping**

General rotation damping

**Type** float in [0, 1], default 0.0

**sensors**

Game engine sensor to detect events

**Type** `bpy_prop_collection` of `Sensor`, (readonly)

**show\_actuators**

Shows actuators for this object in the user interface

**Type** boolean, default False

**show\_controllers**

Shows controllers for this object in the user interface

**Type** boolean, default False

**show\_debug\_state**

Print state debug info in the game engine

**Type** boolean, default False

**show\_sensors**

Shows sensors for this object in the user interface

**Type** boolean, default False

**show\_state\_panel**

Show state panel

**Type** boolean, default False

**soft\_body**

Settings for Bullet soft body simulation

**Type** `GameSoftBodySettings`, (readonly)

**states\_initial**

Initial state when the game starts

**Type** boolean array of 30 items, default (False, False)

**states\_visible**

State determining which controllers are displayed

**Type** boolean array of 30 items, default (False, False)

**use\_activity\_culling**

Disable simulation of angular motion along the Z axis

**Type** boolean, default False

**use\_actor**

Object is detected by the Near and Radar sensor

**Type** boolean, default False

**use\_all\_states**

Set all state bits

**Type** boolean, default False

**use\_anisotropic\_friction**

Enable anisotropic friction

**Type** boolean, default False

**use\_collision\_bounds**

Specify a collision bounds type other than the default

**Type** boolean, default False

**use\_collision\_compound**

Add children to form a compound collision object

**Type** boolean, default False

**use\_ghost**

Object does not retribute collisions, like a ghost

**Type** boolean, default False

**use\_material\_physics\_fh**

React to force field physics settings in materials

**Type** boolean, default False

**use\_obstacle\_create**

Create representation for obstacle simulation

**Type** boolean, default False

**use\_rotate\_from\_normal**

Use face normal to rotate object, so that it points away from the surface

**Type** boolean, default False

**use\_sleep**

Disable auto (de)activation in physics simulation

**Type** boolean, default False

**used\_states**

States which are being used by controllers

**Type** boolean array of 30 items, default (False, False), (readonly)

**velocity\_max**

Clamp velocity to this maximum speed

**Type** float in [0, 1000], default 0.0

**velocity\_min**

Clamp velocity to this minimum speed (except when totally still)

**Type** float in [0, 1000], default 0.0

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`



- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Object.game`

### 2.4.239 GameProperty(bpy\_struct)

base class — `bpy_struct`

subclasses — `GameStringProperty`, `GameIntProperty`, `GameBooleanProperty`, `GameFloatProperty`, `GameTimerProperty`

**class** `bpy.types.GameProperty` (*bpy\_struct*)

Game engine user defined object property

#### **name**

Available as `GameObject` attributes in the game engine's python API

**Type** string, default ""

#### **show\_debug**

Print debug information for this property

**Type** boolean, default False

#### **type**

- **BOOL** Boolean, Boolean Property.
- **INT** Integer, Integer Property.
- **FLOAT** Float, Floating-Point Property.
- **STRING** String, String Property.
- **TIMER** Timer, Timer Property.

**Type** enum in ['BOOL', 'INT', 'FLOAT', 'STRING', 'TIMER'], default 'BOOL'

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `GameObjectSettings.properties`

### 2.4.240 GameSoftBodySettings(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.GameSoftBodySettings` (*bpy\_struct*)  
Soft body simulation settings for an object in the game engine

**cluster\_iterations**

Number of cluster iterations

**Type** int in [1, 128], default 0

**collision\_margin**

Collision margin for soft body. Small value makes the algorithm unstable

**Type** float in [0.01, 1], default 0.0

**dynamic\_friction**

Dynamic Friction

**Type** float in [0, 1], default 0.0

**linear\_stiffness**

Linear stiffness of the soft body links

**Type** float in [0, 1], default 0.0

**location\_iterations**

Position solver iterations

**Type** int in [0, 10], default 0

**shape\_threshold**

Shape matching threshold

**Type** float in [0, 1], default 0.0

**use\_bending\_constraints**

Enable bending constraints

**Type** boolean, default False

**use\_cluster\_rigid\_to\_softbody**

Enable cluster collision between soft and rigid body

**Type** boolean, default False

**use\_cluster\_soft\_to\_softbody**

Enable cluster collision between soft and soft body

**Type** boolean, default False

**use\_shape\_match**

Enable soft body shape matching goal

**Type** boolean, default False

**weld\_threshold**

Welding threshold: distance between nearby vertices to be considered equal => set to 0.0 to disable welding test and speed up scene loading (ok if the mesh has no duplicates)

**Type** float in [0, 0.01], default 0.0

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `GameObjectSettings.soft_body`

### 2.4.241 GameStringProperty(GameProperty)

base classes — `bpy_struct`, `GameProperty`

**class** `bpy.types.GameStringProperty` (*GameProperty*)  
Game engine user defined text string property

**value**

Property value

**Type** string, default ""

#### Inherited Properties

- `bpy_struct.id_data`
- `GameProperty.name`
- `GameProperty.show_debug`
- `GameProperty.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.242 GameTimerProperty(GameProperty)

base classes — `bpy_struct`, `GameProperty`

**class** `bpy.types.GameTimerProperty` (*GameProperty*)  
Game engine user defined timer property

**value**

Property value

**Type** float in [-10000, 10000], default 0.0

#### Inherited Properties

- `bpy_struct.id_data`

- `GameProperty.name`
- `GameProperty.show_debug`
- `GameProperty.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.243 GlowSequence(EffectSequence)

base classes — `bpy_struct`, `Sequence`, `EffectSequence`

**class** `bpy.types.GlowSequence` (*EffectSequence*)  
Sequence strip creating a glow effect

**blur\_radius**

Radius of glow effect

**Type** float in [0.5, 20], default 0.0

**boost\_factor**

Brightness multiplier

**Type** float in [0, 10], default 0.0

**clamp**

rightness limit of intensity

**Type** float in [0, 1], default 0.0

**quality**

Accuracy of the blur effect

**Type** int in [1, 5], default 0

**threshold**

Minimum intensity to trigger a glow

**Type** float in [0, 1], default 0.0

**use\_only\_boost**

Show the glow buffer only

**Type** boolean, default False

**Inherited Properties**

- `bpy_struct.id_data`
- `Sequence.name`
- `Sequence.blend_type`
- `Sequence.blend_alpha`
- `Sequence.channel`
- `Sequence.waveform`
- `Sequence.effect_fader`
- `Sequence.frame_final_end`
- `Sequence.frame_offset_end`
- `Sequence.frame_still_end`
- `Sequence.input_1`
- `Sequence.input_2`
- `Sequence.input_3`
- `Sequence.select_left_handle`
- `Sequence.frame_final_duration`
- `Sequence.frame_duration`
- `Sequence.lock`
- `Sequence.mute`
- `Sequence.select_right_handle`
- `Sequence.select`
- `Sequence.speed_factor`
- `Sequence.frame_start`
- `Sequence.frame_final_start`
- `Sequence.frame_offset_start`
- `Sequence.frame_still_start`
- `Sequence.type`
- `Sequence.use_default_fade`
- `Sequence.input_count`
- `EffectSequence.color_balance`
- `EffectSequence.use_float`
- `EffectSequence.crop`
- `EffectSequence.use_deinterlace`
- `EffectSequence.use_reverse_frames`
- `EffectSequence.use_flip_x`
- `EffectSequence.use_flip_y`
- `EffectSequence.color_multiply`
- `EffectSequence.use_premultiply`
- `EffectSequence.proxy`
- `EffectSequence.use_proxy_custom_directory`
- `EffectSequence.use_proxy_custom_file`
- `EffectSequence.color_saturation`
- `EffectSequence.strobe`
- `EffectSequence.transform`
- `EffectSequence.use_color_balance`
- `EffectSequence.use_crop`
- `EffectSequence.use_proxy`
- `EffectSequence.use_translation`

**Inherited Functions**

- `bpy_struct.as_pointer`

- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Sequence.getStripElem`
- `Sequence.swap`

### 2.4.244 GreasePencil(ID)

base classes — `bpy_struct`, `ID`

**class** `bpy.types.GreasePencil` (*ID*)

Freehand annotation sketchbook

#### **draw\_mode**

- **CURSOR** Cursor, Draw stroke at the 3D cursor.
- **VIEW** View, Stick stroke to the view .
- **SURFACE** Surface, Stick stroke to surfaces.
- **STROKE** Stroke, Stick stroke to other strokes.

**Type** enum in ['CURSOR', 'VIEW', 'SURFACE', 'STROKE'], default 'VIEW'

#### **layers**

**Type** `GreasePencilLayers` `bpy_prop_collection` of `GPencilLayer`, (readonly)

#### **use\_stroke\_endpoints**

Only use the first and last parts of the stroke for snapping

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`

### References

- `BlendData.grease_pencil`
- `GreasePencilLayers.active`
- `NodeTree.grease_pencil`
- `Object.grease_pencil`
- `Scene.grease_pencil`
- `SpaceImageEditor.grease_pencil`

### 2.4.245 GreasePencilLayers(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.GreasePencilLayers` (*bpy\_struct*)

Collection of grease pencil layers

**active**

Active grease pencil layer

**Type** `GreasePencil`

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`



- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `GreasePencil.layers`

## 2.4.246 Group(ID)

base classes — `bpy_struct`, `ID`

**class** `bpy.types.Group` (*ID*)  
Group of Object datablocks

### **dupli\_offset**

Offset from the origin to use when instancing as DupliGroup

**Type** float array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`

### **layers**

Layers visible when this groups is instanced as a dupli

**Type** boolean array of 20 items, default `(False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False)`

### **objects**

A collection of this groups objects

**Type** `GroupObjects` `bpy_prop_collection` of `Object`, (readonly)

### **users\_dupli\_group**

The dupli group this group is used in (readonly)

## Inherited Properties

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`

- `ID.users`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`

### References

- `BlendData.groups`
- `BlendDataGroups.new`
- `BlendDataGroups.remove`
- `ClothCollisionSettings.group`
- `DopeSheet.filter_group`
- `EffectorWeights.group`
- `Material.light_group`
- `Object.dupli_group`
- `ParticleSettings.dupli_group`
- `RenderLayer.light_override`
- `SceneRenderLayer.light_override`
- `SmokeDomainSettings.collision_group`
- `SmokeDomainSettings.effector_group`
- `SmokeDomainSettings.fluid_group`

## 2.4.247 GroupInputs(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.GroupInputs` (*bpy\_struct*)  
Collection of group sockets

**new** (*name*="Socket", *type*='VALUE')  
Add a socket to the group tree

**Parameters**

- **name** (*string, (optional)*) – Name, Name of the socket
- **type** (*enum in ['VALUE', 'VECTOR', 'RGBA'], (optional)*) – Type, Type of socket

**Returns** New socket**Return type** `NodeSocket`**expose** (*sock=None, add\_link=True*)

Expose an internal socket in the group tree

**Parameters**

- **sock** (`NodeSocket`, *(optional)*) – Socket, Internal node socket to expose
- **add\_link** (*boolean, (optional)*) – Add Link, If TRUE, adds a link to the internal socket

**Returns** New socket**Return type** `NodeSocket`**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `NodeTree.inputs`

**2.4.248 GroupObjects(`bpy_struct`)**base class — `bpy_struct`**class** `bpy.types.GroupObjects` (*`bpy_struct`*)

Collection of group objects

**link** (*object*)

Add this object to a group

**Parameters** **object** (*Object*, (never None)) – Object to add

**unlink** (*object*)

Remove this object to a group

**Parameters** **object** (*Object*) – Object to remove

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Group.objects`

## 2.4.249 GroupOutputs(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.GroupOutputs` (*bpy\_struct*)

Collection of group sockets

**new** (*name*="Socket", *type*='VALUE')

Add a socket to the group tree

#### Parameters

- **name** (*string*, (*optional*)) – Name, Name of the socket
- **type** (*enum* in [`'VALUE'`, `'VECTOR'`, `'RGBA'`], (*optional*)) – Type, Type of socket

**Returns** New socket

**Return type** `NodeSocket`

**expose** (*sock=None, add\_link=True*)

Expose an internal socket in the group tree

**Parameters**

- **sock** (`NodeSocket`, (optional)) – Socket, Internal node socket to expose
- **add\_link** (*boolean, (optional)*) – Add Link, If TRUE, adds a link to the internal socket

**Returns** New socket

**Return type** `NodeSocket`

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `NodeTree.outputs`

## 2.4.250 Header(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.Header` (*bpy\_struct*)

Editor header containing UI elements

**bl\_idname**

If this is set, the header gets a custom ID, otherwise it takes the name of the class used to define the panel; for example, if the class name is “OBJECT\_HT\_hello”, and `bl_idname` is not set by the script, then `bl_idname = “OBJECT_HT_hello”`

**Type** string, default “”

### **bl\_space\_type**

The space where the header is going to be used in

**Type** enum in ['EMPTY', 'VIEW\_3D', 'GRAPH\_EDITOR', 'OUTLINER', 'PROPERTIES', 'FILE\_BROWSER', 'IMAGE\_EDITOR', 'INFO', 'SEQUENCE\_EDITOR', 'TEXT\_EDITOR', 'DOPE SHEET\_EDITOR', 'NLA\_EDITOR', 'TIMELINE', 'NODE\_EDITOR', 'LOGIC\_EDITOR', 'CONSOLE', 'USER\_PREFERENCES'], default 'EMPTY'

### **layout**

Structure of the header in the UI

**Type** `UILayout`, (readonly)

### **draw** (*context*)

Draw UI elements into the header UI layout

### **classmethod append** (*draw\_func*)

Append a draw function to this menu, takes the same arguments as the menu draw function

### **classmethod prepend** (*draw\_func*)

Prepend a draw function to this menu, takes the same arguments as the menu draw function

### **classmethod remove** (*draw\_func*)

Remove a draw function that has been added to this menu

## **Inherited Properties**

- `bpy_struct.id_data`

## **Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## **2.4.251 HemiLamp(Lamp)**

base classes — `bpy_struct`, `ID`, `Lamp`

```
class bpy.types.HemiLamp(Lamp)
    180 degree constant lamp
```

### Inherited Properties

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`
- `Lamp.active_texture`
- `Lamp.active_texture_index`
- `Lamp.animation_data`
- `Lamp.color`
- `Lamp.use_diffuse`
- `Lamp.distance`
- `Lamp.energy`
- `Lamp.use_own_layer`
- `Lamp.use_negative`
- `Lamp.node_tree`
- `Lamp.use_specular`
- `Lamp.texture_slots`
- `Lamp.type`
- `Lamp.use_nodes`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`

### 2.4.252 Histogram(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.Histogram` (*bpy\_struct*)  
Statistical view of the levels of color in an image

**mode**

Channels to display when drawing the histogram

**Type** enum in ['LUMA', 'RGB', 'R', 'G', 'B'], default 'LUMA'

#### Inherited Properties

- `bpy_struct.id_data`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

#### References

- `Scopes.histogram`
- `SpaceImageEditor.sample_histogram`

### 2.4.253 HookModifier(`Modifier`)

base classes — `bpy_struct`, `Modifier`

**class** `bpy.types.HookModifier` (*Modifier*)  
Hook modifier to modify the location of vertices

**falloff**

If not zero, the distance from the hook where influence ends

**Type** float in [0, inf], default 0.0

**force**

Relative force of the hook



**Type** float in [0, 1], default 0.0

**object**

Parent Object for hook, also recalculates and clears offset

**Type** Object

**subtarget**

Name of Parent Bone for hook (if applicable), also recalculates and clears offset

**Type** string, default ""

**vertex\_group**

Name of Vertex Group which determines influence of modifier per point

**Type** string, default ""

### Inherited Properties

- `bpy_struct.id_data`
- `Modifier.name`
- `Modifier.use_apply_on_spline`
- `Modifier.show_in_editmode`
- `Modifier.show_expanded`
- `Modifier.show_on_cage`
- `Modifier.show_viewport`
- `Modifier.show_render`
- `Modifier.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.254 ID(`bpy_struct`)

base class — `bpy_struct`

subclasses — `Lattice`, `Library`, `Key`, `NodeTree`, `MetaBall`, `Text`, `Lamp`, `World`, `Brush`, `Object`, `Armature`, `Mesh`, `VectorFont`, `GreasePencil`, `Sound`, `ParticleSettings`, `Scene`, `WindowManager`, `Texture`, `Curve`, `Action`, `Group`, `Screen`, `Speaker`, `Material`, `Image`, `Camera`

**class** `bpy.types.ID` (*bpy\_struct*)

Base type for datablocks, defining a unique name, linking from other libraries and garbage collection

**is\_updated**

Datablock is tagged for recalculation

**Type** boolean, default False, (readonly)

**is\_updated\_data**

Datablock data is tagged for recalculation

**Type** boolean, default False, (readonly)

**library**

Library file the datablock is linked from

**Type** `Library`, (readonly)

**name**

Unique datablock ID name

**Type** string, default ""

**tag**

Tools can use this to tag data (initial state is undefined)

**Type** boolean, default False

**use\_fake\_user**

Save this datablock even if it has no users

**Type** boolean, default False

**users**

Number of times this datablock is referenced

**Type** int in [0, 32767], default 0, (readonly)

**copy** ()

Create a copy of this datablock (not supported for all datablocks)

**Returns** New copy of the ID

**Return type** `ID`

**user\_clear** ()

Clear the user count of a datablock so its not saved, on reload the data will be removed

This function is for advanced use only, misuse can crash blender since the user count is used to prevent data being removed when it is used.

*# This example shows what \_not\_ to do, and will crash blender.*

```
import bpy
```

```
# object which is in the scene.
```

```
obj = bpy.data.objects["Cube"]
```

```
# without this, removal would raise an error.
```

```
obj.user_clear()
```

```
# runs without an exception
```

```
# but will crash on redraw.
```

```
bpy.data.objects.remove(obj)
```

**animation\_data\_create()**

Create animation data to this ID, note that not all ID types support this

**Returns** New animation data or NULL

**Return type** `AnimData`

**animation\_data\_clear()**

Clear animation on this this ID

**update\_tag(*refresh=set()*)**

Tag the ID to update its display data

**Parameters** **refresh** (*enum set in {'OBJECT', 'DATA', 'TIME'}, (optional)*) – Type of updates to perform

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `BlendData.scripts`
- `BlendDataObjects.new`
- `DopeSheet.source`
- `DriverTarget.id`
- `ID.copy`
- `Key.user`
- `KeyingSetPath.id`
- `KeyingSetPaths.add`
- `Object.data`
- `SpaceNodeEditor.id`
- `SpaceNodeEditor.id_from`
- `SpaceProperties.pin_id`
- `UILayout.template_path_builder`

- `UILayout.template_preview`
- `UILayout.template_preview`

### 2.4.255 IDMaterials(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.IDMaterials` (*bpy\_struct*)  
Collection of materials

**append** (*material*)

Add a new material to the data block

**Parameters** **material** (`Material`) – Material to add

**pop** (*index, update\_data=False*)

Remove a material from the data block

**Parameters**

- **index** (*int in [0, 32766]*) – Index of material to remove
- **update\_data** (*boolean, (optional)*) – Update data by re-adjusting the material slots assigned

**Returns** Material to remove

**Return type** `Material`

#### Inherited Properties

- `bpy_struct.id_data`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

#### References

- `Curve.materials`

- `Mesh.materials`
- `MetaBall.materials`

### 2.4.256 IKParam(bpy\_struct)

base class — `bpy_struct`

subclasses — `Itasc`

**class** `bpy.types.IKParam` (*bpy\_struct*)

Base type for IK solver parameters

**ik\_solver**

IK solver for which these parameters are defined, 0 for Legacy, 1 for iTaSC

- LEGACY Legacy, Original IK solver.
- ITASC iTaSC, Multi constraint, stateful IK solver.

**Type** enum in ['LEGACY', 'ITASC'], default 'LEGACY', (readonly)

#### Inherited Properties

- `bpy_struct.id_data`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

#### References

- `Pose.ik_param`

### 2.4.257 Image(ID)

base classes — `bpy_struct`, `ID`

**class** `bpy.types.Image` (*ID*)

Image datablock referencing an external or packed image

**bindcode**

OpenGL bindcode

**Type** int in [0, inf], default 0, (readonly)

**depth**

Image bit depth

**Type** int in [0, inf], default 0, (readonly)

**display\_aspect**

Display Aspect for this image, does not affect rendering

**Type** float array of 2 items in [0.1, 5000], default (0.0, 0.0)

**field\_order**

Order of video fields. Select which lines are displayed first

- EVEN Upper First, Upper field first.
- ODD Lower First, Lower field first.

**Type** enum in ['EVEN', 'ODD'], default 'EVEN'

**file\_format**

Format used for re-saving this file

- BMP BMP, Output image in bitmap format.
- IRIS Iris, Output image in (old!) SGI IRIS format.
- PNG PNG, Output image in PNG format.
- JPEG JPEG, Output image in JPEG format.
- TARGA Targa, Output image in Targa format.
- TARGA\_RAW Targa Raw, Output image in uncompressed Targa format.
- MULTILAYER MultiLayer, Output image in multilayer OpenEXR format.
- OPEN\_EXR OpenEXR, Output image in OpenEXR format.
- TIFF TIFF, Output image in TIFF format.
- AVI\_JPEG AVI JPEG, Output video in AVI JPEG format.
- AVI\_RAW AVI Raw, Output video in AVI Raw format.

**Type** enum in ['BMP', 'IRIS', 'PNG', 'JPEG', 'TARGA', 'TARGA\_RAW', 'MULTILAYER', 'OPEN\_EXR', 'TIFF', 'AVI\_JPEG', 'AVI\_RAW'], default 'TARGA'

**filepath**

Image/Movie file name

**Type** string, default ""

**filepath\_raw**

Image/Movie file name (without data refreshing)

**Type** string, default ""

**fps**

Speed of the animation in frames per second

**Type** int in [1, 100], default 0

**frame\_end**

End frame of an animated texture

**Type** int in [0, 128], default 0

**frame\_start**

Start frame of an animated texture

**Type** int in [0, 128], default 0

**generated\_height**

Generated image height

**Type** int in [1, 16384], default 0

**generated\_type**

Generated image type

- BLANK Blank, Generate a blank image.
- UV\_GRID UV Grid, Generated grid to test UV mappings.
- COLOR\_GRID Color Grid, Generated improved UV grid to test UV mappings.

**Type** enum in ['BLANK', 'UV\_GRID', 'COLOR\_GRID'], default 'BLANK'

**generated\_width**

Generated image width

**Type** int in [1, 16384], default 0

**has\_data**

True if this image has data

**Type** boolean, default False, (readonly)

**is\_dirty**

Image has changed and is not saved

**Type** boolean, default False, (readonly)

**mapping**

Mapping type to use for this image in the game engine

- UV UV Coordinates, Use UV coordinates for mapping the image.
- REFLECTION Reflection, Use reflection mapping for mapping the image.

**Type** enum in ['UV', 'REFLECTION'], default 'UV'

**packed\_file**

**Type** PackedFile, (readonly)

**pixels**

Image pixels in floating point values

**Type** float in [-inf, inf], default 0.0

**resolution**

X/Y pixels per meter

**Type** float array of 2 items in [-inf, inf], default (0.0, 0.0)

**size**

Width and height in pixels, zero when image data cant be loaded

**Type** int array of 2 items in [-inf, inf], default (0, 0), (readonly)

**source**

Where the image comes from

- FILE Single Image, Single image file.
- SEQUENCE Image Sequence, Multiple image files, as a sequence.
- MOVIE Movie File, Movie file.
- GENERATED Generated, Generated image.
- VIEWER Viewer, Compositing node viewer.

**Type** enum in ['FILE', 'SEQUENCE', 'MOVIE', 'GENERATED', 'VIEWER'], default 'FILE'

**tiles\_x**

Degree of repetition in the X direction

**Type** int in [1, 16], default 0

**tiles\_y**

Degree of repetition in the Y direction

**Type** int in [1, 16], default 0

**type**

How to generate the image

**Type** enum in ['IMAGE', 'MULTILAYER', 'UV\_TEST', 'RENDER\_RESULT', 'COMPOSITING'], default 'IMAGE', (readonly)

**use\_animation**

Use as animated texture in the game engine

**Type** boolean, default False

**use\_clamp\_x**

Disable texture repeating horizontally

**Type** boolean, default False

**use\_clamp\_y**

Disable texture repeating vertically

**Type** boolean, default False

**use\_fields**

Use fields of the image

**Type** boolean, default False

**use\_generated\_float**

Generate floating point buffer

**Type** boolean, default False



**use\_premultiply**

Convert RGB from key alpha to premultiplied alpha

**Type** boolean, default False

**use\_tiles**

Use of tilemode for faces (default shift-LMB to pick the tile for selected faces)

**Type** boolean, default False

**save\_render** (*filepath, scene=None*)

Save image to a specific path using a scenes render settings

**Parameters**

- **filepath** (*string*) – Save path
- **scene** (*Scene*, (optional)) – Scene to take image parameters from

**save** ()

Save image to its source path

**pack** (*as\_png=False*)

Pack an image as embedded data into the .blend file

**Parameters** **as\_png** (*boolean*, (optional)) – as\_png, Pack the image as PNG (needed for generated/dirty images)

**unpack** (*method='USE\_LOCAL'*)

Save an image packed in the .blend file to disk

**Parameters** **method** (*enum in ['USE\_LOCAL', 'WRITE\_LOCAL', 'USE\_ORIGINAL', 'WRITE\_ORIGINAL']*, (optional)) – method, How to unpack

**reload** ()

Reload the image from its source path

**update** ()

Update the display image from the floating point buffer

**gl\_load** (*filter=9985, mag=9729*)

Load the image into OpenGL graphics memory

**Parameters**

- **filter** (*int in [-inf, inf]*, (optional)) – Filter, The texture minifying function
- **mag** (*int in [-inf, inf]*, (optional)) – Magnification, The texture magnification function

**Returns** Error, OpenGL error value

**Return type** int in [-inf, inf]

**gl\_free** ()

Free the image from OpenGL graphics memory

**Inherited Properties**

- bpy\_struct.id\_data
- ID.name
- ID.use\_fake\_user
- ID.is\_updated
- ID.is\_updated\_data

- `ID.library`
- `ID.tag`
- `ID.users`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`

### References

- `BackgroundImage.image`
- `BlendData.images`
- `BlendDataImages.load`
- `BlendDataImages.new`
- `BlendDataImages.remove`
- `Brush.clone_image`
- `CompositorNodeImage.image`
- `EnvironmentMapTexture.image`
- `ImageTexture.image`
- `MeshTextureFace.image`
- `SpaceImageEditor.image`
- `TextureNodeImage.image`
- `UILayout.template_image_layers`
- `UVProjectModifier.image`
- `VoxelDataTexture.image`

## 2.4.258 ImagePaint(Paint)

base classes — `bpy_struct`, `Paint`

**class** `bpy.types.ImagePaint` (*Paint*)

Properties of image and texture painting mode

**invert\_stencil**

Invert the stencil layer

**Type** boolean, default False

**normal\_angle**

Paint most on faces pointing towards the view according to this angle

**Type** int in [0, 90], default 0

**screen\_grab\_size**

Size to capture the image for re-projecting

**Type** int array of 2 items in [512, 16384], default (0, 0)

**seam\_bleed**

Extend paint beyond the faces UVs to reduce seams (in pixels, slower)

**Type** int in [0, 32767], default 0

**use\_backface\_culling**

Ignore faces pointing away from the view (faster)

**Type** boolean, default False

**use\_clone\_layer**

Use another UV layer as clone source, otherwise use 3D the cursor as the source

**Type** boolean, default False

**use\_normal\_falloff**

Paint most on faces pointing towards the view

**Type** boolean, default False

**use\_occlude**

Only paint onto the faces directly under the brush (slower)

**Type** boolean, default False

**use\_projection**

Use projection painting for improved consistency in the brush strokes

**Type** boolean, default False

**use\_stencil\_layer**

Set the mask layer from the UV layer buttons

**Type** boolean, default False

**Inherited Properties**

- `bpy_struct.id_data`
- `Paint.brush`
- `Paint.show_low_resolution`
- `Paint.show_brush`
- `Paint.show_brush_on_surface`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`

- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `ToolSettings.image_paint`

## 2.4.259 ImageSequence(Sequence)

base classes — `bpy_struct`, `Sequence`

**class** `bpy.types.ImageSequence` (*Sequence*)

Sequence strip to load one or more images

**animation\_offset\_end**

Animation end offset (trim end)

**Type** int in [0, inf], default 0

**animation\_offset\_start**

Animation start offset (trim start)

**Type** int in [0, inf], default 0

**color\_balance**

**Type** `SequenceColorBalance`, (readonly)

**color\_multiply**

**Type** float in [0, 20], default 0.0

**color\_saturation**

**Type** float in [0, 20], default 0.0

**crop**

**Type** `SequenceCrop`, (readonly)

**directory**

**Type** string, default ""

**elements**

**Type** `bpy_prop_collection` of `SequenceElement`, (readonly)

**proxy**

**Type** `SequenceProxy`, (readonly)

**strobe**

Only display every nth frame

**Type** float in [1, 30], default 0.0

**ttransform**

**Type** `SequenceTransform`, (readonly)

**use\_color\_balance**

(3-Way color correction) on input

**Type** boolean, default False

**use\_crop**

Crop image before processing

**Type** boolean, default False

**use\_deinterlace**

For video movies to remove fields

**Type** boolean, default False

**use\_flip\_x**

Flip on the X axis

**Type** boolean, default False

**use\_flip\_y**

Flip on the Y axis

**Type** boolean, default False

**use\_float**

Convert input to float data

**Type** boolean, default False

**use\_premultiply**

Convert RGB from key alpha to premultiplied alpha

**Type** boolean, default False

**use\_proxy**

Use a preview proxy and/or timecode index for this strip

**Type** boolean, default False

**use\_proxy\_custom\_directory**

Use a custom directory to store data

**Type** boolean, default False

**use\_proxy\_custom\_file**

Use a custom file to read proxy data from

**Type** boolean, default False

**use\_reverse\_frames**

Reverse frame order

**Type** boolean, default False

**use\_translation**

Translate image before processing

**Type** boolean, default False

**Inherited Properties**

- `bpy_struct.id_data`
- `Sequence.name`
- `Sequence.blend_type`
- `Sequence.blend_alpha`
- `Sequence.channel`
- `Sequence.waveform`
- `Sequence.effect_fader`
- `Sequence.frame_final_end`
- `Sequence.frame_offset_end`
- `Sequence.frame_still_end`
- `Sequence.input_1`
- `Sequence.input_2`
- `Sequence.input_3`
- `Sequence.select_left_handle`
- `Sequence.frame_final_duration`
- `Sequence.frame_duration`
- `Sequence.lock`
- `Sequence.mute`
- `Sequence.select_right_handle`
- `Sequence.select`
- `Sequence.speed_factor`
- `Sequence.frame_start`
- `Sequence.frame_final_start`
- `Sequence.frame_offset_start`
- `Sequence.frame_still_start`
- `Sequence.type`
- `Sequence.use_default_fade`
- `Sequence.input_count`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`

- `bpy_struct.values`
- `Sequence.getStripElem`
- `Sequence.swap`

## 2.4.260 ImageTexture(Texture)

base classes — `bpy_struct`, `ID`, `Texture`

**class** `bpy.types.ImageTexture` (*Texture*)

### **checker\_distance**

Sets distance between checker tiles

**Type** float in [0, 0.99], default 0.0

### **crop\_max\_x**

Sets maximum X value to crop the image

**Type** float in [-10, 10], default 0.0

### **crop\_max\_y**

Sets maximum Y value to crop the image

**Type** float in [-10, 10], default 0.0

### **crop\_min\_x**

Sets minimum X value to crop the image

**Type** float in [-10, 10], default 0.0

### **crop\_min\_y**

Sets minimum Y value to crop the image

**Type** float in [-10, 10], default 0.0

### **extension**

Sets how the image is extrapolated past its original bounds

- **EXTEND** Extend, Extend by repeating edge pixels of the image.
- **CLIP** Clip, Clip to image size and sets exterior pixels as transparent.
- **CLIP\_CUBE** Clip Cube, Clip to cubic-shaped area around the image and sets exterior pixels as transparent.
- **REPEAT** Repeat, Cause the image to repeat horizontally and vertically.
- **CHECKER** Checker, Cause the image to repeat in checker board pattern.

**Type** enum in ['EXTEND', 'CLIP', 'CLIP\_CUBE', 'REPEAT', 'CHECKER'], default 'EXTEND'

### **filter\_eccentricity**

Maximum eccentricity. Higher gives less blur at distant/oblique angles, but is also slower

**Type** int in [1, 256], default 0

### **filter\_probes**

Maximum number of samples. Higher gives less blur at distant/oblique angles, but is also slower

**Type** int in [1, 256], default 0

**filter\_size**

Multiply the filter size used by MIP Map and Interpolation

**Type** float in [0.1, 50], default 0.0

**filter\_type**

Texture filter to use for sampling image

**Type** enum in ['BOX', 'EWA', 'FELINE', 'AREA'], default 'BOX'

**image**

**Type** [Image](#)

**image\_user**

Parameters defining which layer, pass and frame of the image is displayed

**Type** [ImageUser](#), (readonly)

**invert\_alpha**

Inverts all the alpha values in the image

**Type** boolean, default False

**repeat\_x**

Sets a repetition multiplier in the X direction

**Type** int in [1, 512], default 0

**repeat\_y**

Sets a repetition multiplier in the Y direction

**Type** int in [1, 512], default 0

**use\_alpha**

Uses the alpha channel information in the image

**Type** boolean, default False

**use\_calculate\_alpha**

Calculates an alpha channel based on RGB values in the image

**Type** boolean, default False

**use\_checker\_even**

Sets even checker tiles

**Type** boolean, default False

**use\_checker\_odd**

Sets odd checker tiles

**Type** boolean, default False

**use\_derivative\_map**

Uses red and green as derivative values

**Type** boolean, default False

**use\_filter\_size\_min**

Use Filter Size as a minimal filter value in pixels

**Type** boolean, default False

**use\_flip\_axis**

Flips the texture's X and Y axis



**Type** boolean, default False

**use\_interpolation**

Interpolates pixels using selected filter

**Type** boolean, default False

**use\_mipmap**

Uses auto-generated MIP maps for the image

**Type** boolean, default False

**use\_mipmap\_gauss**

Use Gauss filter to sample down MIP maps

**Type** boolean, default False

**use\_mirror\_x**

Mirrors the image repetition on the X direction

**Type** boolean, default False

**use\_mirror\_y**

Mirrors the image repetition on the Y direction

**Type** boolean, default False

**use\_normal\_map**

Uses image RGB values for normal mapping

**Type** boolean, default False

**users\_material**

Materials that use this texture (readonly)

**users\_object\_modifier**

Object modifiers that use this texture (readonly)

### Inherited Properties

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`
- `Texture.animation_data`
- `Texture.intensity`
- `Texture.color_ramp`
- `Texture.contrast`
- `Texture.factor_blue`
- `Texture.factor_green`
- `Texture.factor_red`
- `Texture.node_tree`
- `Texture.saturation`
- `Texture.use_preview_alpha`
- `Texture.type`
- `Texture.use_color_ramp`

- `Texture.use_nodes`
- `Texture.users_material`
- `Texture.users_object_modifier`
- `Texture.users_material`
- `Texture.users_object_modifier`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`
- `Texture.evaluate`

## 2.4.261 ImageUser(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.ImageUser` (*bpy\_struct*)

Parameters defining how an Image datablock is used by another datablock

**fields\_per\_frame**

Number of fields per rendered frame (2 fields is 1 image)

**Type** int in [1, 200], default 0

**frame\_duration**

Number of images of a movie to use

**Type** int in [0, 300000], default 0

**frame\_offset**

Offset the number of the frame to use in the animation

**Type** int in [-300000, 300000], default 0

**frame\_start**

Global starting frame of the movie/sequence, assuming first picture has a #1

**Type** int in [-300000, 300000], default 0

**multilayer\_layer**

Layer in multilayer image

**Type** int in [0, 32767], default 0, (readonly)**multilayer\_pass**

Pass in multilayer image

**Type** int in [0, 32767], default 0, (readonly)**use\_auto\_refresh**

Always refresh image on frame changes

**Type** boolean, default False**use\_cyclic**

Cycle the images in the movie

**Type** boolean, default False**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `BackgroundImage.image_user`
- `EnvironmentMapTexture.image_user`
- `ImageTexture.image_user`
- `SpaceImageEditor.image_user`
- `UILayout.template_image`
- `UILayout.template_image_layers`
- `VoxelDataTexture.image_user`

## 2.4.262 InflowFluidSettings(FluidSettings)

base classes — `bpy_struct`, `FluidSettings`

**class** `bpy.types.InflowFluidSettings` (*FluidSettings*)

Fluid simulation settings for objects adding fluids in the simulation

**inflow\_velocity**

Initial velocity of fluid

**Type** float array of 3 items in [-1000.1, 1000.1], default (0.0, 0.0, 0.0)

**use**

Object contributes to the fluid simulation

**Type** boolean, default False

**use\_animated\_mesh**

Export this mesh as an animated one. Slower, only use if really necessary (e.g. armatures or parented objects), animated pos/rot/scale IPOs do not require it

**Type** boolean, default False

**use\_local\_coords**

Use local coordinates for inflow. (e.g. for rotating objects)

**Type** boolean, default False

**volume\_initialization**

Volume initialization type

- **VOLUME** Volume, Use only the inner volume of the mesh.
- **SHELL** Shell, Use only the outer shell of the mesh.
- **BOTH** Both, Use both the inner volume and the outer shell of the mesh.

**Type** enum in ['VOLUME', 'SHELL', 'BOTH'], default 'VOLUME'

### Inherited Properties

- `bpy_struct.id_data`
- `FluidSettings.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`

- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.263 IntProperty(Property)

base classes — `bpy_struct`, `Property`

**class** `bpy.types.IntProperty` (*Property*)

RNA integer number property definition

**array\_length**

Maximum length of the array, 0 means unlimited

**Type** `int` in `[0, inf]`, default 0, (readonly)

**default**

Default value for this number

**Type** `int` in `[-inf, inf]`, default 0, (readonly)

**default\_array**

Default value for this array

**Type** `int` array of 3 items in `[-inf, inf]`, default (0, 0, 0), (readonly)

**hard\_max**

Maximum value used by buttons

**Type** `int` in `[-inf, inf]`, default 0, (readonly)

**hard\_min**

Minimum value used by buttons

**Type** `int` in `[-inf, inf]`, default 0, (readonly)

**soft\_max**

Maximum value used by buttons

**Type** `int` in `[-inf, inf]`, default 0, (readonly)

**soft\_min**

Minimum value used by buttons

**Type** `int` in `[-inf, inf]`, default 0, (readonly)

**step**

Step size used by number buttons, for floats 1/100th of the step size

**Type** `int` in `[0, inf]`, default 0, (readonly)

#### Inherited Properties

- `bpy_struct.id_data`
- `Property.name`
- `Property.is_animatable`
- `Property.srna`
- `Property.description`
- `Property.is_enum_flag`
- `Property.is_hidden`
- `Property.identifier`

- `Property.is_never_none`
- `Property.is_readonly`
- `Property.is_registered`
- `Property.is_registered_optional`
- `Property.is_required`
- `Property.is_output`
- `Property.is_runtime`
- `Property.is_skip_save`
- `Property.subtype`
- `Property.type`
- `Property.unit`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.264 Itasc(IKParam)

base classes — `bpy_struct`, `IKParam`

**class** `bpy.types.Itasc` (*IKParam*)  
Parameters for the iTaSC IK solver

### **damping\_epsilon**

Singular value under which damping is progressively applied (higher values=more stability, less reactivity - default=0.1)

**Type** float in [0, 1], default 0.0

### **damping\_max**

Maximum damping coefficient when singular value is nearly 0 (higher values=more stability, less reactivity - default=0.5)

**Type** float in [0, 1], default 0.0

### **feedback**

Feedback coefficient for error correction. Average response time=1/feedback. Default=20

**Type** float in [0, 100], default 0.0

**iterations**

Maximum number of iterations for convergence in case of reiteration

**Type** int in [1, 1000], default 0

**mode**

- **ANIMATION** Animation, Stateless solver computing pose starting from current action and non-IK constraints.
- **SIMULATION** Simulation, Statefull solver running in real-time context and ignoring actions and non-IK constraints.

**Type** enum in ['ANIMATION', 'SIMULATION'], default 'ANIMATION'

**precision**

Precision of convergence in case of reiteration

**Type** float in [0, 0.1], default 0.0

**reiteration\_method**

Defines if the solver is allowed to reiterate (converges until precision is met) on none, first or all frames

- **NEVER** Never, The solver does not reiterate, not even on first frame (starts from rest pose).
- **INITIAL** Initial, The solver reiterates (converges) on the first frame but not on subsequent frame.
- **ALWAYS** Always, The solver reiterates (converges) on all frames.

**Type** enum in ['NEVER', 'INITIAL', 'ALWAYS'], default 'NEVER'

**solver**

Solving method selection: Automatic damping or manual damping

- **SDLS** SDLS, Selective Damped Least Square.
- **DLS** DLS, Damped Least Square with Numerical Filtering.

**Type** enum in ['SDLS', 'DLS'], default 'SDLS'

**step\_count**

Divides the frame interval into this many steps

**Type** int in [1, 50], default 0

**step\_max**

Higher bound for timestep in second in case of automatic substeps

**Type** float in [0, 1], default 0.0

**step\_min**

Lower bound for timestep in second in case of automatic substeps

**Type** float in [0, 0.1], default 0.0

**use\_auto\_step**

Automatically determine the optimal number of steps for best performance/accuracy trade off

**Type** boolean, default False

**velocity\_max**

Maximum joint velocity in rad/s. Default=50

**Type** float in [0, 100], default 0.0

### Inherited Properties

- `bpy_struct.id_data`
- `IKParam.ik_solver`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.265 JoystickSensor(Sensor)

base classes — `bpy_struct`, `Sensor`

**class** `bpy.types.JoystickSensor` (*Sensor*)

Sensor to detect joystick events

**axis\_direction**

The direction of the axis

**Type** enum in ['RIGHTAXIS', 'UPAXIS', 'LEFTAXIS', 'DOWNAXIS'], default 'RIGHTAXIS'

**axis\_number**

Specify which axis pair to use, 1 is usually the main direction input

**Type** int in [1, 8], default 0

**axis\_threshold**

Specify the precision of the axis

**Type** int in [0, 32768], default 0

**button\_number**

Specify which button to use

**Type** int in [0, 18], default 0

**event\_type**

The type of event this joystick sensor is triggered on

**Type** enum in ['BUTTON', 'AXIS', 'HAT', 'AXIS\_SINGLE'], default 'BUTTON'



**hat\_direction**

Specify hat direction

**Type** enum in ['UP', 'DOWN', 'LEFT', 'RIGHT', 'UPRIGHT', 'DOWNLEFT', 'UPLEFT', 'DOWNRIGHT'], default 'UP'**hat\_number**

Specify which hat to use

**Type** int in [1, 2], default 0**joystick\_index**

Specify which joystick to use

**Type** int in [0, 7], default 0**single\_axis\_number**

Specify a single axis (verticle/horizontal/other) to detect

**Type** int in [1, 16], default 0**use\_all\_events**

Triggered by all events on this joysticks current type (axis/button/hat)

**Type** boolean, default False**Inherited Properties**

- `bpy_struct.id_data`
- `Sensor.name`
- `Sensor.show_expanded`
- `Sensor.frequency`
- `Sensor.invert`
- `Sensor.use_level`
- `Sensor.pin`
- `Sensor.use_pulse_false_level`
- `Sensor.use_pulse_true_level`
- `Sensor.use_tap`
- `Sensor.type`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`

- `bpy_struct.values`
- `Sensor.link`
- `Sensor.unlink`

## 2.4.266 Key(ID)

base classes — `bpy_struct`, `ID`

**class** `bpy.types.Key` (*ID*)

Shape keys datablock containing different shapes of geometric datablocks

### **animation\_data**

Animation data for this datablock

**Type** `AnimData`, (readonly)

### **key\_blocks**

Shape keys

**Type** `bpy_prop_collection` of `ShapeKey`, (readonly)

### **reference\_key**

**Type** `ShapeKey`, (readonly, never `None`)

### **slurph**

Creates a delay in amount of frames in applying keypositions, first vertex goes first

**Type** `int` in `[-500, 500]`, default `0`

### **use\_relative**

Makes shape keys relative

**Type** `boolean`, default `False`

### **user**

Datablock using these shape keys

**Type** `ID`, (readonly, never `None`)

## Inherited Properties

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`

- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`

## References

- `BlendData.shape_keys`
- `Curve.shape_keys`
- `Lattice.shape_keys`
- `Mesh.shape_keys`

### 2.4.267 KeyConfig(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.KeyConfig` (*bpy\_struct*)

Input configuration, including keymaps

**is\_user\_defined**

Indicates that a keyconfig was defined by the user

**Type** boolean, default False, (readonly)

**keymaps**

Key maps configured as part of this configuration

**Type** `KeyMaps` `bpy_prop_collection` of `KeyMap`, (readonly)

**name**

Name of the key configuration

**Type** string, default ""

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`

- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `KeyConfigurations.active`
- `KeyConfigurations.addon`
- `KeyConfigurations.default`
- `KeyConfigurations.new`
- `KeyConfigurations.remove`
- `KeyConfigurations.user`
- `WindowManager.keyconfigs`

## 2.4.268 KeyConfigurations(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.KeyConfigurations` (*bpy\_struct*)

Collection of KeyConfigs

### **active**

Active key configuration (preset)

**Type** `KeyConfig`

### **addon**

Key configuration that can be extended by addons, and is added to the active configuration when handling events

**Type** `KeyConfig`, (readonly)

### **default**

Default builtin key configuration

**Type** `KeyConfig`, (readonly)

### **user**

Final key configuration that combines keymaps from the active and addon configurations, and can be edited by the user

**Type** `KeyConfig`, (readonly)

### **new** (*name*)

new

**Parameters** `name` (*string*) – Name

**Returns** Key Configuration, Added key configuration

**Return type** `KeyConfig`

**remove** (*keyconfig*)  
remove

**Parameters** **keyconfig** (`KeyConfig`) – Key Configuration, Removed key configuration

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `WindowManager.keyconfigs`

## 2.4.269 KeyMap(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.KeyMap` (*bpy\_struct*)  
Input configuration, including keymaps

**is\_modal**

Indicates that a keymap is used for translate modal events for an operator

**Type** boolean, default False, (readonly)

**is\_user\_modified**

Keymap is defined by the user

**Type** boolean, default False

**keymap\_items**

Items in the keymap, linking an operator to an input event

**Type** `KeyMapItems bpy_prop_collection` of `KeyMapItem`, (readonly)

**name**

Name of the key map

**Type** `string`, default `""`, (readonly)

**region\_type**

Optional region type keymap is associated with

**Type** `enum` in `['WINDOW', 'HEADER', 'CHANNELS', 'TEMPORARY', 'UI', 'TOOLS', 'TOOL_PROPS', 'PREVIEW']`, default `'WINDOW'`, (readonly)

**show\_expanded\_children**

Children expanded in the user interface

**Type** `boolean`, default `False`

**show\_expanded\_items**

Expanded in the user interface

**Type** `boolean`, default `False`

**space\_type**

Optional space type keymap is associated with

**Type** `enum` in `['EMPTY', 'VIEW_3D', 'GRAPH_EDITOR', 'OUTLINER', 'PROPERTIES', 'FILE_BROWSER', 'IMAGE_EDITOR', 'INFO', 'SEQUENCE_EDITOR', 'TEXT_EDITOR', 'DOPESHEET_EDITOR', 'NLA_EDITOR', 'TIMELINE', 'NODE_EDITOR', 'LOGIC_EDITOR', 'CONSOLE', 'USER_PREFERENCES']`, default `'EMPTY'`, (readonly)

**active ()**

active

**Returns** `Key Map`, Active key map

**Return type** `KeyMap`

**restore\_to\_default ()**

`restore_to_default`

**restore\_item\_to\_default (item)**

`restore_item_to_default`

**Parameters** `item` (`KeyMapItem`, (never `None`)) – Item

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`

- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `KeyConfig.keymaps`
- `KeyMap.active`
- `KeyMaps.find`
- `KeyMaps.find_modal`
- `KeyMaps.new`

### 2.4.270 KeyMapItem(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.KeyMapItem` (*bpy\_struct*)  
Item in a Key Map

#### **active**

Activate or deactivate item

**Type** boolean, default False

#### **alt**

Alt key pressed

**Type** boolean, default False

#### **any**

Any modifier keys pressed

**Type** boolean, default False

#### **ctrl**

Control key pressed

**Type** boolean, default False

#### **id**

ID of the item

**Type** int in [-32768, 32767], default 0, (readonly)

#### **idname**

Identifier of operator to call on input event

**Type** string, default ""

#### **is\_user\_defined**

Is this keymap item user defined (doesn't just replace a builtin item)

**Type** boolean, default False, (readonly)

**is\_user\_modified**

Is this keymap item modified by the user

**Type** boolean, default False, (readonly)

**key\_modifier**

Regular key pressed as a modifier

**Type** enum in ['NONE', 'LEFTMOUSE', 'MIDDLEMOUSE', 'RIGHTMOUSE', 'BUTTON4MOUSE', 'BUTTON5MOUSE', 'ACTIONMOUSE', 'SELECTMOUSE', 'MOUSEMOVE', 'INBETWEEN\_MOUSEMOVE', 'TRACKPADPAN', 'TRACKPADZOOM', 'MOUSEROTATE', 'WHEELUPMOUSE', 'WHEELDOWNMOUSE', 'WHEELINMOUSE', 'WHEELOUTMOUSE', 'EVT\_TWEAK\_L', 'EVT\_TWEAK\_M', 'EVT\_TWEAK\_R', 'EVT\_TWEAK\_A', 'EVT\_TWEAK\_S', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z', 'ZERO', 'ONE', 'TWO', 'THREE', 'FOUR', 'FIVE', 'SIX', 'SEVEN', 'EIGHT', 'NINE', 'LEFT\_CTRL', 'LEFT\_ALT', 'LEFT\_SHIFT', 'RIGHT\_ALT', 'RIGHT\_CTRL', 'RIGHT\_SHIFT', 'OSKEY', 'GRLESS', 'ESC', 'TAB', 'RET', 'SPACE', 'LINE\_FEED', 'BACK\_SPACE', 'DEL', 'SEMI\_COLON', 'PERIOD', 'COMMA', 'QUOTE', 'ACCENT\_GRAVE', 'MINUS', 'SLASH', 'BACK\_SLASH', 'EQUAL', 'LEFT\_BRACKET', 'RIGHT\_BRACKET', 'LEFT\_ARROW', 'DOWN\_ARROW', 'RIGHT\_ARROW', 'UP\_ARROW', 'NUMPAD\_2', 'NUMPAD\_4', 'NUMPAD\_6', 'NUMPAD\_8', 'NUMPAD\_1', 'NUMPAD\_3', 'NUMPAD\_5', 'NUMPAD\_7', 'NUMPAD\_9', 'NUMPAD\_PERIOD', 'NUMPAD\_SLASH', 'NUMPAD\_ASTERIX', 'NUMPAD\_0', 'NUMPAD\_MINUS', 'NUMPAD\_ENTER', 'NUMPAD\_PLUS', 'F1', 'F2', 'F3', 'F4', 'F5', 'F6', 'F7', 'F8', 'F9', 'F10', 'F11', 'F12', 'F13', 'F14', 'F15', 'F16', 'F17', 'F18', 'F19', 'PAUSE', 'INSERT', 'HOME', 'PAGE\_UP', 'PAGE\_DOWN', 'END', 'MEDIA\_PLAY', 'MEDIA\_STOP', 'MEDIA\_FIRST', 'MEDIA\_LAST', 'WINDOW\_DEACTIVATE', 'TIMER', 'TIMER0', 'TIMER1', 'TIMER2', 'NDOF\_BUTTON\_MENU', 'NDOF\_BUTTON\_FIT', 'NDOF\_BUTTON\_TOP', 'NDOF\_BUTTON\_BOTTOM', 'NDOF\_BUTTON\_LEFT', 'NDOF\_BUTTON\_RIGHT', 'NDOF\_BUTTON\_FRONT', 'NDOF\_BUTTON\_BACK', 'NDOF\_BUTTON\_ISO1', 'NDOF\_BUTTON\_ISO2', 'NDOF\_BUTTON\_ROLL\_CW', 'NDOF\_BUTTON\_ROLL\_CCW', 'NDOF\_BUTTON\_SPIN\_CW', 'NDOF\_BUTTON\_SPIN\_CCW', 'NDOF\_BUTTON\_TILT\_CW', 'NDOF\_BUTTON\_TILT\_CCW', 'NDOF\_BUTTON\_ROTATE', 'NDOF\_BUTTON\_PANZOOM', 'NDOF\_BUTTON\_DOMINANT', 'NDOF\_BUTTON\_PLUS', 'NDOF\_BUTTON\_MINUS', 'NDOF\_BUTTON\_1', 'NDOF\_BUTTON\_2', 'NDOF\_BUTTON\_3', 'NDOF\_BUTTON\_4', 'NDOF\_BUTTON\_5', 'NDOF\_BUTTON\_6', 'NDOF\_BUTTON\_7', 'NDOF\_BUTTON\_8', 'NDOF\_BUTTON\_9', 'NDOF\_BUTTON\_10'], default 'NONE'

**map\_type**

Type of event mapping

**Type** enum in ['KEYBOARD', 'TWEAK', 'MOUSE', 'NDOF', 'TEXTINPUT', 'TIMER'], default 'KEYBOARD'

**name**

Name of operator to call on input event

**Type** string, default "", (readonly)

**oskey**

Operating system key pressed

**Type** boolean, default False



**properties**

Properties to set when the operator is called

**Type** `OperatorProperties`, (readonly)

**propvalue**

The value this event translates to in a modal keymap

**Type** enum in ['NONE'], default 'NONE'

**shift**

Shift key pressed

**Type** boolean, default False

**show\_expanded**

Show key map event and property details in the user interface

**Type** boolean, default False

**type**

Type of event

**Type** enum in ['NONE', 'LEFTMOUSE', 'MIDDLEMOUSE', 'RIGHTMOUSE', 'BUTTON4MOUSE', 'BUTTON5MOUSE', 'ACTIONMOUSE', 'SELECTMOUSE', 'MOUSEMOVE', 'INBETWEEN\_MOUSEMOVE', 'TRACKPADPAN', 'TRACKPADZOOM', 'MOUSEROTATE', 'WHEELUPMOUSE', 'WHEELDOWNMOUSE', 'WHEELINMOUSE', 'WHEELOUTMOUSE', 'EVT\_TWEAK\_L', 'EVT\_TWEAK\_M', 'EVT\_TWEAK\_R', 'EVT\_TWEAK\_A', 'EVT\_TWEAK\_S', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z', 'ZERO', 'ONE', 'TWO', 'THREE', 'FOUR', 'FIVE', 'SIX', 'SEVEN', 'EIGHT', 'NINE', 'LEFT\_CTRL', 'LEFT\_ALT', 'LEFT\_SHIFT', 'RIGHT\_ALT', 'RIGHT\_CTRL', 'RIGHT\_SHIFT', 'OSKEY', 'GRLESS', 'ESC', 'TAB', 'RET', 'SPACE', 'LINE\_FEED', 'BACK\_SPACE', 'DEL', 'SEMI\_COLON', 'PERIOD', 'COMMA', 'QUOTE', 'ACCENT\_GRAVE', 'MINUS', 'SLASH', 'BACK\_SLASH', 'EQUAL', 'LEFT\_BRACKET', 'RIGHT\_BRACKET', 'LEFT\_ARROW', 'DOWN\_ARROW', 'RIGHT\_ARROW', 'UP\_ARROW', 'NUMPAD\_2', 'NUMPAD\_4', 'NUMPAD\_6', 'NUMPAD\_8', 'NUMPAD\_1', 'NUMPAD\_3', 'NUMPAD\_5', 'NUMPAD\_7', 'NUMPAD\_9', 'NUMPAD\_PERIOD', 'NUMPAD\_SLASH', 'NUMPAD\_ASTERIX', 'NUMPAD\_0', 'NUMPAD\_MINUS', 'NUMPAD\_ENTER', 'NUMPAD\_PLUS', 'F1', 'F2', 'F3', 'F4', 'F5', 'F6', 'F7', 'F8', 'F9', 'F10', 'F11', 'F12', 'F13', 'F14', 'F15', 'F16', 'F17', 'F18', 'F19', 'PAUSE', 'INSERT', 'HOME', 'PAGE\_UP', 'PAGE\_DOWN', 'END', 'MEDIA\_PLAY', 'MEDIA\_STOP', 'MEDIA\_FIRST', 'MEDIA\_LAST', 'WINDOW\_DEACTIVATE', 'TIMER', 'TIMER0', 'TIMER1', 'TIMER2', 'NDOF\_BUTTON\_MENU', 'NDOF\_BUTTON\_FIT', 'NDOF\_BUTTON\_TOP', 'NDOF\_BUTTON\_BOTTOM', 'NDOF\_BUTTON\_LEFT', 'NDOF\_BUTTON\_RIGHT', 'NDOF\_BUTTON\_FRONT', 'NDOF\_BUTTON\_BACK', 'NDOF\_BUTTON\_ISO1', 'NDOF\_BUTTON\_ISO2', 'NDOF\_BUTTON\_ROLL\_CW', 'NDOF\_BUTTON\_ROLL\_CCW', 'NDOF\_BUTTON\_SPIN\_CW', 'NDOF\_BUTTON\_SPIN\_CCW', 'NDOF\_BUTTON\_TILT\_CW', 'NDOF\_BUTTON\_TILT\_CCW', 'NDOF\_BUTTON\_ROTATE', 'NDOF\_BUTTON\_PANZOOM', 'NDOF\_BUTTON\_DOMINANT', 'NDOF\_BUTTON\_PLUS', 'NDOF\_BUTTON\_MINUS', 'NDOF\_BUTTON\_1', 'NDOF\_BUTTON\_2', 'NDOF\_BUTTON\_3', 'NDOF\_BUTTON\_4', 'NDOF\_BUTTON\_5', 'NDOF\_BUTTON\_6', 'NDOF\_BUTTON\_7', 'NDOF\_BUTTON\_8', 'NDOF\_BUTTON\_9', 'NDOF\_BUTTON\_10'], default 'NONE'

**value**

**Type** enum in ['ANY', 'NOTHING', 'PRESS', 'RELEASE', 'CLICK', 'DOUBLE\_CLICK'],  
default 'NOTHING'

**compare** (*item*)

compare

**Parameters** *item* (`KeyMapItem`) – Item

**Returns** Comparison result

**Return type** boolean

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `KeyMap.keymap_items`
- `KeyMap.restore_item_to_default`
- `KeyMapItem.compare`
- `KeyMapItems.from_id`
- `KeyMapItems.new`
- `KeyMapItems.new_modal`
- `KeyMapItems.remove`
- `UILayout.template_keymap_item_properties`

## 2.4.271 KeyMapItems(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.KeyMapItems` (*bpy\_struct*)  
Collection of keymap items

**new** (*idname*, *type*, *value*, *any=False*, *shift=False*, *ctrl=False*, *alt=False*, *oskey=False*,  
*key\_modifier='NONE'*)  
 new

### Parameters

- **idname** (*string*) – Operator Identifier
- **type** (*enum in ['NONE', 'LEFTMOUSE', 'MIDDLEMOUSE', 'RIGHTMOUSE', 'BUTTON4MOUSE', 'BUTTON5MOUSE', 'ACTIONMOUSE', 'SELECTMOUSE', 'MOUSEMOVE', 'INBETWEEN\_MOUSEMOVE', 'TRACKPADPAN', 'TRACKPADZOOM', 'MOUSEROTATE', 'WHEELUPMOUSE', 'WHEELDOWNMOUSE', 'WHEELINMOUSE', 'WHEELOUTMOUSE', 'EVT\_TWEAK\_L', 'EVT\_TWEAK\_M', 'EVT\_TWEAK\_R', 'EVT\_TWEAK\_A', 'EVT\_TWEAK\_S', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z', 'ZERO', 'ONE', 'TWO', 'THREE', 'FOUR', 'FIVE', 'SIX', 'SEVEN', 'EIGHT', 'NINE', 'LEFT\_CTRL', 'LEFT\_ALT', 'LEFT\_SHIFT', 'RIGHT\_ALT', 'RIGHT\_CTRL', 'RIGHT\_SHIFT', 'OSKEY', 'GRLESS', 'ESC', 'TAB', 'RET', 'SPACE', 'LINE\_FEED', 'BACK\_SPACE', 'DEL', 'SEMI\_COLON', 'PERIOD', 'COMMA', 'QUOTE', 'ACCENT\_GRAVE', 'MINUS', 'SLASH', 'BACK\_SLASH', 'EQUAL', 'LEFT\_BRACKET', 'RIGHT\_BRACKET', 'LEFT\_ARROW', 'DOWN\_ARROW', 'RIGHT\_ARROW', 'UP\_ARROW', 'NUMPAD\_2', 'NUMPAD\_4', 'NUMPAD\_6', 'NUMPAD\_8', 'NUMPAD\_1', 'NUMPAD\_3', 'NUMPAD\_5', 'NUMPAD\_7', 'NUMPAD\_9', 'NUMPAD\_PERIOD', 'NUMPAD\_SLASH', 'NUMPAD\_ASTERIX', 'NUMPAD\_0', 'NUMPAD\_MINUS', 'NUMPAD\_ENTER', 'NUMPAD\_PLUS', 'F1', 'F2', 'F3', 'F4', 'F5', 'F6', 'F7', 'F8', 'F9', 'F10', 'F11', 'F12', 'F13', 'F14', 'F15', 'F16', 'F17', 'F18', 'F19', 'PAUSE', 'INSERT', 'HOME', 'PAGE\_UP', 'PAGE\_DOWN', 'END', 'MEDIA\_PLAY', 'MEDIA\_STOP', 'MEDIA\_FIRST', 'MEDIA\_LAST', 'WINDOW\_DEACTIVATE', 'TIMER', 'TIMER0', 'TIMER1', 'TIMER2', 'NDOF\_BUTTON\_MENU', 'NDOF\_BUTTON\_FIT', 'NDOF\_BUTTON\_TOP', 'NDOF\_BUTTON\_BOTTOM', 'NDOF\_BUTTON\_LEFT', 'NDOF\_BUTTON\_RIGHT', 'NDOF\_BUTTON\_FRONT', 'NDOF\_BUTTON\_BACK', 'NDOF\_BUTTON\_ISO1', 'NDOF\_BUTTON\_ISO2', 'NDOF\_BUTTON\_ROLL\_CW', 'NDOF\_BUTTON\_ROLL\_CCW', 'NDOF\_BUTTON\_SPIN\_CW', 'NDOF\_BUTTON\_SPIN\_CCW', 'NDOF\_BUTTON\_TILT\_CW', 'NDOF\_BUTTON\_TILT\_CCW', 'NDOF\_BUTTON\_ROTATE', 'NDOF\_BUTTON\_PANZOOM', 'NDOF\_BUTTON\_DOMINANT', 'NDOF\_BUTTON\_PLUS', 'NDOF\_BUTTON\_MINUS', 'NDOF\_BUTTON\_1', 'NDOF\_BUTTON\_2', 'NDOF\_BUTTON\_3', 'NDOF\_BUTTON\_4', 'NDOF\_BUTTON\_5', 'NDOF\_BUTTON\_6', 'NDOF\_BUTTON\_7', 'NDOF\_BUTTON\_8', 'NDOF\_BUTTON\_9', 'NDOF\_BUTTON\_10']*) – Type
- **value** (*enum in ['ANY', 'NOTHING', 'PRESS', 'RELEASE', 'CLICK', 'DOUBLE\_CLICK']*) – Value
- **any** (*boolean, (optional)*) – Any
- **shift** (*boolean, (optional)*) – Shift
- **ctrl** (*boolean, (optional)*) – Ctrl
- **alt** (*boolean, (optional)*) – Alt
- **oskey** (*boolean, (optional)*) – OS Key
- **key\_modifier** (*enum in ['NONE', 'LEFTMOUSE', 'MIDDLEMOUSE', 'RIGHTMOUSE', 'BUTTON4MOUSE', 'BUTTON5MOUSE', 'ACTIONMOUSE', 'SELECTMOUSE', 'MOUSEMOVE', 'INBETWEEN\_MOUSEMOVE', 'TRACKPADPAN', 'TRACKPADZOOM', 'MOUSEROTATE', 'WHEELUPMOUSE', 'WHEEL-*

DOWNMOUSE', 'WHEELINMOUSE', 'WHEELOUTMOUSE', 'EVT\_TWEAK\_L', 'EVT\_TWEAK\_M', 'EVT\_TWEAK\_R', 'EVT\_TWEAK\_A', 'EVT\_TWEAK\_S', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z', 'ZERO', 'ONE', 'TWO', 'THREE', 'FOUR', 'FIVE', 'SIX', 'SEVEN', 'EIGHT', 'NINE', 'LEFT\_CTRL', 'LEFT\_ALT', 'LEFT\_SHIFT', 'RIGHT\_ALT', 'RIGHT\_CTRL', 'RIGHT\_SHIFT', 'OSKEY', 'GRLESS', 'ESC', 'TAB', 'RET', 'SPACE', 'LINE\_FEED', 'BACK\_SPACE', 'DEL', 'SEMI\_COLON', 'PERIOD', 'COMMA', 'QUOTE', 'ACCENT\_GRAVE', 'MINUS', 'SLASH', 'BACK\_SLASH', 'EQUAL', 'LEFT\_BRACKET', 'RIGHT\_BRACKET', 'LEFT\_ARROW', 'DOWN\_ARROW', 'RIGHT\_ARROW', 'UP\_ARROW', 'NUMPAD\_2', 'NUMPAD\_4', 'NUMPAD\_6', 'NUMPAD\_8', 'NUMPAD\_1', 'NUMPAD\_3', 'NUMPAD\_5', 'NUMPAD\_7', 'NUMPAD\_9', 'NUMPAD\_PERIOD', 'NUMPAD\_SLASH', 'NUMPAD\_ASTERIX', 'NUMPAD\_0', 'NUMPAD\_MINUS', 'NUMPAD\_ENTER', 'NUMPAD\_PLUS', 'F1', 'F2', 'F3', 'F4', 'F5', 'F6', 'F7', 'F8', 'F9', 'F10', 'F11', 'F12', 'F13', 'F14', 'F15', 'F16', 'F17', 'F18', 'F19', 'PAUSE', 'INSERT', 'HOME', 'PAGE\_UP', 'PAGE\_DOWN', 'END', 'MEDIA\_PLAY', 'MEDIA\_STOP', 'MEDIA\_FIRST', 'MEDIA\_LAST', 'WINDOW\_DEACTIVATE', 'TIMER', 'TIMER0', 'TIMER1', 'TIMER2', 'NDOF\_BUTTON\_MENU', 'NDOF\_BUTTON\_FIT', 'NDOF\_BUTTON\_TOP', 'NDOF\_BUTTON\_BOTTOM', 'NDOF\_BUTTON\_LEFT', 'NDOF\_BUTTON\_RIGHT', 'NDOF\_BUTTON\_FRONT', 'NDOF\_BUTTON\_BACK', 'NDOF\_BUTTON\_ISO1', 'NDOF\_BUTTON\_ISO2', 'NDOF\_BUTTON\_ROLL\_CW', 'NDOF\_BUTTON\_ROLL\_CCW', 'NDOF\_BUTTON\_SPIN\_CW', 'NDOF\_BUTTON\_SPIN\_CCW', 'NDOF\_BUTTON\_TILT\_CW', 'NDOF\_BUTTON\_TILT\_CCW', 'NDOF\_BUTTON\_ROTATE', 'NDOF\_BUTTON\_PANZOOM', 'NDOF\_BUTTON\_DOMINANT', 'NDOF\_BUTTON\_PLUS', 'NDOF\_BUTTON\_MINUS', 'NDOF\_BUTTON\_1', 'NDOF\_BUTTON\_2', 'NDOF\_BUTTON\_3', 'NDOF\_BUTTON\_4', 'NDOF\_BUTTON\_5', 'NDOF\_BUTTON\_6', 'NDOF\_BUTTON\_7', 'NDOF\_BUTTON\_8', 'NDOF\_BUTTON\_9', 'NDOF\_BUTTON\_10'], (optional)) – Key Modifier

**Returns** Item, Added key map item

**Return type** `KeyMapItem`

**new\_modal** (*propvalue*, *type*, *value*, *any=False*, *shift=False*, *ctrl=False*, *alt=False*, *oskey=False*, *key\_modifier='NONE'*)  
new\_modal

#### Parameters

- **propvalue** (*string*) – Property Value
- **type** (*enum in* [`'NONE'`, `'LEFTMOUSE'`, `'MIDDLEMOUSE'`, `'RIGHTMOUSE'`, `'BUTTON4MOUSE'`, `'BUTTON5MOUSE'`, `'ACTIONMOUSE'`, `'SELECTMOUSE'`, `'MOUSEMOVE'`, `'INBETWEEN_MOUSEMOVE'`, `'TRACKPADPAN'`, `'TRACKPADZOOM'`, `'MOUSEROTATE'`, `'WHEELUPMOUSE'`, `'WHEELDOWNMOUSE'`, `'WHEELINMOUSE'`, `'WHEELOUTMOUSE'`, `'EVT_TWEAK_L'`, `'EVT_TWEAK_M'`, `'EVT_TWEAK_R'`, `'EVT_TWEAK_A'`, `'EVT_TWEAK_S'`, `'A'`, `'B'`, `'C'`, `'D'`, `'E'`, `'F'`, `'G'`, `'H'`, `'I'`, `'J'`, `'K'`, `'L'`, `'M'`, `'N'`, `'O'`, `'P'`, `'Q'`, `'R'`, `'S'`, `'T'`, `'U'`, `'V'`, `'W'`, `'X'`, `'Y'`, `'Z'`, `'ZERO'`, `'ONE'`, `'TWO'`, `'THREE'`, `'FOUR'`, `'FIVE'`, `'SIX'`, `'SEVEN'`, `'EIGHT'`, `'NINE'`, `'LEFT_CTRL'`, `'LEFT_ALT'`, `'LEFT_SHIFT'`, `'RIGHT_ALT'`, `'RIGHT_CTRL'`, `'RIGHT_SHIFT'`, `'OSKEY'`, `'GRLESS'`, `'ESC'`, `'TAB'`, `'RET'`, `'SPACE'`, `'LINE_FEED'`, `'BACK_SPACE'`, `'DEL'`, `'SEMI_COLON'`, `'PERIOD'`, `'COMMA'`, `'QUOTE'`, `'ACCENT_GRAVE'`, `'MINUS'`, `'SLASH'`, `'BACK_SLASH'`, `'EQUAL'`, `'LEFT_BRACKET'`, `'RIGHT_BRACKET'`, `'LEFT_ARROW'`, `'DOWN_ARROW'`, `'RIGHT_ARROW'`, `'UP_ARROW'`, `'NUMPAD_2'`, `'NUMPAD_4'`, `'NUMPAD_6'`,

'NUMPAD\_8', 'NUMPAD\_1', 'NUMPAD\_3', 'NUMPAD\_5', 'NUMPAD\_7',  
 'NUMPAD\_9', 'NUMPAD\_PERIOD', 'NUMPAD\_SLASH', 'NUMPAD\_ASTERIX',  
 'NUMPAD\_0', 'NUMPAD\_MINUS', 'NUMPAD\_ENTER', 'NUMPAD\_PLUS',  
 'F1', 'F2', 'F3', 'F4', 'F5', 'F6', 'F7', 'F8', 'F9', 'F10', 'F11', 'F12',  
 'F13', 'F14', 'F15', 'F16', 'F17', 'F18', 'F19', 'PAUSE', 'INSERT', 'HOME',  
 'PAGE\_UP', 'PAGE\_DOWN', 'END', 'MEDIA\_PLAY', 'MEDIA\_STOP', 'ME-  
 DIA\_FIRST', 'MEDIA\_LAST', 'WINDOW\_DEACTIVATE', 'TIMER', 'TIMER0',  
 'TIMER1', 'TIMER2', 'NDOF\_BUTTON\_MENU', 'NDOF\_BUTTON\_FIT',  
 'NDOF\_BUTTON\_TOP', 'NDOF\_BUTTON\_BOTTOM', 'NDOF\_BUTTON\_LEFT',  
 'NDOF\_BUTTON\_RIGHT', 'NDOF\_BUTTON\_FRONT', 'NDOF\_BUTTON\_BACK',  
 'NDOF\_BUTTON\_ISO1', 'NDOF\_BUTTON\_ISO2', 'NDOF\_BUTTON\_ROLL\_CW',  
 'NDOF\_BUTTON\_ROLL\_CCW', 'NDOF\_BUTTON\_SPIN\_CW',  
 'NDOF\_BUTTON\_SPIN\_CCW', 'NDOF\_BUTTON\_TILT\_CW',  
 'NDOF\_BUTTON\_TILT\_CCW', 'NDOF\_BUTTON\_ROTATE',  
 'NDOF\_BUTTON\_PANZOOM', 'NDOF\_BUTTON\_DOMINANT',  
 'NDOF\_BUTTON\_PLUS', 'NDOF\_BUTTON\_MINUS', 'NDOF\_BUTTON\_1',  
 'NDOF\_BUTTON\_2', 'NDOF\_BUTTON\_3', 'NDOF\_BUTTON\_4',  
 'NDOF\_BUTTON\_5', 'NDOF\_BUTTON\_6', 'NDOF\_BUTTON\_7',  
 'NDOF\_BUTTON\_8', 'NDOF\_BUTTON\_9', 'NDOF\_BUTTON\_10'}) – Type

- **value** (enum in ['ANY', 'NOTHING', 'PRESS', 'RELEASE', 'CLICK', 'DOUBLE\_CLICK']) – Value
- **any** (boolean, (optional)) – Any
- **shift** (boolean, (optional)) – Shift
- **ctrl** (boolean, (optional)) – Ctrl
- **alt** (boolean, (optional)) – Alt
- **oskey** (boolean, (optional)) – OS Key
- **key\_modifier** (enum in ['NONE', 'LEFTMOUSE', 'MIDDLEMOUSE', 'RIGHTMOUSE', 'BUTTON4MOUSE', 'BUTTON5MOUSE', 'ACTIONMOUSE', 'SELECTMOUSE', 'MOUSEMOVE', 'INBETWEEN\_MOUSEMOVE', 'TRACKPADPAN', 'TRACKPADZOOM', 'MOUSEROTATE', 'WHEELUPMOUSE', 'WHEELDOWNMOUSE', 'WHEELINMOUSE', 'WHEELOUTMOUSE', 'EVT\_TWEAK\_L', 'EVT\_TWEAK\_M', 'EVT\_TWEAK\_R', 'EVT\_TWEAK\_A', 'EVT\_TWEAK\_S', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z', 'ZERO', 'ONE', 'TWO', 'THREE', 'FOUR', 'FIVE', 'SIX', 'SEVEN', 'EIGHT', 'NINE', 'LEFT\_CTRL', 'LEFT\_ALT', 'LEFT\_SHIFT', 'RIGHT\_ALT', 'RIGHT\_CTRL', 'RIGHT\_SHIFT', 'OSKEY', 'GRLESS', 'ESC', 'TAB', 'RET', 'SPACE', 'LINE\_FEED', 'BACK\_SPACE', 'DEL', 'SEMI\_COLON', 'PERIOD', 'COMMA', 'QUOTE', 'ACCENT\_GRAVE', 'MINUS', 'SLASH', 'BACK\_SLASH', 'EQUAL', 'LEFT\_BRACKET', 'RIGHT\_BRACKET', 'LEFT\_ARROW', 'DOWN\_ARROW', 'RIGHT\_ARROW', 'UP\_ARROW', 'NUMPAD\_2', 'NUMPAD\_4', 'NUMPAD\_6', 'NUMPAD\_8', 'NUMPAD\_1', 'NUMPAD\_3', 'NUMPAD\_5', 'NUMPAD\_7', 'NUMPAD\_9', 'NUMPAD\_PERIOD', 'NUMPAD\_SLASH', 'NUMPAD\_ASTERIX', 'NUMPAD\_0', 'NUMPAD\_MINUS', 'NUMPAD\_ENTER', 'NUMPAD\_PLUS', 'F1', 'F2', 'F3', 'F4', 'F5', 'F6', 'F7', 'F8', 'F9', 'F10', 'F11', 'F12', 'F13', 'F14', 'F15', 'F16', 'F17', 'F18', 'F19', 'PAUSE', 'INSERT', 'HOME', 'PAGE\_UP', 'PAGE\_DOWN', 'END', 'MEDIA\_PLAY', 'MEDIA\_STOP', 'MEDIA\_FIRST', 'MEDIA\_LAST', 'WINDOW\_DEACTIVATE', 'TIMER', 'TIMER0', 'TIMER1', 'TIMER2', 'NDOF\_BUTTON\_MENU', 'NDOF\_BUTTON\_FIT', 'NDOF\_BUTTON\_TOP', 'NDOF\_BUTTON\_BOTTOM', 'NDOF\_BUTTON\_LEFT', 'NDOF\_BUTTON\_RIGHT', 'NDOF\_BUTTON\_FRONT', 'NDOF\_BUTTON\_BACK',

`'NDOF_BUTTON_ISO1', 'NDOF_BUTTON_ISO2', 'NDOF_BUTTON_ROLL_CW',  
'NDOF_BUTTON_ROLL_CCW', 'NDOF_BUTTON_SPIN_CW',  
'NDOF_BUTTON_SPIN_CCW', 'NDOF_BUTTON_TILT_CW',  
'NDOF_BUTTON_TILT_CCW', 'NDOF_BUTTON_ROTATE',  
'NDOF_BUTTON_PANZOOM', 'NDOF_BUTTON_DOMINANT',  
'NDOF_BUTTON_PLUS', 'NDOF_BUTTON_MINUS', 'NDOF_BUTTON_1',  
'NDOF_BUTTON_2', 'NDOF_BUTTON_3', 'NDOF_BUTTON_4',  
'NDOF_BUTTON_5', 'NDOF_BUTTON_6', 'NDOF_BUTTON_7',  
'NDOF_BUTTON_8', 'NDOF_BUTTON_9', 'NDOF_BUTTON_10']`, (optional) –  
Key Modifier

**Returns** Item, Added key map item

**Return type** `KeyMapItem`

**remove** (*item*)  
remove

**Parameters** *item* (`KeyMapItem`) – Item

**from\_id** (*id*)  
from\_id

**Parameters** *id* (*int in [-inf, inf]*) – id, ID of the item

**Returns** Item

**Return type** `KeyMapItem`

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `KeyMap.keymap_items`

## 2.4.272 KeyMaps(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.KeyMaps` (*bpy\_struct*)

Collection of keymaps

**new** (*name*, *space\_type*='EMPTY', *region\_type*='WINDOW', *modal*=False)  
new

### Parameters

- **name** (*string*) – Name
- **space\_type** (*enum* in [`'EMPTY'`, `'VIEW_3D'`, `'GRAPH_EDITOR'`, `'OUTLINER'`, `'PROPERTIES'`, `'FILE_BROWSER'`, `'IMAGE_EDITOR'`, `'INFO'`, `'SEQUENCE_EDITOR'`, `'TEXT_EDITOR'`, `'DOPESHEET_EDITOR'`, `'NLA_EDITOR'`, `'TIMELINE'`, `'NODE_EDITOR'`, `'LOGIC_EDITOR'`, `'CONSOLE'`, `'USER_PREFERENCES'`], (*optional*)) – Space Type
- **region\_type** (*enum* in [`'WINDOW'`, `'HEADER'`, `'CHANNELS'`, `'TEMPORARY'`, `'UI'`, `'TOOLS'`, `'TOOL_PROPS'`, `'PREVIEW'`], (*optional*)) – Region Type
- **modal** (*boolean*, (*optional*)) – Modal

**Returns** Key Map, Added key map

**Return type** `KeyMap`

**find** (*name*, *space\_type*='EMPTY', *region\_type*='WINDOW')  
find

### Parameters

- **name** (*string*) – Name
- **space\_type** (*enum* in [`'EMPTY'`, `'VIEW_3D'`, `'GRAPH_EDITOR'`, `'OUTLINER'`, `'PROPERTIES'`, `'FILE_BROWSER'`, `'IMAGE_EDITOR'`, `'INFO'`, `'SEQUENCE_EDITOR'`, `'TEXT_EDITOR'`, `'DOPESHEET_EDITOR'`, `'NLA_EDITOR'`, `'TIMELINE'`, `'NODE_EDITOR'`, `'LOGIC_EDITOR'`, `'CONSOLE'`, `'USER_PREFERENCES'`], (*optional*)) – Space Type
- **region\_type** (*enum* in [`'WINDOW'`, `'HEADER'`, `'CHANNELS'`, `'TEMPORARY'`, `'UI'`, `'TOOLS'`, `'TOOL_PROPS'`, `'PREVIEW'`], (*optional*)) – Region Type

**Returns** Key Map, Corresponding key map

**Return type** `KeyMap`

**find\_modal** (*name*)  
find\_modal

**Parameters** **name** (*string*) – Operator Name

**Returns** Key Map, Corresponding key map

**Return type** `KeyMap`

### Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `KeyConfig.keymaps`

### 2.4.273 KeyboardSensor(Sensor)

base classes — `bpy_struct`, `Sensor`

**class** `bpy.types.KeyboardSensor` (*Sensor*)  
Sensor to detect keyboard events

#### **key**

**Type** enum in ['NONE', 'LEFTMOUSE', 'MIDDLEMOUSE', 'RIGHTMOUSE', 'BUTTON4MOUSE', 'BUTTON5MOUSE', 'ACTIONMOUSE', 'SELECTMOUSE', 'MOUSEMOVE', 'INBETWEEN\_MOUSEMOVE', 'TRACKPADPAN', 'TRACKPADZOOM', 'MOUSEROTATE', 'WHEELUPMOUSE', 'WHEELDOWNMOUSE', 'WHEELINMOUSE', 'WHEELOUTMOUSE', 'EVT\_TWEAK\_L', 'EVT\_TWEAK\_M', 'EVT\_TWEAK\_R', 'EVT\_TWEAK\_A', 'EVT\_TWEAK\_S', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z', 'ZERO', 'ONE', 'TWO', 'THREE', 'FOUR', 'FIVE', 'SIX', 'SEVEN', 'EIGHT', 'NINE', 'LEFT\_CTRL', 'LEFT\_ALT', 'LEFT\_SHIFT', 'RIGHT\_ALT', 'RIGHT\_CTRL', 'RIGHT\_SHIFT', 'OSKEY', 'GRLESS', 'ESC', 'TAB', 'RET', 'SPACE', 'LINE\_FEED', 'BACK\_SPACE', 'DEL', 'SEMI\_COLON', 'PERIOD', 'COMMA', 'QUOTE', 'ACCENT\_GRAVE', 'MINUS', 'SLASH', 'BACK\_SLASH', 'EQUAL', 'LEFT\_BRACKET', 'RIGHT\_BRACKET', 'LEFT\_ARROW', 'DOWN\_ARROW', 'RIGHT\_ARROW', 'UP\_ARROW', 'NUMPAD\_2', 'NUMPAD\_4', 'NUMPAD\_6', 'NUMPAD\_8', 'NUMPAD\_1', 'NUMPAD\_3', 'NUMPAD\_5', 'NUMPAD\_7', 'NUMPAD\_9', 'NUMPAD\_PERIOD', 'NUMPAD\_SLASH', 'NUMPAD\_ASTERIX', 'NUMPAD\_0', 'NUMPAD\_MINUS', 'NUMPAD\_ENTER', 'NUMPAD\_PLUS', 'F1', 'F2', 'F3', 'F4', 'F5', 'F6', 'F7', 'F8', 'F9', 'F10', 'F11', 'F12', 'F13', 'F14', 'F15', 'F16', 'F17', 'F18', 'F19', 'PAUSE', 'INSERT', 'HOME', 'PAGE\_UP', 'PAGE\_DOWN', 'END', 'MEDIA\_PLAY', 'MEDIA\_STOP', 'MEDIA\_FIRST', 'MEDIA\_LAST', 'WINDOW\_DEACTIVATE', 'TIMER', 'TIMER0',



```

'TIMER1', 'TIMER2', 'NDOF_BUTTON_MENU', 'NDOF_BUTTON_FIT',
'NDOF_BUTTON_TOP', 'NDOF_BUTTON_BOTTOM', 'NDOF_BUTTON_LEFT',
'NDOF_BUTTON_RIGHT', 'NDOF_BUTTON_FRONT', 'NDOF_BUTTON_BACK',
'NDOF_BUTTON_ISO1', 'NDOF_BUTTON_ISO2', 'NDOF_BUTTON_ROLL_CW',
'NDOF_BUTTON_ROLL_CCW', 'NDOF_BUTTON_SPIN_CW',
'NDOF_BUTTON_SPIN_CCW', 'NDOF_BUTTON_TILT_CW',
'NDOF_BUTTON_TILT_CCW', 'NDOF_BUTTON_ROTATE',
'NDOF_BUTTON_PANZOOM', 'NDOF_BUTTON_DOMINANT',
'NDOF_BUTTON_PLUS', 'NDOF_BUTTON_MINUS', 'NDOF_BUTTON_1',
'NDOF_BUTTON_2', 'NDOF_BUTTON_3', 'NDOF_BUTTON_4',
'NDOF_BUTTON_5', 'NDOF_BUTTON_6', 'NDOF_BUTTON_7',
'NDOF_BUTTON_8', 'NDOF_BUTTON_9', 'NDOF_BUTTON_10'], default 'NONE'

```

**log**

Property that indicates whether to log keystrokes as a string

**Type** string, default ""

**modifier\_key\_1**

Modifier key code

**Type** enum in ['NONE', 'LEFTMOUSE', 'MIDDLEMOUSE', 'RIGHTMOUSE', 'BUTTON4MOUSE', 'BUTTON5MOUSE', 'ACTIONMOUSE', 'SELECTMOUSE', 'MOUSEMOVE', 'INBETWEEN\_MOUSEMOVE', 'TRACKPADPAN', 'TRACKPADZOOM', 'MOUSEROTATE', 'WHEELUPMOUSE', 'WHEELDOWNMOUSE', 'WHEELINMOUSE', 'WHEELOUTMOUSE', 'EVT\_TWEAK\_L', 'EVT\_TWEAK\_M', 'EVT\_TWEAK\_R', 'EVT\_TWEAK\_A', 'EVT\_TWEAK\_S', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z', 'ZERO', 'ONE', 'TWO', 'THREE', 'FOUR', 'FIVE', 'SIX', 'SEVEN', 'EIGHT', 'NINE', 'LEFT\_CTRL', 'LEFT\_ALT', 'LEFT\_SHIFT', 'RIGHT\_ALT', 'RIGHT\_CTRL', 'RIGHT\_SHIFT', 'OSKEY', 'GRLESS', 'ESC', 'TAB', 'RET', 'SPACE', 'LINE\_FEED', 'BACK\_SPACE', 'DEL', 'SEMI\_COLON', 'PERIOD', 'COMMA', 'QUOTE', 'ACCENT\_GRAVE', 'MINUS', 'SLASH', 'BACK\_SLASH', 'EQUAL', 'LEFT\_BRACKET', 'RIGHT\_BRACKET', 'LEFT\_ARROW', 'DOWN\_ARROW', 'RIGHT\_ARROW', 'UP\_ARROW', 'NUMPAD\_2', 'NUMPAD\_4', 'NUMPAD\_6', 'NUMPAD\_8', 'NUMPAD\_1', 'NUMPAD\_3', 'NUMPAD\_5', 'NUMPAD\_7', 'NUMPAD\_9', 'NUMPAD\_PERIOD', 'NUMPAD\_SLASH', 'NUMPAD\_ASTERIX', 'NUMPAD\_0', 'NUMPAD\_MINUS', 'NUMPAD\_ENTER', 'NUMPAD\_PLUS', 'F1', 'F2', 'F3', 'F4', 'F5', 'F6', 'F7', 'F8', 'F9', 'F10', 'F11', 'F12', 'F13', 'F14', 'F15', 'F16', 'F17', 'F18', 'F19', 'PAUSE', 'INSERT', 'HOME', 'PAGE\_UP', 'PAGE\_DOWN', 'END', 'MEDIA\_PLAY', 'MEDIA\_STOP', 'MEDIA\_FIRST', 'MEDIA\_LAST', 'WINDOW\_DEACTIVATE', 'TIMER', 'TIMER0', 'TIMER1', 'TIMER2', 'NDOF\_BUTTON\_MENU', 'NDOF\_BUTTON\_FIT', 'NDOF\_BUTTON\_TOP', 'NDOF\_BUTTON\_BOTTOM', 'NDOF\_BUTTON\_LEFT', 'NDOF\_BUTTON\_RIGHT', 'NDOF\_BUTTON\_FRONT', 'NDOF\_BUTTON\_BACK', 'NDOF\_BUTTON\_ISO1', 'NDOF\_BUTTON\_ISO2', 'NDOF\_BUTTON\_ROLL\_CW', 'NDOF\_BUTTON\_ROLL\_CCW', 'NDOF\_BUTTON\_SPIN\_CW', 'NDOF\_BUTTON\_SPIN\_CCW', 'NDOF\_BUTTON\_TILT\_CW', 'NDOF\_BUTTON\_TILT\_CCW', 'NDOF\_BUTTON\_ROTATE', 'NDOF\_BUTTON\_PANZOOM', 'NDOF\_BUTTON\_DOMINANT', 'NDOF\_BUTTON\_PLUS', 'NDOF\_BUTTON\_MINUS', 'NDOF\_BUTTON\_1', 'NDOF\_BUTTON\_2', 'NDOF\_BUTTON\_3', 'NDOF\_BUTTON\_4', 'NDOF\_BUTTON\_5', 'NDOF\_BUTTON\_6', 'NDOF\_BUTTON\_7', 'NDOF\_BUTTON\_8', 'NDOF\_BUTTON\_9', 'NDOF\_BUTTON\_10'], default 'NONE'

**modifier\_key\_2**

Modifier key code

**Type** enum in ['NONE', 'LEFTMOUSE', 'MIDDLEMOUSE', 'RIGHTMOUSE', 'BUTTON4MOUSE', 'BUTTON5MOUSE', 'ACTIONMOUSE', 'SELECTMOUSE', 'MOUSEMOVE', 'INBETWEEN\_MOUSEMOVE', 'TRACKPADPAN', 'TRACKPADZOOM', 'MOUSEROTATE', 'WHEELUPMOUSE', 'WHEELDOWNMOUSE', 'WHEELINMOUSE', 'WHEELOUTMOUSE', 'EVT\_TWEAK\_L', 'EVT\_TWEAK\_M', 'EVT\_TWEAK\_R', 'EVT\_TWEAK\_A', 'EVT\_TWEAK\_S', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z', 'ZERO', 'ONE', 'TWO', 'THREE', 'FOUR', 'FIVE', 'SIX', 'SEVEN', 'EIGHT', 'NINE', 'LEFT\_CTRL', 'LEFT\_ALT', 'LEFT\_SHIFT', 'RIGHT\_ALT', 'RIGHT\_CTRL', 'RIGHT\_SHIFT', 'OSKEY', 'GRLESS', 'ESC', 'TAB', 'RET', 'SPACE', 'LINE\_FEED', 'BACK\_SPACE', 'DEL', 'SEMI\_COLON', 'PERIOD', 'COMMA', 'QUOTE', 'ACCENT\_GRAVE', 'MINUS', 'SLASH', 'BACK\_SLASH', 'EQUAL', 'LEFT\_BRACKET', 'RIGHT\_BRACKET', 'LEFT\_ARROW', 'DOWN\_ARROW', 'RIGHT\_ARROW', 'UP\_ARROW', 'NUMPAD\_2', 'NUMPAD\_4', 'NUMPAD\_6', 'NUMPAD\_8', 'NUMPAD\_1', 'NUMPAD\_3', 'NUMPAD\_5', 'NUMPAD\_7', 'NUMPAD\_9', 'NUMPAD\_PERIOD', 'NUMPAD\_SLASH', 'NUMPAD\_ASTERIX', 'NUMPAD\_0', 'NUMPAD\_MINUS', 'NUMPAD\_ENTER', 'NUMPAD\_PLUS', 'F1', 'F2', 'F3', 'F4', 'F5', 'F6', 'F7', 'F8', 'F9', 'F10', 'F11', 'F12', 'F13', 'F14', 'F15', 'F16', 'F17', 'F18', 'F19', 'PAUSE', 'INSERT', 'HOME', 'PAGE\_UP', 'PAGE\_DOWN', 'END', 'MEDIA\_PLAY', 'MEDIA\_STOP', 'MEDIA\_FIRST', 'MEDIA\_LAST', 'WINDOW\_DEACTIVATE', 'TIMER', 'TIMER0', 'TIMER1', 'TIMER2', 'NDOF\_BUTTON\_MENU', 'NDOF\_BUTTON\_FIT', 'NDOF\_BUTTON\_TOP', 'NDOF\_BUTTON\_BOTTOM', 'NDOF\_BUTTON\_LEFT', 'NDOF\_BUTTON\_RIGHT', 'NDOF\_BUTTON\_FRONT', 'NDOF\_BUTTON\_BACK', 'NDOF\_BUTTON\_ISO1', 'NDOF\_BUTTON\_ISO2', 'NDOF\_BUTTON\_ROLL\_CW', 'NDOF\_BUTTON\_ROLL\_CCW', 'NDOF\_BUTTON\_SPIN\_CW', 'NDOF\_BUTTON\_SPIN\_CCW', 'NDOF\_BUTTON\_TILT\_CW', 'NDOF\_BUTTON\_TILT\_CCW', 'NDOF\_BUTTON\_ROTATE', 'NDOF\_BUTTON\_PANZOOM', 'NDOF\_BUTTON\_DOMINANT', 'NDOF\_BUTTON\_PLUS', 'NDOF\_BUTTON\_MINUS', 'NDOF\_BUTTON\_1', 'NDOF\_BUTTON\_2', 'NDOF\_BUTTON\_3', 'NDOF\_BUTTON\_4', 'NDOF\_BUTTON\_5', 'NDOF\_BUTTON\_6', 'NDOF\_BUTTON\_7', 'NDOF\_BUTTON\_8', 'NDOF\_BUTTON\_9', 'NDOF\_BUTTON\_10'], default 'NONE'

#### target

Property that receive the keystrokes in case a string is logged

**Type** string, default ""

#### use\_all\_keys

Trigger this sensor on any keystroke

**Type** boolean, default False

#### Inherited Properties

- bpy\_struct.id\_data
- Sensor.name
- Sensor.show\_expanded
- Sensor.frequency
- Sensor.invert
- Sensor.use\_level
- Sensor.pin

- `Sensor.use_pulse_false_level`
- `Sensor.use_pulse_true_level`
- `Sensor.use_tap`
- `Sensor.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Sensor.link`
- `Sensor.unlink`

## 2.4.274 Keyframe(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.Keyframe` (*bpy\_struct*)

Bezier curve point with two handles defining a Keyframe on an F-Curve

**co**

Coordinates of the control point

**Type** float array of 2 items in `[-inf, inf]`, default `(0.0, 0.0)`

**handle\_left**

Coordinates of the left handle (before the control point)

**Type** float array of 2 items in `[-inf, inf]`, default `(0.0, 0.0)`

**handle\_left\_type**

Handle types

- `FREE` Free.
- `VECTOR` Vector.
- `ALIGNED` Aligned.
- `AUTO` Automatic.
- `AUTO_CLAMPED` Auto Clamped, Auto handles clamped to not overshoot.

**Type** enum in `['FREE', 'VECTOR', 'ALIGNED', 'AUTO', 'AUTO_CLAMPED']`, default `'FREE'`

**handle\_right**

Coordinates of the right handle (after the control point)

**Type** float array of 2 items in [-inf, inf], default (0.0, 0.0)

**handle\_right\_type**

Handle types

- FREE Free.
- VECTOR Vector.
- ALIGNED Aligned.
- AUTO Automatic.
- AUTO\_CLAMPED Auto Clamped, Auto handles clamped to not overshoot.

**Type** enum in ['FREE', 'VECTOR', 'ALIGNED', 'AUTO', 'AUTO\_CLAMPED'], default 'FREE'

**interpolation**

Interpolation method to use for segment of the F-Curve from this Keyframe until the next Keyframe

**Type** enum in ['CONSTANT', 'LINEAR', 'BEZIER'], default 'CONSTANT'

**select\_control\_point**

Control point selection status

**Type** boolean, default False

**select\_left\_handle**

Left handle selection status

**Type** boolean, default False

**select\_right\_handle**

Right handle selection status

**Type** boolean, default False

**type**

Type of keyframe (for visual purposes only)

**Type** enum in ['KEYFRAME', 'BREAKDOWN', 'EXTREME', 'JITTER'], default 'KEYFRAME'

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`

- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `FCurve.keyframe_points`
- `FCurveKeyframePoints.insert`
- `FCurveKeyframePoints.remove`

### 2.4.275 KeyingSet(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.KeyingSet` (*bpy\_struct*)  
Settings that should be keyframed together

#### **bl\_options**

Keying set options

- `INSERTKEY_NEEDED` Insert Keyframes - Only Needed, Only insert keyframes where they're needed in the relevant F-Curves.
- `INSERTKEY_VISUAL` Insert Keyframes - Visual, Insert keyframes based on 'visual transforms'.
- `INSERTKEY_XYZ_TO_RGB` F-Curve Colors - XYZ to RGB, Color for newly added transformation F-Curves (Location, Rotation, Scale) and also Color is based on the transform axis.

**Type** enum set in `{'INSERTKEY_NEEDED', 'INSERTKEY_VISUAL', 'INSERTKEY_XYZ_TO_RGB'}`, default `{'INSERTKEY_NEEDED'}`

#### **is\_path\_absolute**

Keying Set defines specific paths/settings to be keyframed (i.e. is not reliant on context info)

**Type** boolean, default `False`, (readonly)

#### **name**

**Type** string, default `""`

#### **paths**

Keying Set Paths to define settings that get keyframed together

**Type** `KeyingSetPaths` `bpy_prop_collection` of `KeyingSetPath`, (readonly)

#### **type\_info**

Callback function defines for built-in Keying Sets

**Type** `KeyingSetInfo`, (readonly)

#### **refresh()**

Refresh Keying Set to ensure that it is valid for the current context. Call before each use of one

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `KeyingSetInfo.generate`
- `KeyingSetInfo.iterator`
- `KeyingSets.active`
- `KeyingSets.new`
- `KeyingSetsAll.active`
- `Scene.keying_sets`
- `Scene.keying_sets_all`

## 2.4.276 KeyingSetInfo(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.KeyingSetInfo` (*bpy\_struct*)  
Callback function defines for builtin Keying Sets

**bl\_idname**

**Type** string, default ""

**bl\_label**

**Type** string, default ""

**bl\_options**

Keying set options

- `INSERTKEY_NEEDED` Insert Keyframes - Only Needed, Only insert keyframes where they're needed in the relevant F-Curves.
- `INSERTKEY_VISUAL` Insert Keyframes - Visual, Insert keyframes based on 'visual transforms'.

- `INSERTKEY_XYZ_TO_RGB` F-Curve Colors - XYZ to RGB, Color for newly added transformation F-Curves (Location, Rotation, Scale) and also Color is based on the transform axis.

**Type** enum set in {'INSERTKEY\_NEEDED', 'INSERTKEY\_VISUAL', 'INSERTKEY\_XYZ\_TO\_RGB'}, default {'INSERTKEY\_NEEDED'}

**poll** (*context*)

Test if Keying Set can be used or not

**Return type** boolean

**iterator** (*context, ks*)

Call generate() on the structs which have properties to be keyframed

**generate** (*context, ks, data*)

Add Paths to the Keying Set to keyframe the properties of the given data

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `KeyingSet.type_info`

## 2.4.277 KeyingSetPath(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.KeyingSetPath` (*bpy\_struct*)

Path to a setting for use in a Keying Set

**array\_index**

Index to the specific setting if applicable

**Type** int in [-inf, inf], default 0

**bl\_options**

Keying set options

- **INSERTKEY\_NEEDED** Insert Keyframes - Only Needed, Only insert keyframes where they're needed in the relevant F-Curves.
- **INSERTKEY\_VISUAL** Insert Keyframes - Visual, Insert keyframes based on 'visual transforms'.
- **INSERTKEY\_XYZ\_TO\_RGB** F-Curve Colors - XYZ to RGB, Color for newly added transformation F-Curves (Location, Rotation, Scale) and also Color is based on the transform axis.

**Type** enum set in {'INSERTKEY\_NEEDED', 'INSERTKEY\_VISUAL', 'INSERTKEY\_XYZ\_TO\_RGB'}, default {'INSERTKEY\_NEEDED'}

**data\_path**

Path to property setting

**Type** string, default ""

**group**

Name of Action Group to assign setting(s) for this path to

**Type** string, default ""

**group\_method**

Method used to define which Group-name to use

**Type** enum in ['NAMED', 'NONE', 'KEYINGSET'], default 'NAMED'

**id**

ID-Block that keyframes for Keying Set should be added to (for Absolute Keying Sets only)

**Type** ID

**id\_type**

Type of ID-block that can be used

**Type** enum in ['ACTION', 'ARMATURE', 'BRUSH', 'CAMERA', 'CURVE', 'FONT', 'GREASEPENCIL', 'GROUP', 'IMAGE', 'KEY', 'LAMP', 'LIBRARY', 'LATTICE', 'MATERIAL', 'META', 'MESH', 'NODETREE', 'OBJECT', 'PARTICLE', 'SCENE', 'SCREEN', 'SPEAKER', 'SOUND', 'TEXT', 'TEXTURE', 'WORLD', 'WINDOWMANAGER'], default 'OBJECT'

**use\_entire\_array**

When an 'array/vector' type is chosen (Location, Rotation, Color, etc.), entire array is to be used

**Type** boolean, default False

**Inherited Properties**

- `bpy_struct.id_data`



## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `KeyingSet.paths`
- `KeyingSetPaths.active`
- `KeyingSetPaths.add`
- `KeyingSetPaths.remove`

### 2.4.278 KeyingSetPaths(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.KeyingSetPaths` (*bpy\_struct*)

Collection of keying set paths

#### **active**

Active Keying Set used to insert/delete keyframes

**Type** `KeyingSetPath`

#### **active\_index**

Current Keying Set index

**Type** `int` in `[-inf, inf]`, default 0

**add** (*target\_id*, *data\_path*, *index=-1*, *group\_method='KEYINGSET'*, *group\_name=""*)

Add a new path for the Keying Set

#### **Parameters**

- **target\_id** (*ID*) – Target ID, ID-Datablock for the destination
- **data\_path** (*string*) – Data-Path, RNA-Path to destination property
- **index** (*int* in `[-1, inf]`, (*optional*)) – Index, The index of the destination property (i.e. axis of Location/Rotation/etc.), or -1 for the entire array
- **group\_method** (*enum* in `['NAMED', 'NONE', 'KEYINGSET']`, (*optional*)) – Grouping Method, Method used to define which Group-name to use

- **group\_name** (*string, (optional)*) – Group Name, Name of Action Group to assign destination to (only if grouping mode is to use this name)

**Returns** New Path, Path created and added to the Keying Set

**Return type** `KeyingSetPath`

**remove** (*path*)

Remove the given path from the Keying Set

**Parameters** **path** (`KeyingSetPath`, (never None)) – Path

**clear** ()

Remove all the paths from the Keying Set

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `KeyingSet.paths`

## 2.4.279 KeyingSets(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.KeyingSets` (*bpy\_struct*)

Scene keying sets

**active**

Active Keying Set used to insert/delete keyframes

**Type** `KeyingSet`

**active\_index**

Current Keying Set index (negative for ‘builtin’ and positive for ‘absolute’)

**Type** int in [-inf, inf], default 0

**new** (*name*="KeyingSet")

Add a new Keying Set to Scene

**Parameters** *name* (*string*, (*optional*)) – Name, Name of Keying Set

**Returns** Newly created Keying Set

**Return type** KeyingSet

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `Scene.keying_sets`

**2.4.280 KeyingSetsAll(bpy\_struct)**

base class — `bpy_struct`

**class** `bpy.types.KeyingSetsAll` (*bpy\_struct*)

All available keying sets

**active**

Active Keying Set used to insert/delete keyframes

**Type** KeyingSet

**active\_index**

Current Keying Set index (negative for ‘builtin’ and positive for ‘absolute’)

**Type** int in [-inf, inf], default 0

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Scene.keying_sets_all`

## 2.4.281 KinematicConstraint(Constraint)

base classes — `bpy_struct`, `Constraint`

**class** `bpy.types.KinematicConstraint` (*Constraint*)

Inverse Kinematics

#### **chain\_count**

How many bones are included in the IK effect - 0 uses all bones

**Type** int in [0, 255], default 0

#### **distance**

Radius of limiting sphere

**Type** float in [0, 100], default 0.0

#### **ik\_type**

**Type** enum in ['COPY\_POSE', 'DISTANCE'], default 'COPY\_POSE'

#### **iterations**

Maximum number of solving iterations

**Type** int in [1, 10000], default 0

**limit\_mode**

Distances in relation to sphere of influence to allow

- **LIMITDIST\_INSIDE** Inside, The object is constrained inside a virtual sphere around the target object, with a radius defined by the limit distance.
- **LIMITDIST\_OUTSIDE** Outside, The object is constrained outside a virtual sphere around the target object, with a radius defined by the limit distance.
- **LIMITDIST\_ONSURFACE** On Surface, The object is constrained on the surface of a virtual sphere around the target object, with a radius defined by the limit distance.

**Type** enum in ['LIMITDIST\_INSIDE', 'LIMITDIST\_OUTSIDE', 'LIMITDIST\_ONSURFACE'], default 'LIMITDIST\_INSIDE'

**lock\_location\_x**

Constraint position along X axis

**Type** boolean, default False

**lock\_location\_y**

Constraint position along Y axis

**Type** boolean, default False

**lock\_location\_z**

Constraint position along Z axis

**Type** boolean, default False

**lock\_rotation\_x**

Constraint rotation along X axis

**Type** boolean, default False

**lock\_rotation\_y**

Constraint rotation along Y axis

**Type** boolean, default False

**lock\_rotation\_z**

Constraint rotation along Z axis

**Type** boolean, default False

**orient\_weight**

For Tree-IK: Weight of orientation control for this target

**Type** float in [0.01, 1], default 0.0

**pole\_angle**

Pole rotation offset

**Type** float in [-3.14159, 3.14159], default 0.0

**pole\_subtarget**

**Type** string, default ""

**pole\_target**

Object for pole rotation

**Type** [Object](#)

**reference\_axis**

Constraint axis Lock options relative to Bone or Target reference

**Type** enum in ['BONE', 'TARGET'], default 'BONE'

**subtarget**

**Type** string, default ""

**target**

Target Object

**Type** Object

**use\_location**

Chain follows position of target

**Type** boolean, default False

**use\_rotation**

Chain follows rotation of target

**Type** boolean, default False

**use\_stretch**

Enable IK Stretching

**Type** boolean, default False

**use\_tail**

Include bone's tail as last element in chain

**Type** boolean, default False

**use\_target**

Disable for targetless IK

**Type** boolean, default False

**weight**

For Tree-IK: Weight of position control for this target

**Type** float in [0.01, 1], default 0.0

**Inherited Properties**

- bpy\_struct.id\_data
- Constraint.name
- Constraint.active
- Constraint.mute
- Constraint.show\_expanded
- Constraint.influence
- Constraint.error\_location
- Constraint.owner\_space
- Constraint.is\_proxy\_local
- Constraint.error\_rotation
- Constraint.target\_space
- Constraint.type
- Constraint.is\_valid

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.282 Lamp(ID)

base classes — `bpy_struct`, `ID`

subclasses — `SpotLamp`, `AreaLamp`, `HemiLamp`, `PointLamp`, `SunLamp`

**class** `bpy.types.Lamp` (*ID*)

Lamp datablock for lighting a scene

**active\_texture**

Active texture slot being displayed

**Type** `Texture`

**active\_texture\_index**

Index of active texture slot

**Type** `int` in `[0, 17]`, default `0`

**animation\_data**

Animation data for this datablock

**Type** `AnimData`, (readonly)

**color**

Light color

**Type** `float` array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`

**distance**

Falloff distance - the light is at half the original intensity at this point

**Type** `float` in `[0, inf]`, default `0.0`

**energy**

Amount of light that the lamp emits

**Type** `float` in `[-inf, inf]`, default `0.0`

**node\_tree**

Node tree for node based lamps

**Type** `NodeTree`, (readonly)

**texture\_slots**

Texture slots defining the mapping and influence of textures

**Type** `LampTextureSlots` `bpy_prop_collection` of `LampTextureSlot`, (readonly)

**type**

Type of Lamp

- `POINT` Point, Omnidirectional point light source.
- `SUN` Sun, Constant direction parallel ray light source.
- `SPOT` Spot, Directional cone light source.
- `HEMI` Hemi, 180 degree constant light source.
- `AREA` Area, Directional area light source.

**Type** enum in [`'POINT'`, `'SUN'`, `'SPOT'`, `'HEMI'`, `'AREA'`], default `'POINT'`

**use\_diffuse**

Lamp does diffuse shading

**Type** boolean, default `False`

**use\_negative**

Lamp casts negative light

**Type** boolean, default `False`

**use\_nodes**

Use shader nodes to render the lamp

**Type** boolean, default `False`

**use\_own\_layer**

Illuminates objects only on the same layer the lamp is on

**Type** boolean, default `False`

**use\_specular**

Lamp creates specular highlights

**Type** boolean, default `False`

**Inherited Properties**

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`



## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`

## References

- `BlendData.lamps`
- `BlendDataLamps.new`
- `BlendDataLamps.remove`

### 2.4.283 LampSkySettings(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.LampSkySettings` (*bpy\_struct*)

Sky related settings for a sun lamp

**atmosphere\_distance\_factor**

Multiplier to convert blender units to physical distance

**Type** float in [0, 500], default 0.0

**atmosphere\_extinction**

Extinction scattering contribution factor

**Type** float in [0, 1], default 0.0

**atmosphere\_inscattering**

Scatter contribution factor

**Type** float in [0, 1], default 0.0

**atmosphere\_turbidity**

Sky turbidity

**Type** float in [1, 30], default 0.0

**backscattered\_light**

Backscattered light

**Type** float in [-1, 1], default 0.0

**horizon\_brightness**

Horizon brightness

**Type** float in [0, 20], default 0.0

**sky\_blend**

Blend factor with sky

**Type** float in [0, 2], default 0.0

**sky\_blend\_type**

Blend mode for combining sun sky with world sky

**Type** enum in ['MIX', 'ADD', 'MULTIPLY', 'SUBTRACT', 'SCREEN', 'DIVIDE', 'DIFFERENCE', 'DARKEN', 'LIGHTEN', 'OVERLAY', 'DODGE', 'BURN', 'HUE', 'SATURATION', 'VALUE', 'COLOR', 'SOFT\_LIGHT', 'LINEAR\_LIGHT'], default 'MIX'

**sky\_color\_space**

Color space to use for internal XYZ->RGB color conversion

**Type** enum in ['SMPTE', 'REC709', 'CIE'], default 'SMPTE'

**sky\_exposure**

Strength of sky shading exponential exposure correction

**Type** float in [0, 20], default 0.0

**spread**

Horizon Spread

**Type** float in [0, 10], default 0.0

**sun\_brightness**

Sun brightness

**Type** float in [0, 10], default 0.0

**sun\_intensity**

Sun intensity

**Type** float in [0, 10], default 0.0

**sun\_size**

Sun size

**Type** float in [0, 10], default 0.0

**use\_atmosphere**

Apply sun effect on atmosphere

**Type** boolean, default False

**use\_sky**

Apply sun effect on sky

**Type** boolean, default False

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `SunLamp.sky`

## 2.4.284 LampTextureSlot(TextureSlot)

base classes — `bpy_struct`, `TextureSlot`

**class** `bpy.types.LampTextureSlot` (*TextureSlot*)

Texture slot for textures in a Lamp datablock

### **color\_factor**

Amount texture affects color values

**Type** float in [-inf, inf], default 0.0

### **object**

Object to use for mapping with Object texture coordinates

**Type** `Object`

### **shadow\_factor**

Amount texture affects shadow

**Type** float in [-inf, inf], default 0.0

### **texture\_coords**

- **GLOBAL** Global, Uses global coordinates for the texture coordinates.
- **VIEW** View, Uses view coordinates for the texture coordinates.
- **OBJECT** Object, Uses linked object's coordinates for texture coordinates.

**Type** enum in ['GLOBAL', 'VIEW', 'OBJECT'], default 'GLOBAL'

**use\_map\_color**

Lets the texture affect the basic color of the lamp

**Type** boolean, default False

**use\_map\_shadow**

Lets the texture affect the shadow color of the lamp

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `TextureSlot.name`
- `TextureSlot.blend_type`
- `TextureSlot.color`
- `TextureSlot.default_value`
- `TextureSlot.invert`
- `TextureSlot.offset`
- `TextureSlot.output_node`
- `TextureSlot.use_rgb_to_intensity`
- `TextureSlot.scale`
- `TextureSlot.use_stencil`
- `TextureSlot.texture`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Lamp.texture_slots`
- `LampTextureSlots.add`
- `LampTextureSlots.create`

## 2.4.285 LampTextureSlots(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.LampTextureSlots` (*bpy\_struct*)

Collection of texture slots

**classmethod** `add()`

add

**Returns** The newly initialized mtex

**Return type** `LampTextureSlot`

**classmethod** `create(index)`

create

**Parameters** `index` (*int in [0, inf]*) – Index, Slot index to initialize

**Returns** The newly initialized mtex

**Return type** `LampTextureSlot`

**classmethod** `clear(index)`

clear

**Parameters** `index` (*int in [0, inf]*) – Index, Slot index to clear

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Lamp.texture_slots`

## 2.4.286 Lattice(ID)

base classes — `bpy_struct`, `ID`

**class** `bpy.types.Lattice` (*ID*)

Lattice datablock defining a grid for deforming other objects

**animation\_data**

Animation data for this datablock

**Type** `AnimData`, (readonly)

**interpolation\_type\_u**

**Type** enum in ['KEY\_LINEAR', 'KEY\_CARDINAL', 'KEY\_BSPLINE'], default 'KEY\_LINEAR'

**interpolation\_type\_v**

**Type** enum in ['KEY\_LINEAR', 'KEY\_CARDINAL', 'KEY\_BSPLINE'], default 'KEY\_LINEAR'

**interpolation\_type\_w**

**Type** enum in ['KEY\_LINEAR', 'KEY\_CARDINAL', 'KEY\_BSPLINE'], default 'KEY\_LINEAR'

**points**

Points of the lattice

**Type** `bpy_prop_collection` of `LatticePoint`, (readonly)

**points\_u**

Points in U direction (can't be changed when there are shape keys)

**Type** int in [1, 64], default 0

**points\_v**

Points in V direction (can't be changed when there are shape keys)

**Type** int in [1, 64], default 0

**points\_w**

Points in W direction (can't be changed when there are shape keys)

**Type** int in [1, 64], default 0

**shape\_keys**

**Type** `Key`, (readonly)

**use\_outside**

Only draw, and take into account, the outer vertices

**Type** boolean, default False

**vertex\_group**

Vertex group to apply the influence of the lattice

**Type** string, default ""

### Inherited Properties

- `bpy_struct.id_data`

- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`

### References

- `BlendData.lattices`
- `BlendDataLattices.new`
- `BlendDataLattices.remove`

## 2.4.287 LatticeModifier(Modifier)

base classes — `bpy_struct`, `Modifier`

**class** `bpy.types.LatticeModifier` (*Modifier*)

Lattice deformation modifier

#### **object**

Lattice object to deform with

**Type** `Object`

#### **vertex\_group**

Name of Vertex Group which determines influence of modifier per point

**Type** `string`, default `""`

### Inherited Properties

- `bpy_struct.id_data`
- `Modifier.name`
- `Modifier.use_apply_on_spline`
- `Modifier.show_in_editmode`
- `Modifier.show_expanded`
- `Modifier.show_on_cage`
- `Modifier.show_viewport`
- `Modifier.show_render`
- `Modifier.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.288 LatticePoint(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.LatticePoint` (*bpy\_struct*)

Point in the lattice grid

**co**

**Type** float array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`, (readonly)

**co\_deform**

**Type** float array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`

**groups**

Weights for the vertex groups this point is member of

**Type** `bpy_prop_collection` of `VertexGroupElement`, (readonly)

### Inherited Properties

- `bpy_struct.id_data`



## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Lattice.points`

## 2.4.289 Library(ID)

base classes — `bpy_struct`, `ID`

**class** `bpy.types.Library` (*ID*)

External .blend file from which data is linked

**filepath**

Path to the library .blend file

**Type** string, default ""

**parent**

**Type** `Library`, (readonly)

**users\_id**

ID data blocks which use this library (readonly)

## Inherited Properties

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`

## References

- `BlendData.libraries`
- `ID.library`
- `Library.parent`

### 2.4.290 LimitDistanceConstraint(Constraint)

base classes — `bpy_struct`, `Constraint`

**class** `bpy.types.LimitDistanceConstraint` (*Constraint*)

Limits the distance from target object

**distance**

Radius of limiting sphere

**Type** float in [0, 100], default 0.0

**head\_tail**

Target along length of bone: Head=0, Tail=1

**Type** float in [0, 1], default 0.0

**limit\_mode**

Distances in relation to sphere of influence to allow

- `LIMITDIST_INSIDE` Inside, The object is constrained inside a virtual sphere around the target object, with a radius defined by the limit distance.
- `LIMITDIST_OUTSIDE` Outside, The object is constrained outside a virtual sphere around the target object, with a radius defined by the limit distance.

- `LIMITDIST_ONSURFACE` On Surface, The object is constrained on the surface of a virtual sphere around the target object, with a radius defined by the limit distance.

**Type** enum in [`LIMITDIST_INSIDE`, `LIMITDIST_OUTSIDE`, `LIMITDIST_ONSURFACE`], default `LIMITDIST_INSIDE`

#### **subtarget**

**Type** string, default ""

#### **target**

Target Object

**Type** Object

#### **use\_transform\_limit**

Transforms are affected by this constraint as well

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `Constraint.name`
- `Constraint.active`
- `Constraint.mute`
- `Constraint.show_expanded`
- `Constraint.influence`
- `Constraint.error_location`
- `Constraint.owner_space`
- `Constraint.is_proxy_local`
- `Constraint.error_rotation`
- `Constraint.target_space`
- `Constraint.type`
- `Constraint.is_valid`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.291 LimitLocationConstraint(Constraint)

base classes — `bpy_struct`, `Constraint`

**class** `bpy.types.LimitLocationConstraint` (*Constraint*)

Limits the location of the constrained object

**max\_x**

Highest X value to allow

**Type** float in [-1000, 1000], default 0.0

**max\_y**

Highest Y value to allow

**Type** float in [-1000, 1000], default 0.0

**max\_z**

Highest Z value to allow

**Type** float in [-1000, 1000], default 0.0

**min\_x**

Lowest X value to allow

**Type** float in [-1000, 1000], default 0.0

**min\_y**

Lowest Y value to allow

**Type** float in [-1000, 1000], default 0.0

**min\_z**

Lowest Z value to allow

**Type** float in [-1000, 1000], default 0.0

**use\_max\_x**

Use the maximum X value

**Type** boolean, default False

**use\_max\_y**

Use the maximum Y value

**Type** boolean, default False

**use\_max\_z**

Use the maximum Z value

**Type** boolean, default False

**use\_min\_x**

Use the minimum X value

**Type** boolean, default False

**use\_min\_y**

Use the minimum Y value

**Type** boolean, default False

**use\_min\_z**

Use the minimum Z value

**Type** boolean, default False

**use\_transform\_limit**

Transforms are affected by this constraint as well

**Type** boolean, default False

**Inherited Properties**

- `bpy_struct.id_data`
- `Constraint.name`
- `Constraint.active`
- `Constraint.mute`
- `Constraint.show_expanded`
- `Constraint.influence`
- `Constraint.error_location`
- `Constraint.owner_space`
- `Constraint.is_proxy_local`
- `Constraint.error_rotation`
- `Constraint.target_space`
- `Constraint.type`
- `Constraint.is_valid`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**2.4.292 LimitRotationConstraint(Constraint)**

base classes — `bpy_struct`, `Constraint`

**class** `bpy.types.LimitRotationConstraint` (*Constraint*)

Limits the rotation of the constrained object

**max\_x**

Highest X value to allow

**Type** float in [-1000, 1000], default 0.0

**max\_y**

Highest Y value to allow

**Type** float in [-1000, 1000], default 0.0

**max\_z**

Highest Z value to allow

**Type** float in [-1000, 1000], default 0.0

**min\_x**

Lowest X value to allow

**Type** float in [-1000, 1000], default 0.0

**min\_y**

Lowest Y value to allow

**Type** float in [-1000, 1000], default 0.0

**min\_z**

Lowest Z value to allow

**Type** float in [-1000, 1000], default 0.0

**use\_limit\_x**

Use the minimum X value

**Type** boolean, default False

**use\_limit\_y**

Use the minimum Y value

**Type** boolean, default False

**use\_limit\_z**

Use the minimum Z value

**Type** boolean, default False

**use\_transform\_limit**

Transforms are affected by this constraint as well

**Type** boolean, default False

## Inherited Properties

- `bpy_struct.id_data`
- `Constraint.name`
- `Constraint.active`
- `Constraint.mute`
- `Constraint.show_expanded`
- `Constraint.influence`
- `Constraint.error_location`
- `Constraint.owner_space`
- `Constraint.is_proxy_local`
- `Constraint.error_rotation`
- `Constraint.target_space`
- `Constraint.type`
- `Constraint.is_valid`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.293 LimitScaleConstraint(Constraint)

base classes — `bpy_struct`, `Constraint`

**class** `bpy.types.LimitScaleConstraint` (*Constraint*)

Limits the scaling of the constrained object

**max\_x**

Highest X value to allow

**Type** float in [-1000, 1000], default 0.0

**max\_y**

Highest Y value to allow

**Type** float in [-1000, 1000], default 0.0

**max\_z**

Highest Z value to allow

**Type** float in [-1000, 1000], default 0.0

**min\_x**

Lowest X value to allow

**Type** float in [-1000, 1000], default 0.0

**min\_y**

Lowest Y value to allow

**Type** float in [-1000, 1000], default 0.0

**min\_z**

Lowest Z value to allow

**Type** float in [-1000, 1000], default 0.0

**use\_max\_x**

Use the maximum X value

**Type** boolean, default False

**use\_max\_y**

Use the maximum Y value

**Type** boolean, default False

**use\_max\_z**

Use the maximum Z value

**Type** boolean, default False

**use\_min\_x**

Use the minimum X value

**Type** boolean, default False

**use\_min\_y**

Use the minimum Y value

**Type** boolean, default False

**use\_min\_z**

Use the minimum Z value

**Type** boolean, default False

**use\_transform\_limit**

Transforms are affected by this constraint as well

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `Constraint.name`
- `Constraint.active`
- `Constraint.mute`
- `Constraint.show_expanded`
- `Constraint.influence`
- `Constraint.error_location`
- `Constraint.owner_space`
- `Constraint.is_proxy_local`
- `Constraint.error_rotation`
- `Constraint.target_space`
- `Constraint.type`
- `Constraint.is_valid`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`



- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.294 LockedTrackConstraint(Constraint)

base classes — `bpy_struct`, `Constraint`

**class** `bpy.types.LockedTrackConstraint` (*Constraint*)

Points toward the target along the track axis, while locking the other axis

### **head\_tail**

Target along length of bone: Head=0, Tail=1

**Type** float in [0, 1], default 0.0

### **lock\_axis**

Axis that points upward

**Type** enum in ['LOCK\_X', 'LOCK\_Y', 'LOCK\_Z'], default 'LOCK\_X'

### **subtarget**

**Type** string, default ""

### **target**

Target Object

**Type** `Object`

### **track\_axis**

Axis that points to the target object

**Type** enum in ['TRACK\_X', 'TRACK\_Y', 'TRACK\_Z', 'TRACK\_NEGATIVE\_X', 'TRACK\_NEGATIVE\_Y', 'TRACK\_NEGATIVE\_Z'], default 'TRACK\_X'

## Inherited Properties

- `bpy_struct.id_data`
- `Constraint.name`
- `Constraint.active`
- `Constraint.mute`
- `Constraint.show_expanded`
- `Constraint.influence`
- `Constraint.error_location`
- `Constraint.owner_space`
- `Constraint.is_proxy_local`
- `Constraint.error_rotation`
- `Constraint.target_space`
- `Constraint.type`
- `Constraint.is_valid`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.295 Macro(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.Macro` (*bpy\_struct*)

Storage of a macro operator being executed, or registered after execution

**bl\_description**

**Type** string, default ""

**bl\_idname**

**Type** string, default ""

**bl\_label**

**Type** string, default ""

**bl\_options**

Options for this operator type

- **REGISTER** Register, Display in the info window and support the redo toolbar panel.
- **UNDO** Undo, Push an undo event (needed for operator redo).
- **BLOCKING** Blocking, Block anything else from using the cursor.
- **MACRO** Macro, Use to check if an operator is a macro.
- **GRAB\_POINTER** Grab Pointer, Use so the operator grabs the mouse focus, enables wrapping when continuous grab is enabled.
- **PRESET** Preset, Display a preset button with the operators settings.
- **INTERNAL** Internal, Removes the operator from search results.

**Type** enum set in { 'REGISTER', 'UNDO', 'BLOCKING', 'MACRO', 'GRAB\_POINTER', 'PRESET', 'INTERNAL' }, default { 'REGISTER' }

**name**

**Type** string, default "", (readonly)

### properties

**Type** `OperatorProperties`, (readonly, never None)

**report** (*type, message*)

report

### Parameters

- **type** (*enum set in {'DEBUG', 'INFO', 'OPERATOR', 'WARNING', 'ERROR', 'ERROR\_INVALID\_INPUT', 'ERROR\_INVALID\_CONTEXT', 'ERROR\_OUT\_OF\_MEMORY'}*) – Type
- **message** (*string*) – Report Message

**classmethod poll** (*context*)

Test if the operator can be called or not

**Return type** boolean

**draw** (*context*)

Draw function for the operator

**define** (*opname*)

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.296 MagicTexture(Texture)

base classes — `bpy_struct`, `ID`, `Texture`

**class** `bpy.types.MagicTexture` (*Texture*)  
Procedural noise texture

**noise\_depth**  
Sets the depth of the cloud calculation  
**Type** int in [0, 30], default 0

**turbulence**  
Sets the turbulence of the bandnoise and ringnoise types  
**Type** float in [0.0001, inf], default 0.0

**users\_material**  
Materials that use this texture (readonly)

**users\_object\_modifier**  
Object modifiers that use this texture (readonly)

### Inherited Properties

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`
- `Texture.animation_data`
- `Texture.intensity`
- `Texture.color_ramp`
- `Texture.contrast`
- `Texture.factor_blue`
- `Texture.factor_green`
- `Texture.factor_red`
- `Texture.node_tree`
- `Texture.saturation`
- `Texture.use_preview_alpha`
- `Texture.type`
- `Texture.use_color_ramp`
- `Texture.use_nodes`
- `Texture.users_material`
- `Texture.users_object_modifier`
- `Texture.users_material`
- `Texture.users_object_modifier`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`

- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`
- `Texture.evaluate`

### 2.4.297 MaintainVolumeConstraint(Constraint)

base classes — `bpy_struct`, `Constraint`

**class** `bpy.types.MaintainVolumeConstraint` (*Constraint*)

Maintains a constant volume along a single scaling axis

**free\_axis**

The free scaling axis of the object

**Type** enum in ['SAMEVOL\_X', 'SAMEVOL\_Y', 'SAMEVOL\_Z'], default 'SAMEVOL\_X'

**volume**

Volume of the bone at rest

**Type** float in [0.001, 100], default 0.0

#### Inherited Properties

- `bpy_struct.id_data`
- `Constraint.name`
- `Constraint.active`
- `Constraint.mute`
- `Constraint.show_expanded`
- `Constraint.influence`
- `Constraint.error_location`
- `Constraint.owner_space`
- `Constraint.is_proxy_local`
- `Constraint.error_rotation`
- `Constraint.target_space`
- `Constraint.type`
- `Constraint.is_valid`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`

- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.298 MarbleTexture(Texture)

base classes — `bpy_struct`, `ID`, `Texture`

**class** `bpy.types.MarbleTexture` (*Texture*)  
Procedural noise texture

**marble\_type**

- **SOFT** Soft, Uses soft marble.
- **SHARP** Sharp, Uses more clearly defined marble.
- **SHARPER** Sharper, Uses very clearly defined marble.

**Type** enum in ['SOFT', 'SHARP', 'SHARPER'], default 'SOFT'

**nabla**

Size of derivative offset used for calculating normal

**Type** float in [0.001, 0.1], default 0.0

**noise\_basis**

Sets the noise basis used for turbulence

- **BLENDER\_ORIGINAL** Blender Original, Noise algorithm - Blender original: Smooth interpolated noise.
- **ORIGINAL\_PERLIN** Original Perlin, Noise algorithm - Original Perlin: Smooth interpolated noise.
- **IMPROVED\_PERLIN** Improved Perlin, Noise algorithm - Improved Perlin: Smooth interpolated noise.
- **VORONOI\_F1** Voronoi F1, Noise algorithm - Voronoi F1: Returns distance to the closest feature point.
- **VORONOI\_F2** Voronoi F2, Noise algorithm - Voronoi F2: Returns distance to the 2nd closest feature point.
- **VORONOI\_F3** Voronoi F3, Noise algorithm - Voronoi F3: Returns distance to the 3rd closest feature point.
- **VORONOI\_F4** Voronoi F4, Noise algorithm - Voronoi F4: Returns distance to the 4th closest feature point.
- **VORONOI\_F2\_F1** Voronoi F2-F1, Noise algorithm - Voronoi F1-F2.

- **VORONOI\_CRACKLE** Voronoi Crackle, Noise algorithm - Voronoi Crackle: Voronoi tessellation with sharp edges.
- **CELL\_NOISE** Cell Noise, Noise algorithm - Cell Noise: Square cell tessellation.

**Type** enum in ['BLENDER\_ORIGINAL', 'ORIGINAL\_PERLIN', 'IMPROVED\_PERLIN', 'VORONOI\_F1', 'VORONOI\_F2', 'VORONOI\_F3', 'VORONOI\_F4', 'VORONOI\_F2\_F1', 'VORONOI\_CRACKLE', 'CELL\_NOISE'], default 'BLENDER\_ORIGINAL'

#### **noise\_basis\_2**

- **SIN** Sin, Uses a sine wave to produce bands.
- **SAW** Saw, Uses a saw wave to produce bands.
- **TRI** Tri, Uses a triangle wave to produce bands.

**Type** enum in ['SIN', 'SAW', 'TRI'], default 'SIN'

#### **noise\_depth**

Sets the depth of the cloud calculation

**Type** int in [0, 30], default 0

#### **noise\_scale**

Sets scaling for noise input

**Type** float in [0.0001, inf], default 0.0

#### **noise\_type**

- **SOFT\_NOISE** Soft, Generate soft noise (smooth transitions).
- **HARD\_NOISE** Hard, Generate hard noise (sharp transitions).

**Type** enum in ['SOFT\_NOISE', 'HARD\_NOISE'], default 'SOFT\_NOISE'

#### **turbulence**

Sets the turbulence of the bandnoise and ringnoise types

**Type** float in [0.0001, inf], default 0.0

#### **users\_material**

Materials that use this texture (readonly)

#### **users\_object\_modifier**

Object modifiers that use this texture (readonly)

### Inherited Properties

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`

- `Texture.animation_data`
- `Texture.intensity`
- `Texture.color_ramp`
- `Texture.contrast`
- `Texture.factor_blue`
- `Texture.factor_green`
- `Texture.factor_red`
- `Texture.node_tree`
- `Texture.saturation`
- `Texture.use_preview_alpha`
- `Texture.type`
- `Texture.use_color_ramp`
- `Texture.use_nodes`
- `Texture.users_material`
- `Texture.users_object_modifier`
- `Texture.users_material`
- `Texture.users_object_modifier`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`
- `Texture.evaluate`

### 2.4.299 MaskModifier(Modifier)

base classes — `bpy_struct`, `Modifier`

**class** `bpy.types.MaskModifier` (*Modifier*)

Mask modifier to hide parts of the mesh

**armature**

Armature to use as source of bones to mask

**Type** Object



**invert\_vertex\_group**

Use vertices that are not part of region defined

**Type** boolean, default False

**mode**

**Type** enum in ['VERTEX\_GROUP', 'ARMATURE'], default 'VERTEX\_GROUP'

**vertex\_group**

Vertex group name

**Type** string, default ""

**Inherited Properties**

- `bpy_struct.id_data`
- `Modifier.name`
- `Modifier.use_apply_on_spline`
- `Modifier.show_in_editmode`
- `Modifier.show_expanded`
- `Modifier.show_on_cage`
- `Modifier.show_viewport`
- `Modifier.show_render`
- `Modifier.type`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**2.4.300 Material(ID)**

base classes — `bpy_struct`, `ID`

**class** `bpy.types.Material` (*ID*)

Material datablock to define the appearance of geometric objects for rendering

**active\_node\_material**

Active node material

**Type** `Material`

**active\_texture**

Active texture slot being displayed

**Type** `Texture`

**active\_texture\_index**

Index of active texture slot

**Type** `int` in [0, 17], default 0

**alpha**

Alpha transparency of the material

**Type** `float` in [0, 1], default 0.0

**ambient**

Amount of global ambient color the material receives

**Type** `float` in [0, 1], default 0.0

**animation\_data**

Animation data for this datablock

**Type** `AnimData`, (readonly)

**darkness**

Minnaert darkness

**Type** `float` in [0, 2], default 0.0

**diffuse\_color**

Diffuse color of the material

**Type** `float` array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**diffuse\_fresnel**

Power of Fresnel

**Type** `float` in [0, 5], default 0.0

**diffuse\_fresnel\_factor**

Blending factor of Fresnel

**Type** `float` in [0, 5], default 0.0

**diffuse\_intensity**

Amount of diffuse reflection

**Type** `float` in [0, 1], default 0.0

**diffuse\_ramp**

Color ramp used to affect diffuse shading

**Type** `ColorRamp`, (readonly)

**diffuse\_ramp\_blend**

Blending method of the ramp and the diffuse color

**Type** `enum` in ['MIX', 'ADD', 'MULTIPLY', 'SUBTRACT', 'SCREEN', 'DIVIDE', 'DIFFERENCE', 'DARKEN', 'LIGHTEN', 'OVERLAY', 'DODGE', 'BURN', 'HUE', 'SATURATION', 'VALUE', 'COLOR', 'SOFT\_LIGHT', 'LINEAR\_LIGHT'], default 'MIX'

**diffuse\_ramp\_factor**

Blending factor (also uses alpha in Colorband)

**Type** float in [0, 1], default 0.0

#### **diffuse\_ramp\_input**

Determines how the ramp maps on the surface

**Type** enum in ['SHADER', 'ENERGY', 'NORMAL', 'RESULT'], default 'SHADER'

#### **diffuse\_shader**

- LAMBERT Lambert, Use a Lambertian shader.
- OREN\_NAYAR Oren-Nayar, Use an Oren-Nayar shader.
- TOON Toon, Use a toon shader.
- MINNAERT Minnaert, Use a Minnaert shader.
- FRESNEL Fresnel, Use a Fresnel shader.

**Type** enum in ['LAMBERT', 'OREN\_NAYAR', 'TOON', 'MINNAERT', 'FRESNEL'], default 'LAMBERT'

#### **diffuse\_toon\_size**

Size of diffuse toon area

**Type** float in [0, 3.14], default 0.0

#### **diffuse\_toon\_smooth**

Smoothness of diffuse toon area

**Type** float in [0, 1], default 0.0

#### **emit**

Amount of light to emit

**Type** float in [0, inf], default 0.0

#### **game\_settings**

Game material settings

**Type** `MaterialGameSettings`, (readonly, never None)

#### **halo**

Halo settings for the material

**Type** `MaterialHalo`, (readonly, never None)

#### **invert\_z**

Render material's faces with an inverted Z buffer (scanline only)

**Type** boolean, default False

#### **light\_group**

Limit lighting to lamps in this Group

**Type** `Group`

#### **mirror\_color**

Mirror color of the material

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

#### **node\_tree**

Node tree for node based materials

**Type** `NodeTree`, (readonly)

**offset\_z**

Give faces an artificial offset in the Z buffer for Z transparency

**Type** float in [-inf, inf], default 0.0

**pass\_index**

Index # for the IndexMA render pass

**Type** int in [0, 32767], default 0

**physics**

Game physics settings

**Type** `MaterialPhysics`, (readonly, never None)

**preview\_render\_type**

Type of preview render

- FLAT Flat, Preview type: Flat XY plane.
- SPHERE Sphere, Preview type: Sphere.
- CUBE Flat, Preview type: Cube.
- MONKEY Flat, Preview type: Monkey.
- HAIR Flat, Preview type: Hair strands.
- SPHERE\_A Flat, Preview type: Large sphere with sky.

**Type** enum in ['FLAT', 'SPHERE', 'CUBE', 'MONKEY', 'HAIR', 'SPHERE\_A'], default 'FLAT'

**raytrace\_mirror**

Raytraced reflection settings for the material

**Type** `MaterialRaytraceMirror`, (readonly, never None)

**raytrace\_transparency**

Raytraced transparency settings for the material

**Type** `MaterialRaytraceTransparency`, (readonly, never None)

**roughness**

Oren-Nayar Roughness

**Type** float in [0, 3.14], default 0.0

**shadow\_buffer\_bias**

Factor to multiply shadow buffer bias with (0 is ignore.)

**Type** float in [0, 10], default 0.0

**shadow\_cast\_alpha**

Shadow casting alpha, in use for Irregular and Deep shadow buffer

**Type** float in [0.001, 1], default 0.0

**shadow\_only\_type**

How to draw shadows

- SHADOW\_ONLY\_OLD Shadow and Distance, Old shadow only method.
- SHADOW\_ONLY Shadow Only, Improved shadow only method.

- SHADOW\_ONLY\_SHADED** Shadow and Shading, Improved shadow only method which also renders lightless areas as shadows.

**Type** enum in ['SHADOW\_ONLY\_OLD', 'SHADOW\_ONLY', 'SHADOW\_ONLY\_SHADED'], default 'SHADOW\_ONLY\_OLD'

#### **shadow\_ray\_bias**

Shadow raytracing bias to prevent terminator problems on shadow boundary

**Type** float in [0, 0.25], default 0.0

#### **specular\_alpha**

Alpha transparency for specular areas

**Type** float in [0, 1], default 0.0

#### **specular\_color**

Specular color of the material

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

#### **specular\_hardness**

How hard (sharp) the specular reflection is

**Type** int in [1, 511], default 0

#### **specular\_intensity**

How intense (bright) the specular reflection is

**Type** float in [0, 1], default 0.0

#### **specular\_ior**

Specular index of refraction

**Type** float in [1, 10], default 0.0

#### **specular\_ramp**

Color ramp used to affect specular shading

**Type** `ColorRamp`, (readonly)

#### **specular\_ramp\_blend**

Blending method of the ramp and the specular color

**Type** enum in ['MIX', 'ADD', 'MULTIPLY', 'SUBTRACT', 'SCREEN', 'DIVIDE', 'DIFFERENCE', 'DARKEN', 'LIGHTEN', 'OVERLAY', 'DODGE', 'BURN', 'HUE', 'SATURATION', 'VALUE', 'COLOR', 'SOFT\_LIGHT', 'LINEAR\_LIGHT'], default 'MIX'

#### **specular\_ramp\_factor**

Blending factor (also uses alpha in Colorband)

**Type** float in [0, 1], default 0.0

#### **specular\_ramp\_input**

Determines how the ramp maps on the surface

**Type** enum in ['SHADER', 'ENERGY', 'NORMAL', 'RESULT'], default 'SHADER'

#### **specular\_shader**

- COOKTORR** CookTorr, Use a Cook-Torrance shader.
- PHONG** Phong, Use a Phong shader.
- BLINN** Blinn, Use a Blinn shader.

- TOON** Toon, Use a toon shader.
- WARDISO** WardIso, Use a Ward anisotropic shader.

**Type** enum in ['COOKTORR', 'PHONG', 'BLINN', 'TOON', 'WARDISO'], default 'COOKTORR'

**specular\_slope**

The standard deviation of surface slope

**Type** float in [0, 0.4], default 0.0

**specular\_toon\_size**

Size of specular toon area

**Type** float in [0, 1.53], default 0.0

**specular\_toon\_smooth**

Smoothness of specular toon area

**Type** float in [0, 1], default 0.0

**strand**

Strand settings for the material

**Type** `MaterialStrand`, (readonly, never None)

**subsurface\_scattering**

Subsurface scattering settings for the material

**Type** `MaterialSubsurfaceScattering`, (readonly, never None)

**texture\_slots**

Texture slots defining the mapping and influence of textures

**Type** `MaterialTextureSlots` `bpy_prop_collection` of `MaterialTextureSlot`, (readonly)

**translucency**

Amount of diffuse shading on the back side

**Type** float in [0, 1], default 0.0

**transparency\_method**

Method to use for rendering transparency

- MASK** Mask, Mask the background.
- Z\_TRANSPARENCY** Z Transparency, Use alpha buffer for transparent faces.
- RAYTRACE** Raytrace, Use raytracing for transparent refraction rendering.

**Type** enum in ['MASK', 'Z\_TRANSPARENCY', 'RAYTRACE'], default 'MASK'

**type**

Material type defining how the object is rendered

- SURFACE** Surface, Render object as a surface.
- WIRE** Wire, Render the edges of faces as wires (not supported in raytracing).
- VOLUME** Volume, Render object as a volume.
- HALO** Halo, Render object as halo particles.

**Type** enum in ['SURFACE', 'WIRE', 'VOLUME', 'HALO'], default 'SURFACE'

**use\_cast\_approximate**

Allow this material to cast shadows when using approximate ambient occlusion

**Type** boolean, default False

**use\_cast\_buffer\_shadows**

Allow this material to cast shadows from shadow buffer lamps

**Type** boolean, default False

**use\_cast\_shadows\_only**

Make objects with this material appear invisible (not rendered), only casting shadows

**Type** boolean, default False

**use\_cubic**

Use cubic interpolation for diffuse values, for smoother transitions

**Type** boolean, default False

**use\_diffuse\_ramp**

Toggle diffuse ramp operations

**Type** boolean, default False

**use\_face\_texture**

Replace the object's base color with color from face assigned image textures

**Type** boolean, default False

**use\_face\_texture\_alpha**

Replace the object's base alpha value with alpha from face assigned image textures

**Type** boolean, default False

**use\_full\_oversampling**

Force this material to render full shading/textures for all anti-aliasing samples

**Type** boolean, default False

**use\_light\_group\_exclusive**

Material uses the light group exclusively - these lamps are excluded from other scene lighting

**Type** boolean, default False

**use\_mist**

Use mist with this material (in world settings)

**Type** boolean, default False

**use\_nodes**

Use shader nodes to render the material

**Type** boolean, default False

**use\_object\_color**

Modulate the result with a per-object color

**Type** boolean, default False

**use\_only\_shadow**

Render shadows as the material's alpha value, making the material transparent except for shadowed areas

**Type** boolean, default False

**use\_ray\_shadow\_bias**

Prevent raytraced shadow errors on surfaces with smooth shaded normals (terminator problem)

**Type** boolean, default False

**use\_raytrace**

Include this material and geometry that uses it in raytracing calculations

**Type** boolean, default False

**use\_shadeless**

Make this material insensitive to light or shadow

**Type** boolean, default False

**use\_shadows**

Allow this material to receive shadows

**Type** boolean, default False

**use\_sky**

Render this material with zero alpha, with sky background in place (scanline only)

**Type** boolean, default False

**use\_specular\_ramp**

Toggle specular ramp operations

**Type** boolean, default False

**use\_tangent\_shading**

Use the material's tangent vector instead of the normal for shading - for anisotropic shading effects

**Type** boolean, default False

**use\_textures**

Enable/Disable each texture

**Type** boolean array of 18 items, default (False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False)

**use\_transparency**

Render material as transparent

**Type** boolean, default False

**use\_transparent\_shadows**

Allow this object to receive transparent shadows cast through other objects

**Type** boolean, default False

**use\_vertex\_color\_light**

Add vertex colors as additional lighting

**Type** boolean, default False

**use\_vertex\_color\_paint**

Replace object base color with vertex colors (multiply with 'texture face' face assigned textures)

**Type** boolean, default False

**volume**

Volume settings for the material

**Type** `MaterialVolume`, (readonly, never None)



## Inherited Properties

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`

## References

- `BlendData.materials`
- `BlendDataMaterials.new`
- `BlendDataMaterials.remove`
- `Curve.materials`
- `IDMaterials.append`
- `IDMaterials.pop`
- `Material.active_node_material`
- `MaterialSlot.material`
- `Mesh.materials`
- `MetaBall.materials`
- `Object.active_material`
- `RenderLayer.material_override`
- `SceneRenderLayer.material_override`
- `ShaderNodeExtendedMaterial.material`
- `ShaderNodeMaterial.material`

- `TouchSensor.material`

### 2.4.301 MaterialGameSettings(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.MaterialGameSettings` (*bpy\_struct*)

Game Engine settings for a Material datablock

#### **alpha\_blend**

Blend Mode for Transparent Faces

- `OPAQUE` Opaque, Render color of textured face as color.
- `ADD` Add, Render face transparent and add color of face.
- `CLIP` Alpha Clip, Use the image alpha values clipped with no blending (binary alpha).
- `ALPHA` Alpha Blend, Render polygon transparent, depending on alpha channel of the texture.
- `ALPHA_SORT` Alpha Sort, Sort faces for correct alpha drawing (slow, use Alpha Clip instead when possible).

**Type** enum in [`'OPAQUE'`, `'ADD'`, `'CLIP'`, `'ALPHA'`, `'ALPHA_SORT'`], default `'OPAQUE'`

#### **face\_orientation**

Especial face orientation options

- `NORMAL` Normal, No tranformation.
- `HALO` Halo, Screen aligned billboard.
- `BILLBOARD` Billboard, Billboard with Z-axis constraint.
- `SHADOW` Shadow, Faces are used for shadow.

**Type** enum in [`'NORMAL'`, `'HALO'`, `'BILLBOARD'`, `'SHADOW'`], default `'NORMAL'`

#### **invisible**

Make face invisible

**Type** boolean, default `False`

#### **physics**

Use physics properties of materials

**Type** boolean, default `False`

#### **text**

Use material as text in Game Engine

**Type** boolean, default `False`

#### **use\_backface\_culling**

Hide Back of the face in Game Engine

**Type** boolean, default `False`

### Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Material.game_settings`

### 2.4.302 MaterialHalo(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.MaterialHalo` (*bpy\_struct*)  
Halo particle effect settings for a Material datablock

#### **add**

Strength of the add effect

**Type** float in [0, 1], default 0.0

#### **flare\_boost**

Give the flare extra strength

**Type** float in [0.1, 10], default 0.0

#### **flare\_seed**

Offset in the flare seed table

**Type** int in [0, 255], default 0

#### **flare\_size**

Factor by which the flare is larger than the halo

**Type** float in [0.1, 25], default 0.0

#### **flare\_subflare\_count**

Number of sub-flares

**Type** int in [1, 32], default 0

#### **flare\_subflare\_size**

Dimension of the sub-flares, dots and circles

**Type** float in [0.1, 25], default 0.0

**hardness**

Hardness of the halo

**Type** int in [0, 127], default 0

**line\_count**

Number of star shaped lines rendered over the halo

**Type** int in [0, 250], default 0

**ring\_count**

Number of rings rendered over the halo

**Type** int in [0, 24], default 0

**seed**

Randomize ring dimension and line location

**Type** int in [0, 255], default 0

**size**

Dimension of the halo

**Type** float in [0, 100], default 0.0

**star\_tip\_count**

Number of points on the star shaped halo

**Type** int in [3, 50], default 0

**use\_extreme\_alpha**

Use extreme alpha

**Type** boolean, default False

**use\_flare\_mode**

Render halo as a lens flare

**Type** boolean, default False

**use\_lines**

Render star shaped lines over halo

**Type** boolean, default False

**use\_ring**

Render rings over halo

**Type** boolean, default False

**use\_shaded**

Let halo receive light and shadows from external objects

**Type** boolean, default False

**use\_soft**

Soften the edges of halos at intersections with other geometry

**Type** boolean, default False

**use\_star**

Render halo as a star

**Type** boolean, default False

**use\_texture**

Give halo a texture

**Type** boolean, default False**use\_vertex\_normal**

Use the vertex normal to specify the dimension of the halo

**Type** boolean, default False**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `Material.halo`

**2.4.303 MaterialPhysics(bpy\_struct)**base class — `bpy_struct`**class** `bpy.types.MaterialPhysics` (*bpy\_struct*)

Physics settings for a Material datablock

**elasticity**

Elasticity of collisions

**Type** float in [0, 1], default 0.0**fh\_damping**

Damping of the spring force, when inside the physics distance area

**Type** float in [0, 1], default 0.0

**fh\_distance**

Distance of the physics area

**Type** float in [0, 20], default 0.0

**fh\_force**

Upward spring force, when inside the physics distance area

**Type** float in [0, 1], default 0.0

**friction**

Coulomb friction coefficient, when inside the physics distance area

**Type** float in [0, 100], default 0.0

**use\_fh\_normal**

Align dynamic game objects along the surface normal, when inside the physics distance area

**Type** boolean, default False

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `Material.physics`

**2.4.304 MaterialRaytraceMirror(`bpy_struct`)**

base class — `bpy_struct`

**class** `bpy.types.MaterialRaytraceMirror` (*bpy\_struct*)  
Raytraced reflection settings for a Material datablock

**depth**

Maximum allowed number of light inter-reflections

**Type** int in [0, 32767], default 0

**distance**

Maximum distance of reflected rays (reflections further than this range fade to sky color or material color)

**Type** float in [0, 10000], default 0.0

**fade\_to**

The color that rays with no intersection within the Max Distance take (material color can be best for indoor scenes, sky color for outdoor)

**Type** enum in ['FADE\_TO\_SKY', 'FADE\_TO\_MATERIAL'], default 'FADE\_TO\_SKY'

**fresnel**

Power of Fresnel for mirror reflection

**Type** float in [0, 5], default 0.0

**fresnel\_factor**

Blending factor for Fresnel

**Type** float in [0, 5], default 0.0

**gloss\_anisotropic**

The shape of the reflection, from 0.0 (circular) to 1.0 (fully stretched along the tangent)

**Type** float in [0, 1], default 0.0

**gloss\_factor**

The shininess of the reflection (values < 1.0 give diffuse, blurry reflections)

**Type** float in [0, 1], default 0.0

**gloss\_samples**

Number of cone samples averaged for blurry reflections

**Type** int in [0, 1024], default 0

**gloss\_threshold**

Threshold for adaptive sampling (if a sample contributes less than this amount [as a percentage], sampling is stopped)

**Type** float in [0, 1], default 0.0

**reflect\_factor**

Amount of mirror reflection for raytrace

**Type** float in [0, 1], default 0.0

**use**

Enable raytraced reflections

**Type** boolean, default False

**Inherited Properties**

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Material.raytrace_mirror`

### 2.4.305 MaterialRaytraceTransparency(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.MaterialRaytraceTransparency` (*bpy\_struct*)

Raytraced refraction settings for a Material datablock

#### **depth**

Maximum allowed number of light inter-refractions

**Type** int in [0, 32767], default 0

#### **depth\_max**

Maximum depth for light to travel through the transparent material before becoming fully filtered (0.0 is disabled)

**Type** float in [0, 100], default 0.0

#### **falloff**

Falloff power for transmissivity filter effect (1.0 is linear)

**Type** float in [0.1, 10], default 0.0

#### **filter**

Amount to blend in the material's diffuse color in raytraced transparency (simulating absorption)

**Type** float in [0, 1], default 0.0

#### **fresnel**

Power of Fresnel for transparency (Ray or ZTransp)

**Type** float in [0, 5], default 0.0

#### **fresnel\_factor**

Blending factor for Fresnel



**Type** float in [1, 5], default 0.0

**gloss\_factor**

The clarity of the refraction. Values < 1.0 give diffuse, blurry refractions

**Type** float in [0, 1], default 0.0

**gloss\_samples**

Number of cone samples averaged for blurry refractions

**Type** int in [0, 1024], default 0

**gloss\_threshold**

Threshold for adaptive sampling. If a sample contributes less than this amount (as a percentage), sampling is stopped

**Type** float in [0, 1], default 0.0

**ior**

Angular index of refraction for raytraced refraction

**Type** float in [0.25, 4], default 0.0

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Material.raytrace_transparency`

## 2.4.306 MaterialSlot(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.MaterialSlot` (*bpy\_struct*)  
Material slot in an object

**link**  
Link material to object or the object's data  
**Type** enum in ['OBJECT', 'DATA'], default 'DATA'

**material**  
Material datablock used by this material slot  
**Type** `Material`

**name**  
Material slot name  
**Type** string, default "", (readonly)

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Object.material_slots`

## 2.4.307 MaterialStrand(*bpy\_struct*)

base class — `bpy_struct`

**class** `bpy.types.MaterialStrand` (*bpy\_struct*)  
Strand settings for a Material datablock

**blend\_distance**  
Worldspace distance over which to blend in the surface normal

**Type** float in [0, 10], default 0.0

**root\_size**

Start size of strands in pixels or Blender units

**Type** float in [0, inf], default 0.0

**shape**

Positive values make strands rounder, negative ones make strands spiky

**Type** float in [-0.9, 0.9], default 0.0

**size\_min**

Minimum size of strands in pixels

**Type** float in [0.001, 10], default 0.0

**tip\_size**

End size of strands in pixels or Blender units

**Type** float in [0, inf], default 0.0

**use\_blender\_units**

Use Blender units for widths instead of pixels

**Type** boolean, default False

**use\_surface\_diffuse**

Make diffuse shading more similar to shading the surface

**Type** boolean, default False, (readonly)

**use\_tangent\_shading**

Uses direction of strands as normal for tangent-shading

**Type** boolean, default False

**uv\_layer**

Name of UV layer to override

**Type** string, default ""

**width\_fade**

Transparency along the width of the strand

**Type** float in [0, 2], default 0.0

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`

- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Material.strand`

### 2.4.308 MaterialSubsurfaceScattering(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.MaterialSubsurfaceScattering` (*bpy\_struct*)

Diffuse subsurface scattering settings for a Material datablock

**back**

Back scattering weight

**Type** float in [0, 10], default 0.0

**color**

Scattering color

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**color\_factor**

Blend factor for SSS colors

**Type** float in [0, 1], default 0.0

**error\_threshold**

Error tolerance (low values are slower and higher quality)

**Type** float in [-inf, inf], default 0.0

**front**

Front scattering weight

**Type** float in [0, 2], default 0.0

**ior**

Index of refraction (higher values are denser)

**Type** float in [-inf, inf], default 0.0

**radius**

Mean red/green/blue scattering path length

**Type** float array of 3 items in [0.001, inf], default (0.0, 0.0, 0.0)

**scale**

Object scale factor

**Type** float in [-inf, inf], default 0.0

**texture\_factor**

Texture scattering blend factor

**Type** float in [0, 1], default 0.0**use**

Enable diffuse subsurface scattering effects in a material

**Type** boolean, default False**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `Material.subsurface_scattering`

**2.4.309 MaterialTextureSlot(TextureSlot)**base classes — `bpy_struct`, `TextureSlot`**class** `bpy.types.MaterialTextureSlot` (*TextureSlot*)

Texture slot for textures in a Material datablock

**alpha\_factor**

Amount texture affects alpha

**Type** float in [-inf, inf], default 0.0**ambient\_factor**

Amount texture affects ambient

**Type** float in [-inf, inf], default 0.0

**bump\_method**

Method to use for bump mapping

**Type** enum in ['BUMP\_ORIGINAL', 'BUMP\_COMPATIBLE', 'BUMP\_DEFAULT', 'BUMP\_BEST\_QUALITY'], default 'BUMP\_ORIGINAL'

**bump\_objectspace**

Space to apply bump mapping in

**Type** enum in ['BUMP\_VIEWSPACE', 'BUMP\_OBJECTSPACE', 'BUMP\_TEXTURESPACE'], default 'BUMP\_VIEWSPACE'

**density\_factor**

Amount texture affects density

**Type** float in [-inf, inf], default 0.0

**diffuse\_color\_factor**

Amount texture affects diffuse color

**Type** float in [-inf, inf], default 0.0

**diffuse\_factor**

Amount texture affects diffuse reflectivity

**Type** float in [-inf, inf], default 0.0

**displacement\_factor**

Amount texture displaces the surface

**Type** float in [-inf, inf], default 0.0

**emission\_color\_factor**

Amount texture affects emission color

**Type** float in [-inf, inf], default 0.0

**emission\_factor**

Amount texture affects emission

**Type** float in [-inf, inf], default 0.0

**emit\_factor**

Amount texture affects emission

**Type** float in [-inf, inf], default 0.0

**hardness\_factor**

Amount texture affects hardness

**Type** float in [-inf, inf], default 0.0

**mapping**

- FLAT Flat, Map X and Y coordinates directly.
- CUBE Cube, Map using the normal vector.
- TUBE Tube, Map with Z as central axis.
- SPHERE Sphere, Map with Z as central axis.

**Type** enum in ['FLAT', 'CUBE', 'TUBE', 'SPHERE'], default 'FLAT'

**mapping\_x**

**Type** enum in ['NONE', 'X', 'Y', 'Z'], default 'NONE'

#### **mapping\_y**

**Type** enum in ['NONE', 'X', 'Y', 'Z'], default 'NONE'

#### **mapping\_z**

**Type** enum in ['NONE', 'X', 'Y', 'Z'], default 'NONE'

#### **mirror\_factor**

Amount texture affects mirror color

**Type** float in [-inf, inf], default 0.0

#### **normal\_factor**

Amount texture affects normal values

**Type** float in [-inf, inf], default 0.0

#### **normal\_map\_space**

Set space of normal map image

**Type** enum in ['CAMERA', 'WORLD', 'OBJECT', 'TANGENT'], default 'CAMERA'

#### **object**

Object to use for mapping with Object texture coordinates

**Type** *Object*

#### **raymir\_factor**

Amount texture affects ray mirror

**Type** float in [-inf, inf], default 0.0

#### **reflection\_color\_factor**

Amount texture affects color of out-scattered light

**Type** float in [-inf, inf], default 0.0

#### **reflection\_factor**

Amount texture affects brightness of out-scattered light

**Type** float in [-inf, inf], default 0.0

#### **scattering\_factor**

Amount texture affects scattering

**Type** float in [-inf, inf], default 0.0

#### **specular\_color\_factor**

Amount texture affects specular color

**Type** float in [-inf, inf], default 0.0

#### **specular\_factor**

Amount texture affects specular reflectivity

**Type** float in [-inf, inf], default 0.0

#### **texture\_coords**

- GLOBAL Global, Uses global coordinates for the texture coordinates.
- OBJECT Object, Uses linked object's coordinates for texture coordinates.
- UV UV, Uses UV coordinates for texture coordinates.

- ORCO Generated, Uses the original undeformed coordinates of the object.
- STRAND Strand / Particle, Uses normalized strand texture coordinate (1D) or particle age (X) and trail position (Y).
- STICKY Sticky, Uses mesh's sticky coordinates for the texture coordinates.
- WINDOW Window, Uses screen coordinates as texture coordinates.
- NORMAL Normal, Uses normal vector as texture coordinates.
- REFLECTION Reflection, Uses reflection vector as texture coordinates.
- STRESS Stress, Uses the difference of edge lengths compared to original coordinates of the mesh.
- TANGENT Tangent, Uses the optional tangent vector as texture coordinates.

**Type** enum in ['GLOBAL', 'OBJECT', 'UV', 'ORCO', 'STRAND', 'STICKY', 'WINDOW', 'NORMAL', 'REFLECTION', 'STRESS', 'TANGENT'], default 'GLOBAL'

**translucency\_factor**

Amount texture affects translucency

**Type** float in [-inf, inf], default 0.0

**transmission\_color\_factor**

Amount texture affects result color after light has been scattered/absorbed

**Type** float in [-inf, inf], default 0.0

**use**

Enable this material texture slot

**Type** boolean, default False

**use\_from\_dupli**

Dupli's instanced from verts, faces or particles, inherit texture coordinate from their parent

**Type** boolean, default False

**use\_from\_original**

Dupli's derive their object coordinates from the original object's transformation

**Type** boolean, default False

**use\_map\_alpha**

Causes the texture to affect the alpha value

**Type** boolean, default False

**use\_map\_ambient**

Causes the texture to affect the value of ambient

**Type** boolean, default False

**use\_map\_color\_diffuse**

Causes the texture to affect basic color of the material

**Type** boolean, default False

**use\_map\_color\_emission**

Causes the texture to affect the color of emission

**Type** boolean, default False



**use\_map\_color\_reflection**

Causes the texture to affect the color of scattered light

**Type** boolean, default False

**use\_map\_color\_spec**

Causes the texture to affect the specular color

**Type** boolean, default False

**use\_map\_color\_transmission**

Causes the texture to affect the result color after other light has been scattered/absorbed

**Type** boolean, default False

**use\_map\_density**

Causes the texture to affect the volume's density

**Type** boolean, default False

**use\_map\_diffuse**

Causes the texture to affect the value of the materials diffuse reflectivity

**Type** boolean, default False

**use\_map\_displacement**

Let the texture displace the surface

**Type** boolean, default False

**use\_map\_emission**

Causes the texture to affect the volume's emission

**Type** boolean, default False

**use\_map\_emit**

Causes the texture to affect the emit value

**Type** boolean, default False

**use\_map\_hardness**

Causes the texture to affect the hardness value

**Type** boolean, default False

**use\_map\_mirror**

Causes the texture to affect the mirror color

**Type** boolean, default False

**use\_map\_normal**

Causes the texture to affect the rendered normal

**Type** boolean, default False

**use\_map\_raymir**

Causes the texture to affect the ray-mirror value

**Type** boolean, default False

**use\_map\_reflect**

Causes the texture to affect the reflected light's brightness

**Type** boolean, default False

**use\_map\_scatter**

Causes the texture to affect the volume's scattering

**Type** boolean, default False

**use\_map\_specular**

Causes the texture to affect the value of specular reflectivity

**Type** boolean, default False

**use\_map\_translucency**

Causes the texture to affect the translucency value

**Type** boolean, default False

**use\_map\_warp**

Let the texture warp texture coordinates of next channels

**Type** boolean, default False

**uv\_layer**

UV layer to use for mapping with UV texture coordinates

**Type** string, default ""

**warp\_factor**

Amount texture affects texture coordinates of next channels

**Type** float in [-inf, inf], default 0.0

### Inherited Properties

- `bpy_struct.id_data`
- `TextureSlot.name`
- `TextureSlot.blend_type`
- `TextureSlot.color`
- `TextureSlot.default_value`
- `TextureSlot.invert`
- `TextureSlot.offset`
- `TextureSlot.output_node`
- `TextureSlot.use_rgb_to_intensity`
- `TextureSlot.scale`
- `TextureSlot.use_stencil`
- `TextureSlot.texture`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`

- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Material.texture_slots`
- `MaterialTextureSlots.add`
- `MaterialTextureSlots.create`

## 2.4.310 MaterialTextureSlots(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.MaterialTextureSlots` (*bpy\_struct*)

Collection of texture slots

**classmethod** `add()`

`add`

**Returns** The newly initialized `mtx`

**Return type** `MaterialTextureSlot`

**classmethod** `create(index)`

`create`

**Parameters** `index` (*int in [0, inf]*) – Index, Slot index to initialize

**Returns** The newly initialized `mtx`

**Return type** `MaterialTextureSlot`

**classmethod** `clear(index)`

`clear`

**Parameters** `index` (*int in [0, inf]*) – Index, Slot index to clear

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`

- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Material.texture_slots`

### 2.4.311 MaterialVolume(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.MaterialVolume` (*bpy\_struct*)

Volume rendering settings for a Material datablock

**asymmetry**

Back scattering (-1.0) to Forward scattering (1.0) and the range in between

**Type** float in [-1, 1], default 0.0

**cache\_resolution**

Resolution of the voxel grid, low resolutions are faster, high resolutions use more memory

**Type** int in [1, 1024], default 0

**density**

The base density of the volume

**Type** float in [0, 1], default 0.0

**density\_scale**

Multiplier for the material's density

**Type** float in [0, inf], default 0.0

**depth\_threshold**

Stop ray marching early if transmission drops below this luminance - higher values give speedups in dense volumes at the expense of accuracy

**Type** float in [0, 1], default 0.0

**emission**

Amount of light that gets emitted by the volume

**Type** float in [0, inf], default 0.0

**emission\_color**

Color of emitted light

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**light\_method**

Method of shading, attenuating, and scattering light through the volume

- **SHADELESS** Shadeless, Do not calculate lighting and shadows.
- **SHADOWED** Shadowed.
- **SHADED** Shaded.

- `MULTIPLE_SCATTERING` Multiple Scattering.
- `SHADED_PLUS_MULTIPLE_SCATTERING` Shaded + Multiple Scattering.

**Type** enum in ['SHADELESS', 'SHADOWED', 'SHADED', 'MULTIPLE\_SCATTERING', 'SHADED\_PLUS\_MULTIPLE\_SCATTERING'], default 'SHADELESS'

#### **ms\_diffusion**

Diffusion factor, the strength of the blurring effect

**Type** float in [0, inf], default 0.0

#### **ms\_intensity**

Multiplier for multiple scattered light energy

**Type** float in [0, inf], default 0.0

#### **ms\_spread**

Proportional distance over which the light is diffused

**Type** float in [0, inf], default 0.0

#### **reflection**

Multiplier to make out-scattered light brighter or darker (non-physically correct)

**Type** float in [0, inf], default 0.0

#### **reflection\_color**

Color of light scattered out of the volume (does not affect transmission)

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

#### **scattering**

Amount of light that gets scattered out by the volume - the more out-scattering, the shallower the light will penetrate

**Type** float in [0, inf], default 0.0

#### **step\_method**

Method of calculating the steps through the volume

**Type** enum in ['RANDOMIZED', 'CONSTANT'], default 'RANDOMIZED'

#### **step\_size**

Distance between subsequent volume depth samples

**Type** float in [0, inf], default 0.0

#### **transmission\_color**

Result color of the volume, after other light has been scattered/absorbed

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

#### **use\_external\_shadows**

Receive shadows from sources outside the volume (temporary)

**Type** boolean, default False

#### **use\_light\_cache**

Pre-calculate the shading information into a voxel grid, speeds up shading at slightly less accuracy

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Material.volume`

## 2.4.312 Menu(`bpy_struct`)

### Basic Menu Example

This script is a simple menu, menus differ from panels in that they must reference from a header, panel or another menu.

Notice the ‘CATEGORY\_MT\_name’ `Menu.bl_idname`, this is a naming convention for menus.

---

**Note:** Menu subclasses must be registered before referencing them from blender.

---

**Note:** Menu’s have their `Layout.operator_context` initialized as ‘EXEC\_REGION\_WIN’ rather than ‘INVOKE\_DEFAULT’, so if the operator context needs to initialize inputs from the `Operator.invoke` function then this needs to be explicitly set.

---

```
import bpy

class BasicMenu(bpy.types.Menu):
    bl_idname = "OBJECT_MT_select_test"
    bl_label = "Select"

    def draw(self, context):
```

```

        layout = self.layout

        layout.operator("object.select_all", text="Select/Deselect All")
        layout.operator("object.select_inverse", text="Inverse")
        layout.operator("object.select_random", text="Random")

bpy.utils.register_class(BasicMenu)

# test call to display immediately.
bpy.ops.wm.call_menu(name="OBJECT_MT_select_test")

```

## Submenus

This menu demonstrates some different functions.

```
import bpy
```

```

class SubMenu(bpy.types.Menu):
    bl_idname = "OBJECT_MT_select_submenu"
    bl_label = "Select"

    def draw(self, context):
        layout = self.layout

        layout.operator("object.select_all", text="Select/Deselect All")
        layout.operator("object.select_inverse", text="Inverse")
        layout.operator("object.select_random", text="Random")

        # access this operator as a submenu
        layout.operator_menu_enum("object.select_by_type", "type", text="Select All by Type...")

        layout.separator()

        # expand each operator option into this menu
        layout.operator_enum("object.lamp_add", "type")

        layout.separator()

        # use existing menu
        layout.menu("VIEW3D_MT_transform")

bpy.utils.register_class(SubMenu)

# test call to display immediately.
bpy.ops.wm.call_menu(name="OBJECT_MT_select_submenu")

```

## Extending Menus

When creating menus for addons you can't reference menus in blenders default scripts.

Instead the addon can add menu items to existing menus.

The function `menu_draw` acts like `Menu.draw`

```
import bpy
```

```
def menu_draw(self, context):  
    self.layout.operator("wm.save_homefile")
```

```
bpy.types.INFO_MT_file.append(menu_draw)
```

base class — `bpy_struct`

**class** `bpy.types.Menu` (*bpy\_struct*)  
Editor menu containing buttons

**bl\_idname**

If this is set, the menu gets a custom ID, otherwise it takes the name of the class used to define the menu (for example, if the class name is “OBJECT\_MT\_hello”, and `bl_idname` is not set by the script, then `bl_idname = “OBJECT_MT_hello”`)

**Type** string, default “”

**bl\_label**

The menu label

**Type** string, default “”

**layout**

Defines the structure of the menu in the UI

**Type** `UILayout`, (readonly)

**classmethod poll** (*context*)

If this method returns a non-null output, then the menu can be drawn

**Return type** boolean

**draw** (*context*)

Draw UI elements into the menu UI layout

**classmethod append** (*draw\_func*)

Append a draw function to this menu, takes the same arguments as the menu's draw function

**draw\_preset** (*context*)

Define these on the subclass - `preset_operator` - `preset_subdir`

**path\_menu** (*searchpaths, operator, props\_default={}*)

**classmethod prepend** (*draw\_func*)

Prepend a draw function to this menu, takes the same arguments as the menu's draw function

**classmethod remove** (*draw\_func*)

Remove a draw function that has been added to this menu

### Inherited Properties

- `bpy_struct.id_data`



## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.313 Mesh(ID)

base classes — `bpy_struct`, `ID`

**class** `bpy.types.Mesh` (*ID*)

Mesh datablock defining geometric surfaces

**animation\_data**

Animation data for this datablock

**Type** `AnimData`, (readonly)

**auto\_smooth\_angle**

Maximum angle between face normals that ‘Auto Smooth’ will operate on

**Type** float in [-inf, inf], default 0.0

**edges**

Edges of the mesh

**Type** `MeshEdges` `bpy_prop_collection` of `MeshEdge`, (readonly)

**faces**

Faces of the mesh

**Type** `MeshFaces` `bpy_prop_collection` of `MeshFace`, (readonly)

**layers\_float**

**Type** `bpy_prop_collection` of `MeshFloatPropertyLayer`, (readonly)

**layers\_int**

**Type** `bpy_prop_collection` of `MeshIntPropertyLayer`, (readonly)

**layers\_string**

**Type** `bpy_prop_collection` of `MeshStringPropertyLayer`, (readonly)

**materials**

**Type** `IDMaterials` `bpy_prop_collection` of `Material`, (readonly)

**shape\_keys**

**Type** `Key`, (readonly)

**show\_all\_edges**

Display all edges for wireframe in all view modes in the 3D view

**Type** boolean, default False

**show\_double\_sided**

Render/display the mesh with double or single sided lighting

**Type** boolean, default False

**show\_edge\_bevel\_weight**

Display weights created for the Bevel modifier

**Type** boolean, default False

**show\_edge\_crease**

Display creases created for subsurf weighting

**Type** boolean, default False

**show\_edge\_seams**

Display UV unwrapping seams

**Type** boolean, default False

**show\_edge\_sharp**

Display sharp edges, used with the EdgeSplit modifier

**Type** boolean, default False

**show\_edges**

Display selected edges using highlights in the 3D view and UV editor

**Type** boolean, default False

**show\_extra\_edge\_length**

Display selected edge lengths, using global values when set in the transform panel

**Type** boolean, default False

**show\_extra\_face\_angle**

Display the angles between the selected edges in degrees, using global values when set in the transform panel

**Type** boolean, default False

**show\_extra\_face\_area**

Display the area of selected faces, using global values when set in the transform panel

**Type** boolean, default False

**show\_faces**

Display all faces as shades in the 3D view and UV editor

**Type** boolean, default False

**show\_normal\_face**

Display face normals as lines

**Type** boolean, default False

**show\_normal\_vertex**

Display vertex normals as lines

**Type** boolean, default False

#### **sticky**

Sticky texture coordinates

**Type** `bpy_prop_collection` of `MeshSticky`, (readonly)

#### **texco\_mesh**

Derive texture coordinates from another mesh

**Type** `Mesh`

#### **texspace\_location**

Texture space location

**Type** float array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`

#### **texspace\_size**

Texture space size

**Type** float array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`

#### **texture\_mesh**

Use another mesh for texture indices (vertex indices must be aligned)

**Type** `Mesh`

#### **total\_edge\_sel**

Selected edge count in editmode

**Type** int in `[0, inf]`, default 0, (readonly)

#### **total\_face\_sel**

Selected face count in editmode

**Type** int in `[0, inf]`, default 0, (readonly)

#### **total\_vert\_sel**

Selected vertex count in editmode

**Type** int in `[0, inf]`, default 0, (readonly)

#### **use\_auto\_smooth**

Treat all set-smoothed faces with angles less than the specified angle as 'smooth' during render

**Type** boolean, default False

#### **use\_auto\_texspace**

Adjust active object's texture space automatically when transforming object

**Type** boolean, default False

#### **use\_mirror\_topology**

Use topology based mirroring (for when both sides of mesh have matching, unique topology)

**Type** boolean, default False

#### **use\_mirror\_x**

X Axis mirror editing

**Type** boolean, default False

#### **use\_paint\_mask**

Face selection masking for painting

**Type** boolean, default False

**use\_paint\_mask\_vertex**

Vertex selection masking for painting (weight paint only)

**Type** boolean, default False**uv\_texture\_clone**

UV texture to be used as cloning source

**Type** `MeshTextureFaceLayer`**uv\_texture\_clone\_index**

Clone UV texture index

**Type** int in [0, inf], default 0**uv\_texture\_stencil**

UV texture to mask the painted area

**Type** `MeshTextureFaceLayer`**uv\_texture\_stencil\_index**

Mask UV texture index

**Type** int in [0, inf], default 0**uv\_textures****Type** `UVTextures` bpy\_prop\_collection of `MeshTextureFaceLayer`, (readonly)**vertex\_colors****Type** `VertexColors` bpy\_prop\_collection of `MeshColorLayer`, (readonly)**vertices**

Vertices of the mesh

**Type** `MeshVertices` bpy\_prop\_collection of `MeshVertex`, (readonly)**edge\_keys**

(readonly)

**ttransform** (*matrix*)

Transform mesh vertices by a matrix

**Parameters** `matrix` (*float array of 16 items in [-inf, inf]*) – Matrix**calc\_normals** ()

Calculate vertex normals

**update** (*calc\_edges=False*)

update

**Parameters** `calc_edges` (*boolean, (optional)*) – Calculate Edges, Force recalculation of edges**validate** (*verbose=False*)

validate geometry, return True when the mesh has had invalid geometry corrected/removed

**Parameters** `verbose` (*boolean, (optional)*) – Verbose, Output information about the errors found**Returns** Result**Return type** boolean**from\_pydata** (*vertices, edges, faces*)

Make a mesh from a list of vertices/edges/faces Until we have a nicer way to make geometry, use this.

**Parameters**

- **vertices** (*iterable object*) – float triplets each representing (X, Y, Z) eg: [(0.0, 1.0, 0.5), ...].
- **edges** (*iterable object*) – int pairs, each pair contains two indices to the *vertices* argument. eg: [(1, 2), ...]
- **faces** (*iterable object*) – iterator of faces, each faces contains three or four indices to the *vertices* argument. eg: [(5, 6, 8, 9), (1, 2, 3), ...]

### Inherited Properties

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`

### References

- `BlendData.meshes`
- `BlendDataMeshes.new`
- `BlendDataMeshes.remove`
- `EditObjectActuator.mesh`
- `Mesh.texco_mesh`
- `Mesh.texture_mesh`
- `Object.to_mesh`

### 2.4.314 MeshColor(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.MeshColor` (*bpy\_struct*)

Vertex colors for a face in a Mesh

**color1**

**Type** float array of 3 items in [0, 1], default (0.0, 0.0, 0.0)

**color2**

**Type** float array of 3 items in [0, 1], default (0.0, 0.0, 0.0)

**color3**

**Type** float array of 3 items in [0, 1], default (0.0, 0.0, 0.0)

**color4**

**Type** float array of 3 items in [0, 1], default (0.0, 0.0, 0.0)

#### Inherited Properties

- `bpy_struct.id_data`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

#### References

- `MeshColorLayer.data`

### 2.4.315 MeshColorLayer(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.MeshColorLayer` (*bpy\_struct*)  
 Layer of vertex colors in a Mesh datablock

**active**  
 Sets the layer as active for display and editing  
**Type** boolean, default False

**active\_render**  
 Sets the layer as active for rendering  
**Type** boolean, default False

**data**  
**Type** `bpy_prop_collection` of `MeshColor`, (readonly)

**name**  
 Name of Vertex color layer  
**Type** string, default ""

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Mesh.vertex_colors`
- `VertexColors.active`
- `VertexColors.new`

## 2.4.316 MeshDeformModifier(Modifier)

base classes — `bpy_struct`, `Modifier`

**class** `bpy.types.MeshDeformModifier` (*Modifier*)  
Mesh deformation modifier to deform with other meshes

**invert\_vertex\_group**  
Invert vertex group influence  
**Type** boolean, default False

**is\_bound**  
Whether geometry has been bound to control cage  
**Type** boolean, default False, (readonly)

**object**  
Mesh object to deform with  
**Type** `Object`

**precision**  
The grid size for binding  
**Type** int in [2, 10], default 0

**use\_dynamic\_bind**  
Recompute binding dynamically on top of other deformers (slower and more memory consuming.)  
**Type** boolean, default False

**vertex\_group**  
Vertex group name  
**Type** string, default ""

### Inherited Properties

- `bpy_struct.id_data`
- `Modifier.name`
- `Modifier.use_apply_on_spline`
- `Modifier.show_in_editmode`
- `Modifier.show_expanded`
- `Modifier.show_on_cage`
- `Modifier.show_viewport`
- `Modifier.show_render`
- `Modifier.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`



- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.317 MeshEdge(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.MeshEdge` (*bpy\_struct*)

Edge in a Mesh datablock

**bevel\_weight**

Weight used by the Bevel modifier

**Type** float in [-inf, inf], default 0.0

**crease**

Weight used by the Subsurf modifier for creasing

**Type** float in [-inf, inf], default 0.0

**hide**

**Type** boolean, default False

**index**

Index number of the vertex

**Type** int in [0, inf], default 0, (readonly)

**is\_fgon**

Fgon edge

**Type** boolean, default False

**is\_loose**

Loose edge

**Type** boolean, default False

**select**

**Type** boolean, default False

**use\_edge\_sharp**

Sharp edge for the EdgeSplit modifier

**Type** boolean, default False

**use\_seam**

Seam edge for UV unwrapping

**Type** boolean, default False

**vertices**

Vertex indices

**Type** int array of 2 items in [0, inf], default (0, 0)

**key**

(readonly)

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Mesh.edges`

## 2.4.318 MeshEdges(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.MeshEdges` (*bpy\_struct*)  
Collection of mesh edges

**add** (*count=0*)  
add

**Parameters** `count` (*int in [0, inf], (optional)*) – Count, Number of vertices to add

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`

- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Mesh.edges`

### 2.4.319 MeshFace(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.MeshFace` (*bpy\_struct*)

Face in a Mesh datablock

#### **area**

Read only area of the face

**Type** float in [0, inf], default 0.0, (readonly)

#### **hide**

**Type** boolean, default False

#### **index**

Index number of the vertex

**Type** int in [0, inf], default 0, (readonly)

#### **material\_index**

**Type** int in [0, 32767], default 0

#### **normal**

Local space unit length normal vector for this face

**Type** float array of 3 items in [-1, 1], default (0.0, 0.0, 0.0), (readonly)

#### **select**

**Type** boolean, default False

#### **use\_smooth**

**Type** boolean, default False

#### **vertices**

Vertex indices

**Type** int array of 4 items in [0, inf], default (0, 0, 0, 0)

#### **vertices\_raw**

Fixed size vertex indices array

**Type** int array of 4 items in [0, inf], default (0, 0, 0, 0)

**center**

The midpoint of the face. (readonly)

**edge\_keys**

(readonly)

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `Mesh.faces`

## 2.4.320 MeshFaces(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.MeshFaces` (*bpy\_struct*)

Collection of mesh faces

**active**

The active face for this mesh

**Type** `int` in `[-inf, inf]`, default 0

**active\_tface**

Active Texture Face

**Type** `MeshTextureFace`, (readonly)

**add** (*count=0*)

add

**Parameters** `count` (*int* in `[0, inf]`, (*optional*)) – Count, Number of vertices to add

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Mesh.faces`

## 2.4.321 MeshFloatProperty(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.MeshFloatProperty` (*bpy\_struct*)

User defined floating point number value in a float properties layer

**value**

**Type** float in [-inf, inf], default 0.0

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`

- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `MeshFloatPropertyLayer.data`

## 2.4.322 MeshFloatPropertyLayer(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.MeshFloatPropertyLayer` (*bpy\_struct*)  
User defined layer of floating pointer number values

### data

**Type** `bpy_prop_collection` of `MeshFloatProperty`, (readonly)

### name

**Type** string, default ""

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Mesh.layers_float`

### 2.4.323 MeshIntProperty(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.MeshIntProperty` (*bpy\_struct*)  
User defined integer number value in an integer properties layer

**value**

**Type** `int` in `[-inf, inf]`, default 0

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `MeshIntPropertyLayer.data`

### 2.4.324 MeshIntPropertyLayer(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.MeshIntPropertyLayer` (*bpy\_struct*)  
User defined layer of integer number values

**data**

**Type** `bpy_prop_collection` of `MeshIntProperty`, (readonly)

**name**

**Type** string, default ""

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Mesh.layers_int`

## 2.4.325 MeshSticky(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.MeshSticky` (*bpy\_struct*)

Sticky texture coordinate

**co**

Sticky texture coordinate location

**Type** float array of 2 items in [-inf, inf], default (0.0, 0.0)

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`



- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Mesh.sticky`

## 2.4.326 MeshStringProperty(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.MeshStringProperty` (*bpy\_struct*)  
User defined string text value in a string properties layer

**value**

**Type** string, default ""

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `MeshStringPropertyLayer.data`

### 2.4.327 MeshStringPropertyLayer(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.MeshStringPropertyLayer` (*bpy\_struct*)  
User defined layer of string text values

#### **data**

**Type** `bpy_prop_collection` of `MeshStringProperty`, (readonly)

#### **name**

**Type** string, default ""

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Mesh.layers_string`

### 2.4.328 MeshTextureFace(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.MeshTextureFace` (*bpy\_struct*)  
UV mapping, texturing and game engine data for a face

**image****Type** `Image`**pin\_uv****Type** boolean array of 4 items, default (False, False, False, False)**select\_uv****Type** boolean array of 4 items, default (False, False, False, False)**uv****Type** float array of 8 items in `[-inf, inf]`, default (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)**uv1****Type** float array of 2 items in `[-inf, inf]`, default (0.0, 0.0)**uv2****Type** float array of 2 items in `[-inf, inf]`, default (0.0, 0.0)**uv3****Type** float array of 2 items in `[-inf, inf]`, default (0.0, 0.0)**uv4****Type** float array of 2 items in `[-inf, inf]`, default (0.0, 0.0)**uv\_raw**

Fixed size UV coordinates array

**Type** float array of 8 items in `[-inf, inf]`, default (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `MeshFaces.active_tface`
- `MeshTextureFaceLayer.data`

## 2.4.329 MeshTextureFaceLayer(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.MeshTextureFaceLayer` (*bpy\_struct*)

Layer of texture faces in a Mesh datablock

### **active**

Set the layer as active for display and editing

**Type** boolean, default False

### **active\_clone**

Set the layer as active for cloning

**Type** boolean, default False

### **active\_render**

Set the layer as active for rendering

**Type** boolean, default False

### **data**

**Type** `bpy_prop_collection` of `MeshTextureFace`, (readonly)

### **name**

Name of UV unwrapping layer

**Type** string, default ""

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`

- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Mesh.uv_texture_clone`
- `Mesh.uv_texture_stencil`
- `Mesh.uv_textures`
- `UVTextures.active`
- `UVTextures.new`

### 2.4.330 MeshVertex(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.MeshVertex` (*bpy\_struct*)

Vertex in a Mesh datablock

#### **bevel\_weight**

Weight used by the Bevel modifier ‘Only Vertices’ option

**Type** float in [-inf, inf], default 0.0

#### **co**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

#### **groups**

Weights for the vertex groups this vertex is member of

**Type** `bpy_prop_collection` of `VertexGroupElement`, (readonly)

#### **hide**

**Type** boolean, default False

#### **index**

Index number of the vertex

**Type** int in [0, inf], default 0, (readonly)

#### **normal**

Vertex Normal

**Type** float array of 3 items in [-1, 1], default (0.0, 0.0, 0.0)

#### **select**

**Type** boolean, default False

## Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Mesh.vertices`

## 2.4.331 MeshVertices(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.MeshVertices` (*bpy\_struct*)  
Collection of mesh vertices

**add** (*count=0*)  
add

**Parameters** `count` (*int in [0, inf], (optional)*) – Count, Number of vertices to add

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`

- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Mesh.vertices`

## 2.4.332 MessageActuator(Actuator)

base classes — `bpy_struct`, `Actuator`

**class** `bpy.types.MessageActuator` (*Actuator*)

### **body\_message**

Optional message body Text

**Type** string, default ""

### **body\_property**

The message body will be set by the Property Value

**Type** string, default ""

### **body\_type**

Toggle message type: either Text or a PropertyName

**Type** enum in ['TEXT', 'PROPERTY'], default 'TEXT'

### **subject**

Optional message subject. This is what can be filtered on

**Type** string, default ""

### **to\_property**

Optional send message to objects with this name only, or empty to broadcast

**Type** string, default ""

## Inherited Properties

- `bpy_struct.id_data`
- `Actuator.name`
- `Actuator.show_expanded`
- `Actuator.pin`
- `Actuator.type`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`

- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Actuator.link`
- `Actuator.unlink`

### 2.4.333 MessageSensor(Sensor)

base classes — `bpy_struct`, `Sensor`

**class** `bpy.types.MessageSensor` (*Sensor*)  
Sensor to detect incoming messages

**subject**

Optional subject filter: only accept messages with this subject, or empty for all

**Type** string, default ""

#### Inherited Properties

- `bpy_struct.id_data`
- `Sensor.name`
- `Sensor.show_expanded`
- `Sensor.frequency`
- `Sensor.invert`
- `Sensor.use_level`
- `Sensor.pin`
- `Sensor.use_pulse_false_level`
- `Sensor.use_pulse_true_level`
- `Sensor.use_tap`
- `Sensor.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`



- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Sensor.link`
- `Sensor.unlink`

### 2.4.334 MetaBall(ID)

base classes — `bpy_struct`, `ID`

**class** `bpy.types.MetaBall` (*ID*)

Metaball datablock to defined blobby surfaces

**animation\_data**

Animation data for this datablock

**Type** `AnimData`, (readonly)

**elements**

Meta elements

**Type** `MetaBallElements` `bpy_prop_collection` of `MetaElement`, (readonly)

**materials**

**Type** `IDMaterials` `bpy_prop_collection` of `Material`, (readonly)

**render\_resolution**

Polygonization resolution in rendering

**Type** float in [0.05, 1], default 0.0

**resolution**

Polygonization resolution in the 3D viewport

**Type** float in [0.05, 1], default 0.0

**texspace\_location**

Texture space location

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**texspace\_size**

Texture space size

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**threshold**

Influence of meta elements

**Type** float in [0, 5], default 0.0

**update\_method**

Metaball edit update behavior

- `UPDATE_ALWAYS` Always, While editing, update metaball always.
- `HALFRES` Half, While editing, update metaball in half resolution.
- `FAST` Fast, While editing, update metaball without polygonization.

- NEVER Never, While editing, don't update metaball at all.

**Type** enum in ['UPDATE\_ALWAYS', 'HALFRES', 'FAST', 'NEVER'], default 'UPDATE\_ALWAYS'

**use\_auto\_texspace**

Adjusts active object's texture space automatically when transforming object

**Type** boolean, default False

**Inherited Properties**

- bpy\_struct.id\_data
- ID.name
- ID.use\_fake\_user
- ID.is\_updated
- ID.is\_updated\_data
- ID.library
- ID.tag
- ID.users

**Inherited Functions**

- bpy\_struct.as\_pointer
- bpy\_struct.callback\_add
- bpy\_struct.callback\_remove
- bpy\_struct.driver\_add
- bpy\_struct.driver\_remove
- bpy\_struct.get
- bpy\_struct.is\_property\_hidden
- bpy\_struct.is\_property\_set
- bpy\_struct.items
- bpy\_struct.keyframe\_delete
- bpy\_struct.keyframe\_insert
- bpy\_struct.keys
- bpy\_struct.path\_from\_id
- bpy\_struct.path\_resolve
- bpy\_struct.type\_recast
- bpy\_struct.values
- ID.copy
- ID.user\_clear
- ID.animation\_data\_create
- ID.animation\_data\_clear
- ID.update\_tag

**References**

- BlendData.metaballs
- BlendDataMetaBalls.new
- BlendDataMetaBalls.remove

### 2.4.335 MetaBallElements(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.MetaBallElements` (*bpy\_struct*)

Collection of metaball elements

**active**

Last selected element

**Type** `MetaElement`, (readonly)

**new** (*type='BALL'*)

Add a new spline to the curve

**Parameters** **type** (*enum in ['BALL', 'CAPSULE', 'PLANE', 'ELLIPSOID', 'CUBE'], (optional)*) – type for the new meta-element

**Returns** The newly created meta-element

**Return type** `MetaElement`

**remove** (*element*)

Remove a spline from a curve

**Parameters** **element** (`MetaElement`, (never None)) – The element to remove

#### Inherited Properties

- `bpy_struct.id_data`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

#### References

- `MetaBall.elements`

## 2.4.336 MetaElement(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.MetaElement` (*bpy\_struct*)

Blobby element in a MetaBall datablock

**co**

**Type** float array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`

**hide**

Hide element

**Type** boolean, default `False`

**radius**

**Type** float in `[0, inf]`, default `0.0`

**rotation**

Normalized quaternion rotation

**Type** float array of 4 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0, 0.0)`

**size\_x**

Size of element, use of components depends on element type

**Type** float in `[0, 20]`, default `0.0`

**size\_y**

Size of element, use of components depends on element type

**Type** float in `[0, 20]`, default `0.0`

**size\_z**

Size of element, use of components depends on element type

**Type** float in `[0, 20]`, default `0.0`

**stiffness**

Stiffness defines how much of the element to fill

**Type** float in `[0, 10]`, default `0.0`

**type**

Metaball types

**Type** enum in `['BALL', 'CAPSULE', 'PLANE', 'ELLIPSOID', 'CUBE']`, default `'BALL'`

**use\_negative**

Set metaball as negative one

**Type** boolean, default `False`

### Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `MetaBall.elements`
- `MetaBallElements.active`
- `MetaBallElements.new`
- `MetaBallElements.remove`

### 2.4.337 MetaSequence(Sequence)

base classes — `bpy_struct`, `Sequence`

**class** `bpy.types.MetaSequence` (*Sequence*)  
Sequence strip to group other strips as a single sequence strip

**animation\_offset\_end**

Animation end offset (trim end)

**Type** `int` in `[0, inf]`, default `0`

**animation\_offset\_start**

Animation start offset (trim start)

**Type** `int` in `[0, inf]`, default `0`

**color\_balance**

**Type** `SequenceColorBalance`, (readonly)

**color\_multiply**

**Type** `float` in `[0, 20]`, default `0.0`

**color\_saturation**

**Type** `float` in `[0, 20]`, default `0.0`

**crop**

**Type** `SequenceCrop`, (readonly)

**proxy**

**Type** `SequenceProxy`, (readonly)

**sequences**

**Type** `bpy_prop_collection` of `Sequence`, (readonly)

**strobe**

Only display every nth frame

**Type** float in [1, 30], default 0.0

**transform**

**Type** `SequenceTransform`, (readonly)

**use\_color\_balance**

(3-Way color correction) on input

**Type** boolean, default False

**use\_crop**

Crop image before processing

**Type** boolean, default False

**use\_deinterlace**

For video movies to remove fields

**Type** boolean, default False

**use\_flip\_x**

Flip on the X axis

**Type** boolean, default False

**use\_flip\_y**

Flip on the Y axis

**Type** boolean, default False

**use\_float**

Convert input to float data

**Type** boolean, default False

**use\_premultiply**

Convert RGB from key alpha to premultiplied alpha

**Type** boolean, default False

**use\_proxy**

Use a preview proxy and/or timecode index for this strip

**Type** boolean, default False

**use\_proxy\_custom\_directory**

Use a custom directory to store data

**Type** boolean, default False

**use\_proxy\_custom\_file**

Use a custom file to read proxy data from

**Type** boolean, default False

**use\_reverse\_frames**

Reverse frame order

**Type** boolean, default False**use\_translation**

Translate image before processing

**Type** boolean, default False**Inherited Properties**

- `bpy_struct.id_data`
- `Sequence.name`
- `Sequence.blend_type`
- `Sequence.blend_alpha`
- `Sequence.channel`
- `Sequence.waveform`
- `Sequence.effect_fader`
- `Sequence.frame_final_end`
- `Sequence.frame_offset_end`
- `Sequence.frame_still_end`
- `Sequence.input_1`
- `Sequence.input_2`
- `Sequence.input_3`
- `Sequence.select_left_handle`
- `Sequence.frame_final_duration`
- `Sequence.frame_duration`
- `Sequence.lock`
- `Sequence.mute`
- `Sequence.select_right_handle`
- `Sequence.select`
- `Sequence.speed_factor`
- `Sequence.frame_start`
- `Sequence.frame_final_start`
- `Sequence.frame_offset_start`
- `Sequence.frame_still_start`
- `Sequence.type`
- `Sequence.use_default_fade`
- `Sequence.input_count`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`

- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Sequence.getStripElem`
- `Sequence.swap`

### 2.4.338 MirrorModifier(Modifier)

base classes — `bpy_struct`, `Modifier`

**class** `bpy.types.MirrorModifier` (*Modifier*)

Mirroring modifier

**merge\_threshold**

Distance from axis within which mirrored vertices are merged

**Type** float in [0, inf], default 0.0

**mirror\_object**

Object to use as mirror

**Type** `Object`

**use\_clip**

Prevents vertices from going through the mirror during transform

**Type** boolean, default False

**use\_mirror\_merge**

Merge vertices within the merge threshold

**Type** boolean, default False

**use\_mirror\_u**

Mirror the U texture coordinate around the 0.5 point

**Type** boolean, default False

**use\_mirror\_v**

Mirror the V texture coordinate around the 0.5 point

**Type** boolean, default False

**use\_mirror\_vertex\_groups**

Mirror vertex groups (e.g. .R->.L)

**Type** boolean, default False

**use\_x**

Enable X axis mirror

**Type** boolean, default False

**use\_y**

Enable Y axis mirror

**Type** boolean, default False

**use\_z**

Enable Z axis mirror

**Type** boolean, default False



## Inherited Properties

- `bpy_struct.id_data`
- `Modifier.name`
- `Modifier.use_apply_on_spline`
- `Modifier.show_in_editmode`
- `Modifier.show_expanded`
- `Modifier.show_on_cage`
- `Modifier.show_viewport`
- `Modifier.show_render`
- `Modifier.type`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.339 Modifier(`bpy_struct`)

base class — `bpy_struct`

subclasses — `FluidSimulationModifier`, `BevelModifier`, `SolidifyModifier`, `SmokeModifier`, `UVProjectModifier`, `VertexWeightProximityModifier`, `VertexWeightEditModifier`, `DecimateModifier`, `ExplodeModifier`, `SmoothModifier`, `HookModifier`, `SoftBodyModifier`, `BooleanModifier`, `ArrayModifier`, `LatticeModifier`, `BuildModifier`, `ClothModifier`, `ParticleSystemModifier`, `SubsurfModifier`, `CurveModifier`, `ScrewModifier`, `MaskModifier`, `ShrinkwrapModifier`, `CollisionModifier`, `CastModifier`, `WaveModifier`, `ParticleInstanceModifier`, `SurfaceModifier`, `SimpleDeformModifier`, `DisplaceModifier`, `ArmatureModifier`, `MeshDeformModifier`, `WarpModifier`, `EdgeSplitModifier`, `MultiresModifier`, `MirrorModifier`, `VertexWeightMixModifier`

**class** `bpy.types.Modifier` (`bpy_struct`)

Modifier affecting the geometry data of an object

**name**

Modifier name

**Type** string, default ""

**show\_expanded**

Set modifier expanded in the user interface

**Type** boolean, default False

**show\_in\_editmode**

Use modifier while in the edit mode

**Type** boolean, default False

**show\_on\_cage**

Enable direct editing of modifier control cage

**Type** boolean, default False

**show\_render**

Use modifier during rendering

**Type** boolean, default False

**show\_viewport**

Realtime display of a modifier

**Type** boolean, default False

**type**

**Type** enum in ['UV\_PROJECT', 'VERTEX\_WEIGHT\_EDIT', 'VERTEX\_WEIGHT\_MIX', 'VERTEX\_WEIGHT\_PROXIMITY', 'ARRAY', 'BEVEL', 'BOOLEAN', 'BUILD', 'DECIMATE', 'EDGE\_SPLIT', 'MASK', 'MIRROR', 'MULTIRES', 'SCREW', 'SOLIDIFY', 'SUBSURF', 'ARMATURE', 'CAST', 'CURVE', 'DISPLACE', 'HOOK', 'LATTICE', 'MESH\_DEFORM', 'SHRINKWRAP', 'SIMPLE\_DEFORM', 'SMOOTH', 'WARP', 'WAVE', 'CLOTH', 'COLLISION', 'EXPLODE', 'FLUID\_SIMULATION', 'PARTICLE\_INSTANCE', 'PARTICLE\_SYSTEM', 'SMOKE', 'SOFT\_BODY', 'SURFACE'], default 'UV\_PROJECT', (readonly)

**use\_apply\_on\_spline**

Apply this and all preceding deformation modifiers on splines' points rather than on filled curve/surface

**Type** boolean, default False

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`

- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Object.modifiers`
- `ObjectModifiers.new`
- `ObjectModifiers.remove`
- `UILayout.template_modifier`

## 2.4.340 MotionPath(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.MotionPath` (*bpy\_struct*)

Cache of the workspace positions of an element over a frame range

### **frame\_end**

End frame of the stored range

**Type** `int` in `[-inf, inf]`, default 0, (readonly)

### **frame\_start**

Starting frame of the stored range

**Type** `int` in `[-inf, inf]`, default 0, (readonly)

### **is\_modified**

Path is being edited

**Type** `boolean`, default False

### **length**

Number of frames cached

**Type** `int` in `[-inf, inf]`, default 0, (readonly)

### **points**

Cached positions per frame

**Type** `bpy_prop_collection` of `MotionPathVert`, (readonly)

### **use\_bone\_head**

For PoseBone paths, use the bone head location when calculating this path

**Type** `boolean`, default False, (readonly)

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`

- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Object.motion_path`
- `PoseBone.motion_path`

## 2.4.341 MotionPathVert(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.MotionPathVert` (*`bpy_struct`*)

    Cached location on path

**co**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**select**

        Path point is selected for editing

**Type** boolean, default False

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`

- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `MotionPath.points`

## 2.4.342 MouseSensor(Sensor)

base classes — `bpy_struct`, `Sensor`

**class** `bpy.types.MouseSensor` (*Sensor*)  
Sensor to detect mouse events

### **mouse\_event**

Specify the type of event this mouse sensor should trigger on

**Type** enum in ['LEFTCLICK', 'MIDDLECLICK', 'RIGHTCLICK', 'WHEELUP', 'WHEEL-DOWN', 'MOVEMENT', 'MOUSEOVER', 'MOUSEOVERANY'], default 'LEFTCLICK'

## Inherited Properties

- `bpy_struct.id_data`
- `Sensor.name`
- `Sensor.show_expanded`
- `Sensor.frequency`
- `Sensor.invert`
- `Sensor.use_level`
- `Sensor.pin`
- `Sensor.use_pulse_false_level`
- `Sensor.use_pulse_true_level`
- `Sensor.use_tap`
- `Sensor.type`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`

- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Sensor.link`
- `Sensor.unlink`

### 2.4.343 MovieSequence(Sequence)

base classes — `bpy_struct`, `Sequence`

**class** `bpy.types.MovieSequence` (*Sequence*)

Sequence strip to load a video

**animation\_offset\_end**

Animation end offset (trim end)

**Type** `int` in `[0, inf]`, default `0`

**animation\_offset\_start**

Animation start offset (trim start)

**Type** `int` in `[0, inf]`, default `0`

**color\_balance**

**Type** `SequenceColorBalance`, (readonly)

**color\_multiply**

**Type** `float` in `[0, 20]`, default `0.0`

**color\_saturation**

**Type** `float` in `[0, 20]`, default `0.0`

**crop**

**Type** `SequenceCrop`, (readonly)

**elements**

**Type** `bpy_prop_collection` of `SequenceElement`, (readonly)

**filepath**

**Type** `string`, default `""`

**mpeg\_preseek**

For MPEG movies, preseek this many frames

**Type** `int` in `[0, 50]`, default `0`

**proxy**

**Type** `SequenceProxy`, (readonly)

**stream\_index**

For files with several movie streams, use the stream with the given index

**Type** `int` in `[0, 20]`, default `0`

**strobe**

Only display every nth frame

**Type** `float` in `[1, 30]`, default `0.0`

**transform**

**Type** `SequenceTransform`, (readonly)

**use\_color\_balance**

(3-Way color correction) on input

**Type** boolean, default False

**use\_crop**

Crop image before processing

**Type** boolean, default False

**use\_deinterlace**

For video movies to remove fields

**Type** boolean, default False

**use\_flip\_x**

Flip on the X axis

**Type** boolean, default False

**use\_flip\_y**

Flip on the Y axis

**Type** boolean, default False

**use\_float**

Convert input to float data

**Type** boolean, default False

**use\_premultiply**

Convert RGB from key alpha to premultiplied alpha

**Type** boolean, default False

**use\_proxy**

Use a preview proxy and/or timecode index for this strip

**Type** boolean, default False

**use\_proxy\_custom\_directory**

Use a custom directory to store data

**Type** boolean, default False

**use\_proxy\_custom\_file**

Use a custom file to read proxy data from

**Type** boolean, default False

**use\_reverse\_frames**

Reverse frame order

**Type** boolean, default False

**use\_translation**

Translate image before processing

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `Sequence.name`
- `Sequence.blend_type`
- `Sequence.blend_alpha`
- `Sequence.channel`
- `Sequence.waveform`
- `Sequence.effect_fader`
- `Sequence.frame_final_end`
- `Sequence.frame_offset_end`
- `Sequence.frame_still_end`
- `Sequence.input_1`
- `Sequence.input_2`
- `Sequence.input_3`
- `Sequence.select_left_handle`
- `Sequence.frame_final_duration`
- `Sequence.frame_duration`
- `Sequence.lock`
- `Sequence.mute`
- `Sequence.select_right_handle`
- `Sequence.select`
- `Sequence.speed_factor`
- `Sequence.frame_start`
- `Sequence.frame_final_start`
- `Sequence.frame_offset_start`
- `Sequence.frame_still_start`
- `Sequence.type`
- `Sequence.use_default_fade`
- `Sequence.input_count`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Sequence.getStripElem`
- `Sequence.swap`



## 2.4.344 MulticamSequence(Sequence)

base classes — `bpy_struct`, `Sequence`

**class** `bpy.types.MulticamSequence` (*Sequence*)

Sequence strip to perform multicam editing: select channel from below

**animation\_offset\_end**

Animation end offset (trim end)

**Type** `int` in `[0, inf]`, default `0`

**animation\_offset\_start**

Animation start offset (trim start)

**Type** `int` in `[0, inf]`, default `0`

**color\_balance**

**Type** `SequenceColorBalance`, (readonly)

**color\_multiply**

**Type** `float` in `[0, 20]`, default `0.0`

**color\_saturation**

**Type** `float` in `[0, 20]`, default `0.0`

**crop**

**Type** `SequenceCrop`, (readonly)

**multicam\_source**

**Type** `int` in `[0, 31]`, default `0`

**proxy**

**Type** `SequenceProxy`, (readonly)

**strobe**

Only display every nth frame

**Type** `float` in `[1, 30]`, default `0.0`

**ttransform**

**Type** `SequenceTransform`, (readonly)

**use\_color\_balance**

(3-Way color correction) on input

**Type** `boolean`, default `False`

**use\_crop**

Crop image before processing

**Type** `boolean`, default `False`

**use\_deinterlace**

For video movies to remove fields

**Type** `boolean`, default `False`

**use\_flip\_x**

Flip on the X axis

**Type** boolean, default False

**use\_flip\_y**

Flip on the Y axis

**Type** boolean, default False

**use\_float**

Convert input to float data

**Type** boolean, default False

**use\_premultiply**

Convert RGB from key alpha to premultiplied alpha

**Type** boolean, default False

**use\_proxy**

Use a preview proxy and/or timecode index for this strip

**Type** boolean, default False

**use\_proxy\_custom\_directory**

Use a custom directory to store data

**Type** boolean, default False

**use\_proxy\_custom\_file**

Use a custom file to read proxy data from

**Type** boolean, default False

**use\_reverse\_frames**

Reverse frame order

**Type** boolean, default False

**use\_translation**

Translate image before processing

**Type** boolean, default False

## Inherited Properties

- `bpy_struct.id_data`
- `Sequence.name`
- `Sequence.blend_type`
- `Sequence.blend_alpha`
- `Sequence.channel`
- `Sequence.waveform`
- `Sequence.effect_fader`
- `Sequence.frame_final_end`
- `Sequence.frame_offset_end`
- `Sequence.frame_still_end`
- `Sequence.input_1`
- `Sequence.input_2`
- `Sequence.input_3`
- `Sequence.select_left_handle`
- `Sequence.frame_final_duration`
- `Sequence.frame_duration`
- `Sequence.lock`

- `Sequence.mute`
- `Sequence.select_right_handle`
- `Sequence.select`
- `Sequence.speed_factor`
- `Sequence.frame_start`
- `Sequence.frame_final_start`
- `Sequence.frame_offset_start`
- `Sequence.frame_still_start`
- `Sequence.type`
- `Sequence.use_default_fade`
- `Sequence.input_count`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Sequence.getStripElem`
- `Sequence.swap`

## 2.4.345 MultiresModifier(Modifier)

base classes — `bpy_struct`, `Modifier`

**class** `bpy.types.MultiresModifier` (*Modifier*)

Multiresolution mesh modifier

#### **filepath**

Path to external displacements file

**Type** string, default ""

#### **is\_external**

Store multires displacements outside the .blend file, to save memory

**Type** boolean, default False, (readonly)

#### **levels**

Number of subdivisions to use in the viewport

**Type** int in [0, 255], default 0

**render\_levels**

The subdivision level visible at render time

**Type** int in [0, 255], default 0

**sculpt\_levels**

Number of subdivisions to use in sculpt mode

**Type** int in [0, 255], default 0

**show\_only\_control\_edges**

Skip drawing/rendering of interior subdivided edges

**Type** boolean, default False

**subdivision\_type**

Selects type of subdivision algorithm

**Type** enum in ['CATMULL\_CLARK', 'SIMPLE'], default 'CATMULL\_CLARK'

**total\_levels**

Number of subdivisions for which displacements are stored

**Type** int in [0, 255], default 0, (readonly)

**use\_subsurf\_uv**

Use subsurf to subdivide UVs

**Type** boolean, default False

**Inherited Properties**

- `bpy_struct.id_data`
- `Modifier.name`
- `Modifier.use_apply_on_spline`
- `Modifier.show_in_editmode`
- `Modifier.show_expanded`
- `Modifier.show_on_cage`
- `Modifier.show_viewport`
- `Modifier.show_render`
- `Modifier.type`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`

- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.346 MusgraveTexture(Texture)

base classes — `bpy_struct`, `ID`, `Texture`

**class** `bpy.types.MusgraveTexture` (*Texture*)  
Procedural musgrave texture

### **dimension\_max**

Highest fractal dimension

**Type** float in [0.0001, 2], default 0.0

### **gain**

The gain multiplier

**Type** float in [0, 6], default 0.0

### **lacunarity**

Gap between successive frequencies

**Type** float in [0, 6], default 0.0

### **musgrave\_type**

- `MULTIFRACTAL` Multifractal, Fractal noise algorithm. Multifractal: Uses Perlin noise as a basis.
- `RIDGED_MULTIFRACTAL` Ridged Multifractal, Fractal noise algorithm. Ridged Multifractal: Uses Perlin noise with inflection as a basis.
- `HYBRID_MULTIFRACTAL` Hybrid Multifractal, Fractal noise algorithm. Hybrid Multifractal: Uses Perlin noise as a basis, with extended controls.
- `FBM` fBM, Fractal noise algorithm. Fractal Brownian Motion: Uses Brownian noise as a basis.
- `HETERO_TERRAIN` Hetero Terrain, Fractal noise algorithm. Hetero Terrain: similar to multifractal.

**Type** enum in ['MULTIFRACTAL', 'RIDGED\_MULTIFRACTAL', 'HYBRID\_MULTIFRACTAL', 'FBM', 'HETERO\_TERRAIN'], default 'MULTIFRACTAL'

### **nabla**

Size of derivative offset used for calculating normal

**Type** float in [0.001, 0.1], default 0.0

### **noise\_basis**

Sets the noise basis used for turbulence

- `BLENDER_ORIGINAL` Blender Original, Noise algorithm - Blender original: Smooth interpolated noise.
- `ORIGINAL_PERLIN` Original Perlin, Noise algorithm - Original Perlin: Smooth interpolated noise.
- `IMPROVED_PERLIN` Improved Perlin, Noise algorithm - Improved Perlin: Smooth interpolated noise.
- `VORONOI_F1` Voronoi F1, Noise algorithm - Voronoi F1: Returns distance to the closest feature point.
- `VORONOI_F2` Voronoi F2, Noise algorithm - Voronoi F2: Returns distance to the 2nd closest feature point.

- VORONOI\_F3 Voronoi F3, Noise algorithm - Voronoi F3: Returns distance to the 3rd closest feature point.
- VORONOI\_F4 Voronoi F4, Noise algorithm - Voronoi F4: Returns distance to the 4th closest feature point.
- VORONOI\_F2\_F1 Voronoi F2-F1, Noise algorithm - Voronoi F1-F2.
- VORONOI\_CRACKLE Voronoi Crackle, Noise algorithm - Voronoi Crackle: Voronoi tessellation with sharp edges.
- CELL\_NOISE Cell Noise, Noise algorithm - Cell Noise: Square cell tessellation.

**Type** enum in ['BLENDER\_ORIGINAL', 'ORIGINAL\_PERLIN', 'IMPROVED\_PERLIN', 'VORONOI\_F1', 'VORONOI\_F2', 'VORONOI\_F3', 'VORONOI\_F4', 'VORONOI\_F2\_F1', 'VORONOI\_CRACKLE', 'CELL\_NOISE'], default 'BLENDER\_ORIGINAL'

**noise\_intensity**

Scales the intensity of the noise

**Type** float in [0, 10], default 0.0

**noise\_scale**

Sets scaling for noise input

**Type** float in [0.0001, inf], default 0.0

**octaves**

Number of frequencies used

**Type** float in [0, 8], default 0.0

**offset**

The fractal offset

**Type** float in [0, 6], default 0.0

**users\_material**

Materials that use this texture (readonly)

**users\_object\_modifier**

Object modifiers that use this texture (readonly)

**Inherited Properties**

- bpy\_struct.id\_data
- ID.name
- ID.use\_fake\_user
- ID.is\_updated
- ID.is\_updated\_data
- ID.library
- ID.tag
- ID.users
- Texture.animation\_data
- Texture.intensity
- Texture.color\_ramp
- Texture.contrast
- Texture.factor\_blue

- `Texture.factor_green`
- `Texture.factor_red`
- `Texture.node_tree`
- `Texture.saturation`
- `Texture.use_preview_alpha`
- `Texture.type`
- `Texture.use_color_ramp`
- `Texture.use_nodes`
- `Texture.users_material`
- `Texture.users_object_modifier`
- `Texture.users_material`
- `Texture.users_object_modifier`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`
- `Texture.evaluate`

## 2.4.347 NandController(Controller)

base classes — `bpy_struct`, `Controller`

**class** `bpy.types.NandController` (*Controller*)  
 Controller passing on events based on a logical NAND operation

### Inherited Properties

- `bpy_struct.id_data`
- `Controller.name`
- `Controller.states`
- `Controller.show_expanded`
- `Controller.use_priority`

- `Controller.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Controller.link`
- `Controller.unlink`

## 2.4.348 NearSensor(Sensor)

base classes — `bpy_struct`, `Sensor`

**class** `bpy.types.NearSensor` (*Sensor*)

Sensor to detect nearby objects

### **distance**

Trigger distance

**Type** float in [0, 10000], default 0.0

### **property**

Only look for objects with this property (blank = all objects)

**Type** string, default ""

### **reset\_distance**

The distance where the sensor forgets the actor

**Type** float in [0, 10000], default 0.0

### Inherited Properties

- `bpy_struct.id_data`
- `Sensor.name`
- `Sensor.show_expanded`
- `Sensor.frequency`
- `Sensor.invert`
- `Sensor.use_level`
- `Sensor.pin`
- `Sensor.use_pulse_false_level`



- `Sensor.use_pulse_true_level`
- `Sensor.use_tap`
- `Sensor.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Sensor.link`
- `Sensor.unlink`

## 2.4.349 NlaStrip(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.NlaStrip` (*bpy\_struct*)  
A container referencing an existing Action

#### **action**

Action referenced by this strip

**Type** `Action`

#### **action\_frame\_end**

**Type** float in [-inf, inf], default 0.0

#### **action\_frame\_start**

**Type** float in [-inf, inf], default 0.0

#### **active**

NLA Strip is active

**Type** boolean, default False, (readonly)

#### **blend\_in**

Number of frames at start of strip to fade in influence

**Type** float in [-inf, inf], default 0.0

#### **blend\_out**

**Type** float in [-inf, inf], default 0.0

**blend\_type**

Method used for combining strip's result with accumulated result

- REPLACE Replace, Result strip replaces the accumulated results by amount specified by influence.
- ADD Add, Weighted result of strip is added to the accumulated results.
- SUBTRACT Subtract, Weighted result of strip is removed from the accumulated results.
- MULITPLY Multiply, Weighted result of strip is multiplied with the accumulated results.

**Type** enum in ['REPLACE', 'ADD', 'SUBTRACT', 'MULITPLY'], default 'REPLACE'

**extrapolation**

Action to take for gaps past the strip extents

- NOTHING Nothing, Strip has no influence past its extents.
- HOLD Hold, Hold the first frame if no previous strips in track, and always hold last frame.
- HOLD\_FORWARD Hold Forward, Only hold last frame.

**Type** enum in ['NOTHING', 'HOLD', 'HOLD\_FORWARD'], default 'HOLD'

**fcurves**

F-Curves for controlling the strip's influence and timing

**Type** bpy\_prop\_collection of FCurve, (readonly)

**frame\_end**

**Type** float in [-inf, inf], default 0.0

**frame\_start**

**Type** float in [-inf, inf], default 0.0

**influence**

Amount the strip contributes to the current result

**Type** float in [0, 1], default 0.0

**modifiers**

Modifiers affecting all the F-Curves in the referenced Action

**Type** bpy\_prop\_collection of FModifier, (readonly)

**mute**

NLA Strip is not evaluated

**Type** boolean, default False

**name**

**Type** string, default ""

**repeat**

Number of times to repeat the action range

**Type** float in [0.1, 1000], default 0.0

**scale**

Scaling factor for action

**Type** float in [0.0001, 1000], default 0.0

**select**

NLA Strip is selected

**Type** boolean, default False

**strip\_time**

Frame of referenced Action to evaluate

**Type** float in  $[-inf, inf]$ , default 0.0

**strips**

NLA Strips that this strip acts as a container for (if it is of type Meta)

**Type** `bpy_prop_collection` of `NlaStrip`, (readonly)

**type**

Type of NLA Strip

- **CLIP** Action Clip, NLA Strip references some Action.
- **TRANSITION** Transition, NLA Strip ‘transitions’ between adjacent strips.
- **META** Meta, NLA Strip acts as a container for adjacent strips.
- **SOUND** Sound Clip, NLA Strip representing a sound event for speakers.

**Type** enum in  $['CLIP', 'TRANSITION', 'META', 'SOUND']$ , default ‘CLIP’, (readonly)

**use\_animated\_influence**

Influence setting is controlled by an F-Curve rather than automatically determined

**Type** boolean, default False

**use\_animated\_time**

Strip time is controlled by an F-Curve rather than automatically determined

**Type** boolean, default False

**use\_animated\_time\_cyclic**

Cycle the animated time within the action start & end

**Type** boolean, default False

**use\_auto\_blend**

Number of frames for Blending In/Out is automatically determined from overlapping strips

**Type** boolean, default False

**use\_reverse**

NLA Strip is played back in reverse order (only when timing is automatically determined)

**Type** boolean, default False

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`

- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `NlaStrip.strips`
- `NlaStrips.new`
- `NlaStrips.remove`
- `NlaTrack.strips`

## 2.4.350 NlaStrips(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.NlaStrips` (*bpy\_struct*)  
Collection of Nla Strips

**new** (*name, start, action*)  
Add a new Action-Clip strip to the track

### Parameters

- **name** (*string*) – Name for the NLA Strips
- **start** (*int in [-inf, inf]*) – Start Frame, Start frame for this strip
- **action** (*Action, (never None)*) – Action to assign to this strip

**Returns** New NLA Strip

**Return type** `NlaStrip`

**remove** (*strip*)  
Remove a NLA Strip

**Parameters** **strip** (`NlaStrip, (never None)`) – NLA Strip to remove

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`

- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `NlaTrack.strips`

### 2.4.351 NlaTrack(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.NlaTrack` (*bpy\_struct*)

A animation layer containing Actions referenced as NLA strips

#### **active**

NLA Track is active

**Type** boolean, default False, (readonly)

#### **is\_solo**

NLA Track is evaluated itself (i.e. active Action and all other NLA Tracks in the same AnimData block are disabled)

**Type** boolean, default False, (readonly)

#### **lock**

NLA Track is locked

**Type** boolean, default False

#### **mute**

NLA Track is not evaluated

**Type** boolean, default False

#### **name**

**Type** string, default ""

#### **select**

NLA Track is selected

**Type** boolean, default False

#### **strips**

NLA Strips on this NLA-track

**Type** `NlaStrips bpy_prop_collection` of `NlaStrip`, (readonly)

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `AnimData.nla_tracks`
- `NlaTracks.active`
- `NlaTracks.new`
- `NlaTracks.new`
- `NlaTracks.remove`

## 2.4.352 `NlaTracks(bpy_struct)`

base class — `bpy_struct`

**class** `bpy.types.NlaTracks` (*bpy\_struct*)  
Collection of NLA Tracks

#### **active**

Active Object constraint

**Type** `NlaTrack`

#### **new** (*prev=None*)

Add a new NLA Track

**Parameters** `prev` (`NlaTrack`, (optional)) – NLA Track to add the new one after

**Returns** New NLA Track

**Return type** `NlaTrack`

**remove** (*track*)

Remove a NLA Track

**Parameters** **track** (`NlaTrack`, (never None)) – NLA Track to remove**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `AnimData.nla_tracks`

**2.4.353 Node(`bpy_struct`)**base class — `bpy_struct`subclasses — `NodeFrame`, `NodeWhileLoop`, `TextureNode`, `ShaderNode`, `CompositorNode`, `NodeForLoop`, `NodeGroup`**class** `bpy.types.Node` (*bpy\_struct*)

Node in a node tree

**inputs****Type** `bpy_prop_collection` of `NodeSocket`, (readonly)**label**

Optional custom node label

**Type** string, default ""**location****Type** float array of 2 items in [-10000, 10000], default (0.0, 0.0)

**name**

Unique node identifier

**Type** string, default ""

**outputs**

**Type** bpy\_prop\_collection of NodeSocket, (readonly)

**parent**

Parent this node is attached to

**Type** Node, (readonly)

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `CompositorNodeTree.nodes`
- `CompositorNodes.new`
- `CompositorNodes.remove`
- `Node.parent`
- `NodeLink.from_node`
- `NodeLink.to_node`
- `ShaderNodeTree.nodes`
- `ShaderNodes.new`
- `ShaderNodes.remove`
- `TextureNodeTree.nodes`
- `TextureNodes.new`
- `TextureNodes.remove`



## 2.4.354 NodeForLoop(Node)

base classes — `bpy_struct`, `Node`

**class** `bpy.types.NodeForLoop` (*Node*)

**node\_tree**

**Type** `NodeTree`

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.355 NodeFrame(Node)

base classes — `bpy_struct`, `Node`

**class** `bpy.types.NodeFrame` (*Node*)

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`

- `Node.location`
- `Node.outputs`
- `Node.parent`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.356 NodeGroup(Node)

base classes — `bpy_struct`, `Node`

**class** `bpy.types.NodeGroup` (*Node*)

**node\_tree**

**Type** `NodeTree`

#### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`

- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.357 NodeLink(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.NodeLink` (*bpy\_struct*)

Link between nodes in a node tree

**from\_node**

**Type** `Node`, (readonly)

**from\_socket**

**Type** `NodeSocket`, (readonly)

**to\_node**

**Type** `Node`, (readonly)

**to\_socket**

**Type** `NodeSocket`, (readonly)

#### Inherited Properties

- `bpy_struct.id_data`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `NodeLinks.new`
- `NodeLinks.remove`
- `NodeTree.links`

## 2.4.358 NodeLinks(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.NodeLinks` (*bpy\_struct*)  
Collection of Node Links

**new** (*input*, *output*)  
Add a node link to this node tree

### Parameters

- **input** (`NodeSocket`) – The input socket
- **output** (`NodeSocket`) – The output socket

**Returns** New node link

**Return type** `NodeLink`

**remove** (*link*)  
remove a node link from the node tree

**Parameters** **link** (`NodeLink`) – The node link to remove

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `NodeTree.links`

## 2.4.359 NodeSocket(bpy\_struct)

base class — `bpy_struct`

subclasses — `NodeSocketVectorAcceleration`, `NodeSocketVectorTranslation`,  
`NodeSocketFloatPercentage`, `NodeSocketFloatAngle`, `NodeSocketIntUnsigned`,  
`NodeSocketVectorEuler`, `NodeSocketFloatNone`, `NodeSocketFloatTime`,  
`NodeSocketIntNone`, `NodeSocketFloatFactor`, `NodeSocketVectorXYZ`, `NodeSocketShader`,  
`NodeSocketFloatDistance`, `NodeSocketFloatUnsigned`, `NodeSocketVectorNone`,  
`NodeSocketRGBA`, `NodeSocketBoolean`, `NodeSocketVectorDirection`,  
`NodeSocketVectorVelocity`

**class** `bpy.types.NodeSocket` (*bpy\_struct*)

Input or output socket of a node

**name**

Socket name

**Type** string, default ""

**type**

Node Socket type

**Type** enum in ['VALUE', 'VECTOR', 'RGBA'], default 'VALUE', (readonly)

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `GroupInputs.expose`
- `GroupInputs.expose`
- `GroupInputs.new`
- `GroupOutputs.expose`
- `GroupOutputs.expose`
- `GroupOutputs.new`
- `Node.inputs`
- `Node.outputs`
- `NodeLink.from_socket`
- `NodeLink.to_socket`
- `NodeLinks.new`
- `NodeLinks.new`
- `NodeTree.inputs`
- `NodeTree.outputs`

### 2.4.360 NodeSocketBoolean(NodeSocket)

base classes — `bpy_struct`, `NodeSocket`

**class** `bpy.types.NodeSocketBoolean` (*NodeSocket*)

Input or output socket of a node

**default\_value**

**Type** boolean, default False

#### Inherited Properties

- `bpy_struct.id_data`
- `NodeSocket.name`
- `NodeSocket.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.361 NodeSocketFloatAngle(NodeSocket)

base classes — `bpy_struct`, `NodeSocket`

**class** `bpy.types.NodeSocketFloatAngle` (*NodeSocket*)

Input or output socket of a node

**default\_value**

**Type** float in [-inf, inf], default 0.0

**subtype**

Subtype defining the socket value details

**Type** enum in ['NONE', 'UNSIGNED', 'NONE', 'UNSIGNED', 'PERCENTAGE', 'FACTOR', 'ANGLE', 'TIME', 'DISTANCE', 'NONE', 'TRANSLATION', 'DIRECTION', 'VELOCITY', 'ACCELERATION', 'EULER', 'XYZ'], default 'NONE'

#### Inherited Properties

- `bpy_struct.id_data`
- `NodeSocket.name`
- `NodeSocket.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.362 NodeSocketFloatDistance(NodeSocket)

base classes — `bpy_struct`, `NodeSocket`

**class** `bpy.types.NodeSocketFloatDistance` (*NodeSocket*)

Input or output socket of a node

**default\_value**

**Type** float in [-inf, inf], default 0.0

**subtype**

Subtype defining the socket value details

**Type** enum in ['NONE', 'UNSIGNED', 'NONE', 'UNSIGNED', 'PERCENTAGE', 'FACTOR', 'ANGLE', 'TIME', 'DISTANCE', 'NONE', 'TRANSLATION', 'DIRECTION', 'VELOCITY', 'ACCELERATION', 'EULER', 'XYZ'], default 'NONE'

#### Inherited Properties

- `bpy_struct.id_data`
- `NodeSocket.name`
- `NodeSocket.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.363 NodeSocketFloatFactor(NodeSocket)

base classes — `bpy_struct`, `NodeSocket`

**class** `bpy.types.NodeSocketFloatFactor` (*NodeSocket*)

Input or output socket of a node

**default\_value**

**Type** float in [0, 1], default 0.0

**subtype**

Subtype defining the socket value details

**Type** enum in ['NONE', 'UNSIGNED', 'NONE', 'UNSIGNED', 'PERCENTAGE', 'FACTOR', 'ANGLE', 'TIME', 'DISTANCE', 'NONE', 'TRANSLATION', 'DIRECTION', 'VELOCITY', 'ACCELERATION', 'EULER', 'XYZ'], default 'NONE'

#### Inherited Properties

- `bpy_struct.id_data`
- `NodeSocket.name`
- `NodeSocket.type`



## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.364 NodeSocketFloatNone(NodeSocket)

base classes — `bpy_struct`, `NodeSocket`

**class** `bpy.types.NodeSocketFloatNone` (*NodeSocket*)

Input or output socket of a node

**default\_value**

**Type** float in [-inf, inf], default 0.0

**subtype**

Subtype defining the socket value details

**Type** enum in ['NONE', 'UNSIGNED', 'NONE', 'UNSIGNED', 'PERCENTAGE', 'FACTOR', 'ANGLE', 'TIME', 'DISTANCE', 'NONE', 'TRANSLATION', 'DIRECTION', 'VELOCITY', 'ACCELERATION', 'EULER', 'XYZ'], default 'NONE'

## Inherited Properties

- `bpy_struct.id_data`
- `NodeSocket.name`
- `NodeSocket.type`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`

- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.365 NodeSocketFloatPercentage(NodeSocket)

base classes — `bpy_struct`, `NodeSocket`

**class** `bpy.types.NodeSocketFloatPercentage` (*NodeSocket*)

Input or output socket of a node

**default\_value**

**Type** float in [-inf, inf], default 0.0

**subtype**

Subtype defining the socket value details

**Type** enum in ['NONE', 'UNSIGNED', 'NONE', 'UNSIGNED', 'PERCENTAGE', 'FACTOR', 'ANGLE', 'TIME', 'DISTANCE', 'NONE', 'TRANSLATION', 'DIRECTION', 'VELOCITY', 'ACCELERATION', 'EULER', 'XYZ'], default 'NONE'

#### Inherited Properties

- `bpy_struct.id_data`
- `NodeSocket.name`
- `NodeSocket.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.366 NodeSocketFloatTime(NodeSocket)

base classes — `bpy_struct`, `NodeSocket`

**class** `bpy.types.NodeSocketFloatTime` (*NodeSocket*)

Input or output socket of a node

**default\_value**

**Type** float in [-inf, inf], default 0.0

**subtype**

Subtype defining the socket value details

**Type** enum in ['NONE', 'UNSIGNED', 'NONE', 'UNSIGNED', 'PERCENTAGE', 'FACTOR', 'ANGLE', 'TIME', 'DISTANCE', 'NONE', 'TRANSLATION', 'DIRECTION', 'VELOCITY', 'ACCELERATION', 'EULER', 'XYZ'], default 'NONE'

#### Inherited Properties

- `bpy_struct.id_data`
- `NodeSocket.name`
- `NodeSocket.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.367 NodeSocketFloatUnsigned(NodeSocket)

base classes — `bpy_struct`, `NodeSocket`

**class** `bpy.types.NodeSocketFloatUnsigned` (*NodeSocket*)

Input or output socket of a node

**default\_value**

**Type** float in [0, inf], default 0.0

**subtype**

Subtype defining the socket value details

**Type** enum in ['NONE', 'UNSIGNED', 'NONE', 'UNSIGNED', 'PERCENTAGE', 'FACTOR', 'ANGLE', 'TIME', 'DISTANCE', 'NONE', 'TRANSLATION', 'DIRECTION', 'VELOCITY', 'ACCELERATION', 'EULER', 'XYZ'], default 'NONE'

### Inherited Properties

- `bpy_struct.id_data`
- `NodeSocket.name`
- `NodeSocket.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.368 NodeSocketIntNone(NodeSocket)

base classes — `bpy_struct`, `NodeSocket`

**class** `bpy.types.NodeSocketIntNone` (*NodeSocket*)

Input or output socket of a node

**default\_value**

**Type** int in [-inf, inf], default 0

**subtype**

Subtype defining the socket value details

**Type** enum in ['NONE', 'UNSIGNED', 'NONE', 'UNSIGNED', 'PERCENTAGE', 'FACTOR', 'ANGLE', 'TIME', 'DISTANCE', 'NONE', 'TRANSLATION', 'DIRECTION', 'VELOCITY', 'ACCELERATION', 'EULER', 'XYZ'], default 'NONE'

### Inherited Properties

- `bpy_struct.id_data`
- `NodeSocket.name`
- `NodeSocket.type`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.369 NodeSocketIntUnsigned(NodeSocket)

base classes — `bpy_struct`, `NodeSocket`

**class** `bpy.types.NodeSocketIntUnsigned` (*NodeSocket*)

Input or output socket of a node

**default\_value**

**Type** int in [0, inf], default 0

**subtype**

Subtype defining the socket value details

**Type** enum in ['NONE', 'UNSIGNED', 'NONE', 'UNSIGNED', 'PERCENTAGE', 'FACTOR', 'ANGLE', 'TIME', 'DISTANCE', 'NONE', 'TRANSLATION', 'DIRECTION', 'VELOCITY', 'ACCELERATION', 'EULER', 'XYZ'], default 'NONE'

## Inherited Properties

- `bpy_struct.id_data`
- `NodeSocket.name`
- `NodeSocket.type`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`

- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.370 NodeSocketRGBA(NodeSocket)

base classes — `bpy_struct`, `NodeSocket`

**class** `bpy.types.NodeSocketRGBA` (*NodeSocket*)

Input or output socket of a node

**default\_value**

**Type** float array of 4 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0, 0.0)`

#### Inherited Properties

- `bpy_struct.id_data`
- `NodeSocket.name`
- `NodeSocket.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.371 NodeSocketShader(NodeSocket)

base classes — `bpy_struct`, `NodeSocket`

**class** `bpy.types.NodeSocketShader` (*NodeSocket*)

Input or output socket of a node

### Inherited Properties

- `bpy_struct.id_data`
- `NodeSocket.name`
- `NodeSocket.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.372 NodeSocketVectorAcceleration(NodeSocket)

base classes — `bpy_struct`, `NodeSocket`

**class** `bpy.types.NodeSocketVectorAcceleration` (*NodeSocket*)

Input or output socket of a node

#### **default\_value**

**Type** float array of 3 items in  $[-\text{inf}, \text{inf}]$ , default (0.0, 0.0, 0.0)

#### **subtype**

Subtype defining the socket value details

**Type** enum in ['NONE', 'UNSIGNED', 'NONE', 'UNSIGNED', 'PERCENTAGE', 'FACTOR', 'ANGLE', 'TIME', 'DISTANCE', 'NONE', 'TRANSLATION', 'DIRECTION', 'VELOCITY', 'ACCELERATION', 'EULER', 'XYZ'], default 'NONE'

### Inherited Properties

- `bpy_struct.id_data`
- `NodeSocket.name`
- `NodeSocket.type`

### Inherited Functions

- `bpy_struct.as_pointer`

- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.373 NodeSocketVectorDirection(NodeSocket)

base classes — `bpy_struct`, `NodeSocket`

**class** `bpy.types.NodeSocketVectorDirection` (*NodeSocket*)

Input or output socket of a node

**default\_value**

**Type** float array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`

**subtype**

Subtype defining the socket value details

**Type** enum in `['NONE', 'UNSIGNED', 'NONE', 'UNSIGNED', 'PERCENTAGE', 'FACTOR', 'ANGLE', 'TIME', 'DISTANCE', 'NONE', 'TRANSLATION', 'DIRECTION', 'VELOCITY', 'ACCELERATION', 'EULER', 'XYZ']`, default `'NONE'`

#### Inherited Properties

- `bpy_struct.id_data`
- `NodeSocket.name`
- `NodeSocket.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`



- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.374 NodeSocketVectorEuler(NodeSocket)

base classes — `bpy_struct`, `NodeSocket`

**class** `bpy.types.NodeSocketVectorEuler` (*NodeSocket*)

Input or output socket of a node

**default\_value**

**Type** float array of 3 items in  $[-inf, inf]$ , default (0.0, 0.0, 0.0)

**subtype**

Subtype defining the socket value details

**Type** enum in ['NONE', 'UNSIGNED', 'NONE', 'UNSIGNED', 'PERCENTAGE', 'FACTOR', 'ANGLE', 'TIME', 'DISTANCE', 'NONE', 'TRANSLATION', 'DIRECTION', 'VELOCITY', 'ACCELERATION', 'EULER', 'XYZ'], default 'NONE'

#### Inherited Properties

- `bpy_struct.id_data`
- `NodeSocket.name`
- `NodeSocket.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.375 NodeSocketVectorNone(NodeSocket)

base classes — `bpy_struct`, `NodeSocket`

**class** `bpy.types.NodeSocketVectorNone` (*NodeSocket*)

Input or output socket of a node

**default\_value**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**subtype**

Subtype defining the socket value details

**Type** enum in ['NONE', 'UNSIGNED', 'NONE', 'UNSIGNED', 'PERCENTAGE', 'FACTOR', 'ANGLE', 'TIME', 'DISTANCE', 'NONE', 'TRANSLATION', 'DIRECTION', 'VELOCITY', 'ACCELERATION', 'EULER', 'XYZ'], default 'NONE'

### Inherited Properties

- `bpy_struct.id_data`
- `NodeSocket.name`
- `NodeSocket.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.376 NodeSocketVectorTranslation(NodeSocket)

base classes — `bpy_struct`, `NodeSocket`

**class** `bpy.types.NodeSocketVectorTranslation` (*NodeSocket*)

Input or output socket of a node

**default\_value**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**subtype**

Subtype defining the socket value details

**Type** enum in ['NONE', 'UNSIGNED', 'NONE', 'UNSIGNED', 'PERCENTAGE', 'FACTOR', 'ANGLE', 'TIME', 'DISTANCE', 'NONE', 'TRANSLATION', 'DIRECTION', 'VELOCITY', 'ACCELERATION', 'EULER', 'XYZ'], default 'NONE'

### Inherited Properties

- `bpy_struct.id_data`
- `NodeSocket.name`
- `NodeSocket.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.377 NodeSocketVectorVelocity(NodeSocket)

base classes — `bpy_struct`, `NodeSocket`

**class** `bpy.types.NodeSocketVectorVelocity` (*NodeSocket*)

Input or output socket of a node

#### **default\_value**

**Type** float array of 3 items in  $[-inf, inf]$ , default (0.0, 0.0, 0.0)

#### **subtype**

Subtype defining the socket value details

**Type** enum in ['NONE', 'UNSIGNED', 'NONE', 'UNSIGNED', 'PERCENTAGE', 'FACTOR', 'ANGLE', 'TIME', 'DISTANCE', 'NONE', 'TRANSLATION', 'DIRECTION', 'VELOCITY', 'ACCELERATION', 'EULER', 'XYZ'], default 'NONE'

### Inherited Properties

- `bpy_struct.id_data`
- `NodeSocket.name`
- `NodeSocket.type`

### Inherited Functions

- `bpy_struct.as_pointer`

- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.378 NodeSocketVectorXYZ(NodeSocket)

base classes — `bpy_struct`, `NodeSocket`

**class** `bpy.types.NodeSocketVectorXYZ` (*NodeSocket*)

Input or output socket of a node

**default\_value**

**Type** float array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`

**subtype**

Subtype defining the socket value details

**Type** enum in `['NONE', 'UNSIGNED', 'NONE', 'UNSIGNED', 'PERCENTAGE', 'FACTOR', 'ANGLE', 'TIME', 'DISTANCE', 'NONE', 'TRANSLATION', 'DIRECTION', 'VELOCITY', 'ACCELERATION', 'EULER', 'XYZ']`, default `'NONE'`

#### Inherited Properties

- `bpy_struct.id_data`
- `NodeSocket.name`
- `NodeSocket.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`

- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.379 NodeTree(ID)

base classes — `bpy_struct`, `ID`

subclasses — `CompositorNodeTree`, `ShaderNodeTree`, `TextureNodeTree`

**class** `bpy.types.NodeTree` (*ID*)

Node tree consisting of linked nodes used for materials, textures and compositing

#### **animation\_data**

Animation data for this datablock

**Type** `AnimData`, (readonly)

#### **grease\_pencil**

Grease Pencil datablock

**Type** `GreasePencil`

#### **inputs**

**Type** `GroupInputs` `bpy_prop_collection` of `NodeSocket`, (readonly)

#### **links**

**Type** `NodeLinks` `bpy_prop_collection` of `NodeLink`, (readonly)

#### **outputs**

**Type** `GroupOutputs` `bpy_prop_collection` of `NodeSocket`, (readonly)

#### **type**

Node Tree type

- `SHADER` Shader, Shader nodes.
- `TEXTURE` Texture, Texture nodes.
- `COMPOSITING` Compositing, Compositing nodes.

**Type** `enum` in [`'SHADER'`, `'TEXTURE'`, `'COMPOSITING'`], default `'SHADER'`, (readonly)

#### Inherited Properties

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`

## References

- `BlendData.node_groups`
- `BlendDataNodeTrees.new`
- `BlendDataNodeTrees.remove`
- `CompositorNodes.new`
- `Lamp.node_tree`
- `Material.node_tree`
- `NodeForLoop.node_tree`
- `NodeGroup.node_tree`
- `NodeWhileLoop.node_tree`
- `Scene.node_tree`
- `ShaderNodes.new`
- `SpaceNodeEditor.node_tree`
- `Texture.node_tree`
- `TextureNodes.new`
- `World.node_tree`

### 2.4.380 NodeWhileLoop(Node)

base classes — `bpy_struct`, `Node`

**class** `bpy.types.NodeWhileLoop` (*Node*)

#### **max\_iterations**

Limit for number of iterations

**Type** `int` in `[0, 32767]`, default 0

**node\_tree****Type** `NodeTree`**Inherited Properties**

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**2.4.381 NoiseTexture(Texture)**base classes — `bpy_struct`, `ID`, `Texture`**class** `bpy.types.NoiseTexture` (*Texture*)

Procedural noise texture

**users\_material**

Materials that use this texture (readonly)

**users\_object\_modifier**

Object modifiers that use this texture (readonly)

**Inherited Properties**

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`

- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`
- `Texture.animation_data`
- `Texture.intensity`
- `Texture.color_ramp`
- `Texture.contrast`
- `Texture.factor_blue`
- `Texture.factor_green`
- `Texture.factor_red`
- `Texture.node_tree`
- `Texture.saturation`
- `Texture.use_preview_alpha`
- `Texture.type`
- `Texture.use_color_ramp`
- `Texture.use_nodes`
- `Texture.users_material`
- `Texture.users_object_modifier`
- `Texture.users_material`
- `Texture.users_object_modifier`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`
- `Texture.evaluate`

### 2.4.382 `NorController(Controller)`

base classes — `bpy_struct`, `Controller`

**class** `bpy.types.NorController` (*Controller*)

Controller passing on events based on a logical NOR operation



## Inherited Properties

- `bpy_struct.id_data`
- `Controller.name`
- `Controller.states`
- `Controller.show_expanded`
- `Controller.use_priority`
- `Controller.type`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Controller.link`
- `Controller.unlink`

### 2.4.383 Object(ID)

base classes — `bpy_struct`, `ID`

**class** `bpy.types.Object` (*ID*)

Object datablock defining an object in a scene

**active\_material**

Active material being displayed

**Type** `Material`

**active\_material\_index**

Index of active material slot

**Type** `int` in `[0, inf]`, default `0`

**active\_shape\_key**

Current shape key

**Type** `ShapeKey`, (readonly)

**active\_shape\_key\_index**

Current shape key index

**Type** `int` in `[-32768, 32767]`, default `0`

**animation\_data**

Animation data for this datablock

**Type** `AnimData`, (readonly)

**animation\_visualisation**

Animation data for this datablock

**Type** `AnimViz`, (readonly, never None)

**bound\_box**

Object's bounding box in object-space coordinates, all values are -1.0 when not available

**Type** float array of 24 items in  $[-inf, inf]$ , default (0.0, 0.0), (readonly)

**collision**

Settings for using the object as a collider in physics simulation

**Type** `CollisionSettings`, (readonly)

**color**

Object color and alpha, used when faces have the ObColor mode enabled

**Type** float array of 4 items in  $[-inf, inf]$ , default (0.0, 0.0, 0.0, 0.0)

**constraints**

Constraints affecting the transformation of the object

**Type** `ObjectConstraints` `bpy_prop_collection` of `Constraint`, (readonly)

**data**

Object data

**Type** `ID`

**delta\_location**

Extra translation added to the location of the object

**Type** float array of 3 items in  $[-inf, inf]$ , default (0.0, 0.0, 0.0)

**delta\_rotation\_euler**

Extra rotation added to the rotation of the object (when using Euler rotations)

**Type** float array of 3 items in  $[-inf, inf]$ , default (0.0, 0.0, 0.0)

**delta\_rotation\_quaternion**

Extra rotation added to the rotation of the object (when using Quaternion rotations)

**Type** float array of 4 items in  $[-inf, inf]$ , default (1.0, 0.0, 0.0, 0.0)

**delta\_scale**

Extra scaling added to the scale of the object

**Type** float array of 3 items in  $[-inf, inf]$ , default (0.0, 0.0, 0.0)

**dimensions**

Absolute bounding box dimensions of the object

**Type** float array of 3 items in  $[-inf, inf]$ , default (0.0, 0.0, 0.0)

**draw\_bounds\_type**

Object boundary display type

- BOX Box, Draw bounds as box.
- SPHERE Sphere, Draw bounds as sphere.

- CYLINDER Cylinder, Draw bounds as cylinder.
- CONE Cone, Draw bounds as cone.
- POLYHEDRON Polyhedron, Draw bounds as polyhedron.
- CAPSULE Capsule, Draw bounds as capsule.

**Type** enum in ['BOX', 'SPHERE', 'CYLINDER', 'CONE', 'POLYHEDRON', 'CAPSULE'], default 'BOX'

#### **draw\_type**

Maximum draw type to display object with in viewport

- BOUNDS Bounds, Draw the bounding box of the object.
- WIRE Wire, Draw the object as a wireframe.
- SOLID Solid, Draw the object as a solid (if solid drawing is enabled in the viewport).
- TEXTURED Textured, Draw the object with textures (if textures are enabled in the viewport).

**Type** enum in ['BOUNDS', 'WIRE', 'SOLID', 'TEXTURED'], default 'BOUNDS'

#### **dupli\_faces\_scale**

Scale the DupliFace objects

**Type** float in [0.001, 10000], default 0.0

#### **dupli\_frames\_end**

End frame for DupliFrames

**Type** int in [-300000, 300000], default 0

#### **dupli\_frames\_off**

Recurring frames to exclude from the Dupliframes

**Type** int in [0, 300000], default 0

#### **dupli\_frames\_on**

Number of frames to use between DupOff frames

**Type** int in [0, 300000], default 0

#### **dupli\_frames\_start**

Start frame for DupliFrames

**Type** int in [-300000, 300000], default 0

#### **dupli\_group**

Instance an existing group

**Type** [Group](#)

#### **dupli\_list**

Object duplis

**Type** [bpy\\_prop\\_collection](#) of [DupliObject](#), (readonly)

#### **dupli\_type**

If not None, object duplication method to use

- NONE None.
- FRAMES Frames, Make copy of object for every frame.

- VERTS Verts, Duplicate child objects on all vertices.
- FACES Faces, Duplicate child objects on all faces.
- GROUP Group, Enable group instancing.

**Type** enum in ['NONE', 'FRAMES', 'VERTS', 'FACES', 'GROUP'], default 'NONE'

**empty\_draw\_size**

Size of display for empties in the viewport

**Type** float in [0.0001, 1000], default 0.0

**empty\_draw\_type**

Viewport display style for empties

**Type** enum in ['PLAIN\_AXES', 'ARROWS', 'SINGLE\_ARROW', 'CIRCLE', 'CUBE', 'SPHERE', 'CONE', 'IMAGE'], default 'PLAIN\_AXES'

**empty\_image\_offset**

Origin offset distance

**Type** float array of 2 items in [-inf, inf], default (0.0, 0.0)

**field**

Settings for using the object as a field in physics simulation

**Type** `FieldSettings`, (readonly)

**game**

Game engine related settings for the object

**Type** `GameObjectSettings`, (readonly, never None)

**grease\_pencil**

Grease Pencil datablock

**Type** `GreasePencil`

**hide**

Restrict visibility in the viewport

**Type** boolean, default False

**hide\_render**

Restrict renderability

**Type** boolean, default False

**hide\_select**

Restrict selection in the viewport

**Type** boolean, default False

**is\_duplicator**

**Type** boolean, default False, (readonly)

**layers**

Layers the object is on

**Type** boolean array of 20 items, default (False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False)

**location**

Location of the object

**Type** float array of 3 items in  $[-inf, inf]$ , default (0.0, 0.0, 0.0)

#### **lock\_location**

Lock editing of location in the interface

**Type** boolean array of 3 items, default (False, False, False)

#### **lock\_rotation**

Lock editing of rotation in the interface

**Type** boolean array of 3 items, default (False, False, False)

#### **lock\_rotation\_w**

Lock editing of 'angle' component of four-component rotations in the interface

**Type** boolean, default False

#### **lock\_rotations\_4d**

Lock editing of four component rotations by components (instead of as Eulers)

**Type** boolean, default False

#### **lock\_scale**

Lock editing of scale in the interface

**Type** boolean array of 3 items, default (False, False, False)

#### **material\_slots**

Material slots in the object

**Type** `bpy_prop_collection` of `MaterialSlot`, (readonly)

#### **matrix\_basis**

Matrix access to location, rotation and scale (including deltas), before constraints and parenting are applied

**Type** float array of 16 items in  $[-inf, inf]$ , default (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)

#### **matrix\_local**

Parent relative transformation matrix

**Type** float array of 16 items in  $[-inf, inf]$ , default (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)

#### **matrix\_parent\_inverse**

Inverse of object's parent matrix at time of parenting

**Type** float array of 16 items in  $[-inf, inf]$ , default (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)

#### **matrix\_world**

Worldspace transformation matrix

**Type** float array of 16 items in  $[-inf, inf]$ , default (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)

#### **mode**

Object interaction mode

**Type** enum in ['OBJECT', 'EDIT', 'SCULPT', 'VERTEX\_PAINT', 'WEIGHT\_PAINT', 'TEXTURE\_PAINT', 'PARTICLE\_EDIT', 'POSE'], default 'OBJECT', (readonly)

#### **modifiers**

Modifiers affecting the geometric data of the object

**Type** `ObjectModifiers` `bpy_prop_collection` of `Modifier`, (readonly)

**motion\_path**

Motion Path for this element

**Type** `MotionPath`, (readonly)

**parent**

Parent Object

**Type** `Object`

**parent\_bone**

Name of parent bone in case of a bone parenting relation

**Type** `string`, default ""

**parent\_type**

Type of parent relation

- OBJECT Object, The object is parented to an object.
- CURVE Curve, The object is parented to a curve.
- KEY Key.
- ARMATURE Armature.
- LATTICE Lattice, The object is parented to a lattice.
- VERTEX Vertex, The object is parented to a vertex.
- VERTEX\_3 3 Vertices.
- BONE Bone, The object is parented to a bone.

**Type** `enum` in ['OBJECT', 'CURVE', 'KEY', 'ARMATURE', 'LATTICE', 'VERTEX', 'VERTEX\_3', 'BONE'], default 'OBJECT'

**parent\_vertices**

Indices of vertices in case of a vertex parenting relation

**Type** `int` array of 3 items in [0, inf], default (0, 0, 0)

**particle\_systems**

Particle systems emitted from the object

**Type** `ParticleSystem bpy_prop_collection` of `ParticleSystem`, (readonly)

**pass\_index**

Index # for the IndexOB render pass

**Type** `int` in [0, 32767], default 0

**pose**

Current pose for armatures

**Type** `Pose`, (readonly)

**pose\_library**

Action used as a pose library for armatures

**Type** `Action`

**proxy**

Library object this proxy object controls

**Type** `Object`, (readonly)

**proxy\_group**

Library group duplicator object this proxy object controls

**Type** `Object`, (readonly)

**rotation\_axis\_angle**

Angle of Rotation for Axis-Angle rotation representation

**Type** float array of 4 items in `[-inf, inf]`, default `(0.0, 0.0, 1.0, 0.0)`

**rotation\_euler**

Rotation in Eulers

**Type** float array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`

**rotation\_mode**

- `QUATERNION` Quaternion (WXYZ), No Gimbal Lock.
- `XYZ` XYZ Euler, XYZ Rotation Order - prone to Gimbal Lock (default).
- `XZY` XZY Euler, XZY Rotation Order - prone to Gimbal Lock.
- `YXZ` YXZ Euler, YXZ Rotation Order - prone to Gimbal Lock.
- `YZX` YZX Euler, YZX Rotation Order - prone to Gimbal Lock.
- `ZXY` ZXY Euler, ZXY Rotation Order - prone to Gimbal Lock.
- `ZYX` ZYX Euler, ZYX Rotation Order - prone to Gimbal Lock.
- `AXIS_ANGLE` Axis Angle, Axis Angle (W+XYZ), defines a rotation around some axis defined by 3D-Vector.

**Type** enum in `['QUATERNION', 'XYZ', 'XZY', 'YXZ', 'YZX', 'ZXY', 'ZYX', 'AXIS_ANGLE']`, default `'QUATERNION'`

**rotation\_quaternion**

Rotation in Quaternions

**Type** float array of 4 items in `[-inf, inf]`, default `(1.0, 0.0, 0.0, 0.0)`

**scale**

Scaling of the object

**Type** float array of 3 items in `[-inf, inf]`, default `(1.0, 1.0, 1.0)`

**select**

Object selection state

**Type** boolean, default `False`

**show\_axis**

Display the object's origin and axes

**Type** boolean, default `False`

**show\_bounds**

Display the object's bounds

**Type** boolean, default `False`

**show\_name**

Display the object's name

**Type** boolean, default `False`

**show\_only\_shape\_key**

Always show the current Shape for this Object

**Type** boolean, default False

**show\_texture\_space**

Display the object's texture space

**Type** boolean, default False

**show\_transparent**

Display material transparency in the object (unsupported for duplicator drawing)

**Type** boolean, default False

**show\_wire**

Add the object's wireframe over solid drawing

**Type** boolean, default False

**show\_x\_ray**

Make the object draw in front of others (unsupported for duplicator drawing)

**Type** boolean, default False

**soft\_body**

Settings for soft body simulation

**Type** `SoftBodySettings`, (readonly)

**time\_offset**

Animation offset in frames for F-Curve and dupligroup instances

**Type** float in [-300000, 300000], default 0.0

**track\_axis**

Axis that points in 'forward' direction (applies to DupliFrame when parent 'Follow' is enabled)

**Type** enum in ['POS\_X', 'POS\_Y', 'POS\_Z', 'NEG\_X', 'NEG\_Y', 'NEG\_Z'], default 'POS\_X'

**type**

Type of Object

**Type** enum in ['MESH', 'CURVE', 'SURFACE', 'META', 'FONT', 'ARMATURE', 'LATTICE', 'EMPTY', 'CAMERA', 'LAMP', 'SPEAKER'], default 'EMPTY', (readonly)

**up\_axis**

Axis that points in the upward direction (applies to DupliFrame when parent 'Follow' is enabled)

**Type** enum in ['X', 'Y', 'Z'], default 'X'

**use\_dupli\_faces\_scale**

Scale dupli based on face size

**Type** boolean, default False

**use\_dupli\_frames\_speed**

Set dupliframes to use the frame

**Type** boolean, default False

**use\_dupli\_vertices\_rotation**

Rotate dupli according to vertex normal

**Type** boolean, default False



**use\_shape\_key\_edit\_mode**

Apply shape keys in edit mode (for Meshes only)

**Type** boolean, default False

**use\_slow\_parent**

Create a delay in the parent relationship

**Type** boolean, default False

**use\_time\_offset\_add\_parent**

Add the parent's time offset value

**Type** boolean, default False

**use\_time\_offset\_edit**

Use time offset when inserting keys and display time offset for F-Curve and action views

**Type** boolean, default False

**use\_time\_offset\_parent**

Apply the time offset to this object's parent relationship

**Type** boolean, default False

**use\_time\_offset\_particle**

Let the time offset work on the particle effect

**Type** boolean, default False

**vertex\_groups**

Vertex groups of the object

**Type** `VertexGroups` `bpy_prop_collection` of `VertexGroup`, (readonly)

**children**

All the children of this object (readonly)

**users\_group**

The groups this object is in (readonly)

**users\_scene**

The scenes this object is in (readonly)

**to\_mesh** (*scene, apply\_modifiers, settings*)

Create a Mesh datablock with modifiers applied

**Parameters**

- **scene** (`Scene`, (never None)) – Scene within which to evaluate modifiers
- **apply\_modifiers** (*boolean*) – Apply modifiers
- **settings** (*enum in ['PREVIEW', 'RENDER']*) – Modifier settings to apply
  - PREVIEW Preview, Apply modifier preview settings.
  - RENDER Render, Apply modifier render settings.

**Returns** Mesh created from object, remove it if it is only used for export

**Return type** `Mesh`

**dupli\_list\_create** (*scene*)

Create a list of dupli objects for this object, needs to be freed manually with `free_dupli_list` to restore the objects real matrix and layers

**Parameters** `scene` (`Scene`, (never None)) – Scene within which to evaluate duplis

**dupli\_list\_clear** ()

Free the list of dupli objects

**find\_armature** ()

Find armature influencing this object as a parent or via a modifier

**Returns** Armature object influencing this object or NULL

**Return type** `Object`

**shape\_key\_add** (`name="Key", from_mix=True`)

Add shape key to an object

**Parameters**

- **name** (`string`, (`optional`)) – Unique name for the new keylock
- **from\_mix** (`boolean`, (`optional`)) – Create new shape from existing mix of shapes

**Returns** New shape keyblock

**Return type** `ShapeKey`

**ray\_cast** (`start, end`)

Cast a ray onto in object space

**Return (location, normal, index)** `location`, The hit location of this ray cast, float array of 3 items in [-inf, inf]

`normal`, The face normal at the ray cast hit location, float array of 3 items in [-inf, inf]

`index`, The face index, -1 when no intersection is found, int in [-inf, inf]

**closest\_point\_on\_mesh** (`point, max_dist=1.84467e+19`)

Find the nearest point on the object

**Return (location, normal, index)** `location`, The location on the object closest to the point, float array of 3 items in [-inf, inf]

`normal`, The face normal at the closest point, float array of 3 items in [-inf, inf]

`index`, The face index, -1 when no closest point is found, int in [-inf, inf]

**is\_visible** (`scene`)

Determine if object is visible in a given scene

**Returns** Object visibility

**Return type** `boolean`

**is\_modified** (`scene, settings`)

Determine if this object is modified from the base mesh data

**Parameters** `settings` (`enum in ['PREVIEW', 'RENDER']`) – Modifier settings to apply

- `PREVIEW` Preview, Apply modifier preview settings.
- `RENDER` Render, Apply modifier render settings.

**Returns** Object visibility

**Return type** `boolean`

## Inherited Properties

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`

## References

- `ActionConstraint.target`
- `ArmatureActuator.secondary_target`
- `ArmatureActuator.target`
- `ArmatureModifier.object`
- `ArrayModifier.curve`
- `ArrayModifier.end_cap`
- `ArrayModifier.offset_object`
- `ArrayModifier.start_cap`
- `BlendData.objects`
- `BlendDataObjects.new`
- `BlendDataObjects.remove`
- `BoidRuleAvoid.object`
- `BoidRuleFollowLeader.object`
- `BoidRuleGoal.object`
- `BooleanModifier.object`

- Camera.dof\_object
- CameraActuator.object
- CastModifier.object
- ChildOfConstraint.target
- ClampToConstraint.target
- ConstraintTarget.target
- CopyLocationConstraint.target
- CopyRotationConstraint.target
- CopyScaleConstraint.target
- CopyTransformsConstraint.target
- Curve.bevel\_object
- Curve.taper\_object
- CurveModifier.object
- CurveSplines.active
- DampedTrackConstraint.target
- DisplaceModifier.texture\_coords\_object
- DupliObject.object
- EditObjectActuator.object
- EditObjectActuator.track\_object
- EnvironmentMap.viewpoint\_object
- FloorConstraint.target
- FollowPathConstraint.target
- Group.objects
- GroupObjects.link
- GroupObjects.unlink
- HookModifier.object
- KinematicConstraint.pole\_target
- KinematicConstraint.target
- LampTextureSlot.object
- LatticeModifier.object
- LimitDistanceConstraint.target
- LockedTrackConstraint.target
- MaskModifier.armature
- MaterialTextureSlot.object
- MeshDeformModifier.object
- MirrorModifier.mirror\_object
- Object.find\_armature
- Object.parent
- Object.proxy
- Object.proxy\_group
- ObjectActuator.reference\_object
- ObjectBase.object
- ParentActuator.object
- ParticleEdit.object
- ParticleInstanceModifier.object
- ParticleSettings.billboard\_object
- ParticleSettings.dupli\_object
- ParticleSettingsTextureSlot.object
- ParticleSystem.parent
- ParticleSystem.reactor\_target\_object
- ParticleTarget.object
- PivotConstraint.target
- PointDensity.object
- PoseBone.custom\_shape

- `PropertyActuator.object`
- `RigidBodyJointConstraint.child`
- `RigidBodyJointConstraint.target`
- `Scene.camera`
- `Scene.objects`
- `SceneActuator.camera`
- `SceneObjects.active`
- `SceneObjects.link`
- `SceneObjects.unlink`
- `SceneSequence.scene_camera`
- `ScrewModifier.object`
- `ShrinkwrapConstraint.target`
- `ShrinkwrapModifier.auxiliary_target`
- `ShrinkwrapModifier.target`
- `SimpleDeformModifier.origin`
- `SpaceView3D.camera`
- `SpaceView3D.lock_object`
- `SplineIKConstraint.target`
- `SteeringActuator.navmesh`
- `SteeringActuator.target`
- `StretchToConstraint.target`
- `TextCurve.follow_curve`
- `TimelineMarker.camera`
- `ToolSettings.etch_template`
- `TrackToConstraint.target`
- `TransformConstraint.target`
- `UVProjector.object`
- `VertexWeightEditModifier.mask_tex_map_object`
- `VertexWeightMixModifier.mask_tex_map_object`
- `VertexWeightProximityModifier.mask_tex_map_object`
- `VertexWeightProximityModifier.target`
- `VoxelData.domain_object`
- `WarpModifier.object_from`
- `WarpModifier.object_to`
- `WarpModifier.texture_coords_object`
- `WaveModifier.start_position_object`
- `WaveModifier.texture_coords_object`
- `WorldTextureSlot.object`

## 2.4.384 ObjectActuator(Actuator)

base classes — `bpy_struct`, `Actuator`

**class** `bpy.types.ObjectActuator` (*Actuator*)

Actuator to control the object movement

**angular\_velocity**

Sets the angular velocity

**Type** float array of 3 items in  $[-inf, inf]$ , default (0.0, 0.0, 0.0)

**damping**

Number of frames to reach the target velocity

**Type** int in  $[-32768, 32767]$ , default 0

**derivate\_coefficient**

Not required, high values can cause instability

**Type** float in [-inf, inf], default 0.0

**force**

Sets the force

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**force\_max\_x**

Set the upper limit for force

**Type** float in [-inf, inf], default 0.0

**force\_max\_y**

Set the upper limit for force

**Type** float in [-inf, inf], default 0.0

**force\_max\_z**

Set the upper limit for force

**Type** float in [-inf, inf], default 0.0

**force\_min\_x**

Set the lower limit for force

**Type** float in [-inf, inf], default 0.0

**force\_min\_y**

Set the lower limit for force

**Type** float in [-inf, inf], default 0.0

**force\_min\_z**

Set the lower limit for force

**Type** float in [-inf, inf], default 0.0

**integral\_coefficient**

Low value (0.01) for slow response, high value (0.5) for fast response

**Type** float in [-inf, inf], default 0.0

**linear\_velocity**

Sets the linear velocity (in Servo mode it sets the target relative linear velocity, it will be achieved by automatic application of force. Null velocity is a valid target)

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**mode**

Specify the motion system

**Type** enum in ['OBJECT\_NORMAL', 'OBJECT\_SERVO'], default 'OBJECT\_NORMAL'

**offset\_location**

Sets the location

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**offset\_rotation**

Sets the rotation

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**proportional\_coefficient**

Typical value is 60x integral coefficient

**Type** float in [-inf, inf], default 0.0

**reference\_object**

Reference object for velocity calculation, leave empty for world reference

**Type** Object

**torque**

Sets the torque

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**use\_add\_linear\_velocity**

Toggles between ADD and SET linV

**Type** boolean, default False

**use\_local\_angular\_velocity**

Angular velocity is defined in local coordinates

**Type** boolean, default False

**use\_local\_force**

Force is defined in local coordinates

**Type** boolean, default False

**use\_local\_linear\_velocity**

Velocity is defined in local coordinates

**Type** boolean, default False

**use\_local\_location**

Location is defined in local coordinates

**Type** boolean, default False

**use\_local\_rotation**

Rotation is defined in local coordinates

**Type** boolean, default False

**use\_local\_torque**

Torque is defined in local coordinates

**Type** boolean, default False

**use\_servo\_limit\_x**

Set limit to force along the X axis

**Type** boolean, default False

**use\_servo\_limit\_y**

Set limit to force along the Y axis

**Type** boolean, default False

**use\_servo\_limit\_z**

Set limit to force along the Z axis

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `Actuator.name`
- `Actuator.show_expanded`
- `Actuator.pin`
- `Actuator.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Actuator.link`
- `Actuator.unlink`

## 2.4.385 ObjectBase(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.ObjectBase` (*bpy\_struct*)

An object instance in a scene

#### **layers**

Layers the object base is on

**Type** boolean array of 20 items, default (False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False)

#### **object**

Object this base links to

**Type** `Object`, (readonly)

#### **select**

Object base selection state

**Type** boolean, default False

#### **layers\_from\_view** (*view*)

Sets the object layers from a 3D View (use when adding an object in local view)



## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Scene.object_bases`
- `SceneBases.active`
- `SceneObjects.link`

## 2.4.386 ObjectConstraints(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.ObjectConstraints` (*bpy\_struct*)

Collection of object constraints

### **active**

Active Object constraint

**Type** `Constraint`

### **new** (*type*)

Add a new constraint to this object

**Parameters** `type` (*enum in* [`'COPY_LOCATION'`, `'COPY_ROTATION'`, `'COPY_SCALE'`, `'COPY_TRANSFORMS'`, `'LIMIT_DISTANCE'`, `'LIMIT_LOCATION'`, `'LIMIT_ROTATION'`, `'LIMIT_SCALE'`, `'MAINTAIN_VOLUME'`, `'TRANSFORM'`, `'CLAMP_TO'`, `'DAMPED_TRACK'`, `'IK'`, `'LOCKED_TRACK'`, `'SPLINE_IK'`, `'STRETCH_TO'`, `'TRACK_TO'`, `'ACTION'`, `'CHILD_OF'`, `'FLOOR'`, `'FOLLOW_PATH'`, `'PIVOT'`, `'RIGID_BODY_JOINT'`, `'SCRIPT'`, `'SHRINKWRAP'`]) – Constraint type to add

- `COPY_LOCATION` Copy Location.
- `COPY_ROTATION` Copy Rotation.

- `COPY_SCALE` Copy Scale.
- `COPY_TRANSFORMS` Copy Transforms.
- `LIMIT_DISTANCE` Limit Distance.
- `LIMIT_LOCATION` Limit Location.
- `LIMIT_ROTATION` Limit Rotation.
- `LIMIT_SCALE` Limit Scale.
- `MAINTAIN_VOLUME` Maintain Volume.
- `TRANSFORM` Transformation.
- `CLAMP_TO` Clamp To.
- `DAMPED_TRACK` Damped Track, Tracking by taking the shortest path.
- `IK Inverse Kinematics`.
- `LOCKED_TRACK` Locked Track, Tracking along a single axis.
- `SPLINE_IK` Spline IK.
- `STRETCH_TO` Stretch To.
- `TRACK_TO` Track To, Legacy tracking constraint prone to twisting artifacts.
- `ACTION` Action.
- `CHILD_OF` Child Of.
- `FLOOR` Floor.
- `FOLLOW_PATH` Follow Path.
- `PIVOT` Pivot.
- `RIGID_BODY_JOINT` Rigid Body Joint.
- `SCRIPT` Script.
- `SHRINKWRAP` Shrinkwrap.

**Returns** New constraint

**Return type** `Constraint`

**remove** (*constraint*)

Remove a constraint from this object

**Parameters** `constraint` (`Constraint`, (never None)) – Removed constraint

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`

- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Object.constraints`

### 2.4.387 ObjectModifiers(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.ObjectModifiers` (*bpy\_struct*)

Collection of object modifiers

**new** (*name, type*)

Add a new modifier

#### Parameters

- **name** (*string*) – New name for the bone
- **type** (*enum in* [`'UV_PROJECT'`, `'VERTEX_WEIGHT_EDIT'`, `'VERTEX_WEIGHT_MIX'`, `'VERTEX_WEIGHT_PROXIMITY'`, `'ARRAY'`, `'BEVEL'`, `'BOOLEAN'`, `'BUILD'`, `'DECIMATE'`, `'EDGE_SPLIT'`, `'MASK'`, `'MIRROR'`, `'MULTIRES'`, `'SCREW'`, `'SOLIDIFY'`, `'SUBSURF'`, `'ARMATURE'`, `'CAST'`, `'CURVE'`, `'DISPLACE'`, `'HOOK'`, `'LATTICE'`, `'MESH_DEFORM'`, `'SHRINKWRAP'`, `'SIMPLE_DEFORM'`, `'SMOOTH'`, `'WARP'`, `'WAVE'`, `'CLOTH'`, `'COLLISION'`, `'EXPLODE'`, `'FLUID_SIMULATION'`, `'PARTICLE_INSTANCE'`, `'PARTICLE_SYSTEM'`, `'SMOKE'`, `'SOFT_BODY'`, `'SURFACE'`]) – Modifier type to add

**Returns** Newly created modifier

**Return type** `Modifier`

**remove** (*modifier*)

Remove an existing modifier from the object

**Parameters** **modifier** (`Modifier`, (never `None`)) – Modifier to remove

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Object.modifiers`

### 2.4.388 `ObstacleFluidSettings(FluidSettings)`

base classes — `bpy_struct`, `FluidSettings`

**class** `bpy.types.ObstacleFluidSettings` (*FluidSettings*)

Fluid simulation settings for obstacles in the simulation

#### **impact\_factor**

This is an unphysical value for moving objects - it controls the impact an obstacle has on the fluid, =0 behaves a bit like outflow (deleting fluid), =1 is default, while >1 results in high forces. Can be used to tweak total mass

**Type** float in [-2, 10], default 0.0

#### **partial\_slip\_factor**

Amount of mixing between no- and free-slip, 0 is no slip and 1 is free slip

**Type** float in [0, 1], default 0.0

#### **slip\_type**

- **NOSLIP** No Slip, Obstacle causes zero normal and tangential velocity (=sticky). Default for all. Only option for moving objects.

- **PARTIALSLIP** Partial Slip, Mix between no-slip and free-slip. Non moving objects only!.

- **FREESLIP** Free Slip, Obstacle only causes zero normal velocity (=not sticky). Non moving objects only!.

**Type** enum in ['NOSLIP', 'PARTIALSLIP', 'FREESLIP'], default 'NOSLIP'

#### **use**

Object contributes to the fluid simulation

**Type** boolean, default False

#### **use\_animated\_mesh**

Export this mesh as an animated one. Slower, only use if really necessary (e.g. armatures or parented objects), animated pos/rot/scale IPOs do not require it

**Type** boolean, default False

#### **volume\_initialization**

Volume initialization type

- **VOLUME** Volume, Use only the inner volume of the mesh.
- **SHELL** Shell, Use only the outer shell of the mesh.
- **BOTH** Both, Use both the inner volume and the outer shell of the mesh.

**Type** enum in ['VOLUME', 'SHELL', 'BOTH'], default 'VOLUME'

### Inherited Properties

- `bpy_struct.id_data`
- `FluidSettings.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.389 Operator(`bpy_struct`)

### Basic Operator Example

This script shows simple operator which prints a message.

Since the operator only has an `Operator.execute` function it takes no user input.

---

**Note:** Operator subclasses must be registered before accessing them from blender.

---

```
import bpy

class HelloWorldOperator(bpy.types.Operator):
    bl_idname = "wm.hello_world"
    bl_label = "Minimal Operator"

    def execute(self, context):
        print("Hello World")
        return {'FINISHED'}

bpy.utils.register_class(HelloWorldOperator)

# test call to the newly defined operator
bpy.ops.wm.hello_world()
```

## Invoke Function

`Operator.invoke` is used to initialize the operator from the context at the moment the operator is called. `invoke()` is typically used to assign properties which are then used by `execute()`. Some operators don't have an `execute()` function, removing the ability to be repeated from a script or macro.

This example shows how to define an operator which gets mouse input to execute a function and that this operator can be invoked or executed from the python api.

Also notice this operator defines its own properties, these are different to typical class properties because blender registers them with the operator, to use as arguments when called, saved for operator undo/redo and automatically added into the user interface.

```
import bpy

class SimpleMouseOperator(bpy.types.Operator):
    """ This operator shows the mouse location,
        this string is used for the tooltip and API docs
    """
    bl_idname = "wm.mouse_position"
    bl_label = "Invoke Mouse Operator"

    x = bpy.props.IntProperty()
    y = bpy.props.IntProperty()

    def execute(self, context):
        # rather than printing, use the report function,
        # this way the message appears in the header,
        self.report({'INFO'}, "Mouse coords are %d %d" % (self.x, self.y))
        return {'FINISHED'}

    def invoke(self, context, event):
        self.x = event.mouse_x
        self.y = event.mouse_y
        return self.execute(context)

bpy.utils.register_class(SimpleMouseOperator)

# Test call to the newly defined operator.
# Here we call the operator and invoke it, meaning that the settings are taken
```

```
# from the mouse.
bpy.ops.wm.mouse_position('INVOKE_DEFAULT')

# Another test call, this time call execute() directly with pre-defined settings.
bpy.ops.wm.mouse_position('EXEC_DEFAULT', x=20, y=66)
```

## Calling a File Selector

This example shows how an operator can use the file selector.

Notice the invoke function calls a window manager method and returns `RUNNING_MODAL`, this means the file selector stays open and the operator does not exit immediately after invoke finishes.

The file selector runs the operator, calling `Operator.execute` when the user confirms.

The `Operator.poll` function is optional, used to check if the operator can run.

```
import bpy

class ExportSomeData(bpy.types.Operator):
    """Test exporter which just writes hello world"""
    bl_idname = "export.some_data"
    bl_label = "Export Some Data"

    filepath = bpy.props.StringProperty(subtype="FILE_PATH")

    @classmethod
    def poll(cls, context):
        return context.object is not None

    def execute(self, context):
        file = open(self.filepath, 'w')
        file.write("Hello World " + context.object.name)
        return {'FINISHED'}

    def invoke(self, context, event):
        context.window_manager.fileselect_add(self)
        return {'RUNNING_MODAL'}

# Only needed if you want to add into a dynamic menu
def menu_func(self, context):
    self.layout.operator_context = 'INVOKE_DEFAULT'
    self.layout.operator(ExportSomeData.bl_idname, text="Text Export Operator")

# Register and add to the file selector
bpy.utils.register_class(ExportSomeData)
bpy.types.INFO_MT_file_export.append(menu_func)

# test call
bpy.ops.export.some_data('INVOKE_DEFAULT')
```

## Dialog Box

This operator uses its `Operator.invoke` function to call a popup.

```
import bpy

class DialogOperator(bpy.types.Operator):
    bl_idname = "object.dialog_operator"
    bl_label = "Simple Dialog Operator"

    my_float = bpy.props.FloatProperty(name="Some Floating Point")
    my_bool = bpy.props.BoolProperty(name="Toggle Option")
    my_string = bpy.props.StringProperty(name="String Value")

    def execute(self, context):
        message = "Popup Values: %f, %d, '%s'" % \
            (self.my_float, self.my_bool, self.my_string)
        self.report({'INFO'}, message)
        return {'FINISHED'}

    def invoke(self, context, event):
        wm = context.window_manager
        return wm.invoke_props_dialog(self)

bpy.utils.register_class(DialogOperator)

# test call
bpy.ops.object.dialog_operator('INVOKE_DEFAULT')
```

## Custom Drawing

By default operator properties use an automatic user interface layout. If you need more control you can create your own layout with a `Operator.draw` function.

This works like the `Panel` and `Menu` draw functions, its used for dialogs and file selectors.

```
import bpy

class CustomDrawOperator(bpy.types.Operator):
    bl_idname = "object.custom_draw"
    bl_label = "Simple Modal Operator"

    filepath = bpy.props.StringProperty(subtype="FILE_PATH")

    my_float = bpy.props.FloatProperty(name="Float")
    my_bool = bpy.props.BoolProperty(name="Toggle Option")
    my_string = bpy.props.StringProperty(name="String Value")

    def execute(self, context):
        print()
        return {'FINISHED'}

    def invoke(self, context, event):
        context.window_manager.fileselect_add(self)
        return {'RUNNING_MODAL'}

    def draw(self, context):
        layout = self.layout
```



```

col = layout.column()
col.label(text="Custom Interface!")

row = col.row()
row.prop(self, "my_float")
row.prop(self, "my_bool")

col.prop(self, "my_string")

bpy.utils.register_class(CustomDrawOperator)

# test call
bpy.ops.object.custom_draw('INVOKE_DEFAULT')

```

## Modal Execution

This operator defines a `Operator.modal` function which running, handling events until it returns `{'FINISHED'}` or `{'CANCELLED'}`.

Grab, Rotate, Scale and Fly-Mode are examples of modal operators. They are especially useful for interactive tools, your operator can have its own state where keys toggle options as the operator runs.

`Operator.invoke` is used to initialize the operator as being by returning `{'RUNNING_MODAL'}`, initializing the modal loop.

Notice `__init__()` and `__del__()` are declared. For other operator types they are not useful but for modal operators they will be called before the `Operator.invoke` and after the operator finishes.

```

import bpy

class ModalOperator(bpy.types.Operator):
    bl_idname = "object.modal_operator"
    bl_label = "Simple Modal Operator"

    def __init__(self):
        print("Start")

    def __del__(self):
        print("End")

    def execute(self, context):
        context.object.location.x = self.value / 100.0

    def modal(self, context, event):
        if event.type == 'MOUSEMOVE': # Apply
            self.value = event.mouse_x
            self.execute(context)
        elif event.type == 'LEFTMOUSE': # Confirm
            return {'FINISHED'}
        elif event.type in ('RIGHTMOUSE', 'ESC'): # Cancel
            return {'CANCELLED'}

        return {'RUNNING_MODAL'}

    def invoke(self, context, event):
        self.value = event.mouse_x
        self.execute(context)

```

```
print(context.window_manager.modal_handler_add(self))
return {'RUNNING_MODAL'}
```

```
bpy.utils.register_class(ModalOperator)
```

```
# test call
```

```
bpy.ops.object.modal_operator(' INVOKE_DEFAULT')
```

base class — `bpy_struct`

**class** `bpy.types.Operator` (*bpy\_struct*)

Storage of an operator being executed, or registered after execution

**bl\_description**

**Type** string, default ""

**bl\_idname**

**Type** string, default ""

**bl\_label**

**Type** string, default ""

**bl\_options**

Options for this operator type

- REGISTER Register, Display in the info window and support the redo toolbar panel.
- UNDO Undo, Push an undo event (needed for operator redo).
- BLOCKING Blocking, Block anything else from using the cursor.
- MACRO Macro, Use to check if an operator is a macro.
- GRAB\_POINTER Grab Pointer, Use so the operator grabs the mouse focus, enables wrapping when continuous grab is enabled.
- PRESET Preset, Display a preset button with the operators settings.
- INTERNAL Internal, Removes the operator from search results.

**Type** enum set in {'REGISTER', 'UNDO', 'BLOCKING', 'MACRO', 'GRAB\_POINTER', 'PRESET', 'INTERNAL'}, default {'REGISTER'}

**has\_reports**

Operator has a set of reports (warnings and errors) from last execution

**Type** boolean, default False, (readonly)

**layout**

**Type** `UILayout`, (readonly)

**name**

**Type** string, default "", (readonly)

**properties**

**Type** `OperatorProperties`, (readonly, never None)

**report** (*type, message*)  
report

**Parameters**

- **type** (*enum set in {'DEBUG', 'INFO', 'OPERATOR', 'WARNING', 'ERROR', 'ERROR\_INVALID\_INPUT', 'ERROR\_INVALID\_CONTEXT', 'ERROR\_OUT\_OF\_MEMORY'}*) – Type
- **message** (*string*) – Report Message

**classmethod poll** (*context*)  
Test if the operator can be called or not

**Return type** boolean

**execute** (*context*)  
Execute the operator

**Returns**

result

- **RUNNING\_MODAL** Running Modal, Keep the operator running with blender.
- **CANCELLED** Cancelled, When no action has been taken, operator exits.
- **FINISHED** Finished, When the operator is complete, operator exits.
- **PASS\_THROUGH** Pass Through, Do nothing and pass the event on.

**Return type** enum set in {'RUNNING\_MODAL', 'CANCELLED', 'FINISHED', 'PASS\_THROUGH'}

**check** (*context*)  
Check the operator settings, return True to signal a change to redraw

**Returns** result

**Return type** boolean

**invoke** (*context, event*)  
Invoke the operator

**Returns**

result

- **RUNNING\_MODAL** Running Modal, Keep the operator running with blender.
- **CANCELLED** Cancelled, When no action has been taken, operator exits.
- **FINISHED** Finished, When the operator is complete, operator exits.
- **PASS\_THROUGH** Pass Through, Do nothing and pass the event on.

**Return type** enum set in {'RUNNING\_MODAL', 'CANCELLED', 'FINISHED', 'PASS\_THROUGH'}

**modal** (*context, event*)  
Modal operator function

**Returns**

result

- **RUNNING\_MODAL** Running Modal, Keep the operator running with blender.

- CANCELLED Cancelled, When no action has been taken, operator exits.
- FINISHED Finished, When the operator is complete, operator exits.
- PASS\_THROUGH Pass Through, Do nothing and pass the event on.

**Return type** enum set in {'RUNNING\_MODAL', 'CANCELLED', 'FINISHED', 'PASS\_THROUGH'}

**draw** (*context*)

Draw function for the operator

**cancel** (*context*)

Called when the operator is cancelled

**Returns**

result

- RUNNING\_MODAL Running Modal, Keep the operator running with blender.
- CANCELLED Cancelled, When no action has been taken, operator exits.
- FINISHED Finished, When the operator is complete, operator exits.
- PASS\_THROUGH Pass Through, Do nothing and pass the event on.

**Return type** enum set in {'RUNNING\_MODAL', 'CANCELLED', 'FINISHED', 'PASS\_THROUGH'}

**as\_keywords** (*ignore=()*)

Return a copy of the properties as a dictionary

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `SpaceFileBrowser.operator`
- `WindowManager.fileselect_add`
- `WindowManager.invoke_confirm`
- `WindowManager.invoke_popup`
- `WindowManager.invoke_props_dialog`
- `WindowManager.invoke_props_popup`
- `WindowManager.invoke_search_popup`
- `WindowManager.modal_handler_add`
- `WindowManager.operators`

### 2.4.390 OperatorFileListElement(PropertyGroup)

base classes — `bpy_struct`, `PropertyGroup`

class `bpy.types.OperatorFileListElement` (*PropertyGroup*)

#### name

the name of a file or directory within a file list

**Type** string, default ""

#### Inherited Properties

- `bpy_struct.id_data`
- `PropertyGroup.name`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.391 OperatorMacro(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.OperatorMacro` (*bpy\_struct*)  
Storage of a sub operator in a macro after it has been added

**properties**

**Type** `OperatorProperties`, (readonly, never None)

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.392 OperatorMousePath(PropertyGroup)

base classes — `bpy_struct`, `PropertyGroup`

**class** `bpy.types.OperatorMousePath` (*PropertyGroup*)  
Mouse path values for operators that record such paths

**loc**

Mouse location

**Type** float array of 2 items in `[-inf, inf]`, default `(0.0, 0.0)`

**time**

Time of mouse location

**Type** float in `[-inf, inf]`, default `0.0`

**Inherited Properties**

- `bpy_struct.id_data`
- `PropertyGroup.name`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.393 OperatorProperties(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.OperatorProperties` (*bpy\_struct*)  
Input properties of an Operator

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `KeyMapItem.properties`
- `Macro.properties`
- `Operator.properties`
- `OperatorMacro.properties`
- `UILayout.operator`

## 2.4.394 OperatorStrokeElement(PropertyGroup)

base classes — `bpy_struct`, `PropertyGroup`

class `bpy.types.OperatorStrokeElement` (*PropertyGroup*)

### `is_start`

**Type** boolean, default False

### `location`

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

### `mouse`

**Type** float array of 2 items in [-inf, inf], default (0.0, 0.0)

### `pen_flip`

**Type** boolean, default False

### `pressure`

Tablet pressure

**Type** float in [0, 1], default 0.0

### `time`

**Type** float in [0, inf], default 0.0

## Inherited Properties

- `bpy_struct.id_data`
- `PropertyGroup.name`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`



- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.395 OrController(Controller)

base classes — `bpy_struct`, `Controller`

**class** `bpy.types.OrController` (*Controller*)  
Controller passing on events based on a logical OR operation

#### Inherited Properties

- `bpy_struct.id_data`
- `Controller.name`
- `Controller.states`
- `Controller.show_expanded`
- `Controller.use_priority`
- `Controller.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Controller.link`
- `Controller.unlink`

### 2.4.396 OutflowFluidSettings(FluidSettings)

base classes — `bpy_struct`, `FluidSettings`

**class** `bpy.types.OutflowFluidSettings` (*FluidSettings*)  
Fluid simulation settings for objects removing fluids from the simulation

**use**

Object contributes to the fluid simulation

**Type** boolean, default False

**use\_animated\_mesh**

Export this mesh as an animated one. Slower, only use if really necessary (e.g. armatures or parented objects), animated pos/rot/scale IPOs do not require it

**Type** boolean, default False

**volume\_initialization**

Volume initialization type

- VOLUME Volume, Use only the inner volume of the mesh.
- SHELL Shell, Use only the outer shell of the mesh.
- BOTH Both, Use both the inner volume and the outer shell of the mesh.

**Type** enum in ['VOLUME', 'SHELL', 'BOTH'], default 'VOLUME'

**Inherited Properties**

- `bpy_struct.id_data`
- `FluidSettings.type`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**2.4.397 PackedFile(bpy\_struct)**

base class — `bpy_struct`

**class** `bpy.types.PackedFile` (*bpy\_struct*)

External file packed into the .blend file

**size**

Size of packed file in bytes

**Type** int in [-inf, inf], default 0, (readonly)

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Image.packed_file`
- `Sound.packed_file`
- `VectorFont.packed_file`

## 2.4.398 Paint(`bpy_struct`)

base class — `bpy_struct`

subclasses — `VertexPaint`, `Sculpt`, `ImagePaint`

**class** `bpy.types.Paint` (*`bpy_struct`*)

#### **brush**

Active Brush

**Type** `Brush`

#### **show\_brush**

**Type** boolean, default False

#### **show\_brush\_on\_surface**

**Type** boolean, default False

#### **show\_low\_resolution**

For multires, show low resolution while navigating the view

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.399 Panel(bpy\_struct)

### Basic Panel Example

This script is a simple panel which will draw into the object properties section.

Notice the 'CATEGORY\_PT\_name' `Panel.bl_idname`, this is a naming convention for panels.

---

**Note:** Panel subclasses must be registered for blender to use them.

---

```
import bpy

class HelloWorldPanel(bpy.types.Panel):
    bl_idname = "OBJECT_PT_hello_world"
    bl_label = "Hello World"
    bl_space_type = 'PROPERTIES'
    bl_region_type = 'WINDOW'
    bl_context = "object"

    def draw(self, context):
        self.layout.label(text="Hello World")

bpy.utils.register_class(HelloWorldPanel)
```

## Simple Object Panel

This panel has a `Panel.poll` and `Panel.draw_header` function, even though the contents is basic this closely resembles blenders panels.

```
import bpy

class ObjectSelectPanel(bpy.types.Panel):
    bl_idname = "OBJECT_PT_select"
    bl_label = "Select"
    bl_space_type = 'PROPERTIES'
    bl_region_type = 'WINDOW'
    bl_context = "object"
    bl_options = {'DEFAULT_CLOSED'}

    @classmethod
    def poll(cls, context):
        return (context.object is not None)

    def draw_header(self, context):
        layout = self.layout
        obj = context.object
        layout.prop(obj, "select", text="")

    def draw(self, context):
        layout = self.layout

        obj = context.object
        row = layout.row()
        row.prop(obj, "hide_select")
        row.prop(obj, "hide_render")

        box = layout.box()
        box.label("Selection Tools")
        box.operator("object.select_all")
        row = box.row()
        row.operator("object.select_inverse")
        row.operator("object.select_random")

bpy.utils.register_class(ObjectSelectPanel)
```

## Mix-in Classes

A mix-in parent class can be used to share common properties and `Menu.poll` function.

```
import bpy

class View3DPanel():
    bl_space_type = 'VIEW_3D'
    bl_region_type = 'TOOLS'

    @classmethod
    def poll(cls, context):
        return (context.object is not None)
```

```
class PanelOne(View3DPanel, bpy.types.Panel):
    bl_idname = "VIEW3D_PT_test_1"
    bl_label = "Panel One"

    def draw(self, context):
        self.layout.label("Small Class")

class PanelTwo(View3DPanel, bpy.types.Panel):
    bl_idname = "VIEW3D_PT_test_2"
    bl_label = "Panel Two"

    def draw(self, context):
        self.layout.label("Also Small Class")
```

```
bpy.utils.register_class(PanelOne)
bpy.utils.register_class(PanelTwo)
```

base class — `bpy_struct`

**class** `bpy.types.Panel` (*bpy\_struct*)  
Panel containing UI elements

**bl\_context**

The context in which the panel belongs to. (TODO: explain the possible combinations `bl_context/bl_region_type/bl_space_type`)

**Type** string, default ""

**bl\_idname**

If this is set, the panel gets a custom ID, otherwise it takes the name of the class used to define the panel. For example, if the class name is "OBJECT\_PT\_hello", and `bl_idname` is not set by the script, then `bl_idname` = "OBJECT\_PT\_hello"

**Type** string, default ""

**bl\_label**

The panel label, shows up in the panel header at the right of the triangle used to collapse the panel

**Type** string, default ""

**bl\_options**

Options for this panel type

- `DEFAULT_CLOSED` Default Closed, Defines if the panel has to be open or collapsed at the time of its creation.
- `HIDE_HEADER` Show Header, If set to True, the panel shows a header, which contains a clickable arrow to collapse the panel and the label (see `bl_label`).

**Type** enum set in {`'DEFAULT_CLOSED'`, `'HIDE_HEADER'`}, default {`'DEFAULT_CLOSED'`}

**bl\_region\_type**

The region where the panel is going to be used in

**Type** enum in [`'WINDOW'`, `'HEADER'`, `'CHANNELS'`, `'TEMPORARY'`, `'UI'`, `'TOOLS'`, `'TOOL_PROPS'`, `'PREVIEW'`], default `'WINDOW'`

**bl\_space\_type**

The space where the panel is going to be used in

**Type** enum in ['EMPTY', 'VIEW\_3D', 'GRAPH\_EDITOR', 'OUTLINER', 'PROPERTIES', 'FILE\_BROWSER', 'IMAGE\_EDITOR', 'INFO', 'SEQUENCE\_EDITOR', 'TEXT\_EDITOR', 'DOPESHEET\_EDITOR', 'NLA\_EDITOR', 'TIMELINE', 'NODE\_EDITOR', 'LOGIC\_EDITOR', 'CONSOLE', 'USER\_PREFERENCES'], default 'EMPTY'

**layout**

Defines the structure of the panel in the UI

**Type** `UILayout`, (readonly)

**text**

XXX todo

**Type** string, default ""

**classmethod poll** (*context*)

If this method returns a non-null output, then the panel can be drawn

**Return type** boolean

**draw** (*context*)

Draw UI elements into the panel UI layout

**draw\_header** (*context*)

Draw UI elements into the panel's header UI layout

**classmethod append** (*draw\_func*)

Append a draw function to this menu, takes the same arguments as the menu draw function

**classmethod prepend** (*draw\_func*)

Prepend a draw function to this menu, takes the same arguments as the menu draw function

**classmethod remove** (*draw\_func*)

Remove a draw function that has been added to this menu

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`

- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.400 ParentActuator(Actuator)

base classes — `bpy_struct`, `Actuator`

**class** `bpy.types.ParentActuator` (*Actuator*)

**mode**

**Type** enum in ['SETPARENT', 'REMOVEPARENT'], default 'SETPARENT'

**object**

Set this object as parent

**Type** `Object`

**use\_compound**

Add this object shape to the parent shape (only if the parent shape is already compound)

**Type** boolean, default False

**use\_ghost**

Make this object ghost while parented

**Type** boolean, default False

#### Inherited Properties

- `bpy_struct.id_data`
- `Actuator.name`
- `Actuator.show_expanded`
- `Actuator.pin`
- `Actuator.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`



- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Actuator.link`
- `Actuator.unlink`

### 2.4.401 Particle(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.Particle` (*bpy\_struct*)

Particle in a particle system

**alive\_state**

**Type** enum in ['DEAD', 'UNBORN', 'ALIVE', 'DYING'], default 'DEAD'

**angular\_velocity**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**birth\_time**

**Type** float in [-inf, inf], default 0.0

**die\_time**

**Type** float in [-inf, inf], default 0.0

**hair\_keys**

**Type** `bpy_prop_collection` of `ParticleHairKey`, (readonly)

**is\_exist**

**Type** boolean, default False, (readonly)

**is\_visible**

**Type** boolean, default False, (readonly)

**lifetime**

**Type** float in [-inf, inf], default 0.0

**location**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**particle\_keys**

**Type** `bpy_prop_collection` of `ParticleKey`, (readonly)

**prev\_angular\_velocity**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**prev\_location**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**prev\_rotation**

**Type** float array of 4 items in [-inf, inf], default (0.0, 0.0, 0.0, 0.0)

**prev\_velocity**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**rotation**

**Type** float array of 4 items in [-inf, inf], default (0.0, 0.0, 0.0, 0.0)

**size**

**Type** float in [-inf, inf], default 0.0

**velocity**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `ParticleSystem.particles`

**2.4.402 ParticleBrush(`bpy_struct`)**

base class — `bpy_struct`

**class** `bpy.types.ParticleBrush` (*bpy\_struct*)  
Particle editing brush

**count**

Particle count

**Type** int in [1, 1000], default 0

**curve**

**Type** `CurveMapping`, (readonly)

**length\_mode**

- GROW Grow, Make hairs longer.
- SHRINK Shrink, Make hairs shorter.

**Type** enum in ['GROW', 'SHRINK'], default 'GROW'

#### **puff\_mode**

- ADD Add, Make hairs more puffy.
- SUB Sub, Make hairs less puffy.

**Type** enum in ['ADD', 'SUB'], default 'ADD'

#### **size**

Radius of the brush in pixels

**Type** int in [1, 32767], default 0

#### **steps**

Brush steps

**Type** int in [1, 32767], default 0

#### **strength**

Brush strength

**Type** float in [0.001, 1], default 0.0

#### **use\_puff\_volume**

Apply puff to unselected end-points (helps maintain hair volume when puffing root)

**Type** boolean, default False

### **Inherited Properties**

- `bpy_struct.id_data`

### **Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `ParticleEdit.brush`

### 2.4.403 ParticleDupliWeight(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.ParticleDupliWeight` (*bpy\_struct*)

Weight of a particle dupliobject in a group

**count**

The number of times this object is repeated with respect to other objects

**Type** int in [0, 32767], default 0

**name**

Particle dupliobject name

**Type** string, default "", (readonly)

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `ParticleSettings.active_dupliweight`
- `ParticleSettings.dupli_weights`

## 2.4.404 ParticleEdit(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.ParticleEdit` (*bpy\_struct*)

Properties of particle editing mode

**brush**

**Type** `ParticleBrush`, (readonly)

**default\_key\_count**

How many keys to make new particles with

**Type** `int` in [2, 32767], default 0

**draw\_step**

How many steps to draw the path with

**Type** `int` in [2, 10], default 0

**emitter\_distance**

Distance to keep particles away from the emitter

**Type** `float` in [0, inf], default 0.0

**fade\_frames**

How many frames to fade

**Type** `int` in [2, 100], default 0

**is\_editable**

A valid edit mode exists

**Type** `boolean`, default `False`, (readonly)

**is\_hair**

Editing hair

**Type** `boolean`, default `False`, (readonly)

**object**

The edited object

**Type** `Object`, (readonly)

**select\_mode**

Particle select and display mode

- `PATH` Path, Path edit mode.
- `POINT` Point, Point select mode.
- `TIP` Tip, Tip select mode.

**Type** `enum` in ['PATH', 'POINT', 'TIP'], default 'PATH'

**show\_particles**

Draw actual particles

**Type** `boolean`, default `False`

**tool**

- `NONE` None, Don't use any brush.

- COMB Comb, Comb hairs.
- SMOOTH Smooth, Smooth hairs.
- ADD Add, Add hairs.
- LENGTH Length, Make hairs longer or shorter.
- PUFF Puff, Make hairs stand up.
- CUT Cut, Cut hairs.
- WEIGHT Weight, Weight hair particles.

**Type** enum in ['NONE', 'COMB', 'SMOOTH', 'ADD', 'LENGTH', 'PUFF', 'CUT', 'WEIGHT'], default 'COMB'

**type**

**Type** enum in ['PARTICLES', 'SOFT\_BODY', 'CLOTH'], default 'PARTICLES'

**use\_auto\_velocity**

Calculate point velocities automatically

**Type** boolean, default False

**use\_default\_interpolate**

Interpolate new particles from the existing ones

**Type** boolean, default False

**use\_emitter\_deflect**

Keep paths from intersecting the emitter

**Type** boolean, default False

**use\_fade\_time**

Fade paths and keys further away from current frame

**Type** boolean, default False

**use\_preserve\_length**

Keep path lengths constant

**Type** boolean, default False

**use\_preserve\_root**

Keep root keys unmodified

**Type** boolean, default False

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`

- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `ToolSettings.particle_edit`

### 2.4.405 ParticleFluidSettings(FluidSettings)

base classes — `bpy_struct`, `FluidSettings`

**class** `bpy.types.ParticleFluidSettings` (*FluidSettings*)

Fluid simulation settings for objects storing fluid particles generated by the simulation

#### **alpha\_influence**

Amount of particle alpha change, inverse of size influence: 0=off (all same alpha), 1=full. (large particles get lower alphas, smaller ones higher values)

**Type** float in [0, 2], default 0.0

#### **filepath**

Directory (and/or filename prefix) to store and load particles from

**Type** string, default ""

#### **particle\_influence**

Amount of particle size scaling: 0=off (all same size), 1=full (range 0.2-2.0), >1=stronger

**Type** float in [0, 2], default 0.0

#### **show\_tracer**

Show tracer particles

**Type** boolean, default False

#### **use\_drops**

Show drop particles

**Type** boolean, default False

#### **use\_floats**

Show floating foam particles

**Type** boolean, default False

## Inherited Properties

- `bpy_struct.id_data`
- `FluidSettings.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.406 ParticleHairKey(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.ParticleHairKey` (*`bpy_struct`*)

Particle key for hair particle system

**co**

Location of the hair key in object space

**Type** float array of 3 items in  $[-inf, inf]$ , default (0.0, 0.0, 0.0)

**co\_hair\_space**

Location of the hair key in its internal coordinate system, relative to the emitting face

**Type** float array of 3 items in  $[-inf, inf]$ , default (0.0, 0.0, 0.0)

**time**

Relative time of key over hair length

**Type** float in  $[0, inf]$ , default 0.0

**weight**

Weight for cloth simulation

**Type** float in  $[0, 1]$ , default 0.0

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`



- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Particle.hair_keys`

## 2.4.407 ParticleInstanceModifier(Modifier)

base classes — `bpy_struct`, `Modifier`

**class** `bpy.types.ParticleInstanceModifier` (*Modifier*)  
Particle system instancing modifier

### **axis**

Pole axis for rotation

**Type** enum in ['X', 'Y', 'Z'], default 'X'

### **object**

Object that has the particle system

**Type** `Object`

### **particle\_system\_index**

**Type** int in [1, 10], default 0

### **position**

Position along path

**Type** float in [0, 1], default 0.0

### **random\_position**

Randomize position along path

**Type** float in [0, 1], default 0.0

### **show\_alive**

Show instances when particles are alive

**Type** boolean, default False

### **show\_dead**

Show instances when particles are dead

**Type** boolean, default False

**show\_unborn**

Show instances when particles are unborn

**Type** boolean, default False

**use\_children**

Create instances from child particles

**Type** boolean, default False

**use\_normal**

Create instances from normal particles

**Type** boolean, default False

**use\_path**

Create instances along particle paths

**Type** boolean, default False

**use\_preserve\_shape**

Don't stretch the object

**Type** boolean, default False

**use\_size**

Use particle size to scale the instances

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `Modifier.name`
- `Modifier.use_apply_on_spline`
- `Modifier.show_in_editmode`
- `Modifier.show_expanded`
- `Modifier.show_on_cage`
- `Modifier.show_viewport`
- `Modifier.show_render`
- `Modifier.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`

- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.408 ParticleKey(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.ParticleKey` (*bpy\_struct*)

Key location for a particle over time

**angular\_velocity**

Key angular velocity

**Type** float array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`

**location**

Key location

**Type** float array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`

**rotation**

Key rotation quaternion

**Type** float array of 4 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0, 0.0)`

**time**

Time of key over the simulation

**Type** float in `[0, inf]`, default `0.0`

**velocity**

Key velocity

**Type** float array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`

#### Inherited Properties

- `bpy_struct.id_data`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Particle.particle_keys`

### 2.4.409 ParticleSettings(ID)

base classes — `bpy_struct`, `ID`

**class** `bpy.types.ParticleSettings` (*ID*)

Particle settings, reusable by multiple particle systems

**active\_dupliweight**

**Type** `ParticleDupliWeight`, (readonly)

**active\_dupliweight\_index**

**Type** `int` in `[0, inf]`, default 0

**active\_texture**

Active texture slot being displayed

**Type** `Texture`

**active\_texture\_index**

Index of active texture slot

**Type** `int` in `[0, 17]`, default 0

**adaptive\_angle**

How many degrees path has to curve to make another render segment

**Type** `int` in `[0, 45]`, default 0

**adaptive\_pixel**

How many pixels path has to cover to make another render segment

**Type** `int` in `[0, 50]`, default 0

**adaptive\_subframes**

Automatically set the number of subframes

**Type** `boolean`, default `False`

**angular\_velocity\_factor**

Angular velocity amount

**Type** `float` in `[-200, 200]`, default 0.0

**angular\_velocity\_mode**

Particle angular velocity mode

**Type** `enum` in `['NONE', 'SPIN', 'RAND']`, default `'NONE'`

**animation\_data**

Animation data for this datablock

**Type** `AnimData`, (readonly)

**apply\_effector\_to\_children**

Apply effectors to children

**Type** `boolean`, default `False`

**apply\_guide\_to\_children**

**Type** boolean, default False

#### **billboard\_align**

In respect to what the billboards are aligned

**Type** enum in ['X', 'Y', 'Z', 'VIEW', 'VEL'], default 'X'

#### **billboard\_animation**

How to animate billboard textures

**Type** enum in ['NONE', 'AGE', 'FRAME', 'ANGLE'], default 'NONE'

#### **billboard\_object**

Billboards face this object (default is active camera)

**Type** [Object](#)

#### **billboard\_offset**

**Type** float array of 2 items in [-100, 100], default (0.0, 0.0)

#### **billboard\_offset\_split**

How to offset billboard textures

**Type** enum in ['NONE', 'LINEAR', 'RANDOM'], default 'NONE'

#### **billboard\_size**

Scale billboards relative to particle size

**Type** float array of 2 items in [0.001, 10], default (0.0, 0.0)

#### **billboard\_tilt**

Tilt of the billboards

**Type** float in [-1, 1], default 0.0

#### **billboard\_tilt\_random**

Random tilt of the billboards

**Type** float in [0, 1], default 0.0

#### **billboard\_uv\_split**

Amount of rows/columns to split UV coordinates for billboards

**Type** int in [1, 100], default 0

#### **billboard\_velocity\_head**

Scale billboards by velocity

**Type** float in [0, 10], default 0.0

#### **billboard\_velocity\_tail**

Scale billboards by velocity

**Type** float in [0, 10], default 0.0

#### **boids**

**Type** [BoidSettings](#), (readonly)

#### **branch\_threshold**

Threshold of branching

**Type** float in [0, 1], default 0.0

#### **brownian\_factor**

Specify the amount of Brownian motion

**Type** float in [0, 200], default 0.0

**child\_length**

Length of child paths

**Type** float in [0, 1], default 0.0

**child\_length\_threshold**

Amount of particles left untouched by child path length

**Type** float in [0, 1], default 0.0

**child\_nbr**

Amount of children/parent

**Type** int in [0, 100000], default 0

**child\_parting\_factor**

Create parting in the children based on parent strands

**Type** float in [0, 1], default 0.0

**child\_parting\_max**

Maximum root to tip angle (tip distance/root distance for long hair)

**Type** float in [0, 180], default 0.0

**child\_parting\_min**

Minimum root to tip angle (tip distance/root distance for long hair)

**Type** float in [0, 180], default 0.0

**child\_radius**

Radius of children around parent

**Type** float in [0, 10], default 0.0

**child\_roundness**

Roundness of children around parent

**Type** float in [0, 1], default 0.0

**child\_size**

A multiplier for the child particle size

**Type** float in [0.001, 100000], default 0.0

**child\_size\_random**

Random variation to the size of the child particles

**Type** float in [0, 1], default 0.0

**child\_type**

Create child particles

**Type** enum in ['NONE', 'SIMPLE', 'INTERPOLATED'], default 'NONE'

**clump\_factor**

Amount of clumping

**Type** float in [-1, 1], default 0.0

**clump\_shape**

Shape of clumping

**Type** float in [-0.999, 0.999], default 0.0

**color\_maximum**

Maximum length of the particle color vector

**Type** float in [0.01, 100], default 0.0

**count**

Total number of particles

**Type** int in [0, 10000000], default 0

**courant\_target**

The relative distance a particle can move before requiring more subframes (target Courant number); 0.1-0.3 is the recommended range

**Type** float in [0.01, 10], default 0.2

**create\_long\_hair\_children**

Calculate children that suit long hair well

**Type** boolean, default False

**damping**

Specify the amount of damping

**Type** float in [0, 1], default 0.0

**distribution**

How to distribute particles on selected element

**Type** enum in ['NONE', 'RENDER', 'DOT', 'CIRC', 'CROSS', 'AXIS'], default 'NONE'

**drag\_factor**

Specify the amount of air-drag

**Type** float in [0, 1], default 0.0

**draw\_color**

Draw additional particle data as a color

**Type** enum in ['NONE', 'MATERIAL', 'VELOCITY', 'ACCELERATION'], default 'NONE'

**draw\_method**

How particles are drawn in viewport

**Type** enum in ['NONE', 'RENDER', 'DOT', 'CIRC', 'CROSS', 'AXIS'], default 'NONE'

**draw\_percentage**

Percentage of particles to display in 3D view

**Type** int in [0, 100], default 0

**draw\_size**

Size of particles on viewport in pixels (0=default)

**Type** int in [0, 1000], default 0

**draw\_step**

How many steps paths are drawn with (power of 2)

**Type** int in [0, 10], default 0

**dupli\_group**

Show Objects in this Group in place of particles

**Type** [Group](#)

**dupli\_object**

Show this Object in place of particles

Type `Object`

**dupli\_weights**

Weights for all of the objects in the dupli group

Type `bpy_prop_collection` of `ParticleDupliWeight`, (readonly)

**effect\_hair**

Hair stiffness for effectors

Type float in [0, 1], default 0.0

**effector\_amount**

How many particles are effectors (0 is all particles)

Type int in [0, 10000], default 0

**effector\_weights**

Type `EffectorWeights`, (readonly)

**emit\_from**

Where to emit particles from

Type enum in ['VERT', 'FACE', 'VOLUME'], default 'VERT'

**factor\_random**

Give the starting speed a random variation

Type float in [0, 200], default 0.0

**fluid**

Type `SPHFluidSettings`, (readonly)

**force\_field\_1**

Type `FieldSettings`, (readonly)

**force\_field\_2**

Type `FieldSettings`, (readonly)

**frame\_end**

Frame # to stop emitting particles

Type float in [-300000, 300000], default 0.0

**frame\_start**

Frame # to start emitting particles

Type float in [-300000, 300000], default 0.0

**grid\_random**

Add random offset to the grid locations

Type float in [0, 1], default 0.0

**grid\_resolution**

The resolution of the particle grid

Type int in [1, 250], default 0

**hair\_length**

Length of the hair



**Type** float in [0, 1000], default 0.0

#### **hair\_step**

Number of hair segments

**Type** int in [2, 50], default 0

#### **hexagonal\_grid**

Create the grid in a hexagonal pattern

**Type** boolean, default False

#### **integrator**

Algorithm used to calculate physics. Fastest to most stable/accurate: Midpoint, Euler, Verlet, RK4 (Old)

**Type** enum in ['EULER', 'VERLET', 'MIDPOINT', 'RK4'], default 'EULER'

#### **invert\_grid**

Invert what is considered object and what is not

**Type** boolean, default False

#### **is\_fluid**

Particles were created by a fluid simulation

**Type** boolean, default False, (readonly)

#### **jitter\_factor**

Amount of jitter applied to the sampling

**Type** float in [0, 2], default 0.0

#### **keyed\_loops**

Number of times the keys are looped

**Type** int in [1, 10000], default 0

#### **keys\_step**

**Type** int in [0, 32767], default 0

#### **kink**

Type of periodic offset on the path

**Type** enum in ['NO', 'CURL', 'RADIAL', 'WAVE', 'BRAID'], default 'NO'

#### **kink\_amplitude**

The amplitude of the offset

**Type** float in [-100000, 100000], default 0.0

#### **kink\_amplitude\_clump**

How much clump effects kink amplitude

**Type** float in [0, 1], default 0.0

#### **kink\_axis**

Which axis to use for offset

**Type** enum in ['X', 'Y', 'Z'], default 'X'

#### **kink\_flat**

How flat the hairs are

**Type** float in [0, 1], default 0.0

**kink\_frequency**

The frequency of the offset (1/total length)

**Type** float in [-100000, 100000], default 0.0

**kink\_shape**

Adjust the offset to the beginning/end

**Type** float in [-0.999, 0.999], default 0.0

**length\_random**

Give path length a random variation

**Type** float in [0, 1], default 0.0

**lifetime**

Specify the life span of the particles

**Type** float in [1, 300000], default 0.0

**lifetime\_random**

Give the particle life a random variation

**Type** float in [0, 1], default 0.0

**line\_length\_head**

Length of the line's head

**Type** float in [0, 100000], default 0.0

**line\_length\_tail**

Length of the line's tail

**Type** float in [0, 100000], default 0.0

**lock\_billboard**

Lock the billboards align axis

**Type** boolean, default False

**lock\_boids\_to\_surface**

Constrain boids to a surface

**Type** boolean, default False

**mass**

Specify the mass of the particles

**Type** float in [0.001, 100000], default 0.0

**material**

Specify material used for the particles

**Type** int in [1, 32767], default 0

**normal\_factor**

Let the surface normal give the particle a starting speed

**Type** float in [-1000, 1000], default 0.0

**object\_align\_factor**

Let the emitter object orientation give the particle a starting speed

**Type** float array of 3 items in [-200, 200], default (0.0, 0.0, 0.0)

**object\_factor**

Let the object give the particle a starting speed

**Type** float in [-200, 200], default 0.0

**particle\_factor**

Let the target particle give the particle a starting speed

**Type** float in [-200, 200], default 0.0

**particle\_size**

The size of the particles

**Type** float in [0.001, 100000], default 0.0

**path\_end**

End time of drawn path

**Type** float in [-inf, inf], default 0.0

**path\_start**

Starting time of drawn path

**Type** float in [-inf, inf], default 0.0

**phase\_factor**

Initial rotation phase

**Type** float in [-1, 1], default 0.0

**phase\_factor\_random**

Randomize rotation phase

**Type** float in [0, 1], default 0.0

**physics\_type**

Particle physics type

**Type** enum in ['NO', 'NEWTON', 'KEYED', 'BOIDS', 'FLUID'], default 'NO'

**react\_event**

The event of target particles to react on

**Type** enum in ['DEATH', 'COLLIDE', 'NEAR'], default 'DEATH'

**reactor\_factor**

Let the vector away from the target particles location give the particle a starting speed

**Type** float in [-10, 10], default 0.0

**regrow\_hair**

Regrow hair for each frame

**Type** boolean, default False

**render\_step**

How many steps paths are rendered with (power of 2)

**Type** int in [0, 20], default 0

**render\_type**

How particles are rendered

**Type** enum in ['NONE', 'HALO', 'LINE', 'PATH', 'OBJECT', 'GROUP', 'BILLBOARD'], default 'NONE'

**rendered\_child\_count**

Amount of children/parent for rendering

**Type** int in [0, 100000], default 0

**rotation\_factor\_random**

Randomize rotation

**Type** float in [0, 1], default 0.0

**rotation\_mode**

Particle rotation axis

**Type** enum in ['NONE', 'NOR', 'VEL', 'GLOB\_X', 'GLOB\_Y', 'GLOB\_Z', 'OB\_X', 'OB\_Y', 'OB\_Z'], default 'NONE'

**roughness\_1**

Amount of location dependent rough

**Type** float in [0, 100000], default 0.0

**roughness\_1\_size**

Size of location dependent rough

**Type** float in [0.01, 100000], default 0.0

**roughness\_2**

Amount of random rough

**Type** float in [0, 100000], default 0.0

**roughness\_2\_size**

Size of random rough

**Type** float in [0.01, 100000], default 0.0

**roughness\_2\_threshold**

Amount of particles left untouched by random rough

**Type** float in [0, 1], default 0.0

**roughness\_end\_shape**

Shape of end point rough

**Type** float in [0, 10], default 0.0

**roughness\_endpoint**

Amount of end point rough

**Type** float in [0, 100000], default 0.0

**show\_health**

Draw boid health

**Type** boolean, default False

**show\_number**

Show particle number

**Type** boolean, default False

**show\_size**

Show particle size

**Type** boolean, default False

**show\_unborn**

Show particles before they are emitted

**Type** boolean, default False

- show\_velocity**  
Show particle velocity  
**Type** boolean, default False
- simplify\_rate**  
Speed of simplification  
**Type** float in [0, 1], default 0.0
- simplify\_refsize**  
Reference size in pixels, after which simplification begins  
**Type** int in [1, 32768], default 0
- simplify\_transition**  
Transition period for fading out strands  
**Type** float in [0, 1], default 0.0
- simplify\_viewport**  
Speed of Simplification  
**Type** float in [0, 0.999], default 0.0
- size\_random**  
Give the particle size a random variation  
**Type** float in [0, 1], default 0.0
- subframes**  
Subframes to simulate for improved stability and finer granularity simulations (dt = timestep / (subframes + 1))  
**Type** int in [0, 1000], default 0
- tangent\_factor**  
Let the surface tangent give the particle a starting speed  
**Type** float in [-1000, 1000], default 0.0
- tangent\_phase**  
Rotate the surface tangent  
**Type** float in [-1, 1], default 0.0
- texture\_slots**  
Texture slots defining the mapping and influence of textures  
**Type** `ParticleSettingsTextureSlots` `bpy_prop_collection` of `ParticleSettingsTextureSlot`, (readonly)
- time\_tweak**  
A multiplier for physics timestep (1.0 means one frame = 1/25 seconds)  
**Type** float in [0, 100], default 0.0
- timestep**  
The simulation timestep per frame (seconds per frame)  
**Type** float in [0.0001, 100], default 0.0
- trail\_count**  
Number of trail particles  
**Type** int in [1, 100000], default 0

- type**  
Particle Type  
**Type** enum in ['EMITTER', 'HAIR'], default 'EMITTER'
- use\_absolute\_path\_time**  
Path timing is in absolute frames  
**Type** boolean, default False
- use\_advanced\_hair**  
Use full physics calculations for growing hair  
**Type** boolean, default False
- use\_dead**  
Show particles after they have died  
**Type** boolean, default False
- use\_die\_on\_collision**  
Particles die when they collide with a deflector object  
**Type** boolean, default False
- use\_dynamic\_rotation**  
Sets rotation to dynamic/constant  
**Type** boolean, default False
- use\_emit\_random**  
Emit in random order of elements  
**Type** boolean, default False
- use\_even\_distribution**  
Use even distribution from faces based on face areas or edge lengths  
**Type** boolean, default False
- use\_global\_dupli**  
Use object's global coordinates for duplication  
**Type** boolean, default False
- use\_group\_count**  
Use object multiple times in the same group  
**Type** boolean, default False
- use\_group\_pick\_random**  
Pick objects from group randomly  
**Type** boolean, default False
- use\_hair\_bspline**  
Interpolate hair using B-Splines  
**Type** boolean, default False
- use\_multiply\_size\_mass**  
Multiply mass by particle size  
**Type** boolean, default False
- use\_parent\_particles**  
Render parent particles

**Type** boolean, default False

**use\_react\_multiple**

React multiple times

**Type** boolean, default False

**use\_react\_start\_end**

Give birth to unreacted particles eventually

**Type** boolean, default False

**use\_render\_adaptive**

Draw steps of the particle path

**Type** boolean, default False

**use\_render\_emitter**

Render emitter Object also

**Type** boolean, default False

**use\_rotation\_dupli**

Use object's rotation for duplication (global x-axis is aligned particle rotation axis)

**Type** boolean, default False

**use\_self\_effect**

Particle effectors effect themselves

**Type** boolean, default False

**use\_simplify**

Remove child strands as the object becomes smaller on the screen

**Type** boolean, default False

**use\_simplify\_viewport**

**Type** boolean, default False

**use\_size\_deflect**

Use particle's size in deflection

**Type** boolean, default False

**use\_strand\_primitive**

Use the strand primitive for rendering

**Type** boolean, default False

**use\_velocity\_length**

Multiply line length by particle speed

**Type** boolean, default False

**use\_whole\_group**

Use whole group at once

**Type** boolean, default False

**userjit**

Emission locations / face (0 = automatic)

**Type** int in [0, 1000], default 0

**virtual\_parents**

Relative amount of virtual parents

**Type** float in [0, 1], default 0.0

**Inherited Properties**

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`

**References**

- `BlendData.particles`
- `BlendDataParticles.new`
- `BlendDataParticles.remove`
- `ParticleSystem.settings`

**2.4.410 ParticleSettingsTextureSlot(TextureSlot)**

base classes — `bpy_struct`, `TextureSlot`



**class** `bpy.types.ParticleSettingsTextureSlot` (*TextureSlot*)

Texture slot for textures in a Particle Settings datablock

**clump\_factor**

Amount texture affects child clump

**Type** float in [-inf, inf], default 0.0

**damp\_factor**

Amount texture affects particle damping

**Type** float in [-inf, inf], default 0.0

**density\_factor**

Amount texture affects particle density

**Type** float in [-inf, inf], default 0.0

**field\_factor**

Amount texture affects particle force fields

**Type** float in [-inf, inf], default 0.0

**gravity\_factor**

Amount texture affects particle gravity

**Type** float in [-inf, inf], default 0.0

**kink\_factor**

Amount texture affects child kink

**Type** float in [-inf, inf], default 0.0

**length\_factor**

Amount texture affects child hair length

**Type** float in [-inf, inf], default 0.0

**life\_factor**

Amount texture affects particle life time

**Type** float in [-inf, inf], default 0.0

**mapping**

- FLAT Flat, Maps X and Y coordinates directly.
- CUBE Cube, Maps using the normal vector.
- TUBE Tube, Maps with Z as central axis.
- SPHERE Sphere, Maps with Z as central axis.

**Type** enum in ['FLAT', 'CUBE', 'TUBE', 'SPHERE'], default 'FLAT'

**mapping\_x**

**Type** enum in ['NONE', 'X', 'Y', 'Z'], default 'NONE'

**mapping\_y**

**Type** enum in ['NONE', 'X', 'Y', 'Z'], default 'NONE'

**mapping\_z**

**Type** enum in ['NONE', 'X', 'Y', 'Z'], default 'NONE'

**object**

Object to use for mapping with Object texture coordinates

**Type** Object

**rough\_factor**

Amount texture affects child roughness

**Type** float in [-inf, inf], default 0.0

**size\_factor**

Amount texture affects physical particle size

**Type** float in [-inf, inf], default 0.0

**texture\_coords**

Texture coordinates used to map the texture onto the background

- GLOBAL Global, Uses global coordinates for the texture coordinates.
- OBJECT Object, Uses linked object's coordinates for texture coordinates.
- UV UV, Uses UV coordinates for texture coordinates.
- ORCO Generated, Uses the original undeformed coordinates of the object.
- STRAND Strand / Particle, Uses normalized strand texture coordinate (1D) or particle age (X) and trail position (Y).

**Type** enum in ['GLOBAL', 'OBJECT', 'UV', 'ORCO', 'STRAND'], default 'GLOBAL'

**time\_factor**

Amount texture affects particle emission time

**Type** float in [-inf, inf], default 0.0

**use\_map\_clump**

Affect the child clumping

**Type** boolean, default False

**use\_map\_damp**

Affect the particle velocity damping

**Type** boolean, default False

**use\_map\_density**

Affect the density of the particles

**Type** boolean, default False

**use\_map\_field**

Affect the particle force fields

**Type** boolean, default False

**use\_map\_gravity**

Affect the particle gravity

**Type** boolean, default False

**use\_map\_kink**

Affect the child kink

**Type** boolean, default False

**use\_map\_length**

Affect the child hair length

**Type** boolean, default False**use\_map\_life**

Affect the life time of the particles

**Type** boolean, default False**use\_map\_rough**

Affect the child rough

**Type** boolean, default False**use\_map\_size**

Affect the particle size

**Type** boolean, default False**use\_map\_time**

Affect the emission time of the particles

**Type** boolean, default False**use\_map\_velocity**

Affect the particle initial velocity

**Type** boolean, default False**uv\_layer**

UV layer to use for mapping with UV texture coordinates

**Type** string, default ""**velocity\_factor**

Amount texture affects particle initial velocity

**Type** float in [-inf, inf], default 0.0**Inherited Properties**

- `bpy_struct.id_data`
- `TextureSlot.name`
- `TextureSlot.blend_type`
- `TextureSlot.color`
- `TextureSlot.default_value`
- `TextureSlot.invert`
- `TextureSlot.offset`
- `TextureSlot.output_node`
- `TextureSlot.use_rgb_to_intensity`
- `TextureSlot.scale`
- `TextureSlot.use_stencil`
- `TextureSlot.texture`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`

- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `ParticleSettings.texture_slots`
- `ParticleSettingsTextureSlots.add`
- `ParticleSettingsTextureSlots.create`

### 2.4.411 ParticleSettingsTextureSlots(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.ParticleSettingsTextureSlots` (*bpy\_struct*)

Collection of texture slots

**classmethod** `add()`

`add`

**Returns** The newly initialized `mtx`

**Return type** `ParticleSettingsTextureSlot`

**classmethod** `create(index)`

`create`

**Parameters** `index` (*int in [0, inf]*) – Index, Slot index to initialize

**Returns** The newly initialized `mtx`

**Return type** `ParticleSettingsTextureSlot`

**classmethod** `clear(index)`

`clear`

**Parameters** `index` (*int in [0, inf]*) – Index, Slot index to clear

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`

- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `ParticleSystem.texture_slots`

### 2.4.412 ParticleSystem(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.ParticleSystem` (*`bpy_struct`*)

Particle system in an object

**active\_particle\_target**

**Type** `ParticleTarget`, (readonly)

**active\_particle\_target\_index**

**Type** `int` in `[0, inf]`, default `0`

**billboard\_normal\_uv**

UV Layer to control billboard normals

**Type** `string`, default `""`

**billboard\_split\_uv**

UV Layer to control billboard splitting

**Type** `string`, default `""`

**billboard\_time\_index\_uv**

UV Layer to control billboard time index (X-Y)

**Type** `string`, default `""`

**child\_particles**

Child particles generated by the particle system

**Type** `bpy_prop_collection` of `ChildParticle`, (readonly)

**child\_seed**

Offset in the random number table for child particles, to get a different randomized result

**Type** `int` in `[0, inf]`, default `0`

**cloth**

Cloth dynamics for hair

**Type** `ClothModifier`, (readonly, never None)

**dt\_frac**

The current simulation time step size, as a fraction of a frame

**Type** float in [0.00990099, 1], default 0.0, (readonly)

**has\_multiple\_caches**

Particle system has multiple point caches

**Type** boolean, default False, (readonly)

**invert\_vertex\_group\_clump**

Negate the effect of the clump vertex group

**Type** boolean, default False

**invert\_vertex\_group\_density**

Negate the effect of the density vertex group

**Type** boolean, default False

**invert\_vertex\_group\_field**

Negate the effect of the field vertex group

**Type** boolean, default False

**invert\_vertex\_group\_kink**

Negate the effect of the kink vertex group

**Type** boolean, default False

**invert\_vertex\_group\_length**

Negate the effect of the length vertex group

**Type** boolean, default False

**invert\_vertex\_group\_rotation**

Negate the effect of the rotation vertex group

**Type** boolean, default False

**invert\_vertex\_group\_roughness\_1**

Negate the effect of the roughness 1 vertex group

**Type** boolean, default False

**invert\_vertex\_group\_roughness\_2**

Negate the effect of the roughness 2 vertex group

**Type** boolean, default False

**invert\_vertex\_group\_roughness\_end**

Negate the effect of the roughness end vertex group

**Type** boolean, default False

**invert\_vertex\_group\_size**

Negate the effect of the size vertex group

**Type** boolean, default False

**invert\_vertex\_group\_tangent**

Negate the effect of the tangent vertex group

**Type** boolean, default False

**invert\_vertex\_group\_velocity**

Negate the effect of the velocity vertex group

**Type** boolean, default False

**is\_editable**

Particle system can be edited in particle mode

**Type** boolean, default False, (readonly)

**is\_edited**

Particle system has been edited in particle mode

**Type** boolean, default False, (readonly)

**is\_global\_hair**

Hair keys are in global coordinate space

**Type** boolean, default False, (readonly)

**name**

Particle system name

**Type** string, default ""

**parent**

Use this object's coordinate system instead of global coordinate system

**Type** `Object`

**particles**

Particles generated by the particle system

**Type** `bpy_prop_collection` of `Particle`, (readonly)

**point\_cache**

**Type** `PointCache`, (readonly, never None)

**reactor\_target\_object**

For reactor systems, the object that has the target particle system (empty if same object)

**Type** `Object`

**reactor\_target\_particle\_system**

For reactor systems, index of particle system on the target object

**Type** int in [1, 32767], default 0

**seed**

Offset in the random number table, to get a different randomized result

**Type** int in [0, inf], default 0

**settings**

Particle system settings

**Type** `ParticleSettings`, (never None)

**targets**

Target particle systems

**Type** `bpy_prop_collection` of `ParticleTarget`, (readonly)

- use\_hair\_dynamics**  
Enable hair dynamics using cloth simulation  
**Type** boolean, default False
- use\_keyed\_timing**  
Use key times  
**Type** boolean, default False
- vertex\_group\_clump**  
Vertex group to control clump  
**Type** string, default ""
- vertex\_group\_density**  
Vertex group to control density  
**Type** string, default ""
- vertex\_group\_field**  
Vertex group to control field  
**Type** string, default ""
- vertex\_group\_kink**  
Vertex group to control kink  
**Type** string, default ""
- vertex\_group\_length**  
Vertex group to control length  
**Type** string, default ""
- vertex\_group\_rotation**  
Vertex group to control rotation  
**Type** string, default ""
- vertex\_group\_roughness\_1**  
Vertex group to control roughness 1  
**Type** string, default ""
- vertex\_group\_roughness\_2**  
Vertex group to control roughness 2  
**Type** string, default ""
- vertex\_group\_roughness\_end**  
Vertex group to control roughness end  
**Type** string, default ""
- vertex\_group\_size**  
Vertex group to control size  
**Type** string, default ""
- vertex\_group\_tangent**  
Vertex group to control tangent  
**Type** string, default ""
- vertex\_group\_velocity**  
Vertex group to control velocity



**Type** string, default ""

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Object.particle_systems`
- `ParticleSystemModifier.particle_system`
- `ParticleSystems.active`
- `PointDensity.particle_system`
- `SmokeFlowSettings.particle_system`

## 2.4.413 ParticleSystemModifier(Modifier)

base classes — `bpy_struct`, `Modifier`

**class** `bpy.types.ParticleSystemModifier` (*Modifier*)  
Particle system simulation modifier

### **particle\_system**

Particle System that this modifier controls

**Type** `ParticleSystem`, (readonly, never None)

### Inherited Properties

- `bpy_struct.id_data`
- `Modifier.name`
- `Modifier.use_apply_on_spline`
- `Modifier.show_in_editmode`

- `Modifier.show_expanded`
- `Modifier.show_on_cage`
- `Modifier.show_viewport`
- `Modifier.show_render`
- `Modifier.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.414 ParticleSystems(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.ParticleSystems` (*bpy\_struct*)  
Collection of particle systems

**active**

Active particle system being displayed

**Type** `ParticleSystem`, (readonly)

**active\_index**

Index of active particle system slot

**Type** `int` in `[0, inf]`, default 0

#### Inherited Properties

- `bpy_struct.id_data`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`

- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Object.particle_systems`

### 2.4.415 ParticleTarget(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.ParticleTarget` (*bpy\_struct*)

Target particle system

#### **alliance**

**Type** enum in ['FRIEND', 'NEUTRAL', 'ENEMY'], default 'NEUTRAL'

#### **duration**

**Type** float in [0, 30000], default 0.0

#### **is\_valid**

Keyed particles target is valid

**Type** boolean, default False

#### **name**

Particle target name

**Type** string, default "", (readonly)

#### **object**

The object that has the target particle system (empty if same object)

**Type** `Object`

#### **system**

The index of particle system on the target object

**Type** int in [1, inf], default 0

#### **time**

**Type** float in [0, 30000], default 0.0

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `ParticleSystem.active_particle_target`
- `ParticleSystem.targets`

### 2.4.416 PivotConstraint(Constraint)

base classes — `bpy_struct`, `Constraint`

**class** `bpy.types.PivotConstraint` (*Constraint*)

Rotate around a different point

**head\_tail**

Target along length of bone: Head=0, Tail=1

**Type** float in [0, 1], default 0.0

**offset**

Offset of pivot from target (when set), or from owner's location (when Fixed Position is off), or the absolute pivot point

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**rotation\_range**

Rotation range on which pivoting should occur

- `ALWAYS_ACTIVE` Always, Use the pivot point in every rotation.
- `NX -X Rot`, Use the pivot point in the negative rotation range around the X-axis.
- `NY -Y Rot`, Use the pivot point in the negative rotation range around the Y-axis.
- `NZ -Z Rot`, Use the pivot point in the negative rotation range around the Z-axis.
- `X X Rot`, Use the pivot point in the positive rotation range around the X-axis.
- `Y Y Rot`, Use the pivot point in the positive rotation range around the Y-axis.
- `Z Z Rot`, Use the pivot point in the positive rotation range around the Z-axis.

**Type** enum in ['ALWAYS\_ACTIVE', 'NX', 'NY', 'NZ', 'X', 'Y', 'Z'], default 'NX'

**subtarget**

**Type** string, default ""

**target**

Target Object, defining the position of the pivot when defined

**Type** Object

**use\_relative\_location**

Offset will be an absolute point in space instead of relative to the target

**Type** boolean, default False

### Inherited Properties

- bpy\_struct.id\_data
- Constraint.name
- Constraint.active
- Constraint.mute
- Constraint.show\_expanded
- Constraint.influence
- Constraint.error\_location
- Constraint.owner\_space
- Constraint.is\_proxy\_local
- Constraint.error\_rotation
- Constraint.target\_space
- Constraint.type
- Constraint.is\_valid

### Inherited Functions

- bpy\_struct.as\_pointer
- bpy\_struct.callback\_add
- bpy\_struct.callback\_remove
- bpy\_struct.driver\_add
- bpy\_struct.driver\_remove
- bpy\_struct.get
- bpy\_struct.is\_property\_hidden
- bpy\_struct.is\_property\_set
- bpy\_struct.items
- bpy\_struct.keyframe\_delete
- bpy\_struct.keyframe\_insert
- bpy\_struct.keys
- bpy\_struct.path\_from\_id
- bpy\_struct.path\_resolve
- bpy\_struct.type\_recast
- bpy\_struct.values

## 2.4.417 PluginSequence(EffectSequence)

base classes — bpy\_struct, Sequence, EffectSequence

**class** `bpy.types.PluginSequence` (*EffectSequence*)  
Sequence strip applying an effect, loaded from an external plugin

**filename**

**Type** string, default "", (readonly)

### Inherited Properties

- `bpy_struct.id_data`
- `Sequence.name`
- `Sequence.blend_type`
- `Sequence.blend_alpha`
- `Sequence.channel`
- `Sequence.waveform`
- `Sequence.effect_fader`
- `Sequence.frame_final_end`
- `Sequence.frame_offset_end`
- `Sequence.frame_still_end`
- `Sequence.input_1`
- `Sequence.input_2`
- `Sequence.input_3`
- `Sequence.select_left_handle`
- `Sequence.frame_final_duration`
- `Sequence.frame_duration`
- `Sequence.lock`
- `Sequence.mute`
- `Sequence.select_right_handle`
- `Sequence.select`
- `Sequence.speed_factor`
- `Sequence.frame_start`
- `Sequence.frame_final_start`
- `Sequence.frame_offset_start`
- `Sequence.frame_still_start`
- `Sequence.type`
- `Sequence.use_default_fade`
- `Sequence.input_count`
- `EffectSequence.color_balance`
- `EffectSequence.use_float`
- `EffectSequence.crop`
- `EffectSequence.use_deinterlace`
- `EffectSequence.use_reverse_frames`
- `EffectSequence.use_flip_x`
- `EffectSequence.use_flip_y`
- `EffectSequence.color_multiply`
- `EffectSequence.use_premultiply`
- `EffectSequence.proxy`
- `EffectSequence.use_proxy_custom_directory`
- `EffectSequence.use_proxy_custom_file`
- `EffectSequence.color_saturation`
- `EffectSequence.strobe`
- `EffectSequence.transform`
- `EffectSequence.use_color_balance`
- `EffectSequence.use_crop`

- `EffectSequence.use_proxy`
- `EffectSequence.use_translation`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Sequence.getStripElem`
- `Sequence.swap`

## 2.4.418 PluginTexture(Texture)

base classes — `bpy_struct`, `ID`, `Texture`

**class** `bpy.types.PluginTexture` (*Texture*)

External plugin texture

**users\_material**

Materials that use this texture (readonly)

**users\_object\_modifier**

Object modifiers that use this texture (readonly)

### Inherited Properties

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`
- `Texture.animation_data`
- `Texture.intensity`
- `Texture.color_ramp`
- `Texture.contrast`
- `Texture.factor_blue`
- `Texture.factor_green`

- `Texture.factor_red`
- `Texture.node_tree`
- `Texture.saturation`
- `Texture.use_preview_alpha`
- `Texture.type`
- `Texture.use_color_ramp`
- `Texture.use_nodes`
- `Texture.users_material`
- `Texture.users_object_modifier`
- `Texture.users_material`
- `Texture.users_object_modifier`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`
- `Texture.evaluate`

### 2.4.419 PointCache(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.PointCache` (*bpy\_struct*)

Point cache for physics simulations

#### **compression**

Compression method to be used

- **NO** No, No compression.
- **LIGHT** Light, Fast but not so effective compression.
- **HEAVY** Heavy, Effective but slow compression.

**Type** enum in ['NO', 'LIGHT', 'HEAVY'], default 'NO'



**filepath**

Cache file path

**Type** string, default ""**frame\_end**

Frame on which the simulation stops

**Type** int in [1, 300000], default 0**frame\_start**

Frame on which the simulation starts

**Type** int in [1, 300000], default 0**frame\_step**

Number of frames between cached frames

**Type** int in [1, 20], default 0**frames\_skipped****Type** boolean, default False, (readonly)**index**

Index number of cache files

**Type** int in [-1, 100], default 0**info**

Info on current cache status

**Type** string, default "", (readonly)**is\_baked****Type** boolean, default False, (readonly)**is\_baking****Type** boolean, default False, (readonly)**is\_outdated****Type** boolean, default False, (readonly)**name**

Cache name

**Type** string, default ""**point\_caches**

Point cache list

**Type** [PointCaches](#) [bpy\\_prop\\_collection](#) of [PointCache](#), (readonly)**use\_disk\_cache**

Save cache files to disk (.blend file must be saved first)

**Type** boolean, default False**use\_external**

Read cache from an external location

**Type** boolean, default False**use\_library\_path**

Use this files path when library linked into another file

**Type** boolean, default False

**use\_quick\_cache**

Update simulation with cache steps

**Type** boolean, default False

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `ClothModifier.point_cache`
- `ParticleSystem.point_cache`
- `PointCache.point_caches`
- `SmokeDomainSettings.point_cache`
- `SoftBodyModifier.point_cache`

**2.4.420 PointCaches(`bpy_struct`)**

base class — `bpy_struct`

**class** `bpy.types.PointCaches` (*`bpy_struct`*)  
Collection of point caches

**active\_index**

**Type** int in [0, inf], default 0

**Inherited Properties**

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `PointCache.point_caches`

### 2.4.421 `PointDensity(bpy_struct)`

base class — `bpy_struct`

**class** `bpy.types.PointDensity` (*bpy\_struct*)  
Point density settings

#### **color\_ramp**

**Type** `ColorRamp`, (readonly)

#### **color\_source**

Data to derive color results from

- `CONSTANT` Constant.
- `PARTICLE_AGE` Particle Age, Lifetime mapped as 0.0 - 1.0 intensity.
- `PARTICLE_SPEED` Particle Speed, Particle speed (absolute magnitude of velocity) mapped as 0.0-1.0 intensity.
- `PARTICLE_VELOCITY` Particle Velocity, XYZ velocity mapped to RGB colors.

**Type** enum in [`'CONSTANT'`, `'PARTICLE_AGE'`, `'PARTICLE_SPEED'`, `'PARTICLE_VELOCITY'`], default `'CONSTANT'`

#### **falloff**

Method of attenuating density by distance from the point

- `STANDARD` Standard.
- `SMOOTH` Smooth.
- `SOFT` Soft.

- CONSTANT** Constant, Density is constant within lookup radius.
- ROOT** Root.
- PARTICLE\_AGE** Particle Age.
- PARTICLE\_VELOCITY** Particle Velocity.

**Type** enum in ['STANDARD', 'SMOOTH', 'SOFT', 'CONSTANT', 'ROOT', 'PARTICLE\_AGE', 'PARTICLE\_VELOCITY'], default 'STANDARD'

#### **falloff\_curve**

**Type** `CurveMapping`, (readonly)

#### **falloff\_soft**

Softness of the 'soft' falloff option

**Type** float in [0.01, inf], default 0.0

#### **falloff\_speed\_scale**

Multiplier to bring particle speed within an acceptable range

**Type** float in [0.001, 100], default 0.0

#### **noise\_basis**

Noise formula used for turbulence

- BLENDER\_ORIGINAL** Blender Original, Noise algorithm - Blender original: Smooth interpolated noise.
- ORIGINAL\_PERLIN** Original Perlin, Noise algorithm - Original Perlin: Smooth interpolated noise.
- IMPROVED\_PERLIN** Improved Perlin, Noise algorithm - Improved Perlin: Smooth interpolated noise.
- VORONOI\_F1** Voronoi F1, Noise algorithm - Voronoi F1: Returns distance to the closest feature point.
- VORONOI\_F2** Voronoi F2, Noise algorithm - Voronoi F2: Returns distance to the 2nd closest feature point.
- VORONOI\_F3** Voronoi F3, Noise algorithm - Voronoi F3: Returns distance to the 3rd closest feature point.
- VORONOI\_F4** Voronoi F4, Noise algorithm - Voronoi F4: Returns distance to the 4th closest feature point.
- VORONOI\_F2\_F1** Voronoi F2-F1, Noise algorithm - Voronoi F1-F2.
- VORONOI\_CRACKLE** Voronoi Crackle, Noise algorithm - Voronoi Crackle: Voronoi tessellation with sharp edges.
- CELL\_NOISE** Cell Noise, Noise algorithm - Cell Noise: Square cell tessellation.

**Type** enum in ['BLENDER\_ORIGINAL', 'ORIGINAL\_PERLIN', 'IMPROVED\_PERLIN', 'VORONOI\_F1', 'VORONOI\_F2', 'VORONOI\_F3', 'VORONOI\_F4', 'VORONOI\_F2\_F1', 'VORONOI\_CRACKLE', 'CELL\_NOISE'], default 'BLENDER\_ORIGINAL'

#### **object**

Object to take point data from

**Type** `Object`

**particle\_cache\_space**

Coordinate system to cache particles in

**Type** enum in ['OBJECT\_LOCATION', 'OBJECT\_SPACE', 'WORLD\_SPACE'], default 'OBJECT\_LOCATION'

**particle\_system**

Particle System to render as points

**Type** `ParticleSystem`

**point\_source**

Point data to use as renderable point density

- PARTICLE\_SYSTEM Particle System, Generate point density from a particle system.
- OBJECT Object Vertices, Generate point density from an object's vertices.

**Type** enum in ['PARTICLE\_SYSTEM', 'OBJECT'], default 'PARTICLE\_SYSTEM'

**radius**

Radius from the shaded sample to look for points within

**Type** float in [0.001, inf], default 0.0

**speed\_scale**

Multiplier to bring particle speed within an acceptable range

**Type** float in [0.001, 100], default 0.0

**turbulence\_depth**

Level of detail in the added turbulent noise

**Type** int in [0, 30], default 0

**turbulence\_influence**

Method for driving added turbulent noise

- STATIC Static, Noise patterns will remain unchanged, faster and suitable for stills.
- PARTICLE\_VELOCITY Particle Velocity, Turbulent noise driven by particle velocity.
- PARTICLE\_AGE Particle Age, Turbulent noise driven by the particle's age between birth and death.
- GLOBAL\_TIME Global Time, Turbulent noise driven by the global current frame.

**Type** enum in ['STATIC', 'PARTICLE\_VELOCITY', 'PARTICLE\_AGE', 'GLOBAL\_TIME'], default 'STATIC'

**turbulence\_scale**

Scale of the added turbulent noise

**Type** float in [0.01, inf], default 0.0

**turbulence\_strength**

**Type** float in [0.01, inf], default 0.0

**use\_falloff\_curve**

Use a custom falloff curve

**Type** boolean, default False

**use\_turbulence**

Add directed noise to the density at render-time

**Type** boolean, default False

**vertex\_cache\_space**

Coordinate system to cache vertices in

**Type** enum in ['OBJECT\_LOCATION', 'OBJECT\_SPACE', 'WORLD\_SPACE'], default 'OBJECT\_LOCATION'

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `PointDensityTexture.point_density`

## 2.4.422 PointDensityTexture(Texture)

base classes — `bpy_struct`, `ID`, `Texture`

**class** `bpy.types.PointDensityTexture` (*Texture*)  
Settings for the Point Density texture

**point\_density**

The point density settings associated with this texture

**Type** `PointDensity`, (readonly)

**users\_material**

Materials that use this texture (readonly)

**users\_object\_modifier**

Object modifiers that use this texture (readonly)

## Inherited Properties

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`
- `Texture.animation_data`
- `Texture.intensity`
- `Texture.color_ramp`
- `Texture.contrast`
- `Texture.factor_blue`
- `Texture.factor_green`
- `Texture.factor_red`
- `Texture.node_tree`
- `Texture.saturation`
- `Texture.use_preview_alpha`
- `Texture.type`
- `Texture.use_color_ramp`
- `Texture.use_nodes`
- `Texture.users_material`
- `Texture.users_object_modifier`
- `Texture.users_material`
- `Texture.users_object_modifier`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`
- `Texture.evaluate`

## 2.4.423 PointLamp(Lamp)

base classes — `bpy_struct`, `ID`, `Lamp`

**class** `bpy.types.PointLamp` (*Lamp*)

Omnidirectional point lamp

**falloff\_curve**

Custom Lamp Falloff Curve

**Type** `CurveMapping`, (readonly)

**falloff\_type**

Intensity Decay with distance

**Type** enum in ['CONSTANT', 'INVERSE\_LINEAR', 'INVERSE\_SQUARE', 'CUSTOM\_CURVE', 'LINEAR\_QUADRATIC\_WEIGHTED'], default 'CONSTANT'

**linear\_attenuation**

Linear distance attenuation

**Type** float in [0, 1], default 0.0

**quadratic\_attenuation**

Quadratic distance attenuation

**Type** float in [0, 1], default 0.0

**shadow\_adaptive\_threshold**

Threshold for Adaptive Sampling (Raytraced shadows)

**Type** float in [0, 1], default 0.0

**shadow\_color**

Color of shadows cast by the lamp

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**shadow\_method**

Method to compute lamp shadow with

- NOSHADOW No Shadow.
- RAY\_SHADOW Ray Shadow, Use ray tracing for shadow.

**Type** enum in ['NOSHADOW', 'RAY\_SHADOW'], default 'NOSHADOW'

**shadow\_ray\_sample\_method**

Method for generating shadow samples: Adaptive QMC is fastest, Constant QMC is less noisy but slower

**Type** enum in ['ADAPTIVE\_QMC', 'CONSTANT\_QMC'], default 'ADAPTIVE\_QMC'

**shadow\_ray\_samples**

Amount of samples taken extra (samples x samples)

**Type** int in [1, 64], default 0

**shadow\_soft\_size**

Light size for ray shadow sampling (Raytraced shadows)

**Type** float in [-inf, inf], default 0.0

**use\_only\_shadow**

Causes light to cast shadows only without illuminating objects



**Type** boolean, default False

**use\_shadow\_layer**

Causes only objects on the same layer to cast shadows

**Type** boolean, default False

**use\_sphere**

Sets light intensity to zero beyond lamp distance

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`
- `Lamp.active_texture`
- `Lamp.active_texture_index`
- `Lamp.animation_data`
- `Lamp.color`
- `Lamp.use_diffuse`
- `Lamp.distance`
- `Lamp.energy`
- `Lamp.use_own_layer`
- `Lamp.use_negative`
- `Lamp.node_tree`
- `Lamp.use_specular`
- `Lamp.texture_slots`
- `Lamp.type`
- `Lamp.use_nodes`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`

- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`

### 2.4.424 PointerProperty(Property)

base classes — `bpy_struct`, `Property`

**class** `bpy.types.PointerProperty` (*Property*)  
RNA pointer property to point to another RNA struct

**fixed\_type**

Fixed pointer type, empty if variable type

**Type** `Struct`, (readonly)

#### Inherited Properties

- `bpy_struct.id_data`
- `Property.name`
- `Property.is_animatable`
- `Property.srna`
- `Property.description`
- `Property.is_enum_flag`
- `Property.is_hidden`
- `Property.identifier`
- `Property.is_never_none`
- `Property.is_readonly`
- `Property.is_registered`
- `Property.is_registered_optional`
- `Property.is_required`
- `Property.is_output`
- `Property.is_runtime`
- `Property.is_skip_save`
- `Property.subtype`
- `Property.type`
- `Property.unit`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`

- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.425 Pose(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.Pose` (*`bpy_struct`*)

A collection of pose channels, including settings for animating bones

**animation\_visualisation**

Animation data for this datablock

**Type** `AnimViz`, (readonly, never None)

**bone\_groups**

Groups of the bones

**Type** `BoneGroups` `bpy_prop_collection` of `BoneGroup`, (readonly)

**bones**

Individual pose bones for the armature

**Type** `bpy_prop_collection` of `PoseBone`, (readonly)

**ik\_param**

Parameters for IK solver

**Type** `IKParam`, (readonly)

**ik\_solver**

Selection of IK solver for IK chain, current choice is 0 for Legacy, 1 for iTaSC

- `LEGACY` Legacy, Original IK solver.
- `ITASC` iTaSC, Multi constraint, stateful IK solver.

**Type** `enum` in [`'LEGACY'`, `'ITASC'`], default `'LEGACY'`

#### Inherited Properties

- `bpy_struct.id_data`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`

- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Object.pose`

### 2.4.426 PoseBone(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.PoseBone` (*bpy\_struct*)

Channel defining pose data for a bone in a Pose

#### **bone**

Bone associated with this PoseBone

**Type** `Bone`, (readonly, never None)

#### **bone\_group**

Bone Group this pose channel belongs to

**Type** `BoneGroup`

#### **bone\_group\_index**

Bone Group this pose channel belongs to (0=no group)

**Type** `int` in `[-32768, 32767]`, default 0

#### **child**

Child of this pose bone

**Type** `PoseBone`, (readonly)

#### **constraints**

Constraints that act on this PoseChannel

**Type** `PoseBoneConstraints` `bpy_prop_collection` of `Constraint`, (readonly)

#### **custom\_shape**

Object that defines custom draw type for this bone

**Type** `Object`

#### **custom\_shape\_transform**

Bone that defines the display transform of this custom shape

**Type** `PoseBone`

#### **head**

Location of head of the channel's bone

**Type** `float` array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`, (readonly)

**ik\_linear\_weight**

Weight of scale constraint for IK

**Type** float in [0, 1], default 0.0**ik\_max\_x**

Maximum angles for IK Limit

**Type** float in [0, 3.14159], default 0.0**ik\_max\_y**

Maximum angles for IK Limit

**Type** float in [0, 3.14159], default 0.0**ik\_max\_z**

Maximum angles for IK Limit

**Type** float in [0, 3.14159], default 0.0**ik\_min\_x**

Minimum angles for IK Limit

**Type** float in [-3.14159, 0], default 0.0**ik\_min\_y**

Minimum angles for IK Limit

**Type** float in [-3.14159, 0], default 0.0**ik\_min\_z**

Minimum angles for IK Limit

**Type** float in [-3.14159, 0], default 0.0**ik\_rotation\_weight**

Weight of rotation constraint for IK

**Type** float in [0, 1], default 0.0**ik\_stiffness\_x**

IK stiffness around the X axis

**Type** float in [0, 0.99], default 0.0**ik\_stiffness\_y**

IK stiffness around the Y axis

**Type** float in [0, 0.99], default 0.0**ik\_stiffness\_z**

IK stiffness around the Z axis

**Type** float in [0, 0.99], default 0.0**ik\_stretch**

Allow scaling of the bone for IK

**Type** float in [0, 1], default 0.0**is\_in\_ik\_chain**

Is part of an IK chain

**Type** boolean, default False, (readonly)**location**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**lock\_ik\_x**

Disallow movement around the X axis

**Type** boolean, default False

**lock\_ik\_y**

Disallow movement around the Y axis

**Type** boolean, default False

**lock\_ik\_z**

Disallow movement around the Z axis

**Type** boolean, default False

**lock\_location**

Lock editing of location in the interface

**Type** boolean array of 3 items, default (False, False, False)

**lock\_rotation**

Lock editing of rotation in the interface

**Type** boolean array of 3 items, default (False, False, False)

**lock\_rotation\_w**

Lock editing of 'angle' component of four-component rotations in the interface

**Type** boolean, default False

**lock\_rotations\_4d**

Lock editing of four component rotations by components (instead of as Eulers)

**Type** boolean, default False

**lock\_scale**

Lock editing of scale in the interface

**Type** boolean array of 3 items, default (False, False, False)

**matrix**

Final 4x4 matrix after constraints and drivers are applied (object space)

**Type** float array of 16 items in [-inf, inf], default (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)

**matrix\_basis**

Alternative access to location/scale/rotation relative to the parent and own rest bone

**Type** float array of 16 items in [-inf, inf], default (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)

**matrix\_channel**

4x4 matrix, before constraints

**Type** float array of 16 items in [-inf, inf], default (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0), (readonly)

**motion\_path**

Motion Path for this element

**Type** `MotionPath`, (readonly)

**name**

**Type** string, default ""

**parent**

Parent of this pose bone

**Type** PoseBone, (readonly)

**rotation\_axis\_angle**

Angle of Rotation for Axis-Angle rotation representation

**Type** float array of 4 items in [-inf, inf], default (0.0, 0.0, 1.0, 0.0)

**rotation\_euler**

Rotation in Eulers

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**rotation\_mode**

- QUATERNION Quaternion (WXYZ), No Gimbal Lock (default).
- XYZ XYZ Euler, XYZ Rotation Order (prone to Gimbal Lock).
- XZY XZY Euler, XZY Rotation Order (prone to Gimbal Lock).
- YXZ YXZ Euler, YXZ Rotation Order (prone to Gimbal Lock).
- YZX YZX Euler, YZX Rotation Order (prone to Gimbal Lock).
- ZXY ZXY Euler, ZXY Rotation Order (prone to Gimbal Lock).
- ZYZ ZYX Euler, ZYX Rotation Order (prone to Gimbal Lock).
- AXIS\_ANGLE Axis Angle, Axis Angle (W+XYZ), defines a rotation around some axis defined by 3D-Vector.

**Type** enum in ['QUATERNION', 'XYZ', 'XZY', 'YXZ', 'YZX', 'ZXY', 'ZYZ', 'AXIS\_ANGLE'], default 'QUATERNION'

**rotation\_quaternion**

Rotation in Quaternions

**Type** float array of 4 items in [-inf, inf], default (1.0, 0.0, 0.0, 0.0)

**scale**

**Type** float array of 3 items in [-inf, inf], default (1.0, 1.0, 1.0)

**tail**

Location of tail of the channel's bone

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0), (readonly)

**use\_ik\_limit\_x**

Limit movement around the X axis

**Type** boolean, default False

**use\_ik\_limit\_y**

Limit movement around the Y axis

**Type** boolean, default False

**use\_ik\_limit\_z**

Limit movement around the Z axis

**Type** boolean, default False

**use\_ik\_linear\_control**

Apply channel size as IK constraint if stretching is enabled

**Type** boolean, default False

**use\_ik\_rotation\_control**

Apply channel rotation as IK constraint

**Type** boolean, default False

**basename**

The name of this bone before any '.' character (readonly)

**center**

The midpoint between the head and the tail. (readonly)

**children**

A list of all the bones children. (readonly)

**children\_recursive**

A list of all children from this bone. (readonly)

**children\_recursive\_basename**

Returns a chain of children with the same base name as this bone. Only direct chains are supported, forks caused by multiple children with matching base names will terminate the function and not be returned. (readonly)

**length**

**The distance from head to tail,** when set the head is moved to fit the length.

**parent\_recursive**

A list of parents, starting with the immediate parent (readonly)

**vector**

**The direction this bone is pointing.** Utility function for (tail - head)

(readonly)

**x\_axis**

Vector pointing down the x-axis of the bone. (readonly)

**y\_axis**

Vector pointing down the x-axis of the bone. (readonly)

**z\_axis**

Vector pointing down the x-axis of the bone. (readonly)

**evaluate\_envelope** (*point*)

Calculate bone envelope at given point

**Parameters** **point** (*float array of 3 items in [-inf, inf]*) – Point, Position in 3d space to evaluate

**Returns** Factor, Envelope factor

**Return type** float in [-inf, inf]

**parent\_index** (*parent\_test*)

The same as 'bone in other\_bone.parent\_recursive' but saved generating a list.

**translate** (*vec*)

Utility function to add *vec* to the head and tail of this bone



## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Pose.bones`
- `PoseBone.child`
- `PoseBone.custom_shape_transform`
- `PoseBone.parent`

### 2.4.427 PoseBoneConstraints(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.PoseBoneConstraints` (*bpy\_struct*)

Collection of pose bone constraints

#### **active**

Active PoseChannel constraint

**Type** `Constraint`

#### **new** (*type*)

Add a constraint to this object

**Parameters** `type` (*enum* in [`'COPY_LOCATION'`, `'COPY_ROTATION'`, `'COPY_SCALE'`, `'COPY_TRANSFORMS'`, `'LIMIT_DISTANCE'`, `'LIMIT_LOCATION'`, `'LIMIT_ROTATION'`, `'LIMIT_SCALE'`, `'MAINTAIN_VOLUME'`, `'TRANSFORM'`, `'CLAMP_TO'`, `'DAMPED_TRACK'`, `'IK'`, `'LOCKED_TRACK'`, `'SPLINE_IK'`, `'STRETCH_TO'`, `'TRACK_TO'`, `'ACTION'`, `'CHILD_OF'`, `'FLOOR'`, `'FOLLOW_PATH'`, `'PIVOT'`, `'RIGID_BODY_JOINT'`, `'SCRIPT'`, `'SHRINKWRAP'`]) – Constraint type to add

- `COPY_LOCATION` Copy Location.

- `COPY_ROTATION` Copy Rotation.
- `COPY_SCALE` Copy Scale.
- `COPY_TRANSFORMS` Copy Transforms.
- `LIMIT_DISTANCE` Limit Distance.
- `LIMIT_LOCATION` Limit Location.
- `LIMIT_ROTATION` Limit Rotation.
- `LIMIT_SCALE` Limit Scale.
- `MAINTAIN_VOLUME` Maintain Volume.
- `TRANSFORM` Transformation.
- `CLAMP_TO` Clamp To.
- `DAMPED_TRACK` Damped Track, Tracking by taking the shortest path.
- `IK` Inverse Kinematics.
- `LOCKED_TRACK` Locked Track, Tracking along a single axis.
- `SPLINE_IK` Spline IK.
- `STRETCH_TO` Stretch To.
- `TRACK_TO` Track To, Legacy tracking constraint prone to twisting artifacts.
- `ACTION` Action.
- `CHILD_OF` Child Of.
- `FLOOR` Floor.
- `FOLLOW_PATH` Follow Path.
- `PIVOT` Pivot.
- `RIGID_BODY_JOINT` Rigid Body Joint.
- `SCRIPT` Script.
- `SHRINKWRAP` Shrinkwrap.

**Returns** New constraint

**Return type** `Constraint`

**remove** (*constraint*)

Remove a constraint from this object

**Parameters** `constraint` (`Constraint`, (never `None`)) – Removed constraint

### Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `PoseBone.constraints`

### 2.4.428 Property(`bpy_struct`)

base class — `bpy_struct`

subclasses — `PointerProperty`, `EnumProperty`, `FloatProperty`, `IntProperty`, `CollectionProperty`, `BooleanProperty`, `StringProperty`

**class** `bpy.types.Property` (*bpy\_struct*)  
RNA property definition

#### **description**

Description of the property for tooltips

**Type** string, default "", (readonly)

#### **identifier**

Unique name used in the code and scripting

**Type** string, default "", (readonly)

#### **is\_animatable**

Property is animatable through RNA

**Type** boolean, default False, (readonly)

#### **is\_enum\_flag**

True when multiple enums

**Type** boolean, default False, (readonly)

#### **is\_hidden**

True when the property is hidden

**Type** boolean, default False, (readonly)

**is\_never\_none**

True when this value can't be set to None

**Type** boolean, default False, (readonly)

**is\_output**

True when this property is an output value from an RNA function

**Type** boolean, default False, (readonly)

**is\_readonly**

Property is editable through RNA

**Type** boolean, default False, (readonly)

**is\_registered**

Property is registered as part of type registration

**Type** boolean, default False, (readonly)

**is\_registered\_optional**

Property is optionally registered as part of type registration

**Type** boolean, default False, (readonly)

**is\_required**

False when this property is an optional argument in an RNA function

**Type** boolean, default False, (readonly)

**is\_runtime**

Property has been dynamically created at runtime

**Type** boolean, default False, (readonly)

**is\_skip\_save**

True when the property is not saved in presets

**Type** boolean, default False, (readonly)

**name**

Human readable name

**Type** string, default "", (readonly)

**srna**

Struct definition used for properties assigned to this item

**Type** *Struct*, (readonly)

**subtype**

Semantic interpretation of the property

**Type** enum in ['NONE', 'FILE\_PATH', 'DIRECTORY\_PATH', 'UNSIGNED', 'PERCENTAGE', 'FACTOR', 'ANGLE', 'TIME', 'DISTANCE', 'COLOR', 'TRANSLATION', 'DIRECTION', 'MATRIX', 'EULER', 'QUATERNION', 'XYZ', 'COLOR\_GAMMA', 'COORDINATES', 'LAYER', 'LAYER\_MEMBERSHIP'], default 'NONE', (readonly)

**type**

Data type of the property

**Type** enum in ['BOOLEAN', 'INT', 'FLOAT', 'STRING', 'ENUM', 'POINTER', 'COLLECTION'], default 'BOOLEAN', (readonly)

**unit**

Type of units for this property

**Type** enum in ['NONE', 'LENGTH', 'AREA', 'VOLUME', 'ROTATION', 'TIME', 'VELOCITY', 'ACCELERATION'], default 'NONE', (readonly)

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Function.parameters`
- `Struct.properties`

## 2.4.429 PropertyActuator(Actuator)

base classes — `bpy_struct`, `Actuator`

**class** `bpy.types.PropertyActuator` (*Actuator*)  
Actuator to handle properties

#### mode

- **ASSIGN** Assign.
- **ADD** Add.
- **COPY** Copy.
- **TOGGLE** Toggle, For bool/int/float/timer properties only.

**Type** enum in ['ASSIGN', 'ADD', 'COPY', 'TOGGLE'], default 'ASSIGN'

#### object

Copy from this Object

**Type** `Object`

**object\_property**

Copy this property

**Type** string, default ""

**property**

The name of the property

**Type** string, default ""

**value**

The name of the property or the value to use (use "" around strings)

**Type** string, default ""

**Inherited Properties**

- `bpy_struct.id_data`
- `Actuator.name`
- `Actuator.show_expanded`
- `Actuator.pin`
- `Actuator.type`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Actuator.link`
- `Actuator.unlink`

**2.4.430 PropertyGroup(bpy\_struct)**

**Custom Properties**

PropertyGroups are the base class for dynamically defined sets of properties.

They can be used to extend existing blender data with your own types which can be animated, accessed from the user interface and from python.

**Note:** The values assigned to blender data are saved to disk but the class definitions are not, this means whenever you load blender the class needs to be registered too.

This is best done by creating an addon which loads on startup and registers your properties.

**Note:** PropertyGroups must be registered before assigning them to blender data.

### See Also:

Property types used in class declarations are all in `bpy.props`

```
import bpy
```

```
class MyPropertyGroup(bpy.types.PropertyGroup):
    custom_1 = bpy.props.FloatProperty(name="My Float")
    custom_2 = bpy.props.IntProperty(name="My Int")

bpy.utils.register_class(MyPropertyGroup)

bpy.types.Object.my_prop_grp = bpy.props.PointerProperty(type=MyPropertyGroup)

# test this worked
bpy.data.objects[0].my_prop_grp.custom_1 = 22.0
```

base class — `bpy_struct`

subclasses — `OperatorFileListElement`, `OperatorMousePath`, `OperatorStrokeElement`

**class** `bpy.types.PropertyGroup` (*bpy\_struct*)

Group of ID properties

**name**

Unique name used in the code and scripting

**Type** string, default ""

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`

- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `PropertyGroupItem.collection`
- `PropertyGroupItem.group`
- `PropertyGroupItem.idp_array`

### 2.4.431 PropertyGroupItem(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.PropertyGroupItem` (*bpy\_struct*)

Property that stores arbitrary, user defined properties

#### **collection**

**Type** `bpy_prop_collection` of `PropertyGroup`, (readonly)

#### **double**

**Type** float in [-inf, inf], default 0.0

#### **double\_array**

**Type** float array of 1 items in [-inf, inf], default (0.0)

#### **float**

**Type** float in [-inf, inf], default 0.0

#### **float\_array**

**Type** float array of 1 items in [-inf, inf], default (0.0)

#### **group**

**Type** `PropertyGroup`, (readonly)

#### **idp\_array**

**Type** `bpy_prop_collection` of `PropertyGroup`, (readonly)

#### **int**

**Type** int in [-inf, inf], default 0

#### **int\_array**

**Type** int array of 1 items in [-inf, inf], default (0,)

#### **string**

**Type** string, default ""

## Inherited Properties

- `bpy_struct.id_data`



## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.432 PropertySensor(Sensor)

base classes — `bpy_struct`, `Sensor`

**class** `bpy.types.PropertySensor` (*Sensor*)

Sensor to detect values and changes in values of properties

### **evaluation\_type**

Type of property evaluation

**Type** enum in ['PROPEQUAL', 'PROPNEQUAL', 'PROPINTERVAL', 'PROPCHANGED'],  
default 'PROPEQUAL'

### **property**

**Type** string, default ""

### **value**

Check for this value in types in Equal or Not Equal types

**Type** string, default ""

### **value\_max**

Specify maximum value in Interval type

**Type** string, default ""

### **value\_min**

Specify minimum value in Interval type

**Type** string, default ""

## Inherited Properties

- `bpy_struct.id_data`
- `Sensor.name`
- `Sensor.show_expanded`
- `Sensor.frequency`

- `Sensor.invert`
- `Sensor.use_level`
- `Sensor.pin`
- `Sensor.use_pulse_false_level`
- `Sensor.use_pulse_true_level`
- `Sensor.use_tap`
- `Sensor.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Sensor.link`
- `Sensor.unlink`

## 2.4.433 PythonConstraint(Constraint)

base classes — `bpy_struct`, `Constraint`

**class** `bpy.types.PythonConstraint` (*Constraint*)  
Uses Python script for constraint evaluation

**has\_script\_error**

The linked Python script has thrown an error

**Type** boolean, default False, (readonly)

**target\_count**

Usually only 1-3 are needed

**Type** int in [-inf, inf], default 0

**targets**

Target Objects

**Type** `bpy_prop_collection` of `ConstraintTarget`, (readonly)

**text**

The text object that contains the Python script

**Type** `Text`

**use\_targets**

Use the targets indicated in the constraint panel

**Type** boolean, default False

**Inherited Properties**

- `bpy_struct.id_data`
- `Constraint.name`
- `Constraint.active`
- `Constraint.mute`
- `Constraint.show_expanded`
- `Constraint.influence`
- `Constraint.error_location`
- `Constraint.owner_space`
- `Constraint.is_proxy_local`
- `Constraint.error_rotation`
- `Constraint.target_space`
- `Constraint.type`
- `Constraint.is_valid`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**2.4.434 PythonController(Controller)**

base classes — `bpy_struct`, `Controller`

**class** `bpy.types.PythonController` (*Controller*)

Controller executing a python script

**mode**

Python script type (textblock or module - faster)

**Type** enum in ['SCRIPT', 'MODULE'], default 'SCRIPT'

**module**

Module name and function to run e.g. “someModule.main”. Internal texts and external python files can be used

**Type** string, default “”

**text**

Text datablock with the python script

**Type** Text

**use\_debug**

Continuously reload the module from disk for editing external modules without restarting

**Type** boolean, default False

**Inherited Properties**

- `bpy_struct.id_data`
- `Controller.name`
- `Controller.states`
- `Controller.show_expanded`
- `Controller.use_priority`
- `Controller.type`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Controller.link`
- `Controller.unlink`

**2.4.435 RadarSensor(Sensor)**

base classes — `bpy_struct`, `Sensor`

**class** `bpy.types.RadarSensor` (*Sensor*)

Sensor to detect objects in a cone shaped radar emanating from the current object

**angle**

Opening angle of the radar cone (in degrees)

**Type** float in [0, 179.9], default 0.0

**axis**

Specify along which axis the radar cone is cast

**Type** enum in ['XAXIS', 'YAXIS', 'ZAXIS', 'NEGXAXIS', 'NEGYAXIS', 'NEGZAXIS'], default 'XAXIS'

**distance**

Depth of the radar cone

**Type** float in [0, 10000], default 0.0

**property**

Only look for Objects with this property (blank = all objects)

**Type** string, default ""

### Inherited Properties

- `bpy_struct.id_data`
- `Sensor.name`
- `Sensor.show_expanded`
- `Sensor.frequency`
- `Sensor.invert`
- `Sensor.use_level`
- `Sensor.pin`
- `Sensor.use_pulse_false_level`
- `Sensor.use_pulse_true_level`
- `Sensor.use_tap`
- `Sensor.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Sensor.link`
- `Sensor.unlink`

## 2.4.436 RandomActuator(Actuator)

base classes — `bpy_struct`, `Actuator`

class `bpy.types.RandomActuator` (*Actuator*)

### **chance**

Pick a number between 0 and 1. Success if you stay below this value

**Type** float in [0, 1], default 0.0

### **distribution**

Choose the type of distribution

**Type** enum in ['BOOL\_CONSTANT', 'BOOL\_UNIFORM', 'BOOL\_BERNOULLI', 'INT\_CONSTANT', 'INT\_UNIFORM', 'INT\_POISSON', 'FLOAT\_CONSTANT', 'FLOAT\_UNIFORM', 'FLOAT\_NORMAL', 'FLOAT\_NEGATIVE\_EXPONENTIAL'], default 'BOOL\_CONSTANT'

### **float\_max**

Choose a number from a range. Upper boundary of the range

**Type** float in [-1000, 1000], default 0.0

### **float\_mean**

A normal distribution. Mean of the distribution

**Type** float in [-1000, 1000], default 0.0

### **float\_min**

Choose a number from a range. Lower boundary of the range

**Type** float in [-1000, 1000], default 0.0

### **float\_value**

Always return this number

**Type** float in [0, 1], default 0.0

### **half\_life\_time**

Negative exponential dropoff

**Type** float in [-1000, 1000], default 0.0

### **int\_max**

Choose a number from a range. Upper boundary of the range

**Type** int in [-1000, 1000], default 0

### **int\_mean**

Expected mean value of the distribution

**Type** float in [0.01, 100], default 0.0

### **int\_min**

Choose a number from a range. Lower boundary of the range

**Type** int in [-1000, 1000], default 0

### **int\_value**

Always return this number

**Type** int in [-inf, inf], default 0

**property**

Assign the random value to this property

**Type** string, default ""

**seed**

Initial seed of the random generator. Use Python for more freedom (choose 0 for not random)

**Type** int in [0, 300000], default 0

**standard\_deviation**

A normal distribution. Standard deviation of the distribution

**Type** float in [-1000, 1000], default 0.0

**use\_always\_true**

Always false or always true

**Type** boolean, default False

**Inherited Properties**

- `bpy_struct.id_data`
- `Actuator.name`
- `Actuator.show_expanded`
- `Actuator.pin`
- `Actuator.type`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Actuator.link`
- `Actuator.unlink`

**2.4.437 RandomSensor(Sensor)**

base classes — `bpy_struct`, `Sensor`

**class** `bpy.types.RandomSensor` (*Sensor*)

Sensor to send random events

**seed**

Initial seed of the generator. (Choose 0 for not random)

**Type** int in [0, 1000], default 0

**Inherited Properties**

- `bpy_struct.id_data`
- `Sensor.name`
- `Sensor.show_expanded`
- `Sensor.frequency`
- `Sensor.invert`
- `Sensor.use_level`
- `Sensor.pin`
- `Sensor.use_pulse_false_level`
- `Sensor.use_pulse_true_level`
- `Sensor.use_tap`
- `Sensor.type`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Sensor.link`
- `Sensor.unlink`

### 2.4.438 RaySensor(Sensor)

base classes — `bpy_struct`, `Sensor`

**class** `bpy.types.RaySensor` (*Sensor*)

Sensor to detect intersections with a ray emanating from the current object

**axis**

Specify along which axis the ray is cast

**Type** enum in ['XAXIS', 'YAXIS', 'ZAXIS', 'NEGXAXIS', 'NEGYAXIS', 'NEGZAXIS'],  
default 'YAXIS'



**material**

Only look for Objects with this material (blank = all objects)

**Type** string, default ""

**property**

Only look for Objects with this property (blank = all objects)

**Type** string, default ""

**range**

Sense objects no farther than this distance

**Type** float in [0.01, 10000], default 0.0

**ray\_type**

Toggle collision on material or property

- PROPERTY Property, Use a material for ray intersections.
- MATERIAL Material, Use a property for ray intersections.

**Type** enum in ['PROPERTY', 'MATERIAL'], default 'PROPERTY'

**use\_x\_ray**

Toggle X-Ray option (see through objects that don't have the property)

**Type** boolean, default False

**Inherited Properties**

- bpy\_struct.id\_data
- Sensor.name
- Sensor.show\_expanded
- Sensor.frequency
- Sensor.invert
- Sensor.use\_level
- Sensor.pin
- Sensor.use\_pulse\_false\_level
- Sensor.use\_pulse\_true\_level
- Sensor.use\_tap
- Sensor.type

**Inherited Functions**

- bpy\_struct.as\_pointer
- bpy\_struct.callback\_add
- bpy\_struct.callback\_remove
- bpy\_struct.driver\_add
- bpy\_struct.driver\_remove
- bpy\_struct.get
- bpy\_struct.is\_property\_hidden
- bpy\_struct.is\_property\_set
- bpy\_struct.items
- bpy\_struct.keyframe\_delete
- bpy\_struct.keyframe\_insert
- bpy\_struct.keys

- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Sensor.link`
- `Sensor.unlink`

### 2.4.439 Region(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.Region` (*bpy\_struct*)  
Region in a subdivided screen area

**height**

Region height

**Type** int in [0, 32767], default 0, (readonly)

**id**

Unique ID for this region

**Type** int in [-32768, 32767], default 0, (readonly)

**type**

Type of this region

**Type** enum in ['WINDOW', 'HEADER', 'CHANNELS', 'TEMPORARY', 'UI', 'TOOLS', 'TOOL\_PROPS', 'PREVIEW'], default 'WINDOW', (readonly)

**width**

Region width

**Type** int in [0, 32767], default 0, (readonly)

**tag\_redraw()**

tag\_redraw

#### Inherited Properties

- `bpy_struct.id_data`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`

- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Area.regions`
- `Context.region`

### 2.4.440 RegionView3D(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.RegionView3D` (*bpy\_struct*)

3D View region data

**is\_perspective**

**Type** boolean, default False

**lock\_rotation**

Lock view rotation in side views

**Type** boolean, default False

**perspective\_matrix**

Current perspective matrix of the 3D region

**Type** float array of 16 items in  $[-inf, inf]$ , default (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0), (readonly)

**show\_sync\_view**

Sync view position between side views

**Type** boolean, default False

**use\_box\_clip**

Clip objects based on what's visible in other side views

**Type** boolean, default False

**view\_distance**

Distance to the view location

**Type** float in  $[0, inf]$ , default 0.0

**view\_location**

View pivot location

**Type** float array of 3 items in  $[-inf, inf]$ , default (0.0, 0.0, 0.0)

**view\_matrix**

Current view matrix of the 3D region

**Type** float array of 16 items in  $[-inf, inf]$ , default (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)

**view\_perspective**

View Perspective

**Type** enum in ['PERSP', 'ORTHO', 'CAMERA'], default 'ORTHO'

**view\_rotation**

Rotation in quaternions (keep normalized)

**Type** float array of 4 items in [-inf, inf], default (0.0, 0.0, 0.0, 0.0)

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `Context.region_data`
- `SpaceView3D.region_3d`
- `SpaceView3D.region_quadview`

## 2.4.441 RenderEngine(bpy\_struct)

### Simple Render Engine

```
import bpy
```

```
class CustomRenderEngine(bpy.types.RenderEngine):  
    # These three members are used by blender to set up the  
    # RenderEngine; define its internal name, visible name and capabilities.  
    bl_idname = 'custom_renderer'  
    bl_label = 'Flat Color Renderer'  
    bl_use_preview = True  
  
    # This is the only method called by blender, in this example  
    # we use it to detect preview rendering and call the implementation  
    # in another method.
```

```

def render(self, scene):
    scale = scene.render.resolution_percentage / 100.0
    self.size_x = int(scene.render.resolution_x * scale)
    self.size_y = int(scene.render.resolution_y * scale)

    if scene.name == 'preview':
        self.render_preview(scene)
    else:
        self.render_scene(scene)

# In this example, we fill the preview renders with a flat green color.
def render_preview(self, scene):
    pixel_count = self.size_x * self.size_y

    # The framebuffer is defined as a list of pixels, each pixel
    # itself being a list of R,G,B,A values
    green_rect = [[0.0, 1.0, 0.0, 1.0]] * pixel_count

    # Here we write the pixel values to the RenderResult
    result = self.begin_result(0, 0, self.size_x, self.size_y)
    layer = result.layers[0]
    layer.rect = green_rect
    self.end_result(result)

# In this example, we fill the full renders with a flat blue color.
def render_scene(self, scene):
    pixel_count = self.size_x * self.size_y

    # The framebuffer is defined as a list of pixels, each pixel
    # itself being a list of R,G,B,A values
    blue_rect = [[0.0, 0.0, 1.0, 1.0]] * pixel_count

    # Here we write the pixel values to the RenderResult
    result = self.begin_result(0, 0, self.size_x, self.size_y)
    layer = result.layers[0]
    layer.rect = blue_rect
    self.end_result(result)

# Register the RenderEngine
bpy.utils.register_class(CustomRenderEngine)

# RenderEngines also need to tell UI Panels that they are compatible
# Otherwise most of the UI will be empty when the engine is selected.
# In this example, we need to see the main render image button and
# the material preview panel.
from bl_ui import properties_render
properties_render.RENDER_PT_render.COMPAT_ENGINES.add('custom_rendrerer')
del properties_render

from bl_ui import properties_material
properties_material.MATERIAL_PT_preview.COMPAT_ENGINES.add('custom_rendrerer')
del properties_material

base class — bpy_struct

class bpy.types.RenderEngine (bpy_struct)
    Render engine

    bl_idname

```

**Type** string, default ""

**bl\_label**

**Type** string, default ""

**bl\_use\_postprocess**

**Type** boolean, default False

**bl\_use\_preview**

**Type** boolean, default False

**bl\_use\_shading\_nodes**

**Type** boolean, default False

**is\_animation**

**Type** boolean, default False

**is\_preview**

**Type** boolean, default False

**update** (*data=None, scene=None*)

Export scene data for render

**render** (*scene=None*)

Render scene into an image

**view\_update** (*context=None*)

Update on data changes for viewport render

**view\_draw** (*context=None*)

Request update call for viewport rendering

**tag\_redraw** ()

tag\_redraw

**tag\_update** ()

tag\_update

**begin\_result** (*x, y, w, h*)

begin\_result

**Parameters**

- **x** (*int in [0, inf]*) – X
- **y** (*int in [0, inf]*) – Y
- **w** (*int in [0, inf]*) – Width
- **h** (*int in [0, inf]*) – Height

**Returns** Result

**Return type** `RenderResult`

**update\_result** (*result*)

update\_result

**Parameters** **result** (`RenderResult`) – Result

**end\_result** (*result*)

end\_result

**Parameters** **result** (`RenderResult`) – Result

**test\_break** ()

test\_break

**Returns** Break

**Return type** boolean

**update\_stats** (*stats, info*)

update\_stats

**Parameters**

- **stats** (*string*) – Stats
- **info** (*string*) – Info

**update\_progress** (*progress*)

update\_progress

**Parameters** **progress** (*float in [0, 1]*) – Percentage of render that's done

**report** (*type, message*)

report

**Parameters**

- **type** (*enum set in {'DEBUG', 'INFO', 'OPERATOR', 'WARNING', 'ERROR', 'ERROR\_INVALID\_INPUT', 'ERROR\_INVALID\_CONTEXT', 'ERROR\_OUT\_OF\_MEMORY'}*) – Type
- **message** (*string*) – Report Message

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`

- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.442 RenderLayer(`bpy_struct`)

base class — `bpy_struct`

`class bpy.types.RenderLayer` (*bpy\_struct*)

**exclude\_ambient\_occlusion**

Exclude AO pass from combined

**Type** boolean, default False, (readonly)

**exclude\_emit**

Exclude emission pass from combined

**Type** boolean, default False, (readonly)

**exclude\_environment**

Exclude environment pass from combined

**Type** boolean, default False, (readonly)

**exclude\_indirect**

Exclude indirect pass from combined

**Type** boolean, default False, (readonly)

**exclude\_reflection**

Exclude raytraced reflection pass from combined

**Type** boolean, default False, (readonly)

**exclude\_refraction**

Exclude raytraced refraction pass from combined

**Type** boolean, default False, (readonly)

**exclude\_shadow**

Exclude shadow pass from combined

**Type** boolean, default False, (readonly)

**exclude\_specular**

Exclude specular pass from combined

**Type** boolean, default False, (readonly)

**invert\_zmask**

For Zmask, only render what is behind solid z values instead of in front

**Type** boolean, default False, (readonly)

**layers**

Scene layers included in this render layer

**Type** boolean array of 20 items, default (False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False), (readonly)



**layers\_zmask**

Zmask scene layers for solid faces

**Type** boolean array of 20 items, default (False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False), (readonly)

**light\_override**

Group to override all other lights in this render layer

**Type** `Group`, (readonly)

**material\_override**

Material to override all other materials in this render layer

**Type** `Material`, (readonly)

**name**

Render layer name

**Type** string, default "", (readonly)

**passes**

**Type** `bpy_prop_collection` of `RenderPass`, (readonly)

**rect**

**Type** float in [-inf, inf], default 0.0

**use**

Disable or enable the render layer

**Type** boolean, default False, (readonly)

**use\_all\_z**

Fill in Z values for solid faces in invisible layers, for masking

**Type** boolean, default False, (readonly)

**use\_edge\_enhance**

Render Edge-enhance in this Layer (only works for Solid faces)

**Type** boolean, default False, (readonly)

**use\_halo**

Render Halos in this Layer (on top of Solid)

**Type** boolean, default False, (readonly)

**use\_pass\_ambient\_occlusion**

Deliver AO pass

**Type** boolean, default False, (readonly)

**use\_pass\_color**

Deliver shade-less color pass

**Type** boolean, default False, (readonly)

**use\_pass\_combined**

Deliver full combined RGBA buffer

**Type** boolean, default False, (readonly)

**use\_pass\_diffuse**

Deliver diffuse pass

**Type** boolean, default False, (readonly)

**use\_pass\_emit**

Deliver emission pass

**Type** boolean, default False, (readonly)

**use\_pass\_environment**

Deliver environment lighting pass

**Type** boolean, default False, (readonly)

**use\_pass\_indirect**

Deliver indirect lighting pass

**Type** boolean, default False, (readonly)

**use\_pass\_material\_index**

Deliver material index pass

**Type** boolean, default False, (readonly)

**use\_pass\_mist**

Deliver mist factor pass (0.0-1.0)

**Type** boolean, default False, (readonly)

**use\_pass\_normal**

Deliver normal pass

**Type** boolean, default False, (readonly)

**use\_pass\_object\_index**

Deliver object index pass

**Type** boolean, default False, (readonly)

**use\_pass\_reflection**

Deliver raytraced reflection pass

**Type** boolean, default False, (readonly)

**use\_pass\_refraction**

Deliver raytraced refraction pass

**Type** boolean, default False, (readonly)

**use\_pass\_shadow**

Deliver shadow pass

**Type** boolean, default False, (readonly)

**use\_pass\_specular**

Deliver specular pass

**Type** boolean, default False, (readonly)

**use\_pass\_uv**

Deliver texture UV pass

**Type** boolean, default False, (readonly)

**use\_pass\_vector**

Deliver speed vector pass

**Type** boolean, default False, (readonly)

**use\_pass\_z**

Deliver Z values pass

**Type** boolean, default False, (readonly)

**use\_sky**

Render Sky in this Layer

**Type** boolean, default False, (readonly)

**use\_solid**

Render Solid faces in this Layer

**Type** boolean, default False, (readonly)

**use\_strand**

Render Strands in this Layer

**Type** boolean, default False, (readonly)

**use\_zmask**

Only render what's in front of the solid z values

**Type** boolean, default False, (readonly)

**use\_ztransp**

Render Z-Transparent faces in this Layer (on top of Solid and Halos)

**Type** boolean, default False, (readonly)

**load\_from\_file** (*filename*, *x=0*, *y=0*)

Copies the pixels of this renderlayer from an image file

**Parameters**

- **filename** (*string*) – Filename, Filename to load into this render tile, must be no smaller than the renderlayer
- **x** (*int in [0, inf], (optional)*) – Offset X, Offset the position to copy from if the image is larger than the render layer
- **y** (*int in [0, inf], (optional)*) – Offset Y, Offset the position to copy from if the image is larger than the render layer

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`

- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `RenderResult.layers`

### 2.4.443 RenderLayers(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.RenderLayers` (*bpy\_struct*)

Collection of render layers

**active**

Active Render Layer

**Type** `SceneRenderLayer`, (never None)

**active\_index**

Active index in render layer array

**Type** `int` in `[-32768, 32767]`, default 0

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `RenderSettings.layers`

### 2.4.444 `RenderPass(bpy_struct)`

base class — `bpy_struct`

class `bpy.types.RenderPass` (*bpy\_struct*)

#### `channel_id`

**Type** string, default "", (readonly)

#### `channels`

**Type** int in [-inf, inf], default 0, (readonly)

#### `name`

**Type** string, default "", (readonly)

#### `rect`

**Type** float in [-inf, inf], default 0.0

#### `type`

**Type** enum in ['COMBINED', 'Z', 'COLOR', 'DIFFUSE', 'SPECULAR', 'SHADOW', 'AO', 'REFLECTION', 'NORMAL', 'VECTOR', 'REFRACTION', 'OBJECT\_INDEX', 'UV', 'MIST', 'EMIT', 'ENVIRONMENT', 'MATERIAL\_INDEX'], default 'COMBINED', (readonly)

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `RenderLayer.passes`

## 2.4.445 `RenderResult(bpy_struct)`

base class — `bpy_struct`

**class** `bpy.types.RenderResult` (*bpy\_struct*)

Result of rendering, including all layers and passes

### **layers**

**Type** `bpy_prop_collection` of `RenderLayer`, (readonly)

### **resolution\_x**

**Type** `int` in `[-inf, inf]`, default 0, (readonly)

### **resolution\_y**

**Type** `int` in `[-inf, inf]`, default 0, (readonly)

### **load\_from\_file** (*filename*)

Copies the pixels of this render result from an image file

**Parameters** **filename** (*string*) – File Name, Filename to load into this render tile, must be no smaller than the render result

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `RenderEngine.begin_result`
- `RenderEngine.end_result`
- `RenderEngine.update_result`

2.4.446 `RenderSettings(bpy_struct)`

base class — `bpy_struct`

class `bpy.types.RenderSettings` (*bpy\_struct*)

Rendering settings for a Scene datablock

**alpha\_mode**

Representation of alpha information in the RGBA pixels

- `SKY` Sky, Transparent pixels are filled with sky color.
- `PREMUL` Premultiplied, Transparent RGB pixels are multiplied by the alpha channel.
- `STRAIGHT` Straight Alpha, Transparent RGB and alpha pixels are unmodified.

**Type** enum in [`'SKY'`, `'PREMUL'`, `'STRAIGHT'`], default `'SKY'`

**antialiasing\_samples**

Amount of anti-aliasing samples per pixel

**Type** enum in [`'5'`, `'8'`, `'11'`, `'16'`], default `'5'`

**bake\_aa\_mode**

**Type** enum in [`'5'`, `'8'`, `'11'`, `'16'`], default `'5'`

**bake\_bias**

Bias towards faces further away from the object (in blender units)

**Type** float in [0, 1000], default 0.0

**bake\_distance**

Maximum distance from active object to other object (in blender units)

**Type** float in [0, 1000], default 0.0

**bake\_margin**

Amount of pixels to extend the baked result with, as post process filter

**Type** int in [0, 64], default 0

**bake\_normal\_space**

Choose normal space for baking

- `CAMERA` Camera, Bake the normals in camera space.
- `WORLD` World, Bake the normals in world space.
- `OBJECT` Object, Bake the normals in object space.
- `TANGENT` Tangent, Bake the normals in tangent space.

**Type** enum in [`'CAMERA'`, `'WORLD'`, `'OBJECT'`, `'TANGENT'`], default `'CAMERA'`

**bake\_quad\_split**

Choose the method used to split a quad into 2 triangles for baking

- AUTO Automatic, Split quads to give the least distortion while baking.
- FIXED Fixed, Split quads predictably (0,1,2) (0,2,3).
- FIXED\_ALT Fixed Alternate, Split quads predictably (1,2,3) (1,3,0).

**Type** enum in ['AUTO', 'FIXED', 'FIXED\_ALT'], default 'AUTO'

**bake\_type**

Choose shading information to bake into the image

- FULL Full Render, Bake everything.
- AO Ambient Occlusion, Bake ambient occlusion.
- SHADOW Shadow, Bake shadows.
- NORMALS Normals, Bake normals.
- TEXTURE Textures, Bake textures.
- DISPLACEMENT Displacement, Bake displacement.
- EMIT Emission, Bake Emit values (glow).
- ALPHA Alpha, Bake Alpha values (transparency).
- MIRROR\_INTENSITY Mirror Intensity, Bake Mirror values.
- MIRROR\_COLOR Mirror Colors, Bake Mirror colors.
- SPEC\_INTENSITY Specular Intensity, Bake Specular values.
- SPEC\_COLOR Specular Colors, Bake Specular colors.

**Type** enum in ['FULL', 'AO', 'SHADOW', 'NORMALS', 'TEXTURE', 'DISPLACEMENT', 'EMIT', 'ALPHA', 'MIRROR\_INTENSITY', 'MIRROR\_COLOR', 'SPEC\_INTENSITY', 'SPEC\_COLOR'], default 'FULL'

**border\_max\_x**

Maximum X value for the render border

**Type** float in [0, 1], default 0.0

**border\_max\_y**

Maximum Y value for the render border

**Type** float in [0, 1], default 0.0

**border\_min\_x**

Minimum X value to for the render border

**Type** float in [0, 1], default 0.0

**border\_min\_y**

Minimum Y value for the render border

**Type** float in [0, 1], default 0.0

**cineon\_black**

Log conversion reference blackpoint

**Type** int in [0, 1024], default 0



**cineon\_gamma**

Log conversion gamma

**Type** float in [0, 10], default 0.0

**cineon\_white**

Log conversion reference whitepoint

**Type** int in [0, 1024], default 0

**color\_mode**

Choose BW for saving greyscale images, RGB for saving red, green and blue channels, and RGBA for saving red, green, blue and alpha channels

- BW BW, Images get saved in 8 bits grayscale (only PNG, JPEG, TGA, TIF).
- RGB RGB, Images are saved with RGB (color) data.
- RGBA RGBA, Images are saved with RGB and Alpha data (if supported).

**Type** enum in ['BW', 'RGB', 'RGBA'], default 'BW'

**display\_mode**

Select where rendered images will be displayed

- SCREEN Full Screen, Images are rendered in full Screen.
- AREA Image Editor, Images are rendered in Image Editor.
- WINDOW New Window, Images are rendered in new Window.
- NONE Keep UI, Images are rendered without forcing UI changes, optionally showing result.

**Type** enum in ['SCREEN', 'AREA', 'WINDOW', 'NONE'], default 'SCREEN'

**dither\_intensity**

Amount of dithering noise added to the rendered image to break up banding

**Type** float in [0, 2], default 0.0

**edge\_color**

Edge color

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**edge\_threshold**

Threshold for drawing outlines on geometry edges

**Type** int in [0, 255], default 0

**engine**

Engine to use for rendering

- BLENDER\_RENDER Blender Render, Use the Blender internal rendering engine for rendering.

**Type** enum in ['BLENDER\_RENDER'], default 'BLENDER\_RENDER'

**exr\_codec**

Codec settings for OpenEXR

**Type** enum in ['NONE', 'PXR24', 'ZIP', 'PIZ', 'RLE'], default 'NONE'

**exr\_preview**

When rendering animations, save JPG preview images in same directory

**Type** boolean, default False

**exr\_zbuf**

Save the z-depth per pixel (32 bit unsigned int z-buffer)

**Type** boolean, default False

**ffmpeg\_audio\_channels**

Audio channel count

- MONO Mono, Set audio channels to mono.
- STEREO Stereo, Set audio channels to stereo.
- SURROUND4 4 Channels, Set audio channels to 4 channels.
- SURROUND51 5.1 Surround, Set audio channels to 5.1 surround sound.
- SURROUND71 7.1 Surround, Set audio channels to 7.1 surround sound.

**Type** enum in ['MONO', 'STEREO', 'SURROUND4', 'SURROUND51', 'SURROUND71'], default 'MONO'

**ffmpeg\_audio\_mixrate**

Audio samplerate(samples/s)

**Type** int in [8000, 192000], default 0

**field\_order**

Order of video fields (select which lines get rendered first, to create smooth motion for TV output)

- EVEN\_FIRST Upper First, Upper field first.
- ODD\_FIRST Lower First, Lower field first.

**Type** enum in ['EVEN\_FIRST', 'ODD\_FIRST'], default 'EVEN\_FIRST'

**file\_extension**

The file extension used for saving renders

**Type** string, default "", (readonly)

**file\_format**

File format to save the rendered images as

- BMP BMP, Output image in bitmap format.
- IRIS Iris, Output image in (old!) SGI IRIS format.
- PNG PNG, Output image in PNG format.
- JPEG JPEG, Output image in JPEG format.
- TARGA Targa, Output image in Targa format.
- TARGA\_RAW Targa Raw, Output image in uncompressed Targa format.
- MULTILAYER MultiLayer, Output image in multilayer OpenEXR format.
- OPEN\_EXR OpenEXR, Output image in OpenEXR format.
- TIFF TIFF, Output image in TIFF format.
- AVI\_JPEG AVI JPEG, Output video in AVI JPEG format.
- AVI\_RAW AVI Raw, Output video in AVI Raw format.

**Type** enum in ['BMP', 'IRIS', 'PNG', 'JPEG', 'TARGA', 'TARGA\_RAW', 'MULTILAYER', 'OPEN\_EXR', 'TIFF', 'AVI\_JPEG', 'AVI\_RAW'], default 'TARGA'

**file\_quality**

Quality of JPEG images, AVI Jpeg and SGI movies, compression for PNG's

**Type** int in [0, 100], default 0

**filepath**

Directory/name to save animations, # characters defines the position and length of frame numbers

**Type** string, default ""

**filter\_size**

Pixel width over which the reconstruction filter combines samples

**Type** float in [0.5, 1.5], default 0.0

**fps**

Framerate, expressed in frames per second

**Type** int in [1, 120], default 0

**fps\_base**

Framerate base

**Type** float in [0.1, 120], default 0.0

**frame\_map\_new**

How many frames the Map Old will last

**Type** int in [1, 900], default 0

**frame\_map\_old**

Old mapping value in frames

**Type** int in [1, 900], default 0

**has\_multiple\_engines**

More than one rendering engine is available

**Type** boolean, default False, (readonly)

**is\_movie\_format**

When true the format is a movie

**Type** boolean, default False, (readonly)

**layers**

**Type** [RenderLayers bpy\\_prop\\_collection](#) of [SceneRenderLayer](#), (readonly)

**motion\_blur\_samples**

Number of scene samples to take with motion blur

**Type** int in [1, 32], default 0

**motion\_blur\_shutter**

Time taken in frames between shutter open and close

**Type** float in [0.01, 10], default 0.0

**octree\_resolution**

Resolution of raytrace accelerator, use higher resolutions for larger scenes

**Type** enum in ['64', '128', '256', '512'], default '64'

**parts\_x**

Number of horizontal tiles to use while rendering

**Type** int in [1, 512], default 0

**parts\_y**

Number of vertical tiles to use while rendering

**Type** int in [1, 512], default 0

**pixel\_aspect\_x**

Horizontal aspect ratio - for anamorphic or non-square pixel output

**Type** float in [1, 200], default 0.0

**pixel\_aspect\_y**

Vertical aspect ratio - for anamorphic or non-square pixel output

**Type** float in [1, 200], default 0.0

**pixel\_filter\_type**

Reconstruction filter used for combining anti-aliasing samples

- **BOX** Box, Use a box filter for anti-aliasing.
- **TENT** Tent, Use a tent filter for anti-aliasing.
- **QUADRATIC** Quadratic, Use a quadratic filter for anti-aliasing.
- **CUBIC** Cubic, Use a cubic filter for anti-aliasing.
- **CATMULLROM** Catmull-Rom, Use a Catmull-Rom filter for anti-aliasing.
- **GAUSSIAN** Gaussian, Use a Gaussian filter for anti-aliasing.
- **MITCHELL** Mitchell-Netravali, Use a Mitchell-Netravali filter for anti-aliasing.

**Type** enum in ['BOX', 'TENT', 'QUADRATIC', 'CUBIC', 'CATMULLROM', 'GAUSSIAN', 'MITCHELL'], default 'BOX'

**raytrace\_method**

Type of raytrace accelerator structure

- **AUTO** Auto, Automatically select acceleration structure.
- **OCTREE** Octree, Use old Octree structure.
- **BLIBVH** BLI BVH, Use BLI K-Dop BVH.c.
- **VBVH** vBVH, Use vBVH.
- **SIMD\_SVBVH** SIMD SVBVH, Use SIMD SVBVH.
- **SIMD\_QBVH** SIMD QBVH, Use SIMD QBVH.

**Type** enum in ['AUTO', 'OCTREE', 'BLIBVH', 'VBVH', 'SIMD\_SVBVH', 'SIMD\_QBVH'], default 'AUTO'

**resolution\_percentage**

Percentage scale for render resolution

**Type** int in [1, 32767], default 0

**resolution\_x**

Number of horizontal pixels in the rendered image

**Type** int in [4, 10000], default 0

#### **resolution\_y**

Number of vertical pixels in the rendered image

**Type** int in [4, 10000], default 0

#### **sequencer\_gl\_preview**

Method to draw in the sequencer view

- **BOUNDBOX** Bounding Box, Display the object's local bounding boxes only.
- **WIREFRAME** Wireframe, Display the object as wire edges.
- **SOLID** Solid, Display the object solid, lit with default OpenGL lights.
- **TEXTURED** Textured, Display the object solid, with face-assigned textures.
- **RENDERED** Rendered, Display render preview.

**Type** enum in ['BOUNDBOX', 'WIREFRAME', 'SOLID', 'TEXTURED', 'RENDERED'], default 'BOUNDBOX'

#### **sequencer\_gl\_render**

Method to draw in the sequencer view

- **BOUNDBOX** Bounding Box, Display the object's local bounding boxes only.
- **WIREFRAME** Wireframe, Display the object as wire edges.
- **SOLID** Solid, Display the object solid, lit with default OpenGL lights.
- **TEXTURED** Textured, Display the object solid, with face-assigned textures.
- **RENDERED** Rendered, Display render preview.

**Type** enum in ['BOUNDBOX', 'WIREFRAME', 'SOLID', 'TEXTURED', 'RENDERED'], default 'BOUNDBOX'

#### **simplify\_ao\_sss**

Global approximate AO and SSS quality factor

**Type** float in [0, 1], default 0.0

#### **simplify\_child\_particles**

Global child particles percentage

**Type** float in [0, 1], default 0.0

#### **simplify\_shadow\_samples**

Global maximum shadow samples

**Type** int in [0, 32767], default 0

#### **simplify\_subdivision**

Global maximum subdivision level

**Type** int in [0, 32767], default 0

#### **stamp\_background**

Color to use behind stamp text

**Type** float array of 4 items in [0, 1], default (0.0, 0.0, 0.0, 0.0)

**stamp\_font\_size**

Size of the font used when rendering stamp text

**Type** int in [8, 64], default 0

**stamp\_foreground**

Color to use for stamp text

**Type** float array of 4 items in [0, 1], default (0.0, 0.0, 0.0, 0.0)

**stamp\_note\_text**

Custom text to appear in the stamp note

**Type** string, default ""

**threads**

Number of CPU threads to use simultaneously while rendering (for multi-core/CPU systems)

**Type** int in [1, 64], default 0

**threads\_mode**

Determine the amount of render threads used

- AUTO Auto-detect, Automatically determine the number of threads, based on CPUs.
- FIXED Fixed, Manually determine the number of threads.

**Type** enum in ['AUTO', 'FIXED'], default 'AUTO'

**use\_antialiasing**

Render and combine multiple samples per pixel to prevent jagged edges

**Type** boolean, default False

**use\_bake\_antialiasing**

Enables Anti-aliasing

**Type** boolean, default False

**use\_bake\_clear**

Clear Images before baking

**Type** boolean, default False

**use\_bake\_lores\_mesh**

Calculate heights against unsubdivided low resolution mesh

**Type** boolean, default False

**use\_bake\_multires**

Bake directly from multires object

**Type** boolean, default False

**use\_bake\_normalize**

With displacement normalize to the distance, with ambient occlusion normalize without using material settings

**Type** boolean, default False

**use\_bake\_selected\_to\_active**

Bake shading on the surface of selected objects to the active object

**Type** boolean, default False

**use\_border**

Render a user-defined border region, within the frame size (note that this disables `save_buffers` and `full_sample`)

**Type** boolean, default False

**use\_cineon\_log**

Convert to logarithmic color space

**Type** boolean, default False

**use\_color\_management**

Use linear workflow - gamma corrected imaging pipeline

**Type** boolean, default False

**use\_compositing**

Process the render result through the compositing pipeline, if compositing nodes are enabled

**Type** boolean, default False

**use\_crop\_to\_border**

Crop the rendered frame to the defined border size

**Type** boolean, default False

**use\_edge\_enhance**

Create a toon outline around the edges of geometry

**Type** boolean, default False

**use\_envmaps**

Calculate environment maps while rendering

**Type** boolean, default False

**use\_exr\_half**

Use 16 bit floats instead of 32 bit floats per channel

**Type** boolean, default False

**use\_fields**

Render image to two fields per frame, for interlaced TV output

**Type** boolean, default False

**use\_fields\_still**

Disable the time difference between fields

**Type** boolean, default False

**use\_file\_extension**

Add the file format extensions to the rendered file name (eg: filename + .jpg)

**Type** boolean, default False

**use\_free\_image\_textures**

Free all image texture from memory after render, to save memory before compositing

**Type** boolean, default False

**use\_free\_unused\_nodes**

Free Nodes that are not used while compositing, to save memory

**Type** boolean, default False

**use\_full\_sample**

Save for every anti-aliasing sample the entire RenderLayer results (this solves anti-aliasing issues with compositing)

**Type** boolean, default False

**use\_game\_engine**

Current rendering engine is a game engine

**Type** boolean, default False, (readonly)

**use\_instances**

Instance support leads to effective memory reduction when using duplicates

**Type** boolean, default False

**use\_local\_coords**

Vertex coordinates are stored locally on each primitive (increases memory usage, but may have impact on speed)

**Type** boolean, default False

**use\_motion\_blur**

Use multi-sampled 3D scene motion blur

**Type** boolean, default False

**use\_overwrite**

Overwrite existing files while rendering

**Type** boolean, default False

**use\_placeholder**

Create empty placeholder files while rendering frames (similar to Unix 'touch')

**Type** boolean, default False

**use\_radiosity**

Calculate radiosity in a pre-process before rendering

**Type** boolean, default False

**use\_raytrace**

Pre-calculate the raytrace accelerator and render raytracing effects

**Type** boolean, default False

**use\_save\_buffers**

Save tiles for all RenderLayers and SceneNodes to files in the temp directory (saves memory, required for Full Sample)

**Type** boolean, default False

**use\_sequencer**

Process the render (and composited) result through the video sequence editor pipeline, if sequencer strips exist

**Type** boolean, default False

**use\_sequencer\_gl\_preview**

**Type** boolean, default False

**use\_sequencer\_gl\_render**

**Type** boolean, default False



- use\_shading\_nodes**  
Active render engine uses new shading nodes system  
**Type** boolean, default False, (readonly)
- use\_shadows**  
Calculate shadows while rendering  
**Type** boolean, default False
- use\_simplify**  
Enable simplification of scene for quicker preview renders  
**Type** boolean, default False
- use\_simplify\_triangulate**  
Disables non-planer quads being triangulated  
**Type** boolean, default False
- use\_single\_layer**  
Only render the active layer  
**Type** boolean, default False
- use\_sss**  
Calculate sub-surface scattering in materials rendering  
**Type** boolean, default False
- use\_stamp**  
Render the stamp info text in the rendered image  
**Type** boolean, default False
- use\_stamp\_camera**  
Include the name of the active camera in image metadata  
**Type** boolean, default False
- use\_stamp\_date**  
Include the current date in image metadata  
**Type** boolean, default False
- use\_stamp\_filename**  
Include the .blend filename in image metadata  
**Type** boolean, default False
- use\_stamp\_frame**  
Include the frame number in image metadata  
**Type** boolean, default False
- use\_stamp\_lens**  
Include the active camera's lens in image metadata  
**Type** boolean, default False
- use\_stamp\_marker**  
Include the name of the last marker in image metadata  
**Type** boolean, default False
- use\_stamp\_note**  
Include a custom note in image metadata

**Type** boolean, default False

**use\_stamp\_render\_time**

Include the render time in image metadata

**Type** boolean, default False

**use\_stamp\_scene**

Include the name of the active scene in image metadata

**Type** boolean, default False

**use\_stamp\_sequencer\_strip**

Include the name of the foreground sequence strip in image metadata

**Type** boolean, default False

**use\_stamp\_time**

Include the rendered frame timecode as HH:MM:SS.FF in image metadata

**Type** boolean, default False

**use\_textures**

Use textures to affect material properties

**Type** boolean, default False

**use\_tiff\_16bit**

Save TIFF with 16 bits per channel

**Type** boolean, default False

**frame\_path** (*frame=-2147483648*)

Return the absolute path to the filename to be written for a given frame

**Parameters** **frame** (*int in [-inf, inf], (optional)*) – Frame number to use, if unset the current frame will be used

**Returns** File Path, The resulting filepath from the scenes render settings

**Return type** string

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`

- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Scene.render`

### 2.4.447 RigidBodyJointConstraint(Constraint)

base classes — `bpy_struct`, `Constraint`

**class** `bpy.types.RigidBodyJointConstraint` (*Constraint*)

For use with the Game Engine

#### **axis\_x**

Rotate pivot on X axis in degrees

**Type** float in [-6.28319, 6.28319], default 0.0

#### **axis\_y**

Rotate pivot on Y axis in degrees

**Type** float in [-6.28319, 6.28319], default 0.0

#### **axis\_z**

Rotate pivot on Z axis in degrees

**Type** float in [-6.28319, 6.28319], default 0.0

#### **child**

Child object

**Type** `Object`

#### **limit\_angle\_max\_x**

**Type** float in [-6.28319, 6.28319], default 0.0

#### **limit\_angle\_max\_y**

**Type** float in [-6.28319, 6.28319], default 0.0

#### **limit\_angle\_max\_z**

**Type** float in [-6.28319, 6.28319], default 0.0

#### **limit\_angle\_min\_x**

**Type** float in [-6.28319, 6.28319], default 0.0

#### **limit\_angle\_min\_y**

**Type** float in [-6.28319, 6.28319], default 0.0

#### **limit\_angle\_min\_z**

**Type** float in [-6.28319, 6.28319], default 0.0

#### **limit\_max\_x**

**Type** float in [-inf, inf], default 0.0

**limit\_max\_y**

**Type** float in [-inf, inf], default 0.0

**limit\_max\_z**

**Type** float in [-inf, inf], default 0.0

**limit\_min\_x**

**Type** float in [-inf, inf], default 0.0

**limit\_min\_y**

**Type** float in [-inf, inf], default 0.0

**limit\_min\_z**

**Type** float in [-inf, inf], default 0.0

**pivot\_type**

- **BALL** Ball, Allows rotations around all axes.
- **HINGE** Hinge, Works in one plane, allows rotations around one axis only.
- **CONE\_TWIST** Cone Twist, Allows rotations around all axes with limits for the cone and twist axes.
- **GENERIC\_6\_DOF** Generic 6 DoF, No constraints by default, limits can be set individually.

**Type** enum in ['BALL', 'HINGE', 'CONE\_TWIST', 'GENERIC\_6\_DOF'], default 'BALL'

**pivot\_x**

Offset pivot on X

**Type** float in [-1000, 1000], default 0.0

**pivot\_y**

Offset pivot on Y

**Type** float in [-1000, 1000], default 0.0

**pivot\_z**

Offset pivot on Z

**Type** float in [-1000, 1000], default 0.0

**show\_pivot**

Display the pivot point and rotation in 3D view

**Type** boolean, default False

**target**

Target Object

**Type** *Object*

**use\_angular\_limit\_x**

Use minimum/maximum x angular limit

**Type** boolean, default False

**use\_angular\_limit\_y**

Use minimum/maximum y angular limit

**Type** boolean, default False

- use\_angular\_limit\_z**  
Use minimum/maximum z angular limit  
**Type** boolean, default False
- use\_limit\_x**  
Use minimum/maximum x limit  
**Type** boolean, default False
- use\_limit\_y**  
Use minimum/maximum y limit  
**Type** boolean, default False
- use\_limit\_z**  
Use minimum/maximum z limit  
**Type** boolean, default False
- use\_linked\_collision**  
Disable collision between linked bodies  
**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `Constraint.name`
- `Constraint.active`
- `Constraint.mute`
- `Constraint.show_expanded`
- `Constraint.influence`
- `Constraint.error_location`
- `Constraint.owner_space`
- `Constraint.is_proxy_local`
- `Constraint.error_rotation`
- `Constraint.target_space`
- `Constraint.type`
- `Constraint.is_valid`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`

- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.448 SPHFluidSettings(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.SPHFluidSettings` (*bpy\_struct*)  
Settings for particle fluids physics

**buoyancy**

Artificial buoyancy force in negative gravity direction based on pressure differences inside the fluid

**Type** float in [0, 10], default 0.0

**factor\_density**

Density is calculated as a factor of default density (depends on particle size)

**Type** boolean, default False

**factor\_radius**

Interaction radius is a factor of 4 \* particle size

**Type** boolean, default False

**factor\_repulsion**

Repulsion is a factor of stiffness

**Type** boolean, default False

**factor\_rest\_length**

Spring rest length is a factor of 2 \* particle size

**Type** boolean, default False

**factor\_stiff\_viscosity**

Stiff viscosity is a factor of normal viscosity

**Type** boolean, default False

**fluid\_radius**

Fluid interaction radius

**Type** float in [0, 20], default 0.0

**linear\_viscosity**

Linear viscosity

**Type** float in [0, 100], default 0.0

**plasticity**

How much the spring rest length can change after the elastic limit is crossed

**Type** float in [0, 100], default 0.0

**repulsion**

How strongly the fluid tries to keep from clustering (factor of stiffness)

**Type** float in [0, 100], default 0.0

**rest\_density**

Fluid rest density

**Type** float in [0, 100], default 0.0

**rest\_length**

Spring rest length (factor of particle radius)

**Type** float in [0, 2], default 0.0

**spring\_force**

Spring force

**Type** float in [0, 100], default 0.0

**spring\_frames**

Create springs for this number of frames since particles birth (0 is always)

**Type** int in [0, 100], default 0

**stiff\_viscosity**

Creates viscosity for expanding fluid)

**Type** float in [0, 100], default 0.0

**stiffness**

How incompressible the fluid is

**Type** float in [0, 100], default 0.0

**use\_initial\_rest\_length**

Use the initial length as spring rest length instead of 2 \* particle size

**Type** boolean, default False

**use\_viscoelastic\_springs**

Use viscoelastic springs instead of Hooke's springs

**Type** boolean, default False

**yield\_ratio**

How much the spring has to be stretched/compressed in order to change it's rest length

**Type** float in [0, 1], default 0.0

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`

- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `ParticleSettings.fluid`

## 2.4.449 Scene(ID)

base classes — `bpy_struct`, `ID`

**class** `bpy.types.Scene` (*ID*)

Scene data block, consisting in objects and defining time and render related settings

### **animation\_data**

Animation data for this datablock

**Type** `AnimData`, (readonly)

### **audio\_distance\_model**

Distance model for distance attenuation calculation

- `NONE` None, No distance attenuation.
- `INVERSE` Inverse, Inverse distance model.
- `INVERSE_CLAMPED` Inverse Clamped, Inverse distance model with clamping.
- `LINEAR` Linear, Linear distance model.
- `LINEAR_CLAMPED` Linear Clamped, Linear distance model with clamping.
- `EXPONENT` Exponent, Exponent distance model.
- `EXPONENT_CLAMPED` Exponent Clamped, Exponent distance model with clamping.

**Type** `enum` in [`'NONE'`, `'INVERSE'`, `'INVERSE_CLAMPED'`, `'LINEAR'`, `'LINEAR_CLAMPED'`, `'EXPONENT'`, `'EXPONENT_CLAMPED'`], default `'NONE'`

### **audio\_doppler\_factor**

Pitch factor for Doppler effect calculation

**Type** `float` in `[0, inf]`, default `0.0`

### **audio\_doppler\_speed**

Speed of sound for Doppler effect calculation

**Type** `float` in `[0.01, inf]`, default `0.0`

### **audio\_volume**

Audio volume

**Type** `float` in `[0, 1]`, default `0.0`

### **background\_set**

Background set scene

**Type** `Scene`

### **camera**

Active camera, used for rendering the scene



**Type** `Object`

**cursor\_location**

3D cursor location

**Type** float array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`

**frame\_current**

Current Frame, to update animation data from python `frame_set()` instead

**Type** int in `[-300000, 300000]`, default 0

**frame\_end**

Final frame of the playback/rendering range

**Type** int in `[0, 300000]`, default 0

**frame\_preview\_end**

Alternative end frame for UI playback

**Type** int in `[-inf, inf]`, default 0

**frame\_preview\_start**

Alternative start frame for UI playback

**Type** int in `[-inf, inf]`, default 0

**frame\_start**

First frame of the playback/rendering range

**Type** int in `[0, 300000]`, default 0

**frame\_step**

Number of frames to skip forward while rendering/playing back each frame

**Type** int in `[0, 300000]`, default 0

**frame\_subframe**

**Type** float in `[-inf, inf]`, default 0.0, (readonly)

**game\_settings**

**Type** `SceneGameData`, (readonly, never None)

**gravity**

Constant acceleration in a given direction

**Type** float array of 3 items in `[-200, 200]`, default `(0.0, 0.0, 0.0)`

**grease\_pencil**

Grease Pencil datablock

**Type** `GreasePencil`

**is\_nla\_tweakmode**

Whether there is any action referenced by NLA being edited (strictly read-only)

**Type** boolean, default False, (readonly)

**keying\_sets**

Absolute Keying Sets for this Scene

**Type** `KeyingSets` `bpy_prop_collection` of `KeyingSet`, (readonly)

**keying\_sets\_all**

All Keying Sets available for use (Builtins and Absolute Keying Sets for this Scene)

**Type** `KeyingSetsAll bpy_prop_collection` of `KeyingSet`, (readonly)

### **layers**

Layers visible when rendering the scene

**Type** boolean array of 20 items, default (False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False)

### **node\_tree**

Compositing node tree

**Type** `NodeTree`, (readonly)

### **object\_bases**

**Type** `SceneBases bpy_prop_collection` of `ObjectBase`, (readonly)

### **objects**

**Type** `SceneObjects bpy_prop_collection` of `Object`, (readonly)

### **orientations**

**Type** `bpy_prop_collection` of `TransformOrientation`, (readonly)

### **render**

**Type** `RenderSettings`, (readonly, never None)

### **sequence\_editor**

**Type** `SequenceEditor`, (readonly)

### **sync\_mode**

How to sync playback

- NONE No Sync, Do not sync, play every frame.
- FRAME\_DROP Frame Dropping, Drop frames if playback is too slow.
- AUDIO\_SYNC AV-sync, Sync to audio playback, dropping frames.

**Type** enum in ['NONE', 'FRAME\_DROP', 'AUDIO\_SYNC'], default 'NONE'

### **timeline\_markers**

Markers used in all timelines for the current scene

**Type** `TimelineMarkers bpy_prop_collection` of `TimelineMarker`, (readonly)

### **tool\_settings**

**Type** `ToolSettings`, (readonly, never None)

### **unit\_settings**

Unit editing settings

**Type** `UnitSettings`, (readonly, never None)

### **use\_audio**

Play back of audio from Sequence Editor will be muted

**Type** boolean, default False

### **use\_audio\_scrub**

Play audio from Sequence Editor while scrubbing

**Type** boolean, default False

**use\_audio\_sync**

Play back and sync with audio clock, dropping frames if frame display is too slow

**Type** boolean, default False

**use\_frame\_drop**

Play back dropping frames if frame display is too slow

**Type** boolean, default False

**use\_gravity**

Use global gravity for all dynamics

**Type** boolean, default False

**use\_nodes**

Enable the compositing node tree

**Type** boolean, default False

**use\_preview\_range**

Use an alternative start/end frame for UI playback, rather than the scene start/end frame

**Type** boolean, default False

**use\_stamp\_note**

User defined note for the render stamping

**Type** string, default ""

**world**

World used for rendering the scene

**Type** `World`

**statistics ()**

statistics

**Returns** Statistics

**Return type** string

**frame\_set (frame, subframe=0.0)**

Set scene frame updating all objects immediately

**Parameters**

- **frame** (*int in [-300000, 300000]*) – Frame number to set
- **subframe** (*float in [0, 1], (optional)*) – Sub-frame time, between 0.0 and 1.0

**update ()**

Update data tagged to be updated from previous access to data or operators

**Inherited Properties**

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`

## References

- `BlendData.scenes`
- `BlendDataScenes.new`
- `BlendDataScenes.remove`
- `Camera.view_frame`
- `CompositorNodeRLayers.scene`
- `Context.scene`
- `EnvironmentMap.save`
- `Image.save_render`
- `Object.dupli_list_create`
- `Object.is_modified`
- `Object.is_visible`
- `Object.to_mesh`
- `RenderEngine.render`
- `RenderEngine.update`
- `Scene.background_set`
- `SceneActuator.scene`
- `SceneSequence.scene`
- `Screen.scene`

### 2.4.450 SceneActuator(Actuator)

base classes — `bpy_struct`, `Actuator`

class `bpy.types.SceneActuator` (*Actuator*)

**camera**

Set this Camera. Leave empty to refer to self object

**Type** `Object`

**mode**

**Type** enum in ['RESTART', 'SET', 'CAMERA', 'ADDFRONT', 'ADDBACK', 'REMOVE', 'SUSPEND', 'RESUME'], default 'RESTART'

**scene**

Set the Scene to be added/removed/paused/resumed

**Type** `Scene`

**Inherited Properties**

- `bpy_struct.id_data`
- `Actuator.name`
- `Actuator.show_expanded`
- `Actuator.pin`
- `Actuator.type`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Actuator.link`
- `Actuator.unlink`

**2.4.451 SceneBases(`bpy_struct`)**

base class — `bpy_struct`

**class** `bpy.types.SceneBases` (*bpy\_struct*)

Collection of scene bases

**active**

Active object base in the scene

**Type** `ObjectBase`

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Scene.object_bases`

## 2.4.452 SceneGameData(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.SceneGameData` (*bpy\_struct*)

Game data for a Scene datablock

**activity\_culling\_box\_radius**

Radius of the activity bubble, in Manhattan length (objects outside the box are activity-culled)

**Type** float in [0, 1000], default 0.0

**depth**

Display bit depth of full screen display

**Type** int in [8, 32], default 0

**dome\_angle**

Field of View of the Dome - it only works in mode Fisheye and Truncated

**Type** int in [-32768, 32767], default 0

**dome\_buffer\_resolution**

Buffer Resolution - decrease it to increase speed

**Type** float in [-inf, inf], default 0.0

**dome\_mode**

Dome physical configurations

**Type** enum in ['FISHEYE', 'TRUNCATED\_FRONT', 'TRUNCATED\_REAR', 'ENVMAP', 'PANORAM\_SPH'], default 'FISHEYE'

#### **dome\_tessellation**

Tessellation level - check the generated mesh in wireframe mode

**Type** int in [-32768, 32767], default 0

#### **dome\_text**

Custom Warp Mesh data file

**Type** Text

#### **dome\_tilt**

Camera rotation in horizontal axis

**Type** int in [-32768, 32767], default 0

#### **fps**

Nominal number of game frames per second (physics fixed timestep = 1/fps, independently of actual frame rate)

**Type** int in [1, 250], default 0

#### **frame\_color**

Set color of the bars

**Type** float array of 3 items in [0, 1], default (0.0, 0.0, 0.0)

#### **frame\_type**

Select the type of Framing you want

- LETTERBOX Letterbox, Show the entire viewport in the display window, using bar horizontally or vertically.
- EXTEND Extend, Show the entire viewport in the display window, viewing more horizontally or vertically.
- SCALE Scale, Stretch or squeeze the viewport to fill the display window.

**Type** enum in ['LETTERBOX', 'EXTEND', 'SCALE'], default 'LETTERBOX'

#### **frequency**

Display clock frequency of fullscreen display

**Type** int in [4, 2000], default 0

#### **level\_height**

Max difference in heights of obstacles to enable their interaction

**Type** float in [0, 200], default 0.0

#### **logic\_step\_max**

Maximum number of logic frame per game frame if graphics slows down the game, higher value allows better synchronization with physics

**Type** int in [1, 5], default 0

#### **material\_mode**

Material mode to use for rendering

- SINGLETEXTURE Singletexture, Singletexture face materials.
- MULTITEXTURE Multitexture, Multitexture materials.

- GLSL GLSL, OpenGL shading language shaders.

**Type** enum in ['SINGLETEXTURE', 'MULTITEXTURE', 'GLSL'], default 'SINGLETEXTURE'

**obstacle\_simulation**

Simulation used for obstacle avoidance in the game engine

**Type** enum in ['NONE', 'RVO\_RAYS', 'RVO\_CELLS'], default 'NONE'

**occlusion\_culling\_resolution**

Size of the occlusion buffer in pixel, use higher value for better precision (slower)

**Type** int in [128, 1024], default 0

**physics\_engine**

Physics engine used for physics simulation in the game engine

- NONE None, Don't use a physics engine.
- BULLET Bullet, Use the Bullet physics engine.

**Type** enum in ['NONE', 'BULLET'], default 'NONE'

**physics\_gravity**

Gravitational constant used for physics simulation in the game engine

**Type** float in [0, 10000], default 0.0

**physics\_step\_max**

Maximum number of physics step per game frame if graphics slows down the game, higher value allows physics to keep up with realtime

**Type** int in [1, 5], default 0

**physics\_step\_sub**

Number of simulation substep per physic timestep, higher value give better physics precision

**Type** int in [1, 5], default 0

**recast\_data**

**Type** `SceneGameRecastData`, (readonly, never None)

**resolution\_x**

Number of horizontal pixels in the screen

**Type** int in [4, 10000], default 0

**resolution\_y**

Number of vertical pixels in the screen

**Type** int in [4, 10000], default 0

**restrict\_animation\_updates**

Restrict the number of animation updates to the animation FPS (this is better for performance, but can cause issues with smooth playback)

**Type** boolean, default False

**show\_debug\_properties**

Show properties marked for debugging while the game runs

**Type** boolean, default False



**show\_framerate\_profile**

Show framerate and profiling information while the game runs

**Type** boolean, default False

**show\_fullscreen**

Start player in a new fullscreen display

**Type** boolean, default False

**show\_mouse**

Start player with a visible mouse cursor

**Type** boolean, default False

**show\_obstacle\_simulation**

Enable debug visualization for obstacle simulation

**Type** boolean, default False

**show\_physics\_visualization**

Show a visualization of physics bounds and interactions

**Type** boolean, default False

**stereo**

- NONE None, Disable Stereo and Dome environments.
- STEREO Stereo, Enable Stereo environment.
- DOME Dome, Enable Dome environment.

**Type** enum in ['NONE', 'STEREO', 'DOME'], default 'NONE'

**stereo\_eye\_separation**

Set the distance between the eyes - the camera focal length/30 should be fine

**Type** float in [0.01, 5], default 0.0

**stereo\_mode**

Stereographic techniques

**Type** enum in ['QUADBUFFERED', 'ABOVEBELOW', 'INTERLACED', 'ANAGLYPH', 'SIDEBYSIDE', 'VINTERLACE'], default 'QUADBUFFERED'

**use\_activity\_culling**

Activity culling is enabled

**Type** boolean, default False

**use\_animation\_record**

Record animation to F-Curves

**Type** boolean, default False

**use\_auto\_start**

Automatically start game at load time

**Type** boolean, default False

**use\_deprecation\_warnings**

Print warnings when using deprecated features in the python API

**Type** boolean, default False

**use\_display\_lists**

Use display lists to speed up rendering by keeping geometry on the GPU

**Type** boolean, default False

**use\_frame\_rate**

Respect the frame rate rather than rendering as many frames as possible

**Type** boolean, default False

**use\_gls1\_color\_management**

Use color management for GLSL rendering

**Type** boolean, default False

**use\_gls1\_extra\_textures**

Use extra textures like normal or specular maps for GLSL rendering

**Type** boolean, default False

**use\_gls1\_lights**

Use lights for GLSL rendering

**Type** boolean, default False

**use\_gls1\_nodes**

Use nodes for GLSL rendering

**Type** boolean, default False

**use\_gls1\_ramps**

Use ramps for GLSL rendering

**Type** boolean, default False

**use\_gls1\_shaders**

Use shaders for GLSL rendering

**Type** boolean, default False

**use\_gls1\_shadows**

Use shadows for GLSL rendering

**Type** boolean, default False

**use\_occlusion\_culling**

Use optimized Bullet DBVT tree for view frustum and occlusion culling

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`

- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Scene.game_settings`

### 2.4.453 SceneGameRecastData(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.SceneGameRecastData` (*bpy\_struct*)

Recast data for a Game datablock

#### **agent\_height**

Minimum height where the agent can still walk

**Type** float in [-inf, inf], default 0.0

#### **agent\_radius**

Radius of the agent

**Type** float in [-inf, inf], default 0.0

#### **cell\_height**

Rasterized cell height

**Type** float in [-inf, inf], default 0.0

#### **cell\_size**

Rasterized cell size

**Type** float in [-inf, inf], default 0.0

#### **climb\_max**

Maximum height between grid cells the agent can climb

**Type** float in [-inf, inf], default 0.0

#### **edge\_max\_error**

Maximum distance error from contour to cells

**Type** float in [-inf, inf], default 0.0

#### **edge\_max\_len**

Maximum contour edge length

**Type** float in [-inf, inf], default 0.0

#### **region\_merge\_size**

Minimum regions size (smaller regions will be merged)

**Type** float in [-inf, inf], default 0.0

**region\_min\_size**

Minimum regions size (smaller regions will be deleted)

**Type** float in [-inf, inf], default 0.0

**sample\_dist**

Detail mesh sample spacing

**Type** float in [-inf, inf], default 0.0

**sample\_max\_error**

Detail mesh simplification max sample error

**Type** float in [-inf, inf], default 0.0

**slope\_max**

Maximum walkable slope angle in degrees

**Type** float in [0, 1.5708], default 0.0

**verts\_per\_poly**

Max number of vertices per polygon

**Type** int in [-inf, inf], default 0

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `SceneGameData.recast_data`

**2.4.454 SceneObjects(bpy\_struct)**

base class — `bpy_struct`

**class** `bpy.types.SceneObjects` (*bpy\_struct*)

Collection of scene objects

**active**

Active object for this scene

**Type** `Object`

**link** (*object*)

Link object to scene, run `scene.update()` after

**Parameters** `object` (`Object`, (never `None`)) – Object to add to scene

**Returns** The newly created base

**Return type** `ObjectBase`

**unlink** (*object*)

Unlink object from scene

**Parameters** `object` (`Object`, (never `None`)) – Object to remove from scene

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Scene.objects`

## 2.4.455 SceneRenderLayer(*bpy\_struct*)

base class — `bpy_struct`

**class** `bpy.types.SceneRenderLayer` (*bpy\_struct*)

Render layer

**exclude\_ambient\_occlusion**

Exclude AO pass from combined

**Type** boolean, default False

**exclude\_emit**

Exclude emission pass from combined

**Type** boolean, default False

**exclude\_environment**

Exclude environment pass from combined

**Type** boolean, default False

**exclude\_indirect**

Exclude indirect pass from combined

**Type** boolean, default False

**exclude\_reflection**

Exclude raytraced reflection pass from combined

**Type** boolean, default False

**exclude\_refraction**

Exclude raytraced refraction pass from combined

**Type** boolean, default False

**exclude\_shadow**

Exclude shadow pass from combined

**Type** boolean, default False

**exclude\_specular**

Exclude specular pass from combined

**Type** boolean, default False

**invert\_zmask**

For Zmask, only render what is behind solid z values instead of in front

**Type** boolean, default False

**layers**

Scene layers included in this render layer

**Type** boolean array of 20 items, default (False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False)

**layers\_zmask**

Zmask scene layers for solid faces

**Type** boolean array of 20 items, default (False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False)

**light\_override**

Group to override all other lights in this render layer

**Type** [Group](#)

**material\_override**

Material to override all other materials in this render layer

**Type** [Material](#)

**name**  
Render layer name  
**Type** string, default ""

**use**  
Disable or enable the render layer  
**Type** boolean, default False

**use\_all\_z**  
Fill in Z values for solid faces in invisible layers, for masking  
**Type** boolean, default False

**use\_edge\_enhance**  
Render Edge-enhance in this Layer (only works for Solid faces)  
**Type** boolean, default False

**use\_halo**  
Render Halos in this Layer (on top of Solid)  
**Type** boolean, default False

**use\_pass\_ambient\_occlusion**  
Deliver AO pass  
**Type** boolean, default False

**use\_pass\_color**  
Deliver shade-less color pass  
**Type** boolean, default False

**use\_pass\_combined**  
Deliver full combined RGBA buffer  
**Type** boolean, default False

**use\_pass\_diffuse**  
Deliver diffuse pass  
**Type** boolean, default False

**use\_pass\_emit**  
Deliver emission pass  
**Type** boolean, default False

**use\_pass\_environment**  
Deliver environment lighting pass  
**Type** boolean, default False

**use\_pass\_indirect**  
Deliver indirect lighting pass  
**Type** boolean, default False

**use\_pass\_material\_index**  
Deliver material index pass  
**Type** boolean, default False

**use\_pass\_mist**  
Deliver mist factor pass (0.0-1.0)

**Type** boolean, default False

**use\_pass\_normal**

Deliver normal pass

**Type** boolean, default False

**use\_pass\_object\_index**

Deliver object index pass

**Type** boolean, default False

**use\_pass\_reflection**

Deliver raytraced reflection pass

**Type** boolean, default False

**use\_pass\_refraction**

Deliver raytraced refraction pass

**Type** boolean, default False

**use\_pass\_shadow**

Deliver shadow pass

**Type** boolean, default False

**use\_pass\_specular**

Deliver specular pass

**Type** boolean, default False

**use\_pass\_uv**

Deliver texture UV pass

**Type** boolean, default False

**use\_pass\_vector**

Deliver speed vector pass

**Type** boolean, default False

**use\_pass\_z**

Deliver Z values pass

**Type** boolean, default False

**use\_sky**

Render Sky in this Layer

**Type** boolean, default False

**use\_solid**

Render Solid faces in this Layer

**Type** boolean, default False

**use\_strand**

Render Strands in this Layer

**Type** boolean, default False

**use\_zmask**

Only render what's in front of the solid z values

**Type** boolean, default False



**use\_ztransp**

Render Z-Transparent faces in this Layer (on top of Solid and Halos)

**Type** boolean, default False

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `RenderLayers.active`
- `RenderSettings.layers`

**2.4.456 SceneSequence(Sequence)**

base classes — `bpy_struct`, `Sequence`

**class** `bpy.types.SceneSequence` (*Sequence*)  
Sequence strip to used the rendered image of a scene

**animation\_offset\_end**

Animation end offset (trim end)

**Type** int in [0, inf], default 0

**animation\_offset\_start**

Animation start offset (trim start)

**Type** int in [0, inf], default 0

**color\_balance**

**Type** `SequenceColorBalance`, (readonly)

**color\_multiply**

**Type** float in [0, 20], default 0.0

**color\_saturation**

**Type** float in [0, 20], default 0.0

**crop**

**Type** `SequenceCrop`, (readonly)

**proxy**

**Type** `SequenceProxy`, (readonly)

**scene**

Scene that this sequence uses

**Type** `Scene`

**scene\_camera**

Override the scenes active camera

**Type** `Object`

**strobe**

Only display every nth frame

**Type** float in [1, 30], default 0.0

**transform**

**Type** `SequenceTransform`, (readonly)

**use\_color\_balance**

(3-Way color correction) on input

**Type** boolean, default False

**use\_crop**

Crop image before processing

**Type** boolean, default False

**use\_deinterlace**

For video movies to remove fields

**Type** boolean, default False

**use\_flip\_x**

Flip on the X axis

**Type** boolean, default False

**use\_flip\_y**

Flip on the Y axis

**Type** boolean, default False

**use\_float**

Convert input to float data

**Type** boolean, default False

**use\_premultiply**

Convert RGB from key alpha to premultiplied alpha

**Type** boolean, default False

**use\_proxy**

Use a preview proxy and/or timecode index for this strip

**Type** boolean, default False

**use\_proxy\_custom\_directory**

Use a custom directory to store data

**Type** boolean, default False

**use\_proxy\_custom\_file**

Use a custom file to read proxy data from

**Type** boolean, default False

**use\_reverse\_frames**

Reverse frame order

**Type** boolean, default False

**use\_translation**

Translate image before processing

**Type** boolean, default False

**Inherited Properties**

- `bpy_struct.id_data`
- `Sequence.name`
- `Sequence.blend_type`
- `Sequence.blend_alpha`
- `Sequence.channel`
- `Sequence.waveform`
- `Sequence.effect_fader`
- `Sequence.frame_final_end`
- `Sequence.frame_offset_end`
- `Sequence.frame_still_end`
- `Sequence.input_1`
- `Sequence.input_2`
- `Sequence.input_3`
- `Sequence.select_left_handle`
- `Sequence.frame_final_duration`
- `Sequence.frame_duration`
- `Sequence.lock`
- `Sequence.mute`
- `Sequence.select_right_handle`
- `Sequence.select`
- `Sequence.speed_factor`
- `Sequence.frame_start`
- `Sequence.frame_final_start`
- `Sequence.frame_offset_start`
- `Sequence.frame_still_start`
- `Sequence.type`
- `Sequence.use_default_fade`
- `Sequence.input_count`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Sequence.getStripElem`
- `Sequence.swap`

### 2.4.457 `Scopes(bpy_struct)`

base class — `bpy_struct`

**class** `bpy.types.Scopes` (*bpy\_struct*)

Scopes for statistical view of an image

**accuracy**

Proportion of original image source pixel lines to sample

**Type** float in [0, 100], default 0.0

**histogram**

Histogram for viewing image statistics

**Type** `Histogram`, (readonly)

**use\_full\_resolution**

Sample every pixel of the image

**Type** boolean, default False

**vectorscope\_alpha**

Opacity of the points

**Type** float in [0, 1], default 0.0

**waveform\_alpha**

Opacity of the points

**Type** float in [0, 1], default 0.0

**waveform\_mode**

**Type** enum in ['LUMA', 'RGB', 'YCBCR601', 'YCBCR709', 'YCBCRJPG'], default 'LUMA'

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `SpaceImageEditor.scopes`

## 2.4.458 Screen(ID)

base classes — `bpy_struct`, `ID`

**class** `bpy.types.Screen` (*ID*)

Screen datablock, defining the layout of areas in a window

### **areas**

Areas the screen is subdivided into

**Type** `bpy_prop_collection` of `Area`, (readonly)

### **is\_animation\_playing**

Animation playback is active

**Type** `boolean`, default `False`, (readonly)

### **scene**

Active scene to be edited in the screen

**Type** `Scene`, (never `None`)

### **show\_fullscreen**

An area is maximised, filling this screen

**Type** `boolean`, default `False`, (readonly)

### **use\_play\_3d\_editors**

**Type** boolean, default False

**use\_play\_animation\_editors**

**Type** boolean, default False

**use\_play\_image\_editors**

**Type** boolean, default False

**use\_play\_node\_editors**

**Type** boolean, default False

**use\_play\_properties\_editors**

**Type** boolean, default False

**use\_play\_sequence\_editors**

**Type** boolean, default False

**use\_play\_top\_left\_3d\_editor**

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`

- `ID.animation_data_clear`
- `ID.update_tag`

## References

- `BlendData.screens`
- `Context.screen`
- `Window.screen`

## 2.4.459 ScrewModifier(Modifier)

base classes — `bpy_struct`, `Modifier`

**class** `bpy.types.ScrewModifier` (*Modifier*)  
Revolve edges

### **angle**

Angle of revolution

**Type** float in [-inf, inf], default 0.0

### **axis**

Screw axis

**Type** enum in ['X', 'Y', 'Z'], default 'X'

### **iterations**

Number of times to apply the screw operation

**Type** int in [1, 10000], default 0

### **object**

Object to define the screw axis

**Type** `Object`

### **render\_steps**

Number of steps in the revolution

**Type** int in [2, 10000], default 0

### **screw\_offset**

Offset the revolution along its axis

**Type** float in [-inf, inf], default 0.0

### **steps**

Number of steps in the revolution

**Type** int in [2, 10000], default 0

### **use\_normal\_calculate**

Calculate the order of edges (needed for meshes, but not curves)

**Type** boolean, default False

### **use\_normal\_flip**

Flip normals of lathed faces

**Type** boolean, default False

**use\_object\_screw\_offset**

Use the distance between the objects to make a screw

**Type** boolean, default False

**Inherited Properties**

- bpy\_struct.id\_data
- Modifier.name
- Modifier.use\_apply\_on\_spline
- Modifier.show\_in\_editmode
- Modifier.show\_expanded
- Modifier.show\_on\_cage
- Modifier.show\_viewport
- Modifier.show\_render
- Modifier.type

**Inherited Functions**

- bpy\_struct.as\_pointer
- bpy\_struct.callback\_add
- bpy\_struct.callback\_remove
- bpy\_struct.driver\_add
- bpy\_struct.driver\_remove
- bpy\_struct.get
- bpy\_struct.is\_property\_hidden
- bpy\_struct.is\_property\_set
- bpy\_struct.items
- bpy\_struct.keyframe\_delete
- bpy\_struct.keyframe\_insert
- bpy\_struct.keys
- bpy\_struct.path\_from\_id
- bpy\_struct.path\_resolve
- bpy\_struct.type\_recast
- bpy\_struct.values

## 2.4.460 Sculpt(Paint)

base classes — bpy\_struct, Paint

**class** bpy.types.Sculpt (*Paint*)

**lock\_x**

Disallow changes to the X axis of vertices

**Type** boolean, default False

**lock\_y**

Disallow changes to the Y axis of vertices

**Type** boolean, default False

**lock\_z**

Disallow changes to the Z axis of vertices



**Type** boolean, default False

**radial\_symmetry**

Number of times to copy strokes across the surface

**Type** int array of 3 items in [1, 64], default (1, 1, 1)

**use\_deform\_only**

Use only deformation modifiers (temporary disable all constructive modifiers except multi-resolution)

**Type** boolean, default False

**use\_symmetry\_feather**

Reduce the strength of the brush where it overlaps symmetrical daubs

**Type** boolean, default False

**use\_symmetry\_x**

Mirror brush across the X axis

**Type** boolean, default False

**use\_symmetry\_y**

Mirror brush across the Y axis

**Type** boolean, default False

**use\_symmetry\_z**

Mirror brush across the Z axis

**Type** boolean, default False

**use\_threaded**

Take advantage of multiple CPU cores to improve sculpting performance

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `Paint.brush`
- `Paint.show_low_resolution`
- `Paint.show_brush`
- `Paint.show_brush_on_surface`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`

- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `ToolSettings.sculpt`

### 2.4.461 Sensor(`bpy_struct`)

base class — `bpy_struct`

subclasses — `RaySensor`, `PropertySensor`, `MessageSensor`, `TouchSensor`, `KeyboardSensor`, `CollisionSensor`, `RadarSensor`, `DelaySensor`, `RandomSensor`, `AlwaysSensor`, `ActuatorSensor`, `JoystickSensor`, `MouseSensor`, `ArmatureSensor`, `NearSensor`

**class** `bpy.types.Sensor` (*`bpy_struct`*)

Game engine logic brick to detect events

**frequency**

Delay between repeated pulses(in logic tics, 0=no delay)

**Type** int in [0, 10000], default 0

**invert**

Invert the level(output) of this sensor

**Type** boolean, default False

**name**

Sensor name

**Type** string, default ""

**pin**

Display when not linked to a visible states controller

**Type** boolean, default False

**show\_expanded**

Set sensor expanded in the user interface

**Type** boolean, default False

**type**

**Type** enum in ['ACTUATOR', 'ALWAYS', 'ARMATURE', 'COLLISION', 'DELAY', 'JOY-STICK', 'KEYBOARD', 'MESSAGE', 'MOUSE', 'NEAR', 'PROPERTY', 'RADAR', 'RANDOM', 'RAY', 'TOUCH'], default 'ALWAYS'

**use\_level**

Level detector, trigger controllers of new states(only applicable upon logic state transition)

**Type** boolean, default False

**use\_pulse\_false\_level**

Activate FALSE level triggering (pulse mode)

**Type** boolean, default False

**use\_pulse\_true\_level**

Activate TRUE level triggering (pulse mode)

**Type** boolean, default False

**use\_tap**

Trigger controllers only for an instant, even while the sensor remains true

**Type** boolean, default False

**link** (*controller*)

Link the sensor to a controller

**Parameters** **controller** (*Controller*) – Controller to link to

**unlink** (*controller*)

Unlink the sensor from a controller

**Parameters** **controller** (*Controller*) – Controller to unlink from

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `Controller.link`
- `Controller.unlink`
- `GameObjectSettings.sensors`

**2.4.462 Sequence(*bpy\_struct*)**

base class — `bpy_struct`

subclasses — `MulticamSequence`, `ImageSequence`, `SceneSequence`, `AdjustmentSequence`, `MetaSequence`, `SoundSequence`, `MovieSequence`, `EffectSequence`

**class** `bpy.types.Sequence` (*bpy\_struct*)

Sequence strip in the sequence editor

**blend\_alpha**

**Type** float in [0, 1], default 0.0

**blend\_type**

**Type** enum in ['REPLACE', 'CROSS', 'ADD', 'SUBTRACT', 'ALPHA\_OVER', 'ALPHA\_UNDER', 'GAMMA\_CROSS', 'MULTIPLY', 'OVER\_DROP'], default 'REPLACE'

**channel**

Y position of the sequence strip

**Type** int in [0, 31], default 0

**effect\_fader**

**Type** float in [0, 1], default 0.0

**frame\_duration**

The length of the contents of this strip before the handles are applied

**Type** int in [1, 300000], default 0, (readonly)

**frame\_final\_duration**

The length of the contents of this strip before the handles are applied

**Type** int in [1, 300000], default 0

**frame\_final\_end**

End frame displayed in the sequence editor after offsets are applied

**Type** int in [-inf, inf], default 0

**frame\_final\_start**

Start frame displayed in the sequence editor after offsets are applied, setting this is equivalent to moving the handle, not the actual start frame

**Type** int in [-inf, inf], default 0

**frame\_offset\_end**

**Type** int in [-inf, inf], default 0, (readonly)

**frame\_offset\_start**

**Type** int in [-inf, inf], default 0, (readonly)

**frame\_start**

**Type** int in [-inf, inf], default 0

**frame\_still\_end**

**Type** int in [0, 300000], default 0, (readonly)

**frame\_still\_start**

**Type** int in [0, 300000], default 0, (readonly)

**input\_1**

First input for the effect strip

**Type** `Sequence`, (readonly)

**input\_2**

Second input for the effect strip

**Type** `Sequence`, (readonly)

**input\_3**

Third input for the effect strip

**Type** `Sequence`, (readonly)

**input\_count**

**Type** `int` in `[0, inf]`, default `0`, (readonly)

**lock**

Lock strip so that it can't be transformed

**Type** `boolean`, default `False`

**mute**

**Type** `boolean`, default `False`

**name**

**Type** `string`, default `""`

**select**

**Type** `boolean`, default `False`

**select\_left\_handle**

**Type** `boolean`, default `False`

**select\_right\_handle**

**Type** `boolean`, default `False`

**speed\_factor**

Multiply the current speed of the sequence with this number or remap current frame to this frame

**Type** `float` in `[-inf, inf]`, default `0.0`

**type**

**Type** `enum` in `['IMAGE', 'META', 'SCENE', 'MOVIE', 'SOUND', 'CROSS', 'ADD', 'SUBTRACT', 'ALPHA_OVER', 'ALPHA_UNDER', 'GAMMA_CROSS', 'MULTIPLY', 'OVER_DROP', 'PLUGIN', 'WIPE', 'GLOW', 'TRANSFORM', 'COLOR', 'SPEED', 'MULTICAM', 'ADJUSTMENT']`, default `'IMAGE'`, (readonly)

**use\_default\_fade**

Fade effect using the built-in default (usually make transition as long as effect strip)

**Type** `boolean`, default `False`

**waveform**

Whether to draw the sound's waveform

**Type** `boolean`, default `False`

**getStripElem** (*frame*)

Return the strip element from a given frame or None

**Parameters** **frame** (*int* in `[-300000, 300000]`) – Frame, The frame to get the strip element from

**Returns** strip element of the current frame

**Return type** `SequenceElement`

**swap** (*other*)

swap

**Parameters** **other** (`Sequence`, (never `None`)) – Other

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `MetaSequence.sequences`
- `Sequence.input_1`
- `Sequence.input_2`
- `Sequence.input_3`
- `Sequence.swap`
- `SequenceEditor.active_strip`
- `SequenceEditor.meta_stack`
- `SequenceEditor.sequences`
- `SequenceEditor.sequences_all`

## 2.4.463 SequenceColorBalance(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.SequenceColorBalance` (*bpy\_struct*)

Color balance parameters for a sequence strip

**gain**

Color balance gain (highlights)

**Type** float array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`

**gamma**

Color balance gamma (midtones)

**Type** float array of 3 items in  $[-\text{inf}, \text{inf}]$ , default (0.0, 0.0, 0.0)**invert\_gain****Type** boolean, default False**invert\_gamma****Type** boolean, default False**invert\_lift****Type** boolean, default False**lift**

Color balance lift (shadows)

**Type** float array of 3 items in  $[-\text{inf}, \text{inf}]$ , default (0.0, 0.0, 0.0)**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `AdjustmentSequence.color_balance`
- `EffectSequence.color_balance`
- `ImageSequence.color_balance`
- `MetaSequence.color_balance`
- `MovieSequence.color_balance`
- `MulticamSequence.color_balance`
- `SceneSequence.color_balance`

## 2.4.464 SequenceCrop(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.SequenceCrop` (*bpy\_struct*)

Cropping parameters for a sequence strip

**max\_x**

**Type** int in [0, inf], default 0

**max\_y**

**Type** int in [0, inf], default 0

**min\_x**

**Type** int in [0, inf], default 0

**min\_y**

**Type** int in [0, inf], default 0

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `AdjustmentSequence.crop`
- `EffectSequence.crop`
- `ImageSequence.crop`
- `MetaSequence.crop`
- `MovieSequence.crop`
- `MulticamSequence.crop`
- `SceneSequence.crop`



## 2.4.465 SequenceEditor(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.SequenceEditor` (*bpy\_struct*)

Sequence editing data for a Scene datablock

**active\_strip**

**Type** `Sequence`

**meta\_stack**

Meta strip stack, last is currently edited meta strip

**Type** `bpy_prop_collection` of `Sequence`, (readonly)

**overlay\_frame**

Sequencer's active strip

**Type** `int` in `[-inf, inf]`, default `0`

**overlay\_lock**

**Type** `boolean`, default `False`

**sequences**

**Type** `bpy_prop_collection` of `Sequence`, (readonly)

**sequences\_all**

**Type** `bpy_prop_collection` of `Sequence`, (readonly)

**show\_overlay**

Partial overlay on top of the sequencer

**Type** `boolean`, default `False`

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Scene.sequence_editor`

## 2.4.466 SequenceElement(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.SequenceElement` (*bpy\_struct*)  
Sequence strip data for a single frame

**filename**

**Type** string, default ""

**orig\_height**

Original image height

**Type** int in [-inf, inf], default 0, (readonly)

**orig\_width**

Original image width

**Type** int in [-inf, inf], default 0, (readonly)

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `ImageSequence.elements`
- `MovieSequence.elements`
- `Sequence.getStripElem`

## 2.4.467 SequenceProxy(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.SequenceProxy` (*bpy\_struct*)

Proxy parameters for a sequence strip

**build\_100**

Build 100% proxy resolution

**Type** boolean, default False

**build\_25**

Build 25% proxy resolution

**Type** boolean, default False

**build\_50**

Build 50% proxy resolution

**Type** boolean, default False

**build\_75**

Build 75% proxy resolution

**Type** boolean, default False

**build\_free\_run**

Build free run time code index

**Type** boolean, default False

**build\_free\_run\_rec\_date**

Build free run time code index using Record Date/Time

**Type** boolean, default False

**build\_record\_run**

Build record run time code index

**Type** boolean, default False

**directory**

Location to store the proxy files

**Type** string, default ""

**filepath**

Location of custom proxy file

**Type** string, default ""

**quality**

JPEG Quality of proxies to build

**Type** int in [0, 32767], default 0

**timecode**

- NONE No TC in use.
- RECORD\_RUN Record Run, Use images in the order as they are recorded.
- FREE\_RUN Free Run, Use global timestamp written by recording device.
- FREE\_RUN\_REC\_DATE Free Run (rec date), Interpolate a global timestamp using the record date and time written by recording device.

**Type** enum in ['NONE', 'RECORD\_RUN', 'FREE\_RUN', 'FREE\_RUN\_REC\_DATE'], default 'NONE'

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `AdjustmentSequence.proxy`
- `EffectSequence.proxy`
- `ImageSequence.proxy`
- `MetaSequence.proxy`
- `MovieSequence.proxy`
- `MulticamSequence.proxy`
- `SceneSequence.proxy`

## 2.4.468 SequenceTransform(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.SequenceTransform` (*bpy\_struct*)

Transform parameters for a sequence strip

**offset\_x**

**Type** int in [-inf, inf], default 0

**offset\_y**

**Type** int in [-inf, inf], default 0

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `AdjustmentSequence.transform`
- `EffectSequence.transform`
- `ImageSequence.transform`
- `MetaSequence.transform`
- `MovieSequence.transform`
- `MulticamSequence.transform`
- `SceneSequence.transform`

### 2.4.469 ShaderNode(Node)

base classes — `bpy_struct`, `Node`

subclasses — `ShaderNodeHueSaturation`, `ShaderNodeVectorMath`, `ShaderNodeRGB`, `ShaderNodeCameraData`, `ShaderNodeMath`, `ShaderNodeRGBToBW`, `ShaderNodeRGBCurve`, `ShaderNodeInvert`, `ShaderNodeExtendedMaterial`, `ShaderNodeMaterial`, `ShaderNodeValue`, `ShaderNodeMixRGB`, `ShaderNodeSeparateRGB`, `ShaderNodeVectorCurve`, `ShaderNodeCombineRGB`, `ShaderNodeOutput`, `ShaderNodeValToRGB`, `ShaderNodeGeometry`, `ShaderNodeTexture`, `ShaderNodeMapping`, `ShaderNodeSqueeze`, `ShaderNodeNormal`

**class** `bpy.types.ShaderNode` (*Node*)  
Material shader node

#### type

**Type** enum in `['OUTPUT', 'MATERIAL', 'RGB', 'VALUE', 'MIX_RGB', 'VAL-TORGB', 'RGBTOBW', 'TEXTURE', 'NORMAL', 'GEOMETRY', 'MAPPING', 'CURVE_VEC', 'CURVE_RGB', 'CAMERA', 'MATH', 'VECT_MATH', 'SQUEEZE',`

‘MATERIAL\_EXT’, ‘INVERT’, ‘SEPRGB’, ‘COMBRGB’, ‘HUE\_SAT’, ‘SCRIPT’, ‘GROUP’], default ‘OUTPUT’, (readonly)

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.470 ShaderNodeCameraData(ShaderNode)

base classes — `bpy_struct`, `Node`, `ShaderNode`

class `bpy.types.ShaderNodeCameraData` (*ShaderNode*)

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `ShaderNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.471 ShaderNodeCombineRGB(ShaderNode)

base classes — `bpy_struct`, `Node`, `ShaderNode`

**class** `bpy.types.ShaderNodeCombineRGB` (*ShaderNode*)

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `ShaderNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`

- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.472 ShaderNodeExtendedMaterial(ShaderNode)

base classes — `bpy_struct`, `Node`, `ShaderNode`

**class** `bpy.types.ShaderNodeExtendedMaterial` (*ShaderNode*)

### **invert\_normal**

Material Node uses inverted normal

**Type** boolean, default False

### **material**

**Type** `Material`

### **use\_diffuse**

Material Node outputs Diffuse

**Type** boolean, default False

### **use\_specular**

Material Node outputs Specular

**Type** boolean, default False

## Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `ShaderNode.type`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`



- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.473 ShaderNodeGeometry(ShaderNode)

base classes — `bpy_struct`, `Node`, `ShaderNode`

**class** `bpy.types.ShaderNodeGeometry` (*ShaderNode*)

#### **color\_layer**

**Type** string, default ""

#### **uv\_layer**

**Type** string, default ""

#### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `ShaderNode.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.474 ShaderNodeHueSaturation(ShaderNode)

base classes — `bpy_struct`, `Node`, `ShaderNode`

`class bpy.types.ShaderNodeHueSaturation` (*ShaderNode*)

#### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `ShaderNode.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.475 ShaderNodeInvert(ShaderNode)

base classes — `bpy_struct`, `Node`, `ShaderNode`

`class bpy.types.ShaderNodeInvert` (*ShaderNode*)

#### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `ShaderNode.type`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.476 ShaderNodeMapping(ShaderNode)

base classes — `bpy_struct`, `Node`, `ShaderNode`

**class** `bpy.types.ShaderNodeMapping` (*ShaderNode*)

### **location**

Location offset for the input coordinate

**Type** float array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`

### **max**

Maximum value to clamp coordinate to

**Type** float array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`

### **min**

Minimum value to clamp coordinate to

**Type** float array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`

### **rotation**

Rotation offset for the input coordinate

**Type** float array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`

### **scale**

Scale adjustment for the input coordinate

**Type** float array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`

### **use\_max**

Clamp the output coordinate to a maximum value

**Type** boolean, default `False`

### **use\_min**

Clamp the output coordinate to a minimum value

**Type** boolean, default `False`

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `ShaderNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.477 ShaderNodeMaterial(ShaderNode)

base classes — `bpy_struct`, `Node`, `ShaderNode`

**class** `bpy.types.ShaderNodeMaterial` (*ShaderNode*)

**invert\_normal**

Material Node uses inverted normal

**Type** boolean, default False

**material**

**Type** `Material`

**use\_diffuse**

Material Node outputs Diffuse

**Type** boolean, default False

**use\_specular**

Material Node outputs Specular

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `ShaderNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.478 ShaderNodeMath(ShaderNode)

base classes — `bpy_struct`, `Node`, `ShaderNode`

**class** `bpy.types.ShaderNodeMath` (*ShaderNode*)

#### operation

**Type** enum in ['ADD', 'SUBTRACT', 'MULTIPLY', 'DIVIDE', 'SINE', 'COSINE', 'TANGENT', 'ARCSINE', 'ARCCOSINE', 'ARCTANGENT', 'POWER', 'LOGARITHM', 'MINIMUM', 'MAXIMUM', 'ROUND', 'LESS\_THAN', 'GREATER\_THAN'], default 'ADD'

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`

- `Node.parent`
- `ShaderNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.479 ShaderNodeMixRGB(ShaderNode)

base classes — `bpy_struct`, `Node`, `ShaderNode`

**class** `bpy.types.ShaderNodeMixRGB` (*ShaderNode*)

#### **blend\_type**

**Type** enum in ['MIX', 'ADD', 'MULTIPLY', 'SUBTRACT', 'SCREEN', 'DIVIDE', 'DIFFERENCE', 'DARKEN', 'LIGHTEN', 'OVERLAY', 'DODGE', 'BURN', 'HUE', 'SATURATION', 'VALUE', 'COLOR', 'SOFT\_LIGHT', 'LINEAR\_LIGHT'], default 'MIX'

#### **use\_alpha**

Include alpha of second input in this operation

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `ShaderNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.480 ShaderNodeNormal(ShaderNode)

base classes — `bpy_struct`, `Node`, `ShaderNode`

class `bpy.types.ShaderNodeNormal` (*ShaderNode*)

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `ShaderNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`

- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.481 ShaderNodeOutput(ShaderNode)

base classes — `bpy_struct`, `Node`, `ShaderNode`

**class** `bpy.types.ShaderNodeOutput` (*ShaderNode*)

#### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `ShaderNode.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.482 ShaderNodeRGB(ShaderNode)

base classes — `bpy_struct`, `Node`, `ShaderNode`

**class** `bpy.types.ShaderNodeRGB` (*ShaderNode*)

#### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`



- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `ShaderNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.483 ShaderNodeRGBCurve(ShaderNode)

base classes — `bpy_struct`, `Node`, `ShaderNode`

class `bpy.types.ShaderNodeRGBCurve` (*ShaderNode*)

### mapping

**Type** `CurveMapping`, (readonly)

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `ShaderNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`

- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.484 ShaderNodeRGBToBW(ShaderNode)

base classes — `bpy_struct`, `Node`, `ShaderNode`

**class** `bpy.types.ShaderNodeRGBToBW` (*ShaderNode*)

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `ShaderNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.485 ShaderNodeSeparateRGB(ShaderNode)

base classes — `bpy_struct`, `Node`, `ShaderNode`

**class** `bpy.types.ShaderNodeSeparateRGB` (*ShaderNode*)

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `ShaderNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.486 ShaderNodeSqueeze(ShaderNode)

base classes — `bpy_struct`, `Node`, `ShaderNode`

**class** `bpy.types.ShaderNodeSqueeze` (*ShaderNode*)

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`

- `ShaderNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.487 ShaderNodeTexture(ShaderNode)

base classes — `bpy_struct`, `Node`, `ShaderNode`

**class** `bpy.types.ShaderNodeTexture` (*ShaderNode*)

### **node\_output**

For node-based textures, which output node to use

**Type** `int` in `[-32768, 32767]`, default `0`

### **texture**

**Type** `Texture`

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `ShaderNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`

- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.488 ShaderNodeTree(NodeTree)

base classes — `bpy_struct`, `ID`, `NodeTree`

**class** `bpy.types.ShaderNodeTree` (*NodeTree*)  
Node tree consisting of linked nodes used for materials

#### **nodes**

**Type** `ShaderNodes` `bpy_prop_collection` of `Node`, (readonly)

#### Inherited Properties

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`
- `NodeTree.animation_data`
- `NodeTree.grease_pencil`
- `NodeTree.inputs`
- `NodeTree.links`
- `NodeTree.outputs`
- `NodeTree.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`

- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`

### 2.4.489 ShaderNodeValToRGB(ShaderNode)

base classes — `bpy_struct`, `Node`, `ShaderNode`

**class** `bpy.types.ShaderNodeValToRGB` (*ShaderNode*)

#### **color\_ramp**

**Type** `ColorRamp`, (readonly)

#### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `ShaderNode.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.490 ShaderNodeValue(ShaderNode)

base classes — `bpy_struct`, `Node`, `ShaderNode`

**class** `bpy.types.ShaderNodeValue` (*ShaderNode*)

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `ShaderNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.491 ShaderNodeVectorCurve(ShaderNode)

base classes — `bpy_struct`, `Node`, `ShaderNode`

**class** `bpy.types.ShaderNodeVectorCurve` (*ShaderNode*)

**mapping**

**Type** `CurveMapping`, (readonly)

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`

- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `ShaderNode.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.492 ShaderNodeVectorMath(ShaderNode)

base classes — `bpy_struct`, `Node`, `ShaderNode`

class `bpy.types.ShaderNodeVectorMath` (*ShaderNode*)

#### operation

**Type** enum in ['ADD', 'SUBTRACT', 'AVERAGE', 'DOT\_PRODUCT', 'CROSS\_PRODUCT', 'NORMALIZE'], default 'ADD'

#### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `ShaderNode.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`



- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.493 ShaderNodes(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.ShaderNodes` (*bpy\_struct*)  
Collection of Shader Nodes

**new** (*type*, *group=None*)  
Add a node to this node tree

#### Parameters

- **type** (*enum in ['OUTPUT', 'MATERIAL', 'RGB', 'VALUE', 'MIX\_RGB', 'VALTORGB', 'RGBTOBW', 'TEXTURE', 'NORMAL', 'GEOMETRY', 'MAPPING', 'CURVE\_VEC', 'CURVE\_RGB', 'CAMERA', 'MATH', 'VECT\_MATH', 'SQUEEZE', 'MATERIAL\_EXT', 'INVERT', 'SEPRGB', 'COMBRGB', 'HUE\_SAT', 'SCRIPT', 'GROUP']*) – Type, Type of node to add
- **group** (`NodeTree`, (optional)) – The group tree

**Returns** New node

**Return type** `Node`

**remove** (*node*)  
Remove a node from this node tree

**Parameters** **node** (`Node`) – The node to remove

#### Inherited Properties

- `bpy_struct.id_data`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`

- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `ShaderNodeTree.nodes`

## 2.4.494 ShapeActionActuator(Actuator)

base classes — `bpy_struct`, `Actuator`

**class** `bpy.types.ShapeActionActuator` (*Actuator*)

Actuator to control shape key animations

### **action**

**Type** `Action`

### **frame\_blend\_in**

Number of frames of motion blending

**Type** `int` in `[0, 32767]`, default `0`

### **frame\_end**

**Type** `float` in `[-inf, inf]`, default `0.0`

### **frame\_property**

Assign the action's current frame number to this property

**Type** `string`, default `""`

### **frame\_start**

**Type** `float` in `[-inf, inf]`, default `0.0`

### **mode**

Action playback type

**Type** `enum` in `['PLAY', 'PINGPONG', 'FLIPPER', 'LOOPSTOP', 'LOOPEND', 'PROPERTY']`, default `'PLAY'`

### **priority**

Execution priority - lower numbers will override actions with higher numbers. With 2 or more actions at once, the overriding channels must be lower in the stack

**Type** `int` in `[0, 100]`, default `0`

### **property**

Use this property to define the Action position

**Type** `string`, default `""`

**use\_continue\_last\_frame**

Restore last frame when switching on/off, otherwise play from the start each time

**Type** boolean, default False

**Inherited Properties**

- `bpy_struct.id_data`
- `Actuator.name`
- `Actuator.show_expanded`
- `Actuator.pin`
- `Actuator.type`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Actuator.link`
- `Actuator.unlink`

**2.4.495 ShapeKey(`bpy_struct`)**

base class — `bpy_struct`

**class** `bpy.types.ShapeKey` (*bpy\_struct*)

Shape key in a shape keys datablock

**data**

**Type** `bpy_prop_collection` of `UnknownType`, (readonly)

**frame**

Frame for absolute keys

**Type** float in `[-inf, inf]`, default `0.0`, (readonly)

**interpolation**

Interpolation type

**Type** enum in `['KEY_LINEAR', 'KEY_CARDINAL', 'KEY_BSPLINE']`, default `'KEY_LINEAR'`

**mute**  
Mute this shape key  
**Type** boolean, default False

**name**  
Name of Shape Key  
**Type** string, default ""

**relative\_key**  
Shape used as a relative key  
**Type** ShapeKey

**slider\_max**  
Maximum for slider  
**Type** float in [-10, 10], default 1.0

**slider\_min**  
Minimum for slider  
**Type** float in [-10, 10], default 0.0

**value**  
Value of shape key at the current frame  
**Type** float in [0, 1], default 0.0

**vertex\_group**  
Vertex weight group, to blend with basis shape  
**Type** string, default ""

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `ClothSettings.rest_shape_key`
- `Key.key_blocks`
- `Key.reference_key`
- `Object.active_shape_key`
- `Object.shape_key_add`
- `ShapeKey.relative_key`

### 2.4.496 ShapeKeyBezierPoint(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.ShapeKeyBezierPoint` (*bpy\_struct*)

Point in a shape key for Bezier curves

**co**

**Type** float array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`

**handle\_left**

**Type** float array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`

**handle\_right**

**Type** float array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`

#### Inherited Properties

- `bpy_struct.id_data`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.497 ShapeKeyCurvePoint(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.ShapeKeyCurvePoint` (*bpy\_struct*)

Point in a shape key for curves

**co**

**Type** float array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`

**tilt**

**Type** float in `[-inf, inf]`, default `0.0`

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.498 ShapeKeyPoint(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.ShapeKeyPoint` (*bpy\_struct*)

Point in a shape key

**co**

**Type** float array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`

### Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.499 ShrinkwrapConstraint(Constraint)

base classes — `bpy_struct`, `Constraint`

**class** `bpy.types.ShrinkwrapConstraint` (*Constraint*)

Creates constraint-based shrinkwrap relationship

#### **distance**

Distance to Target

**Type** float in [0, 100], default 0.0

#### **shrinkwrap\_type**

Selects type of shrinkwrap algorithm for target position

- `NEAREST_SURFACE` Nearest Surface Point, Shrinks the location to the nearest target surface.
- `PROJECT` Project, Shrinks the location to the nearest target surface along a given axis.
- `NEAREST_VERTEX` Nearest Vertex, Shrinks the location to the nearest target vertex.

**Type** enum in ['NEAREST\_SURFACE', 'PROJECT', 'NEAREST\_VERTEX'], default 'NEAREST\_SURFACE'

#### **target**

Target Object

**Type** `Object`

#### **use\_x**

Projection over X Axis

**Type** boolean, default False

#### **use\_y**

Projection over Y Axis

**Type** boolean, default False

**use\_z**

Projection over Z Axis

**Type** boolean, default False

**Inherited Properties**

- `bpy_struct.id_data`
- `Constraint.name`
- `Constraint.active`
- `Constraint.mute`
- `Constraint.show_expanded`
- `Constraint.influence`
- `Constraint.error_location`
- `Constraint.owner_space`
- `Constraint.is_proxy_local`
- `Constraint.error_rotation`
- `Constraint.target_space`
- `Constraint.type`
- `Constraint.is_valid`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.500 ShrinkwrapModifier(Modifier)

base classes — `bpy_struct`, `Modifier`

**class** `bpy.types.ShrinkwrapModifier` (*Modifier*)

Shrink wrapping modifier to shrink wrap and object to a target

**auxiliary\_target**

Additional mesh target to shrink to

**Type** `Object`

**cull\_face**

Stop vertices from projecting to a the face on the target when facing towards/away



**Type** enum in ['OFF', 'FRONT', 'BACK'], default 'OFF'

**offset**

Distance to keep from the target

**Type** float in [-inf, inf], default 0.0

**subsurf\_levels**

Number of subdivisions that must be performed before extracting vertices' positions and normals

**Type** int in [0, 6], default 0

**target**

Mesh target to shrink to

**Type** Object

**use\_keep\_above\_surface**

**Type** boolean, default False

**use\_negative\_direction**

Allow vertices to move in the negative direction of axis

**Type** boolean, default False

**use\_positive\_direction**

Allow vertices to move in the positive direction of axis

**Type** boolean, default False

**use\_project\_x**

**Type** boolean, default False

**use\_project\_y**

**Type** boolean, default False

**use\_project\_z**

**Type** boolean, default False

**vertex\_group**

Vertex group name

**Type** string, default ""

**wrap\_method**

- NEAREST\_SURFACEPOINT Nearest Surface Point, Shrinks the mesh to the nearest target surface.
- PROJECT Project, Shrinks the mesh to the nearest target surface along a given axis.
- NEAREST\_VERTEX Nearest Vertex, Shrinks the mesh to the nearest target vertex.

**Type** enum in ['NEAREST\_SURFACEPOINT', 'PROJECT', 'NEAREST\_VERTEX'], default 'NEAREST\_SURFACEPOINT'

### Inherited Properties

- `bpy_struct.id_data`
- `Modifier.name`
- `Modifier.use_apply_on_spline`
- `Modifier.show_in_editmode`

- `Modifier.show_expanded`
- `Modifier.show_on_cage`
- `Modifier.show_viewport`
- `Modifier.show_render`
- `Modifier.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.501 SimpleDeformModifier(Modifier)

base classes — `bpy_struct`, `Modifier`

**class** `bpy.types.SimpleDeformModifier` (*Modifier*)

Simple deformation modifier to apply effects such as twisting and bending

### **deform\_method**

- **TWIST** Twist, Rotates around the Z axis of the modifier space.
- **BEND** Bend, Bends the mesh over the Z axis of the modifier space.
- **TAPER** Taper, Linearly scales along Z axis of the modifier space.
- **STRETCH** Stretch, Stretches the object along the Z axis of the modifier space.

**Type** enum in ['TWIST', 'BEND', 'TAPER', 'STRETCH'], default 'TWIST'

### **factor**

Amount to deform object

**Type** float in [-inf, inf], default 0.0

### **limits**

Lower/Upper limits for deform

**Type** float array of 2 items in [0, 1], default (0.0, 0.0)

### **lock\_x**

Do not allow tapering along the X axis

**Type** boolean, default False

**lock\_y**

Do not allow tapering along the Y axis

**Type** boolean, default False

**origin**

Origin of modifier space coordinates

**Type** Object

**use\_relative**

Sets the origin of deform space to be relative to the object

**Type** boolean, default False

**vertex\_group**

Vertex group name

**Type** string, default ""

**Inherited Properties**

- bpy\_struct.id\_data
- Modifier.name
- Modifier.use\_apply\_on\_spline
- Modifier.show\_in\_editmode
- Modifier.show\_expanded
- Modifier.show\_on\_cage
- Modifier.show\_viewport
- Modifier.show\_render
- Modifier.type

**Inherited Functions**

- bpy\_struct.as\_pointer
- bpy\_struct.callback\_add
- bpy\_struct.callback\_remove
- bpy\_struct.driver\_add
- bpy\_struct.driver\_remove
- bpy\_struct.get
- bpy\_struct.is\_property\_hidden
- bpy\_struct.is\_property\_set
- bpy\_struct.items
- bpy\_struct.keyframe\_delete
- bpy\_struct.keyframe\_insert
- bpy\_struct.keys
- bpy\_struct.path\_from\_id
- bpy\_struct.path\_resolve
- bpy\_struct.type\_recast
- bpy\_struct.values

**2.4.502 SmokeCollSettings(bpy\_struct)**

base class — bpy\_struct

**class** `bpy.types.SmokeCollSettings` (*bpy\_struct*)  
Smoke collision settings

#### Inherited Properties

- `bpy_struct.id_data`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

#### References

- `SmokeModifier.coll_settings`

### 2.4.503 SmokeDomainSettings(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.SmokeDomainSettings` (*bpy\_struct*)  
Smoke domain settings

#### **alpha**

How much density affects smoke motion (higher value results in faster rising smoke)

**Type** float in [-5, 5], default 0.0

#### **amplify**

Enhance the resolution of smoke by this factor using noise

**Type** int in [1, 10], default 0

#### **beta**

How much heat affects smoke motion (higher value results in faster rising smoke)

**Type** float in [-5, 5], default 0.0

#### **collision\_extents**

Select which domain border will be treated as collision object

- BORDEROPEN** Open, Smoke doesn't collide with any border.
- BORDERVERTICAL** Vertically Open, Smoke doesn't collide with top and bottom sides.
- BORDERCLOSED** Collide All, Smoke collides with every side.

**Type** enum in ['BORDEROPEN', 'BORDERVERTICAL', 'BORDERCLOSED'], default 'BORDEROPEN'

#### **collision\_group**

Limit collisions to this group

**Type** Group

#### **dissolve\_speed**

Dissolve Speed

**Type** int in [1, 10000], default 0

#### **effector\_group**

Limit effectors to this group

**Type** Group

#### **effector\_weights**

**Type** EffectorWeights, (readonly)

#### **fluid\_group**

Limit fluid objects to this group

**Type** Group

#### **noise\_type**

Noise method which is used for creating the high resolution

**Type** enum in ['NOISEWAVE'], default 'NOISEWAVE'

#### **point\_cache**

**Type** PointCache, (readonly, never None)

#### **point\_cache\_compress\_type**

Compression method to be used

- CACHELIGHT** Light, Fast but not so effective compression.
- CACHEHEAVY** Heavy, Effective but slow compression.

**Type** enum in ['CACHELIGHT', 'CACHEHEAVY'], default 'CACHELIGHT'

#### **resolution\_max**

Maximal resolution used in the fluid domain

**Type** int in [24, 512], default 0

#### **show\_high\_resolution**

Show high resolution (using amplification)

**Type** boolean, default False

#### **smooth\_emitter**

Smoothen emitted smoke to avoid blockiness

**Type** boolean, default False

**strength**

Strength of noise

**Type** float in [0, 10], default 0.0

**time\_scale**

Adjust simulation speed

**Type** float in [0.2, 1.5], default 0.0

**use\_dissolve\_smoke**

Enable smoke to disappear over time

**Type** boolean, default False

**use\_dissolve\_smoke\_log**

Using 1/x

**Type** boolean, default False

**use\_high\_resolution**

Enable high resolution (using amplification)

**Type** boolean, default False

**vorticity**

Amount of turbulence/rotation in fluid

**Type** float in [0.01, 4], default 0.0

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `SmokeModifier.domain_settings`

## 2.4.504 SmokeFlowSettings(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.SmokeFlowSettings` (*bpy\_struct*)

Smoke flow settings

**density**

**Type** float in [0.001, 1], default 0.0

**initial\_velocity**

Smoke inherits its velocity from the emitter particle

**Type** boolean, default False

**particle\_system**

Particle systems emitted from the object

**Type** `ParticleSystem`

**temperature**

Temperature difference to ambient temperature

**Type** float in [-10, 10], default 0.0

**use\_absolute**

Only allow given density value in emitter area

**Type** boolean, default False

**use\_outflow**

Delete smoke from simulation

**Type** boolean, default False

**velocity\_factor**

Multiplier to adjust velocity passed to smoke

**Type** float in [-2, 2], default 0.0

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`

- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `SmokeModifier.flow_settings`

## 2.4.505 SmokeModifier(Modifier)

base classes — `bpy_struct`, `Modifier`

**class** `bpy.types.SmokeModifier` (*Modifier*)

Smoke simulation modifier

### **coll\_settings**

**Type** `SmokeCollSettings`, (readonly)

### **domain\_settings**

**Type** `SmokeDomainSettings`, (readonly)

### **flow\_settings**

**Type** `SmokeFlowSettings`, (readonly)

### **smoke\_type**

- `NONE` None.
- `DOMAIN` Domain.
- `FLOW` Flow, Inflow/Outflow.
- `COLLISION` Collision.

**Type** enum in [`'NONE'`, `'DOMAIN'`, `'FLOW'`, `'COLLISION'`], default `'NONE'`

## Inherited Properties

- `bpy_struct.id_data`
- `Modifier.name`
- `Modifier.use_apply_on_spline`
- `Modifier.show_in_editmode`
- `Modifier.show_expanded`
- `Modifier.show_on_cage`
- `Modifier.show_viewport`
- `Modifier.show_render`
- `Modifier.type`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`



- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.506 SmoothModifier(Modifier)

base classes — `bpy_struct`, `Modifier`

**class** `bpy.types.SmoothModifier` (*Modifier*)

Smoothing effect modifier

**factor**

Strength of modifier effect

**Type** float in `[-inf, inf]`, default 0.0

**iterations**

**Type** int in `[-32768, 32767]`, default 0

**use\_x**

Smooth object along X axis

**Type** boolean, default False

**use\_y**

Smooth object along Y axis

**Type** boolean, default False

**use\_z**

Smooth object along Z axis

**Type** boolean, default False

**vertex\_group**

Name of Vertex Group which determines influence of modifier per point

**Type** string, default ""

#### Inherited Properties

- `bpy_struct.id_data`
- `Modifier.name`
- `Modifier.use_apply_on_spline`
- `Modifier.show_in_editmode`
- `Modifier.show_expanded`
- `Modifier.show_on_cage`
- `Modifier.show_viewport`

- `Modifier.show_render`
- `Modifier.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.507 SoftBodyModifier(Modifier)

base classes — `bpy_struct`, `Modifier`

**class** `bpy.types.SoftBodyModifier` (*Modifier*)  
Soft body simulation modifier

### **point\_cache**

**Type** `PointCache`, (readonly, never None)

### **settings**

**Type** `SoftBodySettings`, (readonly, never None)

### Inherited Properties

- `bpy_struct.id_data`
- `Modifier.name`
- `Modifier.use_apply_on_spline`
- `Modifier.show_in_editmode`
- `Modifier.show_expanded`
- `Modifier.show_on_cage`
- `Modifier.show_viewport`
- `Modifier.show_render`
- `Modifier.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`

- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.508 SoftBodySettings(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.SoftBodySettings` (*bpy\_struct*)  
Soft body simulation settings for an object

**aero**

Make edges ‘sail’

**Type** int in [0, 30000], default 0

**aerodynamics\_type**

Method of calculating aerodynamic interaction

- **SIMPLE** Simple, Edges receive a drag force from surrounding media.
- **LIFT\_FORCE** Lift Force, Edges receive a lift force when passing through surrounding media.

**Type** enum in ['SIMPLE', 'LIFT\_FORCE'], default ‘SIMPLE’

**ball\_damp**

Blending to inelastic collision

**Type** float in [0.001, 1], default 0.0

**ball\_size**

Absolute ball size or factor if not manual adjusted

**Type** float in [-10, 10], default 0.0

**ball\_stiff**

Ball inflating pressure

**Type** float in [0.001, 100], default 0.0

**bend**

Bending Stiffness

**Type** float in [0, 10], default 0.0

**choke**

‘Viscosity’ inside collision target

**Type** int in [0, 100], default 0

**collision\_type**

Choose Collision Type

- MANUAL** Manual, Manual adjust.
- AVERAGE** Average, Average Spring length \* Ball Size.
- MINIMAL** Minimal, Minimal Spring length \* Ball Size.
- MAXIMAL** Maximal, Maximal Spring length \* Ball Size.
- MINMAX** AvMinMax, (Min+Max)/2 \* Ball Size.

**Type** enum in ['MANUAL', 'AVERAGE', 'MINIMAL', 'MAXIMAL', 'MINMAX'], default 'MANUAL'

**damping**

Edge spring friction

**Type** float in [0, 50], default 0.0

**effector\_weights**

**Type** `EffectorWeights`, (readonly)

**error\_threshold**

The Runge-Kutta ODE solver error limit, low value gives more precision, high values speed

**Type** float in [0.001, 10], default 0.0

**friction**

General media friction for point movements

**Type** float in [0, 50], default 0.0

**fuzzy**

Fuzziness while on collision, high values make collision handling faster but less stable

**Type** int in [1, 100], default 0

**goal\_default**

Default Goal (vertex target position) value, when no Vertex Group used

**Type** float in [0, 1], default 0.0

**goal\_friction**

Goal (vertex target position) friction

**Type** float in [0, 50], default 0.0

**goal\_max**

Goal maximum, vertex weights are scaled to match this range

**Type** float in [0, 1], default 0.0

**goal\_min**

Goal minimum, vertex weights are scaled to match this range

**Type** float in [0, 1], default 0.0

**goal\_spring**

Goal (vertex target position) spring stiffness

**Type** float in [0, 0.999], default 0.0

**gravity**

Apply gravitation to point movement

**Type** float in [-10, 10], default 0.0**location\_mass\_center**

Location of Center of mass

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)**mass**

General Mass value

**Type** float in [0, 50000], default 0.0**plastic**

Permanent deform

**Type** int in [0, 100], default 0**pull**

Edge spring stiffness when longer than rest length

**Type** float in [0, 0.999], default 0.0**push**

Edge spring stiffness when shorter than rest length

**Type** float in [0, 0.999], default 0.0**rotation\_estimate**

Estimated rotation matrix

**Type** float array of 9 items in [-inf, inf], default (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)**scale\_estimate**

Estimated scale matrix

**Type** float array of 9 items in [-inf, inf], default (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)**shear**

Shear Stiffness

**Type** float in [0, 1], default 0.0**speed**

Tweak timing for physics to control frequency and speed

**Type** float in [0.01, 100], default 0.0**spring\_length**

Alter spring length to shrink/blow up (unit %) 0 to disable

**Type** int in [0, 200], default 0**step\_max**

Maximal # solver steps/frame

**Type** int in [0, 30000], default 0**step\_min**

Minimal # solver steps/frame

**Type** int in [0, 30000], default 0**use\_auto\_step**

Use velocities for automagic step sizes

**Type** boolean, default False

**use\_diagnose**

Turn on SB diagnose console prints

**Type** boolean, default False

**use\_edge\_collision**

Edges collide too

**Type** boolean, default False

**use\_edges**

Use Edges as springs

**Type** boolean, default False

**use\_estimate\_matrix**

Estimate matrix... split to COM, ROT, SCALE

**Type** boolean, default False

**use\_face\_collision**

Faces collide too, can be very slow

**Type** boolean, default False

**use\_goal**

Define forces for vertices to stick to animated position

**Type** boolean, default False

**use\_self\_collision**

Enable naive vertex ball self collision

**Type** boolean, default False

**use\_stiff\_quads**

Add diagonal springs on 4-gons

**Type** boolean, default False

**vertex\_group\_goal**

Control point weight values

**Type** string, default ""

**vertex\_group\_mass**

Control point mass values

**Type** string, default ""

**vertex\_group\_spring**

Control point spring strength values

**Type** string, default ""

### Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Object.soft_body`
- `SoftBodyModifier.settings`

## 2.4.509 SolidifyModifier(Modifier)

base classes — `bpy_struct`, `Modifier`

**class** `bpy.types.SolidifyModifier` (*Modifier*)

Create a solid skin by extruding, compensating for sharp angles

### **edge\_crease\_inner**

Assign a crease to inner edges

**Type** float in [0, 1], default 0.0

### **edge\_crease\_outer**

Assign a crease to outer edges

**Type** float in [0, 1], default 0.0

### **edge\_crease\_rim**

Assign a crease to the edges making up the rim

**Type** float in [0, 1], default 0.0

### **invert\_vertex\_group**

Invert the vertex group influence

**Type** boolean, default False

### **material\_offset**

Offset material index of generated faces

**Type** int in [-32768, 32767], default 0

### **material\_offset\_rim**

Offset material index of generated rim faces

**Type** int in [-32768, 32767], default 0

**offset**

Offset the thickness from the center

**Type** float in [-inf, inf], default 0.0

**thickness**

Thickness of the shell

**Type** float in [-inf, inf], default 0.0

**thickness\_vertex\_group**

Thickness factor to use for zero vertex group influence

**Type** float in [0, 1], default 0.0

**use\_even\_offset**

Maintain thickness by adjusting for sharp corners (slow, disable when not needed)

**Type** boolean, default False

**use\_quality\_normals**

Calculate normals which result in more even thickness (slow, disable when not needed)

**Type** boolean, default False

**use\_rim**

Create edge loops between the inner and outer surfaces on face edges (slow, disable when not needed)

**Type** boolean, default False

**vertex\_group**

Vertex group name

**Type** string, default ""

### Inherited Properties

- `bpy_struct.id_data`
- `Modifier.name`
- `Modifier.use_apply_on_spline`
- `Modifier.show_in_editmode`
- `Modifier.show_expanded`
- `Modifier.show_on_cage`
- `Modifier.show_viewport`
- `Modifier.show_render`
- `Modifier.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`



- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.510 Sound(ID)

base classes — `bpy_struct`, `ID`

**class** `bpy.types.Sound` (*ID*)

Sound datablock referencing an external or packed sound file

**filepath**

Sound sample file used by this Sound datablock

**Type** string, default ""

**packed\_file**

**Type** `PackedFile`, (readonly)

**use\_memory\_cache**

The sound file is decoded and loaded into RAM

**Type** boolean, default False

**use\_mono**

If the file contains multiple audio channels they are rendered to a single one

**Type** boolean, default False

**factory**

The `aud.Factory` object of the sound. (readonly)

#### Inherited Properties

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`

- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`

## References

- `BlendData.sounds`
- `SoundActuator.sound`
- `SoundSequence.sound`
- `Speaker.sound`

## 2.4.511 SoundActuator(Actuator)

base classes — `bpy_struct`, `Actuator`

**class** `bpy.types.SoundActuator` (*Actuator*)  
Sound file

**cone\_inner\_angle\_3d**

The angle of the inner cone

**Type** float in [-inf, inf], default 0.0

**cone\_outer\_angle\_3d**

The angle of the outer cone

**Type** float in [-inf, inf], default 0.0

**cone\_outer\_gain\_3d**

The gain outside the outer cone. The gain in the outer cone will be interpolated between this value and the normal gain in the inner cone

**Type** float in [-inf, inf], default 0.0

**distance\_3d\_max**

The maximum distance at which you can hear the sound

**Type** float in [-inf, inf], default 0.0

**distance\_3d\_reference**

The distance where the sound has a gain of 1.0

**Type** float in [-inf, inf], default 0.0

**gain\_3d\_max**

The maximum gain of the sound, no matter how near it is

**Type** float in [-inf, inf], default 0.0

**gain\_3d\_min**

The minimum gain of the sound, no matter how far it is away

**Type** float in [-inf, inf], default 0.0

**mode**

**Type** enum in ['PLAYSTOP', 'PLAYEND', 'LOOPSTOP', 'LOOPEND', 'LOOPBIDIRECTIONAL', 'LOOPBIDIRECTIONALSTOP'], default 'PLAYSTOP'

**pitch**

Sets the pitch of the sound

**Type** float in [-inf, inf], default 0.0

**rolloff\_factor\_3d**

The influence factor on volume depending on distance

**Type** float in [-inf, inf], default 0.0

**sound**

**Type** Sound

**use\_sound\_3d**

Enable/Disable 3D Sound

**Type** boolean, default False

**volume**

Sets the initial volume of the sound

**Type** float in [0, 2], default 0.0

**Inherited Properties**

- `bpy_struct.id_data`
- `Actuator.name`
- `Actuator.show_expanded`
- `Actuator.pin`
- `Actuator.type`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`

- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Actuator.link`
- `Actuator.unlink`

### 2.4.512 SoundSequence(Sequence)

base classes — `bpy_struct`, `Sequence`

**class** `bpy.types.SoundSequence` (*Sequence*)

Sequence strip defining a sound to be played over a period of time

**animation\_offset\_end**

Animation end offset (trim end)

**Type** `int` in `[0, inf]`, default `0`

**animation\_offset\_start**

Animation start offset (trim start)

**Type** `int` in `[0, inf]`, default `0`

**filepath**

**Type** `string`, default `""`

**pan**

Playback panning of the sound (only for Mono sources)

**Type** `float` in `[-2, 2]`, default `0.0`

**pitch**

Playback pitch of the sound

**Type** `float` in `[0.1, 10]`, default `0.0`

**sound**

Sound datablock used by this sequence

**Type** `Sound`, (readonly)

**volume**

Playback volume of the sound

**Type** `float` in `[0, 100]`, default `0.0`

#### Inherited Properties

- `bpy_struct.id_data`
- `Sequence.name`
- `Sequence.blend_type`
- `Sequence.blend_alpha`
- `Sequence.channel`
- `Sequence.waveform`
- `Sequence.effect_fader`
- `Sequence.frame_final_end`
- `Sequence.frame_offset_end`

- `Sequence.frame_still_end`
- `Sequence.input_1`
- `Sequence.input_2`
- `Sequence.input_3`
- `Sequence.select_left_handle`
- `Sequence.frame_final_duration`
- `Sequence.frame_duration`
- `Sequence.lock`
- `Sequence.mute`
- `Sequence.select_right_handle`
- `Sequence.select`
- `Sequence.speed_factor`
- `Sequence.frame_start`
- `Sequence.frame_final_start`
- `Sequence.frame_offset_start`
- `Sequence.frame_still_start`
- `Sequence.type`
- `Sequence.use_default_fade`
- `Sequence.input_count`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Sequence.getStripElem`
- `Sequence.swap`

### 2.4.513 Space(`bpy_struct`)

base class — `bpy_struct`

subclasses — `SpaceUserPreferences`, `SpaceOutliner`, `SpaceImageEditor`, `SpaceSequenceEditor`, `SpaceProperties`, `SpaceGraphEditor`, `SpaceLogicEditor`, `SpaceInfo`, `SpaceConsole`, `SpaceNodeEditor`, `SpaceView3D`, `SpaceDopeSheetEditor`, `SpaceTextEditor`, `SpaceFileBrowser`, `SpaceNLA`, `SpaceTimeline`

**class** `bpy.types.Space` (`bpy_struct`)  
 Space data for a screen area

**type**

Space data type

**Type** enum in ['EMPTY', 'VIEW\_3D', 'GRAPH\_EDITOR', 'OUTLINER', 'PROPERTIES', 'FILE\_BROWSER', 'IMAGE\_EDITOR', 'INFO', 'SEQUENCE\_EDITOR', 'TEXT\_EDITOR', 'DOPE SHEET\_EDITOR', 'NLA\_EDITOR', 'TIMELINE', 'NODE\_EDITOR', 'LOGIC\_EDITOR', 'CONSOLE', 'USER\_PREFERENCES'], default 'EMPTY', (readonly)

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `Area.spaces`
- `AreaSpaces.active`
- `Context.space_data`

**2.4.514 SpaceConsole(Space)**

base classes — `bpy_struct`, `Space`

**class** `bpy.types.SpaceConsole` (*Space*)

Interactive python console

**font\_size**

Font size to use for displaying the text

**Type** int in [8, 32], default 0

**history**

Command history

**Type** `bpy_prop_collection` of `ConsoleLine`, (readonly)

### **language**

Command line prompt language

**Type** `string`, default `""`

### **prompt**

Command line prompt

**Type** `string`, default `""`

### **scrollback**

Command output

**Type** `bpy_prop_collection` of `ConsoleLine`, (readonly)

### **select\_end**

**Type** `int` in `[0, inf]`, default `0`

### **select\_start**

**Type** `int` in `[0, inf]`, default `0`

## Inherited Properties

- `bpy_struct.id_data`
- `Space.type`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.515 SpaceDopeSheetEditor(Space)

base classes — `bpy_struct`, `Space`

**class** `bpy.types.SpaceDopeSheetEditor` (*Space*)  
DopeSheet space data

**action**

Action displayed and edited in this space

**Type** `Action`

**auto\_snap**

Automatic time snapping settings for transformations

- NONE No Auto-Snap.
- STEP Time Step, Snap to 1.0 frame/second intervals.
- FRAME Nearest Frame, Snap to actual frames/seconds (nla-action time).
- MARKER Nearest Marker, Snap to nearest marker.

**Type** enum in ['NONE', 'STEP', 'FRAME', 'MARKER'], default 'NONE'

**dopesheet**

Settings for filtering animation data

**Type** `DopeSheet`, (readonly)

**mode**

Editing context being displayed

- DOPESHEET DopeSheet, DopeSheet Editor.
- ACTION Action Editor, Action Editor.
- SHAPEKEY ShapeKey Editor, ShapeKey Editor.
- GPENCIL Grease Pencil, Grease Pencil.

**Type** enum in ['DOPESHEET', 'ACTION', 'SHAPEKEY', 'GPENCIL'], default 'ACTION'

**show\_frame\_indicator**

Show frame number beside the current frame indicator line

**Type** boolean, default False

**show\_pose\_markers**

Show markers belonging to the active action instead of Scene markers (Action and Shape Key Editors only)

**Type** boolean, default False

**show\_seconds**

Show timing in seconds not frames

**Type** boolean, default False, (readonly)

**show\_sliders**

Show sliders beside F-Curve channels

**Type** boolean, default False

**use\_auto\_merge\_keyframes**

Automatically merge nearby keyframes

**Type** boolean, default False

**use\_marker\_sync**

Sync Markers with keyframe edits



**Type** boolean, default False

**use\_realtime\_update**

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

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `Space.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.516 SpaceFileBrowser(Space)

base classes — `bpy_struct`, `Space`

**class** `bpy.types.SpaceFileBrowser` (*Space*)

File browser space data

**operator**

**Type** `Operator`, (readonly)

**params**

Parameters and Settings for the Filebrowser

**Type** `FileSelectParams`, (readonly)

### Inherited Properties

- `bpy_struct.id_data`
- `Space.type`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.517 SpaceGraphEditor(Space)

base classes — `bpy_struct`, `Space`

**class** `bpy.types.SpaceGraphEditor` (*Space*)

Graph Editor space data

**auto\_snap**

Automatic time snapping settings for transformations

- **NONE** No Auto-Snap.
- **STEP** Time Step, Snap to 1.0 frame/second intervals.
- **FRAME** Nearest Frame, Snap to actual frames/seconds (nla-action time).
- **MARKER** Nearest Marker, Snap to nearest marker.

**Type** enum in ['NONE', 'STEP', 'FRAME', 'MARKER'], default 'NONE'

**cursor\_position\_y**

Graph Editor 2D-Value cursor - Y-Value component

**Type** float in [-inf, inf], default 0.0

**dopesheet**

Settings for filtering animation data

**Type** `DopeSheet`, (readonly)

**has\_ghost\_curves**

Graph Editor instance has some ghost curves stored

**Type** boolean, default False

**mode**

Editing context being displayed

- **FCURVES** F-Curve Editor, Edit animation/keyframes displayed as 2D curves.
- **DRIVERS** Drivers, Edit drivers.

**Type** enum in ['FCURVES', 'DRIVERS'], default 'FCURVES'

**pivot\_point**

Pivot center for rotation/scaling

**Type** enum in ['BOUNDING\_BOX\_CENTER', 'CURSOR', 'INDIVIDUAL\_ORIGINS'], default 'BOUNDING\_BOX\_CENTER'

**show\_cursor**

Show 2D cursor

**Type** boolean, default False

**show\_frame\_indicator**

Show frame number beside the current frame indicator line

**Type** boolean, default False

**show\_handles**

Show handles of Bezier control points

**Type** boolean, default False

**show\_seconds**

Show timing in seconds not frames

**Type** boolean, default False, (readonly)

**show\_sliders**

Show sliders beside F-Curve channels

**Type** boolean, default False

**use\_auto\_merge\_keyframes**

Automatically merge nearby keyframes

**Type** boolean, default False

**use\_beauty\_drawing**

Draw F-Curves using Anti-Aliasing and other fancy effects (disable for better performance)

**Type** boolean, default False

**use\_only\_selected\_curves\_handles**

Only keyframes of selected F-Curves are visible and editable

**Type** boolean, default False

**use\_only\_selected\_keyframe\_handles**

Only show and edit handles of selected keyframes

**Type** boolean, default False

**use\_realtime\_update**

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

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `Space.type`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.518 SpaceImageEditor(Space)

base classes — `bpy_struct`, `Space`

**class** `bpy.types.SpaceImageEditor` (*Space*)

Image and UV editor space data

#### **curve**

Color curve mapping to use for displaying the image

**Type** `CurveMapping`, (readonly)

#### **draw\_channels**

Channels of the image to draw

- `COLOR` Color, Draw image with RGB colors.
- `COLOR_ALPHA` Color and Alpha, Draw image with RGB colors and alpha transparency.
- `ALPHA` Alpha, Draw alpha transparency channel.
- `Z_BUFFER` Z-Buffer, Draw Z-buffer associated with image (mapped from camera clip start to end).

**Type** enum in [`'COLOR'`, `'COLOR_ALPHA'`, `'ALPHA'`, `'Z_BUFFER'`], default `'COLOR'`

#### **grease\_pencil**

Grease pencil data for this space

**Type** `GreasePencil`

#### **image**

Image displayed and edited in this space

**Type** `Image`

#### **image\_user**

Parameters defining which layer, pass and frame of the image is displayed

**Type** `ImageUser`, (readonly, never None)

**sample\_histogram**

Sampled colors along line

**Type** `Histogram`, (readonly)

**scopes**

Scopes to visualize image statistics

**Type** `Scopes`, (readonly)

**show\_paint**

Show paint related properties

**Type** boolean, default `False`, (readonly)

**show\_render**

Show render related properties

**Type** boolean, default `False`, (readonly)

**show\_repeat**

Draw the image repeated outside of the main view

**Type** boolean, default `False`

**show\_uvedit**

Show UV editing related properties

**Type** boolean, default `False`, (readonly)

**use\_grease\_pencil**

Display and edit the grease pencil freehand annotations overlay

**Type** boolean, default `False`

**use\_image\_paint**

Enable image painting mode

**Type** boolean, default `False`

**use\_image\_pin**

Display current image regardless of object selection

**Type** boolean, default `False`

**use\_realtime\_update**

Update other affected window spaces automatically to reflect changes during interactive operations such as transform

**Type** boolean, default `False`

**uv\_editor**

UV editor settings

**Type** `SpaceUVEditor`, (readonly, never `None`)

**zoom**

Zoom factor

**Type** float array of 2 items in `[-inf, inf]`, default `(0.0, 0.0)`, (readonly)

**Inherited Properties**

- `bpy_struct.id_data`

- `Space.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.519 SpaceInfo(Space)

base classes — `bpy_struct`, `Space`

**class** `bpy.types.SpaceInfo` (*Space*)

Info space data

**show\_report\_debug**

Display debug reporting info

**Type** boolean, default False

**show\_report\_error**

Display error text

**Type** boolean, default False

**show\_report\_info**

Display general information

**Type** boolean, default False

**show\_report\_operator**

Display the operator log

**Type** boolean, default False

**show\_report\_warning**

Display warnings

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `Space.type`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.520 SpaceLogicEditor(Space)

base classes — `bpy_struct`, `Space`

**class** `bpy.types.SpaceLogicEditor` (*Space*)

Logic editor space data

**show\_actuators\_active\_object**

Show actuators of active object

**Type** boolean, default False

**show\_actuators\_active\_states**

Show only actuators connected to active states

**Type** boolean, default False

**show\_actuators\_linked\_controller**

Show linked objects to the actuator

**Type** boolean, default False

**show\_actuators\_selected\_objects**

Show actuators of all selected objects

**Type** boolean, default False

**show\_controllers\_active\_object**

Show controllers of active object

**Type** boolean, default False

**show\_controllers\_linked\_controller**

Show linked objects to sensor/actuator

**Type** boolean, default False

**show\_controllers\_selected\_objects**

Show controllers of all selected objects

**Type** boolean, default False

**show\_sensors\_active\_object**

Show sensors of active object

**Type** boolean, default False

**show\_sensors\_active\_states**

Show only sensors connected to active states

**Type** boolean, default False

**show\_sensors\_linked\_controller**

Show linked objects to the controller

**Type** boolean, default False

**show\_sensors\_selected\_objects**

Show sensors of all selected objects

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `Space.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.521 SpaceNLA(Space)

base classes — `bpy_struct`, `Space`

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

NLA editor space data

**auto\_snap**

Automatic time snapping settings for transformations

- **NONE** No Auto-Snap.
- **STEP** Time Step, Snap to 1.0 frame/second intervals.



- FRAME Nearest Frame, Snap to actual frames/seconds (nla-action time).
- MARKER Nearest Marker, Snap to nearest marker.

**Type** enum in ['NONE', 'STEP', 'FRAME', 'MARKER'], default 'NONE'

### dopesheet

Settings for filtering animation data

**Type** `DopeSheet`, (readonly)

### show\_frame\_indicator

Show frame number beside the current frame indicator line

**Type** boolean, default False

### show\_seconds

Show timing in seconds not frames

**Type** boolean, default False, (readonly)

### 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

## Inherited Properties

- `bpy_struct.id_data`
- `Space.type`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.522 SpaceNodeEditor(Space)

base classes — `bpy_struct`, `Space`

**class** `bpy.types.SpaceNodeEditor` (*Space*)

Node editor space data

**backdrop\_channels**

Channels of the image to draw

- `COLOR` Color, Draw image with RGB colors.
- `COLOR_ALPHA` Color and Alpha, Draw image with RGB colors and alpha transparency.
- `ALPHA` Alpha, Draw alpha transparency channel.

**Type** enum in [`'COLOR'`, `'COLOR_ALPHA'`, `'ALPHA'`], default `'COLOR'`

**backdrop\_x**

Backdrop X offset

**Type** float in `[-inf, inf]`, default `0.0`

**backdrop\_y**

Backdrop Y offset

**Type** float in `[-inf, inf]`, default `0.0`

**backdrop\_zoom**

Backdrop zoom factor

**Type** float in `[0.01, inf]`, default `1.0`

**id**

Datablock whose nodes are being edited

**Type** `ID`, (readonly)

**id\_from**

Datablock from which the edited datablock is linked

**Type** `ID`, (readonly)

**node\_tree**

Node tree being displayed and edited

**Type** `NodeTree`

**shader\_type**

Type of data to take shader from

- `OBJECT` Object, Edit shader nodes from Object.
- `WORLD` World, Edit shader nodes from World.

**Type** enum in [`'OBJECT'`, `'WORLD'`], default `'OBJECT'`

**show\_backdrop**

Use active Viewer Node output as backdrop for compositing nodes

**Type** boolean, default `False`

**texture\_type**

Type of data to take texture from

- OBJECT Object, Edit texture nodes from Object.
- WORLD World, Edit texture nodes from World.
- BRUSH Brush, Edit texture nodes from Brush.

**Type** enum in ['OBJECT', 'WORLD', 'BRUSH'], default 'OBJECT'

#### **tree\_type**

Node tree type to display and edit

- SHADER Shader, Shader nodes.
- TEXTURE Texture, Texture nodes.
- COMPOSITING Compositing, Compositing nodes.

**Type** enum in ['SHADER', 'TEXTURE', 'COMPOSITING'], default 'SHADER'

#### **use\_auto\_render**

Re-render and composite changed layers on 3D edits

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `Space.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.523 SpaceOutliner(Space)

base classes — `bpy_struct`, `Space`

**class** `bpy.types.SpaceOutliner` (*Space*)  
Outliner space data

### **display\_mode**

Type of information to display

- **ALL\_SCENES** All Scenes, Display datablocks in all scenes.
- **CURRENT\_SCENE** Current Scene, Display datablocks in current scene.
- **VISIBLE\_LAYERS** Visible Layers, Display datablocks in visible layers.
- **SELECTED** Selected, Display datablocks of selected objects.
- **ACTIVE** Active, Display datablocks of active object.
- **SAME\_TYPES** Same Types, Display datablocks of all objects of same type as selected object.
- **GROUPS** Groups, Display groups and their datablocks.
- **LIBRARIES** Libraries, Display libraries.
- **SEQUENCE** Sequence, Display sequence datablocks.
- **DATABLOCKS** Datablocks, Display raw datablocks.
- **USER\_PREFERENCES** User Preferences, Display the user preference datablocks.
- **KEYMAPS** Key Maps, Display keymap datablocks.

**Type** enum in ['ALL\_SCENES', 'CURRENT\_SCENE', 'VISIBLE\_LAYERS', 'SELECTED', 'ACTIVE', 'SAME\_TYPES', 'GROUPS', 'LIBRARIES', 'SEQUENCE', 'DATABLOCKS', 'USER\_PREFERENCES', 'KEYMAPS'], default 'ALL\_SCENES'

### **filter\_text**

Live search filtering string

**Type** string, default ""

### **show\_restrict\_columns**

Show column

**Type** boolean, default False

### **use\_filter\_case\_sensitive**

Only use case sensitive matches of search string

**Type** boolean, default False

### **use\_filter\_complete**

Only use complete matches of search string

**Type** boolean, default False

## **Inherited Properties**

- `bpy_struct.id_data`
- `Space.type`

## **Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`

- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.524 SpaceProperties(Space)

base classes — `bpy_struct`, `Space`

**class** `bpy.types.SpaceProperties` (*Space*)  
Properties space data

### **align**

Arrangement of the panels

**Type** enum in ['HORIZONTAL', 'VERTICAL'], default 'HORIZONTAL'

### **context**

Type of active data to display and edit

- `SCENE` Scene, Scene.
- `RENDER` Render, Render.
- `WORLD` World, World.
- `OBJECT` Object, Object.
- `CONSTRAINT` Constraints, Constraints.
- `MODIFIER` Modifiers, Modifiers.
- `DATA` Data, Data.
- `BONE` Bone, Bone.
- `BONE_CONSTRAINT` Bone Constraints, Bone Constraints.
- `MATERIAL` Material, Material.
- `TEXTURE` Texture, Texture.
- `PARTICLES` Particles, Particle.
- `PHYSICS` Physics, Physics.

**Type** enum in ['SCENE', 'RENDER', 'WORLD', 'OBJECT', 'CONSTRAINT', 'MODIFIER', 'DATA', 'BONE', 'BONE\_CONSTRAINT', 'MATERIAL', 'TEXTURE', 'PARTICLES', 'PHYSICS'], default 'RENDER'

### **pin\_id**

**Type** ID

**texture\_context**

Type of texture data to display and edit

- MATERIAL Material, Material.

**Type** enum in ['MATERIAL'], default 'MATERIAL'

**use\_pin\_id**

Use the pinned context

**Type** boolean, default False

**Inherited Properties**

- bpy\_struct.id\_data
- Space.type

**Inherited Functions**

- bpy\_struct.as\_pointer
- bpy\_struct.callback\_add
- bpy\_struct.callback\_remove
- bpy\_struct.driver\_add
- bpy\_struct.driver\_remove
- bpy\_struct.get
- bpy\_struct.is\_property\_hidden
- bpy\_struct.is\_property\_set
- bpy\_struct.items
- bpy\_struct.keyframe\_delete
- bpy\_struct.keyframe\_insert
- bpy\_struct.keys
- bpy\_struct.path\_from\_id
- bpy\_struct.path\_resolve
- bpy\_struct.type\_recast
- bpy\_struct.values

**2.4.525 SpaceSequenceEditor(Space)**

base classes — bpy\_struct, Space

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

Sequence editor space data

**display\_channel**

The channel number shown in the image preview. 0 is the result of all strips combined

**Type** int in [-5, 32], default 0

**display\_mode**

View mode to use for displaying sequencer output

**Type** enum in ['IMAGE', 'WAVEFORM', 'VECTOR\_SCOPE', 'HISTOGRAM'], default 'IMAGE'

**draw\_overexposed**

Show overexposed areas with zebra stripes

**Type** int in [0, 110], default 0

**grease\_pencil**

Grease pencil data for this space

**Type** `UnknownType`, (readonly)

**offset\_x**

Offset image horizontally from the view center

**Type** float in [-inf, inf], default 0.0

**offset\_y**

Offset image vertically from the view center

**Type** float in [-inf, inf], default 0.0

**proxy\_render\_size**

Draw preview using full resolution or different proxy resolutions

**Type** enum in ['NONE', 'SCENE', 'PROXY\_25', 'PROXY\_50', 'PROXY\_75', 'PROXY\_100', 'FULL'], default 'SCENE'

**show\_frame\_indicator**

Show frame number beside the current frame indicator line

**Type** boolean, default False

**show\_frames**

Draw frames rather than seconds

**Type** boolean, default False

**show\_safe\_margin**

Draw title safe margins in preview

**Type** boolean, default False

**show\_separate\_color**

Separate color channels in preview

**Type** boolean, default False

**use\_grease\_pencil**

Display and edit the grease pencil freehand annotations overlay

**Type** boolean, default False

**use\_marker\_sync**

Transform markers as well as strips

**Type** boolean, default False

**view\_type**

Type of the Sequencer view (sequencer, preview or both)

**Type** enum in ['SEQUENCER', 'PREVIEW', 'SEQUENCER\_PREVIEW'], default 'SEQUENCER'

**zoom**

Display zoom level

**Type** float in [-inf, inf], default 0.0

### Inherited Properties

- `bpy_struct.id_data`
- `Space.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.526 SpaceTextEditor(Space)

base classes — `bpy_struct`, `Space`

**class** `bpy.types.SpaceTextEditor` (*Space*)

Text editor space data

**find\_text**

Text to search for with the find tool

**Type** string, default ""

**font\_size**

Font size to use for displaying the text

**Type** int in [8, 32], default 0

**margin\_column**

Column number to show right margin at

**Type** int in [0, 1024], default 0

**replace\_text**

Text to replace selected text with using the replace tool

**Type** string, default ""

**show\_line\_highlight**

Highlight the current line

**Type** boolean, default False

**show\_line\_numbers**

Show line numbers next to the text



**Type** boolean, default False

**show\_margin**

Show right margin

**Type** boolean, default False

**show\_syntax\_highlight**

Syntax highlight for scripting

**Type** boolean, default False

**show\_word\_wrap**

Wrap words if there is not enough horizontal space

**Type** boolean, default False

**tab\_width**

Number of spaces to display tabs with

**Type** int in [2, 8], default 0

**text**

Text displayed and edited in this space

**Type** `Text`

**use\_find\_all**

Search in all text datablocks, instead of only the active one

**Type** boolean, default False

**use\_find\_wrap**

Search again from the start of the file when reaching the end

**Type** boolean, default False

**use\_live\_edit**

Run python while editing

**Type** boolean, default False

**use\_match\_case**

Search string is sensitive to uppercase and lowercase letters

**Type** boolean, default False

**use\_overwrite**

Overwrite characters when typing rather than inserting them

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `Space.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`

- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.527 SpaceTimeline(Space)

base classes — `bpy_struct`, `Space`

**class** `bpy.types.SpaceTimeline` (*Space*)

Timeline editor space data

**cache\_cloth**

Show the active object's cloth point cache

**Type** boolean, default False

**cache\_particles**

Show the active object's particle point cache

**Type** boolean, default False

**cache\_smoke**

Show the active object's smoke cache

**Type** boolean, default False

**cache\_softbody**

Show the active object's softbody point cache

**Type** boolean, default False

**show\_cache**

Show the status of cached frames in the timeline

**Type** boolean, default False

**show\_frame\_indicator**

Show frame number beside the current frame indicator line

**Type** boolean, default False

**show\_only\_selected**

Show keyframes for active Object and/or its selected bones only

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `Space.type`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.528 SpaceUVEditor(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.SpaceUVEditor` (*bpy\_struct*)

UV editor data for the image editor space

### **cursor\_location**

2D cursor location for this view

**Type** float array of 2 items in [-inf, inf], default (0.0, 0.0)

### **draw\_stretch\_type**

Type of stretch to draw

- **ANGLE** Angle, Angular distortion between UV and 3D angles.
- **AREA** Area, Area distortion between UV and 3D faces.

**Type** enum in ['ANGLE', 'AREA'], default 'ANGLE'

### **edge\_draw\_type**

Draw type for drawing UV edges

- **OUTLINE** Outline, Draw white edges with black outline.
- **DASH** Dash, Draw dashed black-white edges.
- **BLACK** Black, Draw black edges.
- **WHITE** White, Draw white edges.

**Type** enum in ['OUTLINE', 'DASH', 'BLACK', 'WHITE'], default 'OUTLINE'

### **lock\_bounds**

Constraint to stay within the image bounds while editing

**Type** boolean, default False

**pivot\_point**

Rotation/Scaling Pivot

**Type** enum in ['CENTER', 'MEDIAN', 'CURSOR'], default 'CENTER'

**show\_faces**

Draw faces over the image

**Type** boolean, default False

**show\_modified\_edges**

Draw edges after modifiers are applied

**Type** boolean, default False

**show\_normalized\_coords**

Display UV coordinates from 0.0 to 1.0 rather than in pixels

**Type** boolean, default False

**show\_other\_objects**

Draw other selected objects that share the same image

**Type** boolean, default False

**show\_smooth\_edges**

Draw UV edges anti-aliased

**Type** boolean, default False

**show\_stretch**

Draw faces colored according to the difference in shape between UVs and their 3D coordinates (blue for low distortion, red for high distortion)

**Type** boolean, default False

**sticky\_select\_mode**

Automatically select also UVs sharing the same vertex as the ones being selected

- **DISABLED** Disabled, Sticky vertex selection disabled.
- **SHARED\_LOCATION** Shared Location, Select UVs that are at the same location and share a mesh vertex.
- **SHARED\_VERTEX** Shared Vertex, Select UVs that share mesh vertex, irrespective if they are in the same location.

**Type** enum in ['DISABLED', 'SHARED\_LOCATION', 'SHARED\_VERTEX'], default 'SHARED\_LOCATION'

**use\_live\_unwrap**

Continuously unwrap the selected UV island while transforming pinned vertices

**Type** boolean, default False

**use\_snap\_to\_pixels**

Snap UVs to pixel locations while editing

**Type** boolean, default False

**Inherited Properties**

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `SpaceImageEditor.uv_editor`

## 2.4.529 SpaceUserPreferences(Space)

base classes — `bpy_struct`, `Space`

**class** `bpy.types.SpaceUserPreferences` (*Space*)

User preferences space data

**filter\_text**

Search term for filtering in the UI

**Type** string, default ""

### Inherited Properties

- `bpy_struct.id_data`
- `Space.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`

- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.530 SpaceView3D(Space)

base classes — `bpy_struct`, `Space`

**class** `bpy.types.SpaceView3D` (*Space*)

3D View space data

**background\_images**

List of background images

**Type** `bpy_prop_collection` of `BackgroundImage`, (readonly)

**camera**

Active camera used in this view (when unlocked from the scene's active camera)

**Type** `Object`

**clip\_end**

3D View far clipping distance

**Type** float in [1, inf], default 0.0

**clip\_start**

3D View near clipping distance

**Type** float in [0.001, inf], default 0.0

**current\_orientation**

Current transformation orientation

**Type** `TransformOrientation`, (readonly)

**cursor\_location**

3D cursor location for this view (dependent on local view setting)

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**grid\_lines**

Number of grid lines to display in perspective view

**Type** int in [0, 1024], default 0

**grid\_scale**

Distance between 3D View grid lines

**Type** float in [0, inf], default 0.0

**grid\_subdivisions**

Number of subdivisions between grid lines

**Type** int in [1, 1024], default 0

**layers**

Layers visible in this 3D View

**Type** boolean array of 20 items, default (False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False)

**layers\_used**

Layers that contain something

**Type** boolean array of 20 items, default (False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False, False), (readonly)

**lens**

Lens angle (mm) in perspective view

**Type** float in [1, 250], default 0.0

**local\_view**

Display an isolated sub-set of objects, apart from the scene visibility

**Type** `SpaceView3D`, (readonly)

**lock\_bone**

3D View center is locked to this bone's position

**Type** string, default ""

**lock\_camera**

Enable view navigation within the camera view

**Type** boolean, default False

**lock\_camera\_and\_layers**

Use the scene's active camera and layers in this view, rather than local layers

**Type** boolean, default False

**lock\_cursor**

3D View center is locked to the cursor's position

**Type** boolean, default False

**lock\_object**

3D View center is locked to this object's position

**Type** `Object`

**pivot\_point**

Pivot center for rotation/scaling

- `BOUNDING_BOX_CENTER` Bounding Box Center, Pivot around bounding box center of selected object(s).
- `CURSOR` 3D Cursor, Pivot around the 3D cursor.
- `INDIVIDUAL_ORIGINS` Individual Origins, Pivot around each object's own origin.
- `MEDIAN_POINT` Median Point, Pivot around the median point of selected objects.
- `ACTIVE_ELEMENT` Active Element, Pivot around active object.

**Type** enum in ['`BOUNDING_BOX_CENTER`', '`CURSOR`', '`INDIVIDUAL_ORIGINS`', '`MEDIAN_POINT`', '`ACTIVE_ELEMENT`'], default '`BOUNDING_BOX_CENTER`'

**region\_3d**

3D region in this space, in case of quad view the camera region

**Type** `RegionView3D`, (readonly)

**region\_quadview**

3D region that defines the quad view settings

**Type** `RegionView3D`, (readonly)

**show\_all\_objects\_origin**

Show the object origin center dot for all (selected and unselected) objects

**Type** boolean, default False

**show\_axis\_x**

Show the X axis line in perspective view

**Type** boolean, default False

**show\_axis\_y**

Show the Y axis line in perspective view

**Type** boolean, default False

**show\_axis\_z**

Show the Z axis line in perspective view

**Type** boolean, default False

**show\_background\_images**

Display reference images behind objects in the 3D View

**Type** boolean, default False

**show\_floor**

Show the ground plane grid in perspective view

**Type** boolean, default False

**show\_manipulator**

Use a 3D manipulator widget for controlling transforms

**Type** boolean, default False

**show\_only\_render**

Display only objects which will be rendered

**Type** boolean, default False

**show\_outline\_selected**

Show an outline highlight around selected objects in non-wireframe views

**Type** boolean, default False

**show\_relationship\_lines**

Show dashed lines indicating parent or constraint relationships

**Type** boolean, default False

**show\_textured\_solid**

Display face-assigned textures in solid view

**Type** boolean, default False

**transform\_orientation**

Transformation orientation

- GLOBAL Global, Align the transformation axes to world space.
- LOCAL Local, Align the transformation axes to the selected objects' local space.
- GIMBAL Gimbal, Align each axis to the Euler rotation axis as used for input.



- NORMAL** Normal, Align the transformation axes to average normal of selected elements (bone Y axis for pose mode).
- VIEW** View, Align the transformation axes to the window.
- CUSTOM** Custom, Use a custom transform orientation.

**Type** enum in ['GLOBAL', 'LOCAL', 'GIMBAL', 'NORMAL', 'VIEW', 'CUSTOM'], default 'GLOBAL'

#### **use\_manipulator\_rotate**

Use the manipulator for rotation transformations

**Type** boolean, default False

#### **use\_manipulator\_scale**

Use the manipulator for scale transformations

**Type** boolean, default False

#### **use\_manipulator\_translate**

Use the manipulator for movement transformations

**Type** boolean, default False

#### **use\_occlude\_geometry**

Limit selection to visible (clipped with depth buffer)

**Type** boolean, default False

#### **use\_pivot\_point\_align**

Manipulate center points (object and pose mode only)

**Type** boolean, default False

#### **viewport\_shade**

Method to display/shade objects in the 3D View

- BOUNDBOX** Bounding Box, Display the object's local bounding boxes only.
- WIREFRAME** Wireframe, Display the object as wire edges.
- SOLID** Solid, Display the object solid, lit with default OpenGL lights.
- TEXTURED** Textured, Display the object solid, with face-assigned textures.
- RENDERED** Rendered, Display render preview.

**Type** enum in ['BOUNDBOX', 'WIREFRAME', 'SOLID', 'TEXTURED', 'RENDERED'], default 'BOUNDBOX'

### **Inherited Properties**

- `bpy_struct.id_data`
- `Space.type`

### **Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`

- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `ObjectBase.layers_from_view`
- `SpaceView3D.local_view`

## 2.4.531 Speaker(ID)

base classes — `bpy_struct`, `ID`

**class** `bpy.types.Speaker` (*ID*)

Speaker datablock for 3D audio speaker objects

**animation\_data**

Animation data for this datablock

**Type** `AnimData`, (readonly)

**attenuation**

How strong the distance affects volume, depending on distance model

**Type** float in [0, inf], default 0.0

**cone\_angle\_inner**

Inner angle of the cone in degrees, inside the cone the volume is 100 %

**Type** float in [0, 360], default 0.0

**cone\_angle\_outer**

Outer angle of the cone in degrees, outside this cone the volume is the outer cone volume, between inner and outer cone the volume is interpolated

**Type** float in [0, 360], default 0.0

**cone\_volume\_outer**

Volume outside the outer cone

**Type** float in [0, 1], default 0.0

**distance\_max**

Maximum distance for volume calculation, no matter how far away the object is

**Type** float in [0, inf], default 0.0

**distance\_reference**

Reference distance at which volume is 100 %

**Type** float in [0, inf], default 0.0

**muted**

Mutes the speaker

**Type** boolean, default False

**pitch**

Playback pitch of the sound

**Type** float in [0.1, 10], default 0.0

**sound**

Sound datablock used by this speaker

**Type** [Sound](#)

**volume**

How loud the sound is

**Type** float in [0, 1], default 0.0

**volume\_max**

Maximum volume, no matter how near the object is

**Type** float in [0, 1], default 0.0

**volume\_min**

Minimum volume, no matter how far away the object is

**Type** float in [0, 1], default 0.0

### Inherited Properties

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`

- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`

## References

- `BlendData.speakers`
- `BlendDataSpeakers.new`
- `BlendDataSpeakers.remove`

## 2.4.532 SpeedControlSequence(EffectSequence)

base classes — `bpy_struct`, `Sequence`, `EffectSequence`

**class** `bpy.types.SpeedControlSequence` (*EffectSequence*)

Sequence strip to control the speed of other strips

### **multiply\_speed**

Multiply the resulting speed after the speed factor

**Type** float in [0, inf], default 0.0

### **scale\_to\_length**

Scale values from 0.0 to 1.0 to target sequence length

**Type** boolean, default False

### **use\_as\_speed**

Interpret the value as speed instead of a frame number

**Type** boolean, default False

### **use\_frame\_blend**

Blend two frames into the target for a smoother result

**Type** boolean, default False

## Inherited Properties

- `bpy_struct.id_data`
- `Sequence.name`
- `Sequence.blend_type`
- `Sequence.blend_alpha`
- `Sequence.channel`
- `Sequence.waveform`
- `Sequence.effect_fader`
- `Sequence.frame_final_end`
- `Sequence.frame_offset_end`
- `Sequence.frame_still_end`
- `Sequence.input_1`
- `Sequence.input_2`

- `Sequence.input_3`
- `Sequence.select_left_handle`
- `Sequence.frame_final_duration`
- `Sequence.frame_duration`
- `Sequence.lock`
- `Sequence.mute`
- `Sequence.select_right_handle`
- `Sequence.select`
- `Sequence.speed_factor`
- `Sequence.frame_start`
- `Sequence.frame_final_start`
- `Sequence.frame_offset_start`
- `Sequence.frame_still_start`
- `Sequence.type`
- `Sequence.use_default_fade`
- `Sequence.input_count`
- `EffectSequence.color_balance`
- `EffectSequence.use_float`
- `EffectSequence.crop`
- `EffectSequence.use_deinterlace`
- `EffectSequence.use_reverse_frames`
- `EffectSequence.use_flip_x`
- `EffectSequence.use_flip_y`
- `EffectSequence.color_multiply`
- `EffectSequence.use_premultiply`
- `EffectSequence.proxy`
- `EffectSequence.use_proxy_custom_directory`
- `EffectSequence.use_proxy_custom_file`
- `EffectSequence.color_saturation`
- `EffectSequence.strobe`
- `EffectSequence.ttransform`
- `EffectSequence.use_color_balance`
- `EffectSequence.use_crop`
- `EffectSequence.use_proxy`
- `EffectSequence.use_translation`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`

- `bpy_struct.values`
- `Sequence.getStripElem`
- `Sequence.swap`

### 2.4.533 Spline(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.Spline` (*bpy\_struct*)

Element of a curve, either NURBS, Bezier or Polyline or a character with text objects

**bezier\_points**

Collection of points for Bezier curves only

**Type** `SplineBezierPoints` `bpy_prop_collection` of `BezierSplinePoint`,  
(readonly)

**character\_index**

Location of this character in the text data (only for text curves)

**Type** `int` in `[0, inf]`, default `0`, (readonly)

**hide**

Hide this curve in editmode

**Type** `boolean`, default `False`

**material\_index**

**Type** `int` in `[0, 32767]`, default `0`

**order\_u**

NURBS order in the U direction (for splines and surfaces, higher values let points influence a greater area)

**Type** `int` in `[2, 6]`, default `0`

**order\_v**

NURBS order in the V direction (for surfaces only, higher values let points influence a greater area)

**Type** `int` in `[2, 6]`, default `0`

**point\_count\_u**

Total number points for the curve or surface in the U direction

**Type** `int` in `[0, 32767]`, default `0`, (readonly)

**point\_count\_v**

Total number points for the surface on the V direction

**Type** `int` in `[0, 32767]`, default `0`, (readonly)

**points**

Collection of points that make up this poly or nurbs spline

**Type** `SplinePoints` `bpy_prop_collection` of `SplinePoint`, (readonly)

**radius\_interpolation**

The type of radius interpolation for Bezier curves

**Type** `enum` in `['LINEAR', 'CARDINAL', 'BSPLINE', 'EASE']`, default `'LINEAR'`

**resolution\_u**

Curve or Surface subdivisions per segment

**Type** `int` in `[1, 32767]`, default `0`

**resolution\_v**

Surface subdivisions per segment

**Type** int in [1, 32767], default 0

**tilt\_interpolation**

The type of tilt interpolation for 3D, Bezier curves

**Type** enum in ['LINEAR', 'CARDINAL', 'BSPLINE', 'EASE'], default 'LINEAR'

**type**

The interpolation type for this curve element

**Type** enum in ['POLY', 'BEZIER', 'BSPLINE', 'CARDINAL', 'NURBS'], default 'POLY'

**use\_bezier\_u**

Make this nurbs curve or surface act like a Bezier spline in the U direction (Order U must be 3 or 4, Cyclic U must be disabled)

**Type** boolean, default False

**use\_bezier\_v**

Make this nurbs surface act like a Bezier spline in the V direction (Order V must be 3 or 4, Cyclic V must be disabled)

**Type** boolean, default False

**use\_cyclic\_u**

Make this curve or surface a closed loop in the U direction

**Type** boolean, default False

**use\_cyclic\_v**

Make this surface a closed loop in the V direction

**Type** boolean, default False

**use\_endpoint\_u**

Make this nurbs curve or surface meet the endpoints in the U direction (Cyclic U must be disabled)

**Type** boolean, default False

**use\_endpoint\_v**

Make this nurbs surface meet the endpoints in the V direction (Cyclic V must be disabled)

**Type** boolean, default False

**use\_smooth**

Smooth the normals of the surface or beveled curve

**Type** boolean, default False

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`

- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Curve.splines`
- `CurveSplines.new`
- `CurveSplines.remove`

## 2.4.534 SplineBezierPoints(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.SplineBezierPoints` (*bpy\_struct*)

Collection of spline bezirt points

**add** (*count=1*)

Add a number of points to this spline

**Parameters** `count` (*int in [-inf, inf], (optional)*) – Number, Number of points to add to the spline

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`



- `bpy_struct.values`

## References

- `Spline.bezier_points`

### 2.4.535 SplineIKConstraint(Constraint)

base classes — `bpy_struct`, `Constraint`

**class** `bpy.types.SplineIKConstraint` (*Constraint*)

Align 'n' bones along a curve

**chain\_count**

How many bones are included in the chain

**Type** int in [1, 255], default 0

**joint\_bindings**

(EXPERIENCED USERS ONLY) The relative positions of the joints along the chain as percentages

**Type** float array of 32 items in [0, 1], default (0.0, 0.0)

**target**

Curve that controls this relationship

**Type** `Object`

**use\_chain\_offset**

Offset the entire chain relative to the root joint

**Type** boolean, default False

**use\_curve\_radius**

Average radius of the endpoints is used to tweak the X and Z Scaling of the bones, on top of XZ Scale mode

**Type** boolean, default False

**use\_even\_divisions**

Ignore the relative lengths of the bones when fitting to the curve

**Type** boolean, default False

**use\_y\_stretch**

Stretch the Y axis of the bones to fit the curve

**Type** boolean, default False

**xz\_scale\_mode**

Method used for determining the scaling of the X and Z axes of the bones

- `NONE` None, Don't scale the X and Z axes (Default).
- `BONE_ORIGINAL` Bone Original, Use the original scaling of the bones.
- `VOLUME_PRESERVE` Volume Preservation, Scale of the X and Z axes is the inverse of the Y-Scale.

**Type** enum in ['NONE', 'BONE\_ORIGINAL', 'VOLUME\_PRESERVE'], default 'NONE'

### Inherited Properties

- `bpy_struct.id_data`
- `Constraint.name`
- `Constraint.active`
- `Constraint.mute`
- `Constraint.show_expanded`
- `Constraint.influence`
- `Constraint.error_location`
- `Constraint.owner_space`
- `Constraint.is_proxy_local`
- `Constraint.error_rotation`
- `Constraint.target_space`
- `Constraint.type`
- `Constraint.is_valid`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.536 SplinePoint(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.SplinePoint` (*bpy\_struct*)  
Spline point without handles

**co**

Point coordinates

**Type** float array of 4 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0, 0.0)`

**hide**

Visibility status

**Type** boolean, default `False`

**radius**

Radius for bevelling

**Type** float in [0, inf], default 0.0

**select**

Selection status

**Type** boolean, default False

**tilt**

Tilt in 3D View

**Type** float in [-inf, inf], default 0.0

**weight**

NURBS weight

**Type** float in [-inf, inf], default 0.0

**weight\_softbody**

Softbody goal weight

**Type** float in [0.01, 100], default 0.0

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Spline.points`

## 2.4.537 SplinePoints(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.SplinePoints` (*bpy\_struct*)  
Collection of spline points

**add** (*count=1*)

Add a number of points to this spline

**Parameters** **count** (*int in [1, inf], (optional)*) – Number, Number of points to add to the spline

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Spline.points`

## 2.4.538 SpotLamp(Lamp)

base classes — `bpy_struct`, `ID`, `Lamp`

**class** `bpy.types.SpotLamp` (*Lamp*)

Directional cone lamp

**compression\_threshold**

Deep shadow map compression threshold

**Type** float in [0, 1], default 0.0

**falloff\_curve**

Custom Lamp Falloff Curve

**Type** `CurveMapping`, (readonly)

**falloff\_type**

Intensity Decay with distance

**Type** enum in ['CONSTANT', 'INVERSE\_LINEAR', 'INVERSE\_SQUARE', 'CUSTOM\_CURVE', 'LINEAR\_QUADRATIC\_WEIGHTED'], default 'CONSTANT'

**halo\_intensity**

Brightness of the spotlight's halo cone

**Type** float in [-inf, inf], default 0.0

**halo\_step**

Volumetric halo sampling frequency

**Type** int in [0, 12], default 0

**linear\_attenuation**

Linear distance attenuation

**Type** float in [0, 1], default 0.0

**quadratic\_attenuation**

Quadratic distance attenuation

**Type** float in [0, 1], default 0.0

**shadow\_adaptive\_threshold**

Threshold for Adaptive Sampling (Raytraced shadows)

**Type** float in [0, 1], default 0.0

**shadow\_buffer\_bias**

Shadow buffer sampling bias

**Type** float in [0.001, 5], default 0.0

**shadow\_buffer\_clip\_end**

Shadow map clip end beyond which objects will not generate shadows

**Type** float in [0, 9999], default 0.0

**shadow\_buffer\_clip\_start**

Shadow map clip start: objects closer will not generate shadows

**Type** float in [0, 9999], default 0.0

**shadow\_buffer\_samples**

Number of shadow buffer samples

**Type** int in [1, 16], default 0

**shadow\_buffer\_size**

Resolution of the shadow buffer, higher values give crisper shadows but use more memory

**Type** int in [512, 10240], default 0

**shadow\_buffer\_soft**

Size of shadow buffer sampling area

**Type** float in [0, 100], default 0.0

**shadow\_buffer\_type**

Type of shadow buffer

- REGULAR Classical, Classic shadow buffer.
- HALFWAY Classic-Halfway, Regular buffer, averaging the closest and 2nd closest Z value to reducing bias artifacts.
- IRREGULAR Irregular, Irregular buffer produces sharp shadow always, but it doesn't show up for raytracing.

- DEEP Deep, Deep shadow buffer supports transparency and better filtering, at the cost of more memory usage and processing time.

**Type** enum in ['REGULAR', 'HALFWAY', 'IRREGULAR', 'DEEP'], default 'REGULAR'

**shadow\_color**

Color of shadows cast by the lamp

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**shadow\_filter\_type**

Type of shadow filter (Buffer Shadows)

- BOX Box, Apply the Box filter to shadow buffer samples.
- TENT Tent, Apply the Tent Filter to shadow buffer samples.
- GAUSS Gauss, Apply the Gauss filter to shadow buffer samples.

**Type** enum in ['BOX', 'TENT', 'GAUSS'], default 'BOX'

**shadow\_method**

Method to compute lamp shadow with

- NOSHADOW No Shadow.
- BUFFER\_SHADOW Buffer Shadow, Lets spotlight produce shadows using shadow buffer.
- RAY\_SHADOW Ray Shadow, Use ray tracing for shadow.

**Type** enum in ['NOSHADOW', 'BUFFER\_SHADOW', 'RAY\_SHADOW'], default 'NOSHADOW'

**shadow\_ray\_sample\_method**

Method for generating shadow samples: Adaptive QMC is fastest, Constant QMC is less noisy but slower

**Type** enum in ['ADAPTIVE\_QMC', 'CONSTANT\_QMC'], default 'ADAPTIVE\_QMC'

**shadow\_ray\_samples**

Amount of samples taken extra (samples x samples)

**Type** int in [1, 64], default 0

**shadow\_sample\_buffers**

Number of shadow buffers to render for better AA, this increases memory usage

- BUFFERS\_1 1, Only one buffer rendered.
- BUFFERS\_4 4, Renders 4 buffers for better AA, this quadruples memory usage.
- BUFFERS\_9 9, Renders 9 buffers for better AA, this uses nine times more memory.

**Type** enum in ['BUFFERS\_1', 'BUFFERS\_4', 'BUFFERS\_9'], default 'BUFFERS\_1'

**shadow\_soft\_size**

Light size for ray shadow sampling (Raytraced shadows)

**Type** float in [-inf, inf], default 0.0

**show\_cone**

Draw transparent cone in 3D view to visualize which objects are contained in it

**Type** boolean, default False

**spot\_blend**

The softness of the spotlight edge

**Type** float in [0, 1], default 0.0

**spot\_size**

Angle of the spotlight beam in degrees

**Type** float in [0.0174533, 3.14159], default 0.0

**use\_auto\_clip\_end**

Automatic calculation of clipping-end, based on visible vertices

**Type** boolean, default False

**use\_auto\_clip\_start**

Automatic calculation of clipping-start, based on visible vertices

**Type** boolean, default False

**use\_halo**

Renders spotlight with a volumetric halo

**Type** boolean, default False

**use\_only\_shadow**

Causes light to cast shadows only without illuminating objects

**Type** boolean, default False

**use\_shadow\_layer**

Causes only objects on the same layer to cast shadows

**Type** boolean, default False

**use\_sphere**

Sets light intensity to zero beyond lamp distance

**Type** boolean, default False

**use\_square**

Casts a square spot light shape

**Type** boolean, default False

**Inherited Properties**

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`
- `Lamp.active_texture`
- `Lamp.active_texture_index`
- `Lamp.animation_data`
- `Lamp.color`
- `Lamp.use_diffuse`
- `Lamp.distance`

- `Lamp.energy`
- `Lamp.use_own_layer`
- `Lamp.use_negative`
- `Lamp.node_tree`
- `Lamp.use_specular`
- `Lamp.texture_slots`
- `Lamp.type`
- `Lamp.use_nodes`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`

## 2.4.539 StateActuator(Actuator)

base classes — `bpy_struct`, `Actuator`

**class** `bpy.types.StateActuator` (*Actuator*)

Actuator to handle states

#### **operation**

Select the bit operation on object state mask

**Type** enum in ['SET', 'ADD', 'REMOVE', 'CHANGE'], default 'SET'

#### **states**

**Type** boolean array of 30 items, default (False, False)

### Inherited Properties

- `bpy_struct.id_data`



- `Actuator.name`
- `Actuator.show_expanded`
- `Actuator.pin`
- `Actuator.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Actuator.link`
- `Actuator.unlink`

## 2.4.540 SteeringActuator(Actuator)

base classes — `bpy_struct`, `Actuator`

class `bpy.types.SteeringActuator` (*Actuator*)

### **acceleration**

Max acceleration

**Type** float in [0, 1000], default 0.0

### **distance**

Relax distance

**Type** float in [0, 1000], default 0.0

### **facing**

Enable automatic facing

**Type** boolean, default False

### **facing\_axis**

Axis for automatic facing

**Type** enum in ['X', 'Y', 'Z', 'NEG\_X', 'NEG\_Y', 'NEG\_Z'], default 'X'

### **mode**

**Type** enum in ['SEEK', 'FLEE', 'PATHFOLLOWING'], default 'SEEK'

**navmesh**

Navigation mesh

**Type** `Object`

**normal\_up**

Use normal of the navmesh to set “UP” vector

**Type** `boolean`, default `False`

**self\_terminated**

Terminate when target is reached

**Type** `boolean`, default `False`

**show\_visualization**

Enable debug visualization

**Type** `boolean`, default `False`

**target**

Set target object

**Type** `Object`

**turn\_speed**

Max turn speed

**Type** `float` in `[0, 720]`, default `0.0`

**update\_period**

Path update period

**Type** `int` in `[-inf, inf]`, default `0`

**velocity**

Velocity magnitude

**Type** `float` in `[0, 1000]`, default `0.0`

**Inherited Properties**

- `bpy_struct.id_data`
- `Actuator.name`
- `Actuator.show_expanded`
- `Actuator.pin`
- `Actuator.type`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`

- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Actuator.link`
- `Actuator.unlink`

## 2.4.541 StretchToConstraint(Constraint)

base classes — `bpy_struct`, `Constraint`

**class** `bpy.types.StretchToConstraint` (*Constraint*)

Stretches to meet the target object

### **bulge**

Factor between volume variation and stretching

**Type** float in [0, 100], default 0.0

### **head\_tail**

Target along length of bone: Head=0, Tail=1

**Type** float in [0, 1], default 0.0

### **keep\_axis**

Axis to maintain during stretch

• `PLANE_X` X, Keep X Axis.

• `PLANE_Z` Z, Keep Z Axis.

**Type** enum in ['`PLANE_X`', '`PLANE_Z`'], default '`PLANE_X`'

### **rest\_length**

Length at rest position

**Type** float in [0, 100], default 0.0

### **subtarget**

**Type** string, default ""

### **target**

Target Object

**Type** `Object`

### **volume**

Maintain the object's volume as it stretches

**Type** enum in ['`VOLUME_XZX`', '`VOLUME_X`', '`VOLUME_Z`', '`NO_VOLUME`'], default '`VOLUME_XZX`'

## Inherited Properties

- `bpy_struct.id_data`
- `Constraint.name`
- `Constraint.active`

- `Constraint.mute`
- `Constraint.show_expanded`
- `Constraint.influence`
- `Constraint.error_location`
- `Constraint.owner_space`
- `Constraint.is_proxy_local`
- `Constraint.error_rotation`
- `Constraint.target_space`
- `Constraint.type`
- `Constraint.is_valid`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.542 StringProperty(Property)

base classes — `bpy_struct`, `Property`

**class** `bpy.types.StringProperty` (*Property*)  
RNA text string property definition

**default**

string default value

**Type** string, default "", (readonly)

**length\_max**

Maximum length of the string, 0 means unlimited

**Type** int in [0, inf], default 0, (readonly)

### Inherited Properties

- `bpy_struct.id_data`
- `Property.name`
- `Property.is_animatable`
- `Property.srna`
- `Property.description`

- `Property.is_enum_flag`
- `Property.is_hidden`
- `Property.identifier`
- `Property.is_never_none`
- `Property.is_readonly`
- `Property.is_registered`
- `Property.is_registered_optional`
- `Property.is_required`
- `Property.is_output`
- `Property.is_runtime`
- `Property.is_skip_save`
- `Property.subtype`
- `Property.type`
- `Property.unit`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Struct.name_property`

## 2.4.543 Struct(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.Struct` (*bpy\_struct*)  
RNA structure definition

#### base

Struct definition this is derived from

**Type** `Struct`, (readonly)

#### description

Description of the Struct's purpose

**Type** string, default "", (readonly)

**functions**

**Type** `bpy_prop_collection` of `Function`, (readonly)

**identifier**

Unique name used in the code and scripting

**Type** `string`, default `""`, (readonly)

**name**

Human readable name

**Type** `string`, default `""`, (readonly)

**name\_property**

Property that gives the name of the struct

**Type** `StringProperty`, (readonly)

**nested**

Struct in which this struct is always nested, and to which it logically belongs

**Type** `Struct`, (readonly)

**properties**

Properties in the struct

**Type** `bpy_prop_collection` of `Property`, (readonly)

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `BlenderRNA.structs`
- `CollectionProperty.fixed_type`
- `PointerProperty.fixed_type`

- `Property.srna`
- `Struct.base`
- `Struct.nested`

## 2.4.544 StucciTexture(Texture)

base classes — `bpy_struct`, `ID`, `Texture`

**class** `bpy.types.StucciTexture` (*Texture*)

Procedural noise texture

### **noise\_basis**

Sets the noise basis used for turbulence

- `BLENDER_ORIGINAL` Blender Original, Noise algorithm - Blender original: Smooth interpolated noise.
- `ORIGINAL_PERLIN` Original Perlin, Noise algorithm - Original Perlin: Smooth interpolated noise.
- `IMPROVED_PERLIN` Improved Perlin, Noise algorithm - Improved Perlin: Smooth interpolated noise.
- `VORONOI_F1` Voronoi F1, Noise algorithm - Voronoi F1: Returns distance to the closest feature point.
- `VORONOI_F2` Voronoi F2, Noise algorithm - Voronoi F2: Returns distance to the 2nd closest feature point.
- `VORONOI_F3` Voronoi F3, Noise algorithm - Voronoi F3: Returns distance to the 3rd closest feature point.
- `VORONOI_F4` Voronoi F4, Noise algorithm - Voronoi F4: Returns distance to the 4th closest feature point.
- `VORONOI_F2_F1` Voronoi F2-F1, Noise algorithm - Voronoi F1-F2.
- `VORONOI_CRACKLE` Voronoi Crackle, Noise algorithm - Voronoi Crackle: Voronoi tessellation with sharp edges.
- `CELL_NOISE` Cell Noise, Noise algorithm - Cell Noise: Square cell tessellation.

**Type** enum in [`'BLENDER_ORIGINAL'`, `'ORIGINAL_PERLIN'`, `'IMPROVED_PERLIN'`, `'VORONOI_F1'`, `'VORONOI_F2'`, `'VORONOI_F3'`, `'VORONOI_F4'`, `'VORONOI_F2_F1'`, `'VORONOI_CRACKLE'`, `'CELL_NOISE'`], default `'BLENDER_ORIGINAL'`

### **noise\_scale**

Sets scaling for noise input

**Type** float in [0.0001, inf], default 0.0

### **noise\_type**

- `SOFT_NOISE` Soft, Generate soft noise (smooth transitions).
- `HARD_NOISE` Hard, Generate hard noise (sharp transitions).

**Type** enum in [`'SOFT_NOISE'`, `'HARD_NOISE'`], default `'SOFT_NOISE'`

### **stucci\_type**

- `PLASTIC` Plastic, Uses standard stucci.

- WALL\_IN Wall in, Creates Dimples.
- WALL\_OUT Wall out, Creates Ridges.

**Type** enum in ['PLASTIC', 'WALL\_IN', 'WALL\_OUT'], default 'PLASTIC'

**turbulence**

Sets the turbulence of the bandnoise and ringnoise types

**Type** float in [0.0001, inf], default 0.0

**users\_material**

Materials that use this texture (readonly)

**users\_object\_modifier**

Object modifiers that use this texture (readonly)

**Inherited Properties**

- bpy\_struct.id\_data
- ID.name
- ID.use\_fake\_user
- ID.is\_updated
- ID.is\_updated\_data
- ID.library
- ID.tag
- ID.users
- Texture.animation\_data
- Texture.intensity
- Texture.color\_ramp
- Texture.contrast
- Texture.factor\_blue
- Texture.factor\_green
- Texture.factor\_red
- Texture.node\_tree
- Texture.saturation
- Texture.use\_preview\_alpha
- Texture.type
- Texture.use\_color\_ramp
- Texture.use\_nodes
- Texture.users\_material
- Texture.users\_object\_modifier
- Texture.users\_material
- Texture.users\_object\_modifier

**Inherited Functions**

- bpy\_struct.as\_pointer
- bpy\_struct.callback\_add
- bpy\_struct.callback\_remove
- bpy\_struct.driver\_add
- bpy\_struct.driver\_remove
- bpy\_struct.get
- bpy\_struct.is\_property\_hidden



- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`
- `Texture.evaluate`

### 2.4.545 SubsurfModifier(Modifier)

base classes — `bpy_struct`, `Modifier`

**class** `bpy.types.SubsurfModifier` (*Modifier*)

Subdivision surface modifier

#### **levels**

Number of subdivisions to perform

**Type** int in [0, 32767], default 0

#### **render\_levels**

Number of subdivisions to perform when rendering

**Type** int in [0, 32767], default 0

#### **show\_only\_control\_edges**

Skip drawing/rendering of interior subdivided edges

**Type** boolean, default False

#### **subdivision\_type**

Selects type of subdivision algorithm

**Type** enum in ['CATMULL\_CLARK', 'SIMPLE'], default 'CATMULL\_CLARK'

#### **use\_subsurf\_uv**

Use subsurf to subdivide UVs

**Type** boolean, default False

#### Inherited Properties

- `bpy_struct.id_data`
- `Modifier.name`
- `Modifier.use_apply_on_spline`
- `Modifier.show_in_editmode`
- `Modifier.show_expanded`
- `Modifier.show_on_cage`
- `Modifier.show_viewport`
- `Modifier.show_render`

- `Modifier.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.546 SunLamp(Lamp)

base classes — `bpy_struct`, `ID`, `Lamp`

**class** `bpy.types.SunLamp` (*Lamp*)

Constant direction parallel ray lamp

**shadow\_adaptive\_threshold**

Threshold for Adaptive Sampling (Raytraced shadows)

**Type** float in [0, 1], default 0.0

**shadow\_color**

Color of shadows cast by the lamp

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**shadow\_method**

Method to compute lamp shadow with

• `NOSHADOW` No Shadow.

• `RAY_SHADOW` Ray Shadow, Use ray tracing for shadow.

**Type** enum in ['NOSHADOW', 'RAY\_SHADOW'], default 'NOSHADOW'

**shadow\_ray\_sample\_method**

Method for generating shadow samples: Adaptive QMC is fastest, Constant QMC is less noisy but slower

**Type** enum in ['ADAPTIVE\_QMC', 'CONSTANT\_QMC'], default 'ADAPTIVE\_QMC'

**shadow\_ray\_samples**

Amount of samples taken extra (samples x samples)

**Type** int in [1, 64], default 0

**shadow\_soft\_size**

Light size for ray shadow sampling (Raytraced shadows)

**Type** float in  $[-\text{inf}, \text{inf}]$ , default 0.0

**sky**

Sky related settings for sun lamps

**Type** `LampSkySettings`, (readonly, never None)

**use\_only\_shadow**

Causes light to cast shadows only without illuminating objects

**Type** boolean, default False

**use\_shadow\_layer**

Causes only objects on the same layer to cast shadows

**Type** boolean, default False

**Inherited Properties**

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`
- `Lamp.active_texture`
- `Lamp.active_texture_index`
- `Lamp.animation_data`
- `Lamp.color`
- `Lamp.use_diffuse`
- `Lamp.distance`
- `Lamp.energy`
- `Lamp.use_own_layer`
- `Lamp.use_negative`
- `Lamp.node_tree`
- `Lamp.use_specular`
- `Lamp.texture_slots`
- `Lamp.type`
- `Lamp.use_nodes`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`

- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`

### 2.4.547 SurfaceCurve(Curve)

base classes — `bpy_struct`, `ID`, `Curve`

**class** `bpy.types.SurfaceCurve` (*Curve*)  
Curve datablock used for storing surfaces

**use\_uv\_as\_generated**

Uses the UV values as Generated textured coordinates

**Type** boolean, default False

#### Inherited Properties

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`
- `Curve.animation_data`
- `Curve.use_auto_texspace`
- `Curve.bevel_depth`
- `Curve.bevel_object`
- `Curve.bevel_resolution`
- `Curve.use_deform_bounds`
- `Curve.dimensions`
- `Curve.show_handles`
- `Curve.show_normal_face`
- `Curve.eval_time`
- `Curve.extrude`
- `Curve.fill_mode`
- `Curve.use_fill_deform`
- `Curve.use_path_follow`
- `Curve.materials`
- `Curve.offset`
- `Curve.use_time_offset`
- `Curve.use_path`
- `Curve.path_duration`

- `Curve.use_radius`
- `Curve.render_resolution_u`
- `Curve.render_resolution_v`
- `Curve.resolution_u`
- `Curve.resolution_v`
- `Curve.shape_keys`
- `Curve.splines`
- `Curve.use_stretch`
- `Curve.taper_object`
- `Curve.texspace_location`
- `Curve.texspace_size`
- `Curve.twist_mode`
- `Curve.twist_smooth`
- `Curve.use_uv_as_generated`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`

## 2.4.548 SurfaceModifier(Modifier)

base classes — `bpy_struct`, `Modifier`

**class** `bpy.types.SurfaceModifier` (*Modifier*)

Surface modifier defining modifier stack position used for surface fields

### Inherited Properties

- `bpy_struct.id_data`
- `Modifier.name`
- `Modifier.use_apply_on_spline`
- `Modifier.show_in_editmode`

- `Modifier.show_expanded`
- `Modifier.show_on_cage`
- `Modifier.show_viewport`
- `Modifier.show_render`
- `Modifier.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.549 TexMapping(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.TexMapping` (*`bpy_struct`*)

Mapping settings

#### **location**

**Type** float array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`

#### **max**

Maximum value for clipping

**Type** float array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`

#### **min**

Minimum value for clipping

**Type** float array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`

#### **rotation**

**Type** float array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`

#### **scale**

**Type** float array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`

#### **use\_max**

Whether to use maximum clipping value

**Type** boolean, default `False`

**use\_min**

Whether to use minimum clipping value

**Type** boolean, default False**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**2.4.550 Text(ID)**base classes — `bpy_struct`, `ID`**class** `bpy.types.Text` (*ID*)

Text datablock referencing an external or packed text file

**current\_character**

Index of current character in current line, and also start index of character in selection if one exists

**Type** int in [0, inf], default 0, (readonly)**current\_line**

Current line, and start line of selection if one exists

**Type** `TextLine`, (readonly, never None)**filepath**

Filename of the text file

**Type** string, default ""**is\_dirty**

Text file has been edited since last save

**Type** boolean, default False, (readonly)**is\_in\_memory**

Text file is in memory, without a corresponding file on disk

**Type** boolean, default False, (readonly)

**is\_modified**

Text file on disk is different than the one in memory

**Type** boolean, default False, (readonly)

**lines**

Lines of text

**Type** `bpy_prop_collection` of `TextLine`, (readonly)

**markers**

Text markers highlighting part of the text

**Type** `bpy_prop_collection` of `TextMarker`, (readonly)

**select\_end\_character**

Index of character after end of selection in the selection end line

**Type** int in [0, inf], default 0, (readonly)

**select\_end\_line**

End line of selection

**Type** `TextLine`, (readonly, never None)

**use\_module**

Register this text as a module on loading, Text name must end with ".py"

**Type** boolean, default False

**use\_tabs\_as\_spaces**

Automatically converts all new tabs into spaces

**Type** boolean, default False

**users\_logic**

Logic bricks that use this text (readonly)

**clear** ()

clear the text block

**write** (*text*)

write text at the cursor location and advance to the end of the text block

**Parameters** *text* (*string*) – New text for this datablock

**static as\_string** (*self*)

Return the text as a string.

**from\_string** (*string*)

Replace text with this string.

### Inherited Properties

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`



- `ID.users`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`

### References

- `BlendData.texts`
- `BlendDataTexts.load`
- `BlendDataTexts.new`
- `BlendDataTexts.remove`
- `Filter2DActuator.gls1_shader`
- `PythonConstraint.text`
- `PythonController.text`
- `SceneGameData.dome_text`
- `SpaceTextEditor.text`

## 2.4.551 TextBox(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.TextBox` (*bpy\_struct*)

Text bounding box for layout

**height**

**Type** float in [0, 50], default 0.0

**width**

**Type** float in [0, 50], default 0.0

**x**

**Type** float in [-50, 50], default 0.0

**y**

**Type** float in [-50, 50], default 0.0

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `TextCurve.text_boxes`

## 2.4.552 TextCharacterFormat(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.TextCharacterFormat` (*bpy\_struct*)

Text character formatting settings

**use\_bold**

**Type** boolean, default False

**use\_italic**

**Type** boolean, default False

**use\_small\_caps**

**Type** boolean, default False

**use\_underline**

**Type** boolean, default False

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `TextCurve.body_format`
- `TextCurve.edit_format`

## 2.4.553 TextCurve(Curve)

base classes — `bpy_struct`, `ID`, `Curve`

**class** `bpy.types.TextCurve` (*Curve*)  
Curve datablock used for storing text

### **active\_textbox**

**Type** `int` in `[-inf, inf]`, default `0`

### **align**

Text align from the object center

- `LEFT` Left, Align text to the left.
- `CENTER` Center, Center text.
- `RIGHT` Right, Align text to the right.
- `JUSTIFY` Justify, Align to the left and the right.
- `FLUSH` Flush, Align to the left and the right, with equal character spacing.

**Type** `enum` in `['LEFT', 'CENTER', 'RIGHT', 'JUSTIFY', 'FLUSH']`, default `'LEFT'`

- body**  
contents of this text object  
**Type** string, default ""
- body\_format**  
Stores the style of each character  
**Type** bpy\_prop\_collection of TextCharacterFormat, (readonly)
- edit\_format**  
Editing settings character formatting  
**Type** TextCharacterFormat, (readonly)
- family**  
Use Blender Objects as font characters. Give font objects a common name followed by the character it represents, eg. familia, familyb etc, and turn on Verts Duplication  
**Type** string, default ""
- follow\_curve**  
Curve deforming text object  
**Type** Object
- font**  
**Type** VectorFont
- font\_bold**  
**Type** VectorFont
- font\_bold\_italic**  
**Type** VectorFont
- font\_italic**  
**Type** VectorFont
- offset\_x**  
Horizontal offset from the object origin  
**Type** float in [-50, 50], default 0.0
- offset\_y**  
Vertical offset from the object origin  
**Type** float in [-50, 50], default 0.0
- shear**  
Italic angle of the characters  
**Type** float in [-1, 1], default 0.0
- size**  
**Type** float in [0.0001, 10000], default 0.0
- small\_caps\_scale**  
Scale of small capitals  
**Type** float in [-inf, inf], default 0.0
- space\_character**

**Type** float in [0, 10], default 0.0

**space\_line**

**Type** float in [0, 10], default 0.0

**space\_word**

**Type** float in [0, 10], default 0.0

**text\_boxes**

**Type** `bpy_prop_collection` of `TextBox`, (readonly)

**underline\_height**

**Type** float in [-0.2, 0.8], default 0.0

**underline\_position**

Vertical position of underline

**Type** float in [-0.2, 0.8], default 0.0

**use\_fast\_edit**

Don't fill polygons while editing

**Type** boolean, default False

**use\_uv\_as\_generated**

Uses the UV values as Generated textured coordinates

**Type** boolean, default False

## Inherited Properties

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`
- `Curve.animation_data`
- `Curve.use_auto_textrspace`
- `Curve.bevel_depth`
- `Curve.bevel_object`
- `Curve.bevel_resolution`
- `Curve.use_deform_bounds`
- `Curve.dimensions`
- `Curve.show_handles`
- `Curve.show_normal_face`
- `Curve.eval_time`
- `Curve.extrude`
- `Curve.fill_mode`
- `Curve.use_fill_deform`
- `Curve.use_path_follow`
- `Curve.materials`
- `Curve.offset`
- `Curve.use_time_offset`

- `Curve.use_path`
- `Curve.path_duration`
- `Curve.use_radius`
- `Curve.render_resolution_u`
- `Curve.render_resolution_v`
- `Curve.resolution_u`
- `Curve.resolution_v`
- `Curve.shape_keys`
- `Curve.splines`
- `Curve.use_stretch`
- `Curve.taper_object`
- `Curve.texspace_location`
- `Curve.texspace_size`
- `Curve.twist_mode`
- `Curve.twist_smooth`
- `Curve.use_uv_as_generated`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`

### 2.4.554 TextLine(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.TextLine` (*bpy\_struct*)  
Line of text in a Text datablock

**body**

Text in the line

**Type** string, default ""

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Text.current_line`
- `Text.lines`
- `Text.select_end_line`

## 2.4.555 TextMarker(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.TextMarker` (*bpy\_struct*)  
Marker highlighting a portion of text in a Text datablock

**character\_index\_end**

Start position of the marker in the line

**Type** int in [0, inf], default 0, (readonly)

**character\_index\_start**

Start position of the marker in the line

**Type** int in [0, inf], default 0, (readonly)

**color**

Color to display the marker with

**Type** float array of 4 items in [0, 1], default (0.0, 0.0, 0.0, 0.0)

**group**

**Type** int in [0, 65535], default 0, (readonly)

**is\_temporary**

Marker is temporary

**Type** boolean, default False, (readonly)

**line**

Line in which the marker is located

**Type** int in [0, inf], default 0, (readonly)

**use\_edit\_all**

Edit all markers of the same group as one

**Type** boolean, default False, (readonly)

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `Text.markers`

**2.4.556 Texture(ID)**

base classes — `bpy_struct`, `ID`

subclasses — `ImageTexture`, `MarbleTexture`, `EnvironmentMapTexture`, `PointDensityTexture`, `BlendTexture`, `MusgraveTexture`, `StucciTexture`, `VoronoiTexture`, `MagicTexture`, `VoxelDataTexture`, `PluginTexture`, `WoodTexture`, `CloudsTexture`, `DistortedNoiseTexture`, `NoiseTexture`

**class** `bpy.types.Texture` (*ID*)

Texture datablock used by materials, lamps, worlds and brushes



**animation\_data**

Animation data for this datablock

**Type** `AnimData`, (readonly)

**color\_ramp**

**Type** `ColorRamp`, (readonly)

**contrast**

Adjust the contrast of the texture

**Type** float in [0.01, 5], default 0.0

**factor\_blue**

**Type** float in [0, 2], default 0.0

**factor\_green**

**Type** float in [0, 2], default 0.0

**factor\_red**

**Type** float in [0, 2], default 0.0

**intensity**

Adjust the brightness of the texture

**Type** float in [0, 2], default 0.0

**node\_tree**

Node tree for node-based textures

**Type** `NodeTree`, (readonly)

**saturation**

Adjust the saturation of colors in the texture

**Type** float in [0, 2], default 0.0

**type**

- NONE None.
- BLEND Blend, Procedural - create a ramp texture.
- CLOUDS Clouds, Procedural - create a cloud-like fractal noise texture.
- DISTORTED\_NOISE Distorted Noise, Procedural - Noise texture distorted by two noise algorithms.
- ENVIRONMENT\_MAP Environment Map, Create a render of the environment mapped to a texture.
- IMAGE Image or Movie, Allow for images or movies to be used as textures.
- MAGIC Magic, Procedural - Color texture based on trigonometric functions.
- MARBLE Marble, Procedural - Marble-like noise texture with wave generated bands.
- MUSGRAVE Musgrave, Procedural - Highly flexible fractal noise texture.
- NOISE Noise, Procedural - Random noise, gives a different result every time, for every frame, for every pixel.
- POINT\_DENSITY Point Density.
- STUCCI Stucci, Procedural - Create a fractal noise texture.
- VORONOI Voronoi, Procedural - Create cell-like patterns based on Worley noise.

- `VOXEL_DATA` Voxel Data, Create a 3d texture based on volumetric data.
- `WOOD` Wood, Procedural - Wave generated bands or rings, with optional noise.

**Type** enum in ['NONE', 'BLEND', 'CLOUDS', 'DISTORTED\_NOISE', 'ENVIRONMENT\_MAP', 'IMAGE', 'MAGIC', 'MARBLE', 'MUSGRAVE', 'NOISE', 'POINT\_DENSITY', 'STUCCI', 'VORONOI', 'VOXEL\_DATA', 'WOOD'], default 'NONE'

**use\_color\_ramp**

Toggle color ramp operations

**Type** boolean, default False

**use\_nodes**

Make this a node-based texture

**Type** boolean, default False

**use\_preview\_alpha**

Show Alpha in Preview Render

**Type** boolean, default False

**users\_material**

Materials that use this texture (readonly)

**users\_object\_modifier**

Object modifiers that use this texture (readonly)

**evaluate** (*value*)

Evaluate the texture at the coordinates given

**Returns** Result

**Return type** float array of 4 items in [-inf, inf]

### Inherited Properties

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`

- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`

## References

- `BlendData.textures`
- `BlendDataTextures.new`
- `BlendDataTextures.remove`
- `Brush.texture`
- `CompositorNodeTexture.texture`
- `DisplaceModifier.texture`
- `FieldSettings.texture`
- `Lamp.active_texture`
- `Material.active_texture`
- `ParticleSettings.active_texture`
- `ShaderNodeTexture.texture`
- `TextureNodeTexture.texture`
- `TextureSlot.texture`
- `VertexWeightEditModifier.mask_texture`
- `VertexWeightMixModifier.mask_texture`
- `VertexWeightProximityModifier.mask_texture`
- `WarpModifier.texture`
- `WaveModifier.texture`
- `World.active_texture`

## 2.4.557 TextureNode(Node)

base classes — `bpy_struct`, `Node`

subclasses — `TextureNodeScale`, `TextureNodeRGBToBW`, `TextureNodeTranslate`, `TextureNodeTexture`, `TextureNodeCompose`, `TextureNodeValToNor`, `TextureNodeDecompose`, `TextureNodeHueSaturation`, `TextureNodeImage`, `TextureNodeBricks`, `TextureNodeCurveTime`, `TextureNodeInvert`, `TextureNodeCurveRGB`, `TextureNodeDistance`, `TextureNodeRotate`, `TextureNodeMixRGB`, `TextureNodeChecker`, `TextureNodeOutput`, `TextureNodeCoordinates`, `TextureNodeValToRGB`, `TextureNodeMath`, `TextureNodeViewer`

class `bpy.types.TextureNode` (*Node*)

**type**

**Type** enum in ['OUTPUT', 'CHECKER', 'TEXTURE', 'BRICKS', 'MATH', 'MIX\_RGB', 'RGBTOBW', 'VALTORGB', 'IMAGE', 'CURVE\_RGB', 'INVERT', 'HUE\_SAT',

'CURVE\_TIME', 'ROTATE', 'VIEWER', 'TRANSLATE', 'COORD', 'DISTANCE', 'COMPOSE', 'DECOMPOSE', 'VALTONOR', 'SCALE', 'SCRIPT', 'GROUP'], default 'OUTPUT', (readonly)

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.558 TextureNodeBricks(TextureNode)

base classes — `bpy_struct`, `Node`, `TextureNode`

**class** `bpy.types.TextureNodeBricks` (*TextureNode*)

#### **offset**

**Type** float in [0, 1], default 0.0

#### **offset\_frequency**

Offset every N rows

**Type** int in [2, 99], default 0

#### **squash**

**Type** float in [0, 99], default 0.0

#### **squash\_frequency**

Squash every N rows

**Type** int in [2, 99], default 0

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `TextureNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.559 TextureNodeChecker(TextureNode)

base classes — `bpy_struct`, `Node`, `TextureNode`

**class** `bpy.types.TextureNodeChecker` (*TextureNode*)

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `TextureNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.560 TextureNodeCompose(TextureNode)

base classes — `bpy_struct`, `Node`, `TextureNode`

**class** `bpy.types.TextureNodeCompose` (*TextureNode*)

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `TextureNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`

- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.561 TextureNodeCoordinates(TextureNode)

base classes — `bpy_struct`, `Node`, `TextureNode`

**class** `bpy.types.TextureNodeCoordinates` (*TextureNode*)

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `TextureNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.562 TextureNodeCurveRGB(TextureNode)

base classes — `bpy_struct`, `Node`, `TextureNode`

**class** `bpy.types.TextureNodeCurveRGB` (*TextureNode*)

### mapping

**Type** `CurveMapping`, (readonly)

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `TextureNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.563 TextureNodeCurveTime(TextureNode)

base classes — `bpy_struct`, `Node`, `TextureNode`

**class** `bpy.types.TextureNodeCurveTime` (*TextureNode*)

**curve**

**Type** `CurveMapping`, (readonly)

**frame\_end**

**Type** `int` in `[-32768, 32767]`, default 0

**frame\_start**

**Type** `int` in `[-32768, 32767]`, default 0

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`



- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `TextureNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.564 TextureNodeDecompose(TextureNode)

base classes — `bpy_struct`, `Node`, `TextureNode`

class `bpy.types.TextureNodeDecompose` (*TextureNode*)

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `TextureNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`

- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.565 TextureNodeDistance(TextureNode)

base classes — `bpy_struct`, `Node`, `TextureNode`

**class** `bpy.types.TextureNodeDistance` (*TextureNode*)

#### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `TextureNode.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.566 TextureNodeHueSaturation(TextureNode)

base classes — `bpy_struct`, `Node`, `TextureNode`

**class** `bpy.types.TextureNodeHueSaturation` (*TextureNode*)

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `TextureNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.567 TextureNodeImage(TextureNode)

base classes — `bpy_struct`, `Node`, `TextureNode`

**class** `bpy.types.TextureNodeImage` (*TextureNode*)

**image**

**Type** `Image`

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `TextureNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.568 TextureNodeInvert(TextureNode)

base classes — `bpy_struct`, `Node`, `TextureNode`

**class** `bpy.types.TextureNodeInvert` (*TextureNode*)

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `TextureNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`

- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.569 TextureNodeMath(TextureNode)

base classes — `bpy_struct`, `Node`, `TextureNode`

**class** `bpy.types.TextureNodeMath` (*TextureNode*)

### operation

**Type** enum in ['ADD', 'SUBTRACT', 'MULTIPLY', 'DIVIDE', 'SINE', 'COSINE', 'TANGENT', 'ARCSINE', 'ARCCOSINE', 'ARCTANGENT', 'POWER', 'LOGARITHM', 'MINIMUM', 'MAXIMUM', 'ROUND', 'LESS\_THAN', 'GREATER\_THAN'], default 'ADD'

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `TextureNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.570 TextureNodeMixRGB(TextureNode)

base classes — `bpy_struct`, `Node`, `TextureNode`

**class** `bpy.types.TextureNodeMixRGB` (*TextureNode*)

**blend\_type**

**Type** enum in ['MIX', 'ADD', 'MULTIPLY', 'SUBTRACT', 'SCREEN', 'DIVIDE', 'DIFFERENCE', 'DARKEN', 'LIGHTEN', 'OVERLAY', 'DODGE', 'BURN', 'HUE', 'SATURATION', 'VALUE', 'COLOR', 'SOFT\_LIGHT', 'LINEAR\_LIGHT'], default 'MIX'

**use\_alpha**

Include alpha of second input in this operation

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `TextureNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.571 TextureNodeOutput(TextureNode)

base classes — `bpy_struct`, `Node`, `TextureNode`

**class** `bpy.types.TextureNodeOutput` (*TextureNode*)

**filepath**

**Type** string, default ""

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `TextureNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.572 TextureNodeRGBToBW(TextureNode)

base classes — `bpy_struct`, `Node`, `TextureNode`

**class** `bpy.types.TextureNodeRGBToBW` (*TextureNode*)

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `TextureNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`

- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.573 TextureNodeRotate(TextureNode)

base classes — `bpy_struct`, `Node`, `TextureNode`

**class** `bpy.types.TextureNodeRotate` (*TextureNode*)

#### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `TextureNode.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`



## 2.4.574 TextureNodeScale(TextureNode)

base classes — `bpy_struct`, `Node`, `TextureNode`

**class** `bpy.types.TextureNodeScale` (*TextureNode*)

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `TextureNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.575 TextureNodeTexture(TextureNode)

base classes — `bpy_struct`, `Node`, `TextureNode`

**class** `bpy.types.TextureNodeTexture` (*TextureNode*)

### **node\_output**

For node-based textures, which output node to use

**Type** `int` in `[-32768, 32767]`, default `0`

### **texture**

**Type** `Texture`

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `TextureNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.576 TextureNodeTranslate(TextureNode)

base classes — `bpy_struct`, `Node`, `TextureNode`

**class** `bpy.types.TextureNodeTranslate` (*TextureNode*)

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `TextureNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`

- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.577 TextureNodeTree(NodeTree)

base classes — `bpy_struct`, `ID`, `NodeTree`

**class** `bpy.types.TextureNodeTree` (*NodeTree*)  
Node tree consisting of linked nodes used for textures

#### nodes

**Type** `TextureNodes bpy_prop_collection` of `Node`, (readonly)

#### Inherited Properties

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`
- `NodeTree.animation_data`
- `NodeTree.grease_pencil`
- `NodeTree.inputs`
- `NodeTree.links`
- `NodeTree.outputs`
- `NodeTree.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`

- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`

## 2.4.578 TextureNodeValToNor(TextureNode)

base classes — `bpy_struct`, `Node`, `TextureNode`

**class** `bpy.types.TextureNodeValToNor` (*TextureNode*)

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `TextureNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.579 TextureNodeValToRGB(TextureNode)

base classes — `bpy_struct`, `Node`, `TextureNode`

**class** `bpy.types.TextureNodeValToRGB` (*TextureNode*)

**color\_ramp**

**Type** `ColorRamp`, (readonly)

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`
- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `TextureNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.580 TextureNodeViewer(TextureNode)

base classes — `bpy_struct`, `Node`, `TextureNode`

**class** `bpy.types.TextureNodeViewer` (*TextureNode*)

### Inherited Properties

- `bpy_struct.id_data`
- `Node.name`
- `Node.inputs`

- `Node.label`
- `Node.location`
- `Node.outputs`
- `Node.parent`
- `TextureNode.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.581 TextureNodes(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.TextureNodes` (*bpy\_struct*)

Collection of Texture Nodes

**new** (*type*, *group=None*)

Add a node to this node tree

#### Parameters

- **type** (*enum* in [`'OUTPUT'`, `'CHECKER'`, `'TEXTURE'`, `'BRICKS'`, `'MATH'`, `'MIX_RGB'`, `'RGBTOBW'`, `'VALTORGB'`, `'IMAGE'`, `'CURVE_RGB'`, `'INVERT'`, `'HUE_SAT'`, `'CURVE_TIME'`, `'ROTATE'`, `'VIEWER'`, `'TRANSLATE'`, `'COORD'`, `'DISTANCE'`, `'COMPOSE'`, `'DECOMPOSE'`, `'VALTONOR'`, `'SCALE'`, `'SCRIPT'`, `'GROUP'`]) – Type, Type of node to add
- **group** (`NodeTree`, (optional)) – The group tree

**Returns** New node

**Return type** `Node`

**remove** (*node*)

Remove a node from this node tree

**Parameters** **node** (`Node`) – The node to remove

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `TextureNodeTree.nodes`

## 2.4.582 TextureSlot(`bpy_struct`)

base class — `bpy_struct`

subclasses — `ParticleSettingsTextureSlot`, `MaterialTextureSlot`, `LampTextureSlot`, `BrushTextureSlot`, `WorldTextureSlot`

**class** `bpy.types.TextureSlot` (*bpy\_struct*)

Texture slot defining the mapping and influence of a texture

### **blend\_type**

Mode used to apply the texture

**Type** enum in ['MIX', 'ADD', 'SUBTRACT', 'MULTIPLY', 'SCREEN', 'OVERLAY', 'DIFFERENCE', 'DIVIDE', 'DARKEN', 'LIGHTEN', 'HUE', 'SATURATION', 'VALUE', 'COLOR', 'SOFT\_LIGHT', 'LINEAR\_LIGHT'], default 'MIX'

### **color**

Default color for textures that don't return RGB or when RGB to intensity is enabled

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

### **default\_value**

Value to use for Ref, Spec, Amb, Emit, Alpha, RayMir, TransLu and Hard

**Type** float in [-inf, inf], default 0.0

**invert**

Invert the values of the texture to reverse its effect

**Type** boolean, default False

**name**

Texture slot name

**Type** string, default "", (readonly)

**offset**

Fine tune of the texture mapping X, Y and Z locations

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**output\_node**

Which output node to use, for node-based textures

**Type** enum in ['DUMMY'], default 'DUMMY'

**scale**

Set scaling for the texture's X, Y and Z sizes

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**texture**

Texture datablock used by this texture slot

**Type** [Texture](#)

**use\_rgb\_to\_intensity**

Convert texture RGB values to intensity (gray) values

**Type** boolean, default False

**use\_stencil**

Use this texture as a blending value on the next texture

**Type** boolean, default False

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`



- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `UILayout.template_preview`

## 2.4.583 Theme(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.Theme` (*bpy\_struct*)

Theme settings defining draw style and colors in the user interface

**bone\_color\_sets**

**Type** `bpy_prop_collection` of `ThemeBoneColorSet`, (readonly, never None)

**console**

**Type** `ThemeConsole`, (readonly, never None)

**dopesheet\_editor**

**Type** `ThemeDopeSheet`, (readonly, never None)

**file\_browser**

**Type** `ThemeFileBrowser`, (readonly, never None)

**graph\_editor**

**Type** `ThemeGraphEditor`, (readonly, never None)

**image\_editor**

**Type** `ThemeImageEditor`, (readonly, never None)

**info**

**Type** `ThemeInfo`, (readonly, never None)

**logic\_editor**

**Type** `ThemeLogicEditor`, (readonly, never None)

**name**

Name of the theme

**Type** string, default ""

**nla\_editor**

**Type** `ThemeNLAEditor`, (readonly, never None)

**node\_editor**

**Type** `ThemeNodeEditor`, (readonly, never None)

**outliner**

**Type** `ThemeOutliner`, (readonly, never None)

**properties**

**Type** `ThemeProperties`, (readonly, never None)

**sequence\_editor**

**Type** `ThemeSequenceEditor`, (readonly, never None)

**text\_editor**

**Type** `ThemeTextEditor`, (readonly, never None)

**theme\_area**

**Type** enum in ['USER\_INTERFACE', 'BONE\_COLOR\_SETS', 'VIEW\_3D', 'TIMELINE', 'GRAPH\_EDITOR', 'DOPESHEET\_EDITOR', 'NLA\_EDITOR', 'IMAGE\_EDITOR', 'SEQUENCE\_EDITOR', 'TEXT\_EDITOR', 'NODE\_EDITOR', 'LOGIC\_EDITOR', 'PROPERTIES', 'OUTLINER', 'USER\_PREFERENCES', 'INFO', 'FILE\_BROWSER', 'CONSOLE'], default 'USER\_INTERFACE'

**timeline**

**Type** `ThemeTimeline`, (readonly, never None)

**user\_interface**

**Type** `ThemeUserInterface`, (readonly, never None)

**user\_preferences**

**Type** `ThemeUserPreferences`, (readonly, never None)

**view\_3d**

**Type** `ThemeView3D`, (readonly, never None)

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `UserPreferences.themes`

## 2.4.584 ThemeBoneColorSet(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.ThemeBoneColorSet` (*bpy\_struct*)

Theme settings for bone color sets

**active**

Color used for active bones

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**normal**

Color used for the surface of bones

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**select**

Color used for selected bones

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**show\_colored\_constraints**

Allow the use of colors indicating constraints/keyed status

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `BoneGroup.colors`
- `Theme.bone_color_sets`

## 2.4.585 ThemeConsole(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.ThemeConsole` (*bpy\_struct*)

Theme settings for the Console

**back**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button\_text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button\_text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button\_title**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**cursor**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**header**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**header\_text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**header\_text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**line\_error**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**line\_info**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**line\_input**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**line\_output**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**title**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Theme.console`

## 2.4.586 ThemeDopeSheet(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.ThemeDopeSheet` (*bpy\_struct*)

Theme settings for the DopeSheet

**active\_channels\_group**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**back**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button\_text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button\_text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button\_title**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**channel\_group**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**channels**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**channels\_selected**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**dopesheet\_channel**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**dopesheet\_subchannel**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**frame\_current**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**grid**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**header**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**header\_text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**header\_text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**list**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**list\_text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**list\_text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**list\_title**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**long\_key**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**long\_key\_selected**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**title**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**value\_sliders**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**view\_sliders**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `Theme.dopesheet_editor`

**2.4.587 ThemeFileBrowser(`bpy_struct`)**

base class — `bpy_struct`

**class** `bpy.types.ThemeFileBrowser` (*bpy\_struct*)

Theme settings for the File Browser

**active\_file**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**active\_file\_text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**back**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button\_text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button\_text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button\_title**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**header**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**header\_text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**header\_text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**list**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**list\_text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**list\_text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**list\_title**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**scroll\_handle**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**scrollbar**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**selected\_file**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**tiles**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**title**



**Type** float array of 3 items in  $[-\text{inf}, \text{inf}]$ , default (0.0, 0.0, 0.0)

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Theme.file_browser`

## 2.4.588 ThemeFontStyle(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.ThemeFontStyle` (*bpy\_struct*)

Theme settings for Font

#### **font\_kerning\_style**

Which style to use for font kerning

- **UNFITTED** Unfitted, Use scaled but un-grid-fitted kerning distances.
- **DEFAULT** Default, Use scaled and grid-fitted kerning distances.

**Type** enum in ['UNFITTED', 'DEFAULT'], default 'UNFITTED'

#### **points**

**Type** int in [6, 48], default 0

#### **shadow**

Shadow size in pixels (0, 3 and 5 supported)

**Type** int in [0, 5], default 0

**shadow\_offset\_x**

Shadow offset in pixels

**Type** int in [-10, 10], default 0

**shadow\_offset\_y**

Shadow offset in pixels

**Type** int in [-10, 10], default 0

**shadowalpha**

**Type** float in [0, 1], default 0.0

**shadowcolor**

Shadow color in grey value

**Type** float in [0, 1], default 0.0

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `ThemeStyle.panel_title`
- `ThemeStyle.widget`
- `ThemeStyle.widget_label`

**2.4.589 ThemeGraphEditor(`bpy_struct`)**

base class — `bpy_struct`

**class** `bpy.types.ThemeGraphEditor` (*`bpy_struct`*)

Theme settings for the graph editor

**active\_channels\_group**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**back**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button\_text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button\_text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button\_title**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**channel\_group**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**channels\_region**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**dopesheet\_channel**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**dopesheet\_subchannel**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**frame\_current**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**grid**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**handle\_align**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**handle\_auto**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**handle\_auto\_clamped**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**handle\_free**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**handle\_sel\_align**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**handle\_sel\_auto**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**handle\_sel\_auto\_clamped**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**handle\_sel\_free**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**handle\_sel\_vect**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**handle\_vect**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**handle\_vertex**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**handle\_vertex\_select**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**handle\_vertex\_size**

**Type** int in [0, 255], default 0

**header**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**header\_text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**header\_text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**lastsel\_point**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**list**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**list\_text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**list\_text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**list\_title**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**panel**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**title****Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)**vertex****Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)**vertex\_select****Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)**vertex\_size****Type** int in [1, 10], default 0**window\_sliders****Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `Theme.graph_editor`

**2.4.590 ThemeImageEditor(bpy\_struct)**base class — `bpy_struct`**class** `bpy.types.ThemeImageEditor` (*bpy\_struct*)

Theme settings for the Image Editor

**back**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button\_text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button\_text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button\_title**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**editmesh\_active**

**Type** float array of 4 items in [-inf, inf], default (0.0, 0.0, 0.0, 0.0)

**face**

**Type** float array of 4 items in [-inf, inf], default (0.0, 0.0, 0.0, 0.0)

**face\_dot**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**face\_select**

**Type** float array of 4 items in [-inf, inf], default (0.0, 0.0, 0.0, 0.0)

**facedot\_size**

**Type** int in [1, 10], default 0

**header**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**header\_text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**header\_text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**scope\_back**

**Type** float array of 4 items in [-inf, inf], default (0.0, 0.0, 0.0, 0.0)

**text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**title**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**vertex**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**vertex\_select**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**vertex\_size**

**Type** int in [1, 10], default 0

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Theme.image_editor`

## 2.4.591 ThemeInfo(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.ThemeInfo` (*bpy\_struct*)

Theme settings for Info

**back**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button\_text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button\_text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button\_title**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**header**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**header\_text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**header\_text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**title**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `Theme.info`



## 2.4.592 ThemeLogicEditor(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.ThemeLogicEditor` (*bpy\_struct*)

Theme settings for the Logic Editor

**back**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button\_text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button\_text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button\_title**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**header**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**header\_text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**header\_text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**panel**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**title**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`

- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Theme.logic_editor`

## 2.4.593 ThemeNLAEditor(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.ThemeNLAEditor` (*`bpy_struct`*)

Theme settings for the NLA Editor

### **back**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

### **bars**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

### **bars\_selected**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

### **button**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

### **button\_text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

### **button\_text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

### **button\_title**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

### **frame\_current**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

### **grid**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

### **header**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**header\_text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**header\_text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**list**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**list\_text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**list\_text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**list\_title**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**strips**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**strips\_selected**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**title**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**view\_sliders**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`

- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Theme.nla_editor`

## 2.4.594 ThemeNodeEditor(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.ThemeNodeEditor` (*bpy\_struct*)

Theme settings for the Node Editor

### **back**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

### **button**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

### **button\_text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

### **button\_text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

### **button\_title**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

### **converter\_node**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

### **group\_node**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

### **header**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

### **header\_text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

### **header\_text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

### **in\_out\_node**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

### **list**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**list\_text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**list\_text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**list\_title**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**node\_backdrop**

**Type** float array of 4 items in [-inf, inf], default (0.0, 0.0, 0.0, 0.0)

**noodle\_curving**

Curving of the noodle

**Type** int in [0, 10], default 5

**operator\_node**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**selected\_text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**title**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**wire**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**wire\_select**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`

- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Theme.node_editor`

## 2.4.595 ThemeOutliner(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.ThemeOutliner` (*bpy\_struct*)

Theme settings for the Outliner

**back**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button\_text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button\_text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button\_title**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**header**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**header\_text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**header\_text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**title**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Theme.outliner`

## 2.4.596 ThemeProperties(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.ThemeProperties` (*bpy\_struct*)

Theme settings for the Properties

### **back**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

### **button**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

### **button\_text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

### **button\_text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

### **button\_title**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

### **header**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**header\_text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**header\_text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**panel**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**title**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `Theme.properties`

**2.4.597 ThemeSequenceEditor(bpy\_struct)**

base class — `bpy_struct`



---

```
class bpy.types.ThemeSequenceEditor (bpy_struct)
    Theme settings for the Sequence Editor

    audio_strip
        Type float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

    back
        Type float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

    button
        Type float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

    button_text
        Type float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

    button_text_hi
        Type float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

    button_title
        Type float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

    draw_action
        Type float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

    effect_strip
        Type float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

    frame_current
        Type float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

    grid
        Type float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

    header
        Type float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

    header_text
        Type float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

    header_text_hi
        Type float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

    image_strip
        Type float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

    keyframe
        Type float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

    meta_strip
        Type float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

    movie_strip
        Type float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)
```

**plugin\_strip**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**scene\_strip**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**title**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**transition\_strip**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**window\_sliders**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `Theme.sequence_editor`

## 2.4.598 ThemeStyle(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.ThemeStyle` (*bpy\_struct*)

Theme settings for style sets

**panel\_title**

**Type** `ThemeFontStyle`, (readonly, never None)

**widget**

**Type** `ThemeFontStyle`, (readonly, never None)

**widget\_label**

**Type** `ThemeFontStyle`, (readonly, never None)

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `UserPreferences.ui_styles`

## 2.4.599 ThemeTextEditor(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.ThemeTextEditor` (*bpy\_struct*)

Theme settings for the Text Editor

**back**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button\_text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button\_text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button\_title**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**cursor**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**header**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**header\_text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**header\_text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**line\_numbers\_background**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**scroll\_bar**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**selected\_text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**syntax\_builtin**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**syntax\_comment**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**syntax\_numbers**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**syntax\_special**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**syntax\_string**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**title**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Theme.text_editor`

## 2.4.600 ThemeTimeline(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.ThemeTimeline` (*`bpy_struct`*)

Theme settings for the Timeline

**back**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button\_text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button\_text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button\_title**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**frame\_current**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**grid**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**header**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**header\_text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**header\_text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**title**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Theme.timeline`

## 2.4.601 ThemeUserInterface(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.ThemeUserInterface` (*bpy\_struct*)

Theme settings for user interface elements

**icon\_file**

**Type** string, default ""

**wcol\_box**

**Type** `ThemeWidgetColors`, (readonly, never None)

**wcol\_list\_item**

**Type** `ThemeWidgetColors`, (readonly, never None)

**wcol\_menu**

**Type** `ThemeWidgetColors`, (readonly, never None)

**wcol\_menu\_back**

**Type** `ThemeWidgetColors`, (readonly, never None)

**wcol\_menu\_item**

**Type** `ThemeWidgetColors`, (readonly, never None)

**wcol\_num**

**Type** `ThemeWidgetColors`, (readonly, never None)

**wcol\_numslider**

**Type** `ThemeWidgetColors`, (readonly, never None)

**wcol\_option**

**Type** `ThemeWidgetColors`, (readonly, never None)

**wcol\_progress**

**Type** `ThemeWidgetColors`, (readonly, never None)

**wcol\_pulldown**

**Type** `ThemeWidgetColors`, (readonly, never None)

**wcol\_radio**

**Type** `ThemeWidgetColors`, (readonly, never None)

**wcol\_regular**

**Type** `ThemeWidgetColors`, (readonly, never None)

**wcol\_scroll**

**Type** `ThemeWidgetColors`, (readonly, never None)

**wcol\_state**

**Type** `ThemeWidgetStateColors`, (readonly, never None)

**wcol\_text**

**Type** `ThemeWidgetColors`, (readonly, never None)

**wcol\_toggle**

**Type** `ThemeWidgetColors`, (readonly, never None)

**wcol\_tool**

**Type** `ThemeWidgetColors`, (readonly, never None)

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Theme.user_interface`

## 2.4.602 ThemeUserPreferences(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.ThemeUserPreferences` (*`bpy_struct`*)  
Theme settings for the User Preferences

**back**

**Type** float array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`

**button**

**Type** float array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`



**button\_text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button\_text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button\_title**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**header**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**header\_text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**header\_text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**title**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Theme.user_preferences`

### 2.4.603 ThemeView3D(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.ThemeView3D` (*bpy\_struct*)

Theme settings for the 3D View

**act\_spline**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**back**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**bone\_pose**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**bone\_solid**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button\_text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button\_text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**button\_title**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**edge\_crease**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**edge\_facesel**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**edge\_seam**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**edge\_select**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**edge\_sharp**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**editmesh\_active**

**Type** float array of 4 items in [-inf, inf], default (0.0, 0.0, 0.0, 0.0)

**extra\_edge\_len**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**extra\_face\_angle**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**extra\_face\_area**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**face**

**Type** float array of 4 items in [-inf, inf], default (0.0, 0.0, 0.0, 0.0)

**face\_dot**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**face\_select**

**Type** float array of 4 items in [-inf, inf], default (0.0, 0.0, 0.0, 0.0)

**facedot\_size**

**Type** int in [1, 10], default 0

**frame\_current**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**grid**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**handle\_align**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**handle\_auto**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**handle\_free**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**handle\_sel\_align**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**handle\_sel\_auto**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**handle\_sel\_free**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**handle\_sel\_vect**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**handle\_vect**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**header**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**header\_text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**header\_text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**lamp**

**Type** float array of 4 items in [-inf, inf], default (0.0, 0.0, 0.0, 0.0)

**lastsel\_point**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**normal**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**nurb\_sel\_uline**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**nurb\_sel\_vline**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**nurb\_uline**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**nurb\_vline**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**object\_active**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**object\_grouped**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**object\_grouped\_active**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**object\_selected**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**outline\_width**

**Type** int in [1, 5], default 0

**panel**

**Type** float array of 4 items in [-inf, inf], default (0.0, 0.0, 0.0, 0.0)

**speaker**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**text\_hi**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**title**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**transform**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**vertex**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**vertex\_normal**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**vertex\_select**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**vertex\_size**

**Type** int in [1, 10], default 0

**wire**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Theme.view_3d`

## 2.4.604 ThemeWidgetColors(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.ThemeWidgetColors` (*bpy\_struct*)

Theme settings for widget color sets

**inner**

**Type** float array of 4 items in [-inf, inf], default (0.0, 0.0, 0.0, 0.0)

**inner\_sel**

**Type** float array of 4 items in [-inf, inf], default (0.0, 0.0, 0.0, 0.0)

**item**

**Type** float array of 4 items in [-inf, inf], default (0.0, 0.0, 0.0, 0.0)

**outline**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**shadedown**

**Type** int in [-100, 100], default 0

**shadetop**

**Type** int in [-100, 100], default 0

**show\_shaded**

**Type** boolean, default False

**text**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**text\_sel**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`

- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `ThemeUserInterface.wcol_box`
- `ThemeUserInterface.wcol_list_item`
- `ThemeUserInterface.wcol_menu`
- `ThemeUserInterface.wcol_menu_back`
- `ThemeUserInterface.wcol_menu_item`
- `ThemeUserInterface.wcol_num`
- `ThemeUserInterface.wcol_numslider`
- `ThemeUserInterface.wcol_option`
- `ThemeUserInterface.wcol_progress`
- `ThemeUserInterface.wcol_pulldown`
- `ThemeUserInterface.wcol_radio`
- `ThemeUserInterface.wcol_regular`
- `ThemeUserInterface.wcol_scroll`
- `ThemeUserInterface.wcol_text`
- `ThemeUserInterface.wcol_toggle`
- `ThemeUserInterface.wcol_tool`

### 2.4.605 ThemeWidgetStateColors(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.ThemeWidgetStateColors` (*bpy\_struct*)

Theme settings for widget state colors

#### **blend**

**Type** float in [0, 1], default 0.0

#### **inner\_anim**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

#### **inner\_anim\_sel**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

#### **inner\_driven**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

#### **inner\_driven\_sel**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

#### **inner\_key**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

#### **inner\_key\_sel**

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `ThemeUserInterface.wcol_state`

## 2.4.606 TimelineMarker(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.TimelineMarker` (*bpy\_struct*)

Marker for noting points in the timeline

**camera**

Camera this timeline sets to active

**Type** `Object`

**frame**

The frame on which the timeline marker appears

**Type** `int` in `[-inf, inf]`, default `0`

**name**

**Type** `string`, default `""`

**select**

Marker selection state

**Type** `boolean`, default `False`



## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Action.pose_markers`
- `ActionPoseMarkers.active`
- `ActionPoseMarkers.new`
- `ActionPoseMarkers.remove`
- `Scene.timeline_markers`
- `TimelineMarkers.new`
- `TimelineMarkers.remove`

## 2.4.607 TimelineMarkers(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.TimelineMarkers` (*bpy\_struct*)

Collection of timeline markers

**new** (*name*)

Add a keyframe to the curve

**Parameters** *name* (*string*) – New name for the marker (not unique)

**Returns** Newly created timeline marker

**Return type** `TimelineMarker`

**remove** (*marker*)

Remove a timeline marker

**Parameters** *marker* (`TimelineMarker`, (never `None`)) – Timeline marker to remove

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Scene.timeline_markers`

## 2.4.608 Timer(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.Timer` (*bpy\_struct*)  
Window event timer

**time\_delta**

Time since last step in seconds

**Type** float in [-inf, inf], default 0.0, (readonly)

**time\_duration**

Time since last step in seconds

**Type** float in [-inf, inf], default 0.0, (readonly)

**time\_step**

**Type** float in [-inf, inf], default 0.0, (readonly)

### Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `WindowManager.event_timer_add`
- `WindowManager.event_timer_remove`

## 2.4.609 ToolSettings(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.ToolSettings` (*bpy\_struct*)

### **auto\_keying\_mode**

Mode of automatic keyframe insertion for Objects and Bones

**Type** enum in ['ADD\_REPLACE\_KEYS', 'REPLACE\_KEYS'], default 'ADD\_REPLACE\_KEYS'

### **edge\_path\_live\_unwrap**

Changing edges seam re-calculates UV unwrap

**Type** boolean, default False

### **edge\_path\_mode**

The edge flag to tag when selecting the shortest path

**Type** enum in ['SELECT', 'SEAM', 'SHARP', 'CREASE', 'BEVEL'], default 'SELECT'

### **etch\_adaptive\_limit**

Number of bones in the subdivided stroke

**Type** float in [1e-05, 1], default 0.0

### **etch\_convert\_mode**

Method used to convert stroke to bones

- **FIXED** Fixed, Subdivide stroke in fixed number of bones.
- **LENGTH** Length, Subdivide stroke in bones of specific length.

- ADAPTIVE Adaptive, Subdivide stroke adaptively, with more subdivision in curvier parts.
- RETARGET Retarget, Retarget template bone chain to stroke.

**Type** enum in ['FIXED', 'LENGTH', 'ADAPTIVE', 'RETARGET'], default 'FIXED'

**etch\_length\_limit**

Number of bones in the subdivided stroke

**Type** float in [1e-05, 100000], default 0.0

**etch\_number**

DOC BROKEN

**Type** string, default ""

**etch\_roll\_mode**

Method used to adjust the roll of bones when retargeting

- NONE None, Don't adjust roll.
- VIEW View, Roll bones to face the view.
- JOINT Joint, Roll bone to original joint plane offset.

**Type** enum in ['NONE', 'VIEW', 'JOINT'], default 'NONE'

**etch\_side**

DOC BROKEN

**Type** string, default ""

**etch\_subdivision\_number**

Number of bones in the subdivided stroke

**Type** int in [1, 255], default 0

**etch\_template**

Template armature that will be retargeted to the stroke

**Type** `Object`

**image\_paint**

**Type** `ImagePaint`, (readonly)

**mesh\_select\_mode**

Which mesh elements selection works on

**Type** boolean array of 3 items, default (False, False, False)

**normal\_size**

Display size for normals in the 3D view

**Type** float in [1e-05, 1000], default 0.0

**particle\_edit**

**Type** `ParticleEdit`, (readonly)

**proportional\_edit**

Proportional Editing mode, allows transforms with distance fall-off

- DISABLED Disable, Proportional Editing disabled.
- ENABLED Enable, Proportional Editing enabled.

- CONNECTED Connected, Proportional Editing using connected geometry only.

**Type** enum in ['DISABLED', 'ENABLED', 'CONNECTED'], default 'DISABLED'

#### **proportional\_edit\_falloff**

Falloff type for proportional editing mode

- SMOOTH Smooth, Smooth falloff.
- SPHERE Sphere, Spherical falloff.
- ROOT Root, Root falloff.
- SHARP Sharp, Sharp falloff.
- LINEAR Linear, Linear falloff.
- CONSTANT Constant, Constant falloff.
- RANDOM Random, Random falloff.

**Type** enum in ['SMOOTH', 'SPHERE', 'ROOT', 'SHARP', 'LINEAR', 'CONSTANT', 'RANDOM'], default 'SMOOTH'

#### **proportional\_size**

Display size for proportional editing circle

**Type** float in [1e-05, 5000], default 0.0

#### **sculpt**

**Type** `Sculpt`, (readonly)

#### **sculpt\_paint\_use\_unified\_size**

Instead of per brush radius, the radius is shared across brushes

**Type** boolean, default False

#### **sculpt\_paint\_use\_unified\_strength**

Instead of per brush strength, the strength is shared across brushes

**Type** boolean, default False

#### **show\_uv\_local\_view**

Draw only faces with the currently displayed image assigned

**Type** boolean, default False

#### **snap\_element**

Type of element to snap to

- INCREMENT Increment, Snap to increments of grid.
- VERTEX Vertex, Snap to vertices.
- EDGE Edge, Snap to edges.
- FACE Face, Snap to faces.
- VOLUME Volume, Snap to volume.

**Type** enum in ['INCREMENT', 'VERTEX', 'EDGE', 'FACE', 'VOLUME'], default 'INCREMENT'

**snap\_target**

Which part to snap onto the target

- CLOSEST Closest, Snap closest point onto target.
- CENTER Center, Snap center onto target.
- MEDIAN Median, Snap median onto target.
- ACTIVE Active, Snap active onto target.

**Type** enum in ['CLOSEST', 'CENTER', 'MEDIAN', 'ACTIVE'], default 'CLOSEST'

**use\_auto\_normalize**

Ensure all bone-deforming vertex groups add up to 1.0 while weight painting

**Type** boolean, default False

**use\_bone\_sketching**

DOC BROKEN

**Type** boolean, default False

**use\_etch\_autoname**

DOC BROKEN

**Type** boolean, default False

**use\_etch\_overdraw**

DOC BROKEN

**Type** boolean, default False

**use\_etch\_quick**

DOC BROKEN

**Type** boolean, default False

**use\_grease\_pencil\_sessions**

Allow drawing multiple strokes at a time with Grease Pencil

**Type** boolean, default False

**use\_keyframe\_insert\_auto**

Automatic keyframe insertion for Objects and Bones

**Type** boolean, default False

**use\_keyframe\_insert\_keyingset**

Automatic keyframe insertion using active Keying Set only

**Type** boolean, default False

**use\_mesh\_automerge**

Automatically merge vertices moved to the same location

**Type** boolean, default False

**use\_multipaint**

Paint across all selected bones while weight painting

**Type** boolean, default False

**use\_proportional\_edit\_objects**

Proportional editing object mode

**Type** boolean, default False

**use\_record\_with\_nla**

Add a new NLA Track + Strip for every loop/pass made over the animation to allow non-destructive tweaking

**Type** boolean, default False

**use\_snap**

Snap during transform

**Type** boolean, default False

**use\_snap\_align\_rotation**

Align rotation with the snapping target

**Type** boolean, default False

**use\_snap\_peel\_object**

Consider objects as whole when finding volume center

**Type** boolean, default False

**use\_snap\_project**

Project individual elements on the surface of other objects

**Type** boolean, default False

**use\_snap\_self**

Snap onto itself (editmode)

**Type** boolean, default False

**use\_uv\_select\_sync**

Keep UV and edit mode mesh selection in sync

**Type** boolean, default False

**uv\_select\_mode**

UV selection and display mode

- VERTEX Vertex, Vertex selection mode.
- EDGE Edge, Edge selection mode.
- FACE Face, Face selection mode.
- ISLAND Island, Island selection mode.

**Type** enum in ['VERTEX', 'EDGE', 'FACE', 'ISLAND'], default 'VERTEX'

**vertex\_group\_weight**

Weight to assign in vertex groups

**Type** float in [0, 1], default 0.0

**vertex\_paint**

**Type** `VertexPaint`, (readonly)

**weight\_paint**

**Type** `VertexPaint`, (readonly)

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Context.tool_settings`
- `Scene.tool_settings`

## 2.4.610 TouchSensor(Sensor)

base classes — `bpy_struct`, `Sensor`

**class** `bpy.types.TouchSensor` (*Sensor*)  
Sensor to detect objects colliding with the current object

**material**

Only look for objects with this material (blank = all objects)

**Type** `Material`

### Inherited Properties

- `bpy_struct.id_data`
- `Sensor.name`
- `Sensor.show_expanded`
- `Sensor.frequency`
- `Sensor.invert`
- `Sensor.use_level`
- `Sensor.pin`
- `Sensor.use_pulse_false_level`
- `Sensor.use_pulse_true_level`
- `Sensor.use_tap`



- `Sensor.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Sensor.link`
- `Sensor.unlink`

## 2.4.611 TrackToConstraint(Constraint)

base classes — `bpy_struct`, `Constraint`

**class** `bpy.types.TrackToConstraint` (*Constraint*)

Aims the constrained object toward the target

### **head\_tail**

Target along length of bone: Head=0, Tail=1

**Type** float in [0, 1], default 0.0

### **subtarget**

**Type** string, default ""

### **target**

Target Object

**Type** `Object`

### **track\_axis**

Axis that points to the target object

**Type** enum in ['TRACK\_X', 'TRACK\_Y', 'TRACK\_Z', 'TRACK\_NEGATIVE\_X', 'TRACK\_NEGATIVE\_Y', 'TRACK\_NEGATIVE\_Z'], default 'TRACK\_X'

### **up\_axis**

Axis that points upward

**Type** enum in ['UP\_X', 'UP\_Y', 'UP\_Z'], default 'UP\_X'

### **use\_target\_z**

Target's Z axis, not World Z axis, will constraint the Up direction

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `Constraint.name`
- `Constraint.active`
- `Constraint.mute`
- `Constraint.show_expanded`
- `Constraint.influence`
- `Constraint.error_location`
- `Constraint.owner_space`
- `Constraint.is_proxy_local`
- `Constraint.error_rotation`
- `Constraint.target_space`
- `Constraint.type`
- `Constraint.is_valid`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.612 TransformConstraint(Constraint)

base classes — `bpy_struct`, `Constraint`

**class** `bpy.types.TransformConstraint` (*Constraint*)  
Maps transformations of the target to the object

**from\_max\_x**

Top range of X axis source motion

**Type** float in [-inf, inf], default 0.0

**from\_max\_y**

Top range of Y axis source motion

**Type** float in [-inf, inf], default 0.0

**from\_max\_z**

Top range of Z axis source motion

**Type** float in [-inf, inf], default 0.0

**from\_min\_x**

Bottom range of X axis source motion

**Type** float in [-inf, inf], default 0.0

**from\_min\_y**

Bottom range of Y axis source motion

**Type** float in [-inf, inf], default 0.0

**from\_min\_z**

Bottom range of Z axis source motion

**Type** float in [-inf, inf], default 0.0

**map\_from**

The transformation type to use from the target

**Type** enum in ['LOCATION', 'ROTATION', 'SCALE'], default 'LOCATION'

**map\_to**

The transformation type to affect of the constrained object

**Type** enum in ['LOCATION', 'ROTATION', 'SCALE'], default 'LOCATION'

**map\_to\_x\_from**

The source axis constrained object's X axis uses

**Type** enum in ['X', 'Y', 'Z'], default 'X'

**map\_to\_y\_from**

The source axis constrained object's Y axis uses

**Type** enum in ['X', 'Y', 'Z'], default 'X'

**map\_to\_z\_from**

The source axis constrained object's Z axis uses

**Type** enum in ['X', 'Y', 'Z'], default 'X'

**subtarget**

**Type** string, default ""

**target**

Target Object

**Type** [Object](#)

**to\_max\_x**

Top range of X axis destination motion

**Type** float in [-inf, inf], default 0.0

**to\_max\_y**

Top range of Y axis destination motion

**Type** float in [-inf, inf], default 0.0

**to\_max\_z**

Top range of Z axis destination motion

**Type** float in [-inf, inf], default 0.0

**to\_min\_x**

Bottom range of X axis destination motion

**Type** float in [-inf, inf], default 0.0

**to\_min\_y**

Bottom range of Y axis destination motion

**Type** float in [-inf, inf], default 0.0

**to\_min\_z**

Bottom range of Z axis destination motion

**Type** float in [-inf, inf], default 0.0

**use\_motion\_extrapolate**

Extrapolate ranges

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`
- `Constraint.name`
- `Constraint.active`
- `Constraint.mute`
- `Constraint.show_expanded`
- `Constraint.influence`
- `Constraint.error_location`
- `Constraint.owner_space`
- `Constraint.is_proxy_local`
- `Constraint.error_rotation`
- `Constraint.target_space`
- `Constraint.type`
- `Constraint.is_valid`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.613 TransformOrientation(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.TransformOrientation` (*bpy\_struct*)

**matrix**

**Type** float array of 9 items in [-inf, inf], default (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)

**name**

**Type** string, default ""

#### Inherited Properties

- `bpy_struct.id_data`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

#### References

- `Scene.orientations`
- `SpaceView3D.current_orientation`

### 2.4.614 TransformSequence(EffectSequence)

base classes — `bpy_struct`, `Sequence`, `EffectSequence`

**class** `bpy.types.TransformSequence` (*EffectSequence*)  
Sequence strip applying affine transformations to other strips

**interpolation**

- `NONE` None, No interpolation.
- `BILINEAR` Bilinear, Bilinear interpolation.

•BICUBIC Bicubic, Bicubic interpolation.

**Type** enum in ['NONE', 'BILINEAR', 'BICUBIC'], default 'NONE'

**rotation\_start**

**Type** float in [-360, 360], default 0.0

**scale\_start\_x**

**Type** float in [0, inf], default 0.0

**scale\_start\_y**

**Type** float in [0, inf], default 0.0

**translate\_start\_x**

**Type** float in [-inf, inf], default 0.0

**translate\_start\_y**

**Type** float in [-inf, inf], default 0.0

**translation\_unit**

**Type** enum in ['PIXELS', 'PERCENT'], default 'PIXELS'

**use\_uniform\_scale**

Scale uniformly, preserving aspect ratio

**Type** boolean, default False

### Inherited Properties

- bpy\_struct.id\_data
- Sequence.name
- Sequence.blend\_type
- Sequence.blend\_alpha
- Sequence.channel
- Sequence.waveform
- Sequence.effect\_fader
- Sequence.frame\_final\_end
- Sequence.frame\_offset\_end
- Sequence.frame\_still\_end
- Sequence.input\_1
- Sequence.input\_2
- Sequence.input\_3
- Sequence.select\_left\_handle
- Sequence.frame\_final\_duration
- Sequence.frame\_duration
- Sequence.lock
- Sequence.mute
- Sequence.select\_right\_handle
- Sequence.select
- Sequence.speed\_factor
- Sequence.frame\_start
- Sequence.frame\_final\_start
- Sequence.frame\_offset\_start

- `Sequence.frame_still_start`
- `Sequence.type`
- `Sequence.use_default_fade`
- `Sequence.input_count`
- `EffectSequence.color_balance`
- `EffectSequence.use_float`
- `EffectSequence.crop`
- `EffectSequence.use_deinterlace`
- `EffectSequence.use_reverse_frames`
- `EffectSequence.use_flip_x`
- `EffectSequence.use_flip_y`
- `EffectSequence.color_multiply`
- `EffectSequence.use_premultiply`
- `EffectSequence.proxy`
- `EffectSequence.use_proxy_custom_directory`
- `EffectSequence.use_proxy_custom_file`
- `EffectSequence.color_saturation`
- `EffectSequence.strobe`
- `EffectSequence.transform`
- `EffectSequence.use_color_balance`
- `EffectSequence.use_crop`
- `EffectSequence.use_proxy`
- `EffectSequence.use_translation`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Sequence.getStripElem`
- `Sequence.swap`

### 2.4.615 UILayout(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.UILayout` (*`bpy_struct`*)  
 User interface layout in a panel or header

**active**

**Type** boolean, default False

#### **alert**

**Type** boolean, default False

#### **alignment**

**Type** enum in ['EXPAND', 'LEFT', 'CENTER', 'RIGHT'], default 'EXPAND'

#### **enabled**

When false, this (sub)layout is greyed out

**Type** boolean, default False

#### **operator\_context**

**Type** enum in ['INVOKE\_DEFAULT', 'INVOKE\_REGION\_WIN', 'INVOKE\_REGION\_CHANNELS', 'INVOKE\_REGION\_PREVIEW', 'INVOKE\_AREA', 'INVOKE\_SCREEN', 'EXEC\_DEFAULT', 'EXEC\_REGION\_WIN', 'EXEC\_REGION\_CHANNELS', 'EXEC\_REGION\_PREVIEW', 'EXEC\_AREA', 'EXEC\_SCREEN'], default 'INVOKE\_DEFAULT'

#### **scale\_x**

Scale factor along the X for items in this (sub)layout

**Type** float in [0, inf], default 0.0

#### **scale\_y**

Scale factor along the Y for items in this (sub)layout

**Type** float in [0, inf], default 0.0

#### **row** (*align=False*)

Sub-layout. Items placed in this sublayout are placed next to each other in a row

**Parameters** **align** (*boolean, (optional)*) – Align buttons to each other

**Returns** Sub-layout to put items in

**Return type** `UILayout`

#### **column** (*align=False*)

Sub-layout. Items placed in this sublayout are placed under each other in a column

**Parameters** **align** (*boolean, (optional)*) – Align buttons to each other

**Returns** Sub-layout to put items in

**Return type** `UILayout`

#### **column\_flow** (*columns=0, align=False*)

`column_flow`

##### **Parameters**

- **columns** (*int in [0, inf], (optional)*) – Number of columns, 0 is automatic
- **align** (*boolean, (optional)*) – Align buttons to each other

**Returns** Sub-layout to put items in

**Return type** `UILayout`

#### **box** ()

Sublayout (items placed in this sublayout are placed under each other in a column and are surrounded by a box)



**Returns** Sub-layout to put items in

**Return type** `UILayout`

**split** (*percentage=0.0, align=False*)  
split

**Parameters**

- **percentage** (*float in [0, 1], (optional)*) – Percentage, Percentage of width to split at
- **align** (*boolean, (optional)*) – Align buttons to each other

**Returns** Sub-layout to put items in

**Return type** `UILayout`

**prop** (*data, property, text=""*, *icon='NONE', expand=False, slider=False, toggle=False, icon\_only=False, event=False, full\_event=False, emboss=True, index=-1*)  
Item. Exposes an RNA item and places it into the layout

**Parameters**

- **data** (`AnyType`, (never None)) – Data from which to take property
- **property** (*string*) – Identifier of property in data
- **text** (*string, (optional)*) – Override automatic text of the item
- **icon** (*enum in ['NONE', 'QUESTION', 'ERROR', 'CANCEL', 'TRIA\_RIGHT', 'TRIA\_DOWN', 'TRIA\_LEFT', 'TRIA\_UP', 'ARROW\_LEFTRIGHT', 'PLUS', 'DISCLOSURE\_TRI\_DOWN', 'DISCLOSURE\_TRI\_RIGHT', 'RADIOBUT\_OFF', 'RADIOBUT\_ON', 'MENU\_PANEL', 'BLENDER', 'DOT', 'X', 'GO\_LEFT', 'PLUG', 'UI', 'NODE', 'NODE\_SEL', 'FULLSCREEN', 'SPLITSCREEN', 'RIGHTARROW\_THIN', 'BORDERMOVE', 'VIEWZOOM', 'ZOOMIN', 'ZOOMOUT', 'PANEL\_CLOSE', 'COPY\_ID', 'EYEDROPPER', 'LINK\_AREA', 'AUTO', 'CHECKBOX\_DEHLT', 'CHECKBOX\_HLT', 'UNLOCKED', 'LOCKED', 'UNPINNED', 'PINNED', 'SCREEN\_BACK', 'RIGHTARROW', 'DOWNARROW\_HLT', 'DOTSUP', 'DOTSDOWN', 'LINK', 'INLINK', 'PLUGIN', 'HELP', 'GHOST\_ENABLED', 'COLOR', 'LINKED', 'UNLINKED', 'HAND', 'ZOOM\_ALL', 'ZOOM\_SELECTED', 'ZOOM\_PREVIOUS', 'ZOOM\_IN', 'ZOOM\_OUT', 'RENDER\_REGION', 'BORDER\_RECT', 'BORDER\_LASSO', 'FREEZE', 'STYLUS\_PRESSURE', 'GHOST\_DISABLED', 'NEW', 'FILE\_TICK', 'QUIT', 'URL', 'RECOVER\_LAST', 'FULLSCREEN\_ENTER', 'FULLSCREEN\_EXIT', 'BLANK1', 'LAMP', 'MATERIAL', 'TEXTURE', 'ANIM', 'WORLD', 'SCENE', 'EDIT', 'GAME', 'RADIO', 'SCRIPT', 'PARTICLES', 'PHYSICS', 'SPEAKER', 'TEXTURE\_SHADED', 'VIEW3D', 'IPO', 'OOPS', 'BUTS', 'FILESEL', 'IMAGE\_COL', 'INFO', 'SEQUENCE', 'TEXT', 'IMASEL', 'SOUND', 'ACTION', 'NLA', 'SCRIPTWIN', 'TIME', 'NODETREE', 'LOGIC', 'CONSOLE', 'PREFERENCES', 'ASSET\_MANAGER', 'OBJECT\_DATAMODE', 'EDITMODE\_HLT', 'FACESEL\_HLT', 'VPAIN\_T\_HLT', 'TPAIN\_T\_HLT', 'WPAIN\_T\_HLT', 'SCULPTMODE\_HLT', 'POSE\_HLT', 'PARTICLEMODE', 'LIGHTPAINT', 'SCENE\_DATA', 'RENDERLAYERS', 'WORLD\_DATA', 'OBJECT\_DATA', 'MESH\_DATA', 'CURVE\_DATA', 'META\_DATA', 'LATTICE\_DATA', 'LAMP\_DATA', 'MATERIAL\_DATA', 'TEXTURE\_DATA', 'ANIM\_DATA', 'CAMERA\_DATA', 'PARTICLE\_DATA', 'LIBRARY\_DATA\_DIRECT', 'GROUP', 'ARMATURE\_DATA', 'POSE\_DATA', 'BONE\_DATA', 'CONSTRAINT', 'SHAPEKEY\_DATA', 'CONSTRAINT\_BONE', 'PACKAGE', 'UGLYPACKAGE', 'BRUSH\_DATA', 'IMAGE\_DATA', 'FILE', 'FCURVE', 'FONT\_DATA', 'RENDER\_RESULT', 'SURFACE\_DATA', 'EMPTY\_DATA', 'SETTINGS', 'RENDER\_ANIMATION', 'RENDER\_STILL', 'BOIDS', 'STRANDS', 'LIBRARY\_DATA\_INDIRECT', 'GREASEPENCIL', 'GROUP\_BONE',*

'GROUP\_VERTEX', 'GROUP\_VCOL', 'GROUP\_UVS', 'RNA', 'RNA\_ADD', 'OUTLINER\_OB\_EMPTY', 'OUTLINER\_OB\_MESH', 'OUTLINER\_OB\_CURVE', 'OUTLINER\_OB\_LATTICE', 'OUTLINER\_OB\_META', 'OUTLINER\_OB\_LAMP', 'OUTLINER\_OB\_CAMERA', 'OUTLINER\_OB\_ARMATURE', 'OUTLINER\_OB\_FONT', 'OUTLINER\_OB\_SURFACE', 'OUTLINER\_OB\_SPEAKER', 'RESTRICT\_VIEW\_OFF', 'RESTRICT\_VIEW\_ON', 'RESTRICT\_SELECT\_OFF', 'RESTRICT\_SELECT\_ON', 'RESTRICT\_RENDER\_OFF', 'RESTRICT\_RENDER\_ON', 'OUTLINER\_DATA\_EMPTY', 'OUTLINER\_DATA\_MESH', 'OUTLINER\_DATA\_CURVE', 'OUTLINER\_DATA\_LATTICE', 'OUTLINER\_DATA\_META', 'OUTLINER\_DATA\_LAMP', 'OUTLINER\_DATA\_CAMERA', 'OUTLINER\_DATA\_ARMATURE', 'OUTLINER\_DATA\_FONT', 'OUTLINER\_DATA\_SURFACE', 'OUTLINER\_DATA\_SPEAKER', 'OUTLINER\_DATA\_POSE', 'MESH\_PLANE', 'MESH\_CUBE', 'MESH\_CIRCLE', 'MESH\_UVSPHERE', 'MESH\_ICOSPHERE', 'MESH\_GRID', 'MESH\_MONKEY', 'MESH\_CYLINDER', 'MESH\_TORUS', 'MESH\_CONE', 'LAMP\_POINT', 'LAMP\_SUN', 'LAMP\_SPOT', 'LAMP\_HEMI', 'LAMP\_AREA', 'META\_PLANE', 'META\_CUBE', 'META\_BALL', 'META\_ELLIPSOID', 'META\_CAPSULE', 'SURFACE\_NCURVE', 'SURFACE\_NCIRCLE', 'SURFACE\_NSURFACE', 'SURFACE\_NCYLINDER', 'SURFACE\_NSHERE', 'SURFACE\_NTORUS', 'CURVE\_BEZCURVE', 'CURVE\_BEZCIRCLE', 'CURVE\_NCURVE', 'CURVE\_NCIRCLE', 'CURVE\_PATH', 'FORCE\_FORCE', 'FORCE\_WIND', 'FORCE\_VORTEX', 'FORCE\_MAGNETIC', 'FORCE\_HARMONIC', 'FORCE\_CHARGE', 'FORCE\_LENNARDJONES', 'FORCE\_TEXTURE', 'FORCE\_CURVE', 'FORCE\_BOID', 'FORCE\_TURBULENCE', 'FORCE\_DRAG', 'MODIFIER', 'MOD\_WAVE', 'MOD\_BUILD', 'MOD\_DECIM', 'MOD\_MIRROR', 'MOD\_SOFT', 'MOD\_SUBSURF', 'HOOK', 'MOD\_PHYSICS', 'MOD\_PARTICLES', 'MOD\_BOOLEAN', 'MOD\_EDGESPLIT', 'MOD\_ARRAY', 'MOD\_UVPROJECT', 'MOD\_DISPLACE', 'MOD\_CURVE', 'MOD\_LATTICE', 'CONSTRAINT\_DATA', 'MOD\_ARMATURE', 'MOD\_SHRINKWRAP', 'MOD\_CAST', 'MOD\_MESHDEFORM', 'MOD\_BEVEL', 'MOD\_SMOOTH', 'MOD\_SIMPLEDEFORM', 'MOD\_MASK', 'MOD\_CLOTH', 'MOD\_EXPLODE', 'MOD\_FLUIDSIM', 'MOD\_MULTIRES', 'MOD\_SMOKE', 'MOD\_SOLIDIFY', 'MOD\_SCREW', 'MOD\_VERTEX\_WEIGHT', 'REC', 'PLAY', 'FF', 'REW', 'PAUSE', 'PREV\_KEYFRAME', 'NEXT\_KEYFRAME', 'PLAY\_AUDIO', 'PLAY\_REVERSE', 'PREVIEW\_RANGE', 'PMARKER\_ACT', 'PMARKER\_SEL', 'PMARKER', 'MARKER\_HLT', 'MARKER', 'SPACE2', 'SPACE3', 'KEYINGSET', 'KEY\_DEHLT', 'KEY\_HLT', 'MUTE\_IPO\_OFF', 'MUTE\_IPO\_ON', 'VISIBLE\_IPO\_OFF', 'VISIBLE\_IPO\_ON', 'DRIVER', 'SOLO\_OFF', 'SOLO\_ON', 'VERTEXSEL', 'EDGESEL', 'FACESEL', 'ROTATE', 'CURSOR', 'ROTATECOLLECTION', 'ROTATECENTER', 'ROACTIVE', 'ALIGN', 'SMOOTHCURVE', 'SPHERECURVE', 'ROOTCURVE', 'SHARPCURVE', 'LINCURVE', 'NOCURVE', 'RNDCURVE', 'PROP\_OFF', 'PROP\_ON', 'PROP\_CON', 'PARTICLE\_POINT', 'PARTICLE\_TIP', 'PARTICLE\_PATH', 'MAN\_TRANS', 'MAN\_ROT', 'MAN\_SCALE', 'MANIPUL', 'SNAP\_OFF', 'SNAP\_ON', 'SNAP\_NORMAL', 'SNAP\_INCREMENT', 'SNAP\_VERTEX', 'SNAP\_EDGE', 'SNAP\_FACE', 'SNAP\_VOLUME', 'STICKY\_UVS\_LOC', 'STICKY\_UVS\_DISABLE', 'STICKY\_UVS\_VERT', 'CLIPUV\_DEHLT', 'CLIPUV\_HLT', 'SNAP\_PEEL\_OBJECT', 'GRID', 'PASTEDOWN', 'COPYDOWN', 'PASTEFLIPUP', 'PASTEFLIPDOWN', 'SNAP\_SURFACE', 'RETOPO', 'UV\_VERTEXSEL', 'UV\_EDGESEL', 'UV\_FACESEL', 'UV\_ISLANDSEL', 'UV\_SYNC\_SELECT', 'BBOX', 'WIRE', 'SOLID', 'SMOOTH', 'POTATO', 'ORTHO', 'LOCKVIEW\_OFF', 'LOCKVIEW\_ON', 'AXIS\_SIDE', 'AXIS\_FRONT', 'AXIS\_TOP', 'NDOF\_DOM', 'NDOF\_TURN', 'NDOF\_FLY', 'NDOF\_TRANS', 'LAYER\_USED', 'LAYER\_ACTIVE', 'SORTALPHA', 'SORTBYEXT', 'SORTTIME', 'SORTSIZE', 'LONGDISPLAY', 'SHORTDISPLAY', 'GHOST', 'IMGDISPLAY', 'BOOKMARKS', 'FONTPREVIEW', 'FILTER', 'NEWFOLDER', 'FILE\_PARENT', 'FILE\_REFRESH',

'FILE\_FOLDER', 'FILE\_BLANK', 'FILE\_BLEND', 'FILE\_IMAGE', 'FILE\_MOVIE', 'FILE\_SCRIPT', 'FILE\_SOUND', 'FILE\_FONT', 'BACK', 'FORWARD', 'DISK\_DRIVE', 'MATPLANE', 'MATSPHERE', 'MATCUBE', 'MONKEY', 'HAIR', 'ALIASED', 'ANTIALIASED', 'MAT\_SPHERE\_SKY', 'WORDWRAP\_OFF', 'WORDWRAP\_ON', 'SYNTAX\_OFF', 'SYNTAX\_ON', 'LINENUMBERS\_OFF', 'LINENUMBERS\_ON', 'SCRIPTPLUGINS', 'SEQ\_SEQUENCER', 'SEQ\_PREVIEW', 'SEQ\_LUMA\_WAVEFORM', 'SEQ\_CHROMA\_SCOPE', 'SEQ\_HISTOGRAM', 'SEQ\_SPLITVIEW', 'IMAGE\_RGB', 'IMAGE\_RGB\_ALPHA', 'IMAGE\_ALPHA', 'IMAGE\_ZDEPTH', 'IMAGEFILE', 'BRUSH\_ADD', 'BRUSH\_BLOB', 'BRUSH\_BLUR', 'BRUSH\_CLAY', 'BRUSH\_CLONE', 'BRUSH\_CREASE', 'BRUSH\_DARKEN', 'BRUSH\_FILL', 'BRUSH\_FLATTEN', 'BRUSH\_GRAB', 'BRUSH\_INFLATE', 'BRUSH\_LAYER', 'BRUSH\_LIGHTEN', 'BRUSH\_MIX', 'BRUSH\_MULTIPLY', 'BRUSH\_NUDGE', 'BRUSH\_PINCH', 'BRUSH\_SCRAPE', 'BRUSH\_SCULPT\_DRAW', 'BRUSH\_SMEAR', 'BRUSH\_SMOOTH', 'BRUSH\_SNAKE\_HOOK', 'BRUSH\_SOFTEN', 'BRUSH\_SUBTRACT', 'BRUSH\_TEXDRAW', 'BRUSH\_THUMB', 'BRUSH\_ROTATE', 'BRUSH\_VERTEXDRAW', 'VIEW3D\_VEC', 'EDIT\_VEC', 'EDITMODE\_DEHLT', 'EDITMODE\_HLT', 'DISCLOSURE\_TRI\_RIGHT\_VEC', 'DISCLOSURE\_TRI\_DOWN\_VEC', 'MOVE\_UP\_VEC', 'MOVE\_DOWN\_VEC', 'X\_VEC', 'SMALL\_TRI\_RIGHT\_VEC'], (optional)) – Icon, Override automatic icon of the item

- **expand** (*boolean, (optional)*) – Expand button to show more detail
- **slider** (*boolean, (optional)*) – Use slider widget for numeric values
- **toggle** (*boolean, (optional)*) – Use toggle widget for boolean values
- **icon\_only** (*boolean, (optional)*) – Draw only icons in buttons, no text
- **event** (*boolean, (optional)*) – Use button to input key events
- **full\_event** (*boolean, (optional)*) – Use button to input full events including modifiers
- **emboss** (*boolean, (optional)*) – Draw the button itself, just the icon/text
- **index** (*int in [-2, inf], (optional)*) – The index of this button, when set a single member of an array can be accessed, when set to -1 all array members are used

**props\_enum** (*data, property*)

props\_enum

#### Parameters

- **data** (*AnyType, (never None)*) – Data from which to take property
- **property** (*string*) – Identifier of property in data

**prop\_menu\_enum** (*data, property, text=""*, *icon='NONE'*)

prop\_menu\_enum

#### Parameters

- **data** (*AnyType, (never None)*) – Data from which to take property
- **property** (*string*) – Identifier of property in data
- **text** (*string, (optional)*) – Override automatic text of the item
- **icon** (*enum in ['NONE', 'QUESTION', 'ERROR', 'CANCEL', 'TRIA\_RIGHT', 'TRIA\_DOWN', 'TRIA\_LEFT', 'TRIA\_UP', 'ARROW\_LEFTRIGHT', 'PLUS', 'DISCLOSURE\_TRI\_DOWN', 'DISCLOSURE\_TRI\_RIGHT', 'RADIOBUT\_OFF', 'RADIOBUT\_ON', 'MENU\_PANEL', 'BLENDER', 'DOT', 'X', 'GO\_LEFT', 'PLUG', 'UI', 'NODE', 'NODE\_SEL', 'FULLSCREEN', 'SPLITSCREEN', 'RIGHTARROW\_THIN', 'BORDERMOVE', 'VIEWZOOM', 'ZOOMIN', 'ZOOMOUT', 'PANEL\_CLOSE',*

'COPY\_ID', 'EYEDROPPER', 'LINK\_AREA', 'AUTO', 'CHECKBOX\_DEHLT',  
'CHECKBOX\_HLT', 'UNLOCKED', 'LOCKED', 'UNPINNED', 'PINNED',  
'SCREEN\_BACK', 'RIGHTARROW', 'DOWNARROW\_HLT', 'DOTSUP', 'DOTSDOWN',  
'LINK', 'INLINK', 'PLUGIN', 'HELP', 'GHOST\_ENABLED', 'COLOR', 'LINKED',  
'UNLINKED', 'HAND', 'ZOOM\_ALL', 'ZOOM\_SELECTED', 'ZOOM\_PREVIOUS',  
'ZOOM\_IN', 'ZOOM\_OUT', 'RENDER\_REGION', 'BORDER\_RECT', 'BOR-  
DER\_LASSO', 'FREEZE', 'STYLUS\_PRESSURE', 'GHOST\_DISABLED', 'NEW',  
'FILE\_TICK', 'QUIT', 'URL', 'RECOVER\_LAST', 'FULLSCREEN\_ENTER',  
'FULLSCREEN\_EXIT', 'BLANK1', 'LAMP', 'MATERIAL', 'TEXTURE', 'ANIM',  
'WORLD', 'SCENE', 'EDIT', 'GAME', 'RADIO', 'SCRIPT', 'PARTICLES',  
'PHYSICS', 'SPEAKER', 'TEXTURE\_SHADED', 'VIEW3D', 'IPO', 'OOPS',  
'BUTS', 'FILESEL', 'IMAGE\_COL', 'INFO', 'SEQUENCE', 'TEXT', 'IMASEL',  
'SOUND', 'ACTION', 'NLA', 'SCRIPTWIN', 'TIME', 'NODETREE', 'LOGIC',  
'CONSOLE', 'PREFERENCES', 'ASSET\_MANAGER', 'OBJECT\_DATAMODE', 'EDIT-  
MODE\_HLT', 'FACESEL\_HLT', 'VPAINT\_HLT', 'TPAINT\_HLT', 'WPAINT\_HLT',  
'SCULPTMODE\_HLT', 'POSE\_HLT', 'PARTICLEMODE', 'LIGHTPAINT',  
'SCENE\_DATA', 'RENDERLAYERS', 'WORLD\_DATA', 'OBJECT\_DATA',  
'MESH\_DATA', 'CURVE\_DATA', 'META\_DATA', 'LATTICE\_DATA', 'LAMP\_DATA',  
'MATERIAL\_DATA', 'TEXTURE\_DATA', 'ANIM\_DATA', 'CAMERA\_DATA', 'PAR-  
TICLE\_DATA', 'LIBRARY\_DATA\_DIRECT', 'GROUP', 'ARMATURE\_DATA',  
'POSE\_DATA', 'BONE\_DATA', 'CONSTRAINT', 'SHAPEKEY\_DATA', 'CON-  
STRAINT\_BONE', 'PACKAGE', 'UGLYPACKAGE', 'BRUSH\_DATA', 'IMAGE\_DATA',  
'FILE', 'FCURVE', 'FONT\_DATA', 'RENDER\_RESULT', 'SURFACE\_DATA',  
'EMPTY\_DATA', 'SETTINGS', 'RENDER\_ANIMATION', 'RENDER\_STILL', 'BOIDS',  
'STRANDS', 'LIBRARY\_DATA\_INDIRECT', 'GREASEPENCIL', 'GROUP\_BONE',  
'GROUP\_VERTEX', 'GROUP\_VCOL', 'GROUP\_UVS', 'RNA', 'RNA\_ADD', 'OUT-  
LINER\_OB\_EMPTY', 'OUTLINER\_OB\_MESH', 'OUTLINER\_OB\_CURVE', 'OUT-  
LINER\_OB\_LATTICE', 'OUTLINER\_OB\_META', 'OUTLINER\_OB\_LAMP', 'OUT-  
LINER\_OB\_CAMERA', 'OUTLINER\_OB\_ARMATURE', 'OUTLINER\_OB\_FONT',  
'OUTLINER\_OB\_SURFACE', 'OUTLINER\_OB\_SPEAKER', 'RESTRICT\_VIEW\_OFF',  
'RESTRICT\_VIEW\_ON', 'RESTRICT\_SELECT\_OFF', 'RESTRICT\_SELECT\_ON', 'RE-  
STRICT\_RENDER\_OFF', 'RESTRICT\_RENDER\_ON', 'OUTLINER\_DATA\_EMPTY',  
'OUTLINER\_DATA\_MESH', 'OUTLINER\_DATA\_CURVE', 'OUT-  
LINER\_DATA\_LATTICE', 'OUTLINER\_DATA\_META', 'OUTLINER\_DATA\_LAMP',  
'OUTLINER\_DATA\_CAMERA', 'OUTLINER\_DATA\_ARMATURE',  
'OUTLINER\_DATA\_FONT', 'OUTLINER\_DATA\_SURFACE', 'OUT-  
LINER\_DATA\_SPEAKER', 'OUTLINER\_DATA\_POSE', 'MESH\_PLANE',  
'MESH\_CUBE', 'MESH\_CIRCLE', 'MESH\_UVSPHERE', 'MESH\_ICOSPHERE',  
'MESH\_GRID', 'MESH\_MONKEY', 'MESH\_CYLINDER', 'MESH\_TORUS',  
'MESH\_CONE', 'LAMP\_POINT', 'LAMP\_SUN', 'LAMP\_SPOT', 'LAMP\_HEMI',  
'LAMP\_AREA', 'META\_PLANE', 'META\_CUBE', 'META\_BALL',  
'META\_ELLIPSOID', 'META\_CAPSULE', 'SURFACE\_NCURVE', 'SUR-  
FACE\_NCIRCLE', 'SURFACE\_NSURFACE', 'SURFACE\_NCYLINDER',  
'SURFACE\_NSPPHERE', 'SURFACE\_NTORUS', 'CURVE\_BEZCURVE',  
'CURVE\_BEZCIRCLE', 'CURVE\_NCURVE', 'CURVE\_NCIRCLE', 'CURVE\_PATH',  
'FORCE\_FORCE', 'FORCE\_WIND', 'FORCE\_VORTEX', 'FORCE\_MAGNETIC',  
'FORCE\_HARMONIC', 'FORCE\_CHARGE', 'FORCE\_LENNARDJONES',  
'FORCE\_TEXTURE', 'FORCE\_CURVE', 'FORCE\_BOID', 'FORCE\_TURBULENCE',  
'FORCE\_DRAG', 'MODIFIER', 'MOD\_WAVE', 'MOD\_BUILD', 'MOD\_DECIM',  
'MOD\_MIRROR', 'MOD\_SOFT', 'MOD\_SUBSURF', 'HOOK', 'MOD\_PHYSICS',  
'MOD\_PARTICLES', 'MOD\_BOOLEAN', 'MOD\_EDGESPLIT', 'MOD\_ARRAY',  
'MOD\_UVPROJECT', 'MOD\_DISPLACE', 'MOD\_CURVE', 'MOD\_LATTICE',  
'CONSTRAINT\_DATA', 'MOD\_ARMATURE', 'MOD\_SHRINKWRAP',  
'MOD\_CAST', 'MOD\_MESHDEFORM', 'MOD\_BEVEL', 'MOD\_SMOOTH',

'MOD\_SIMPLEDEFORM', 'MOD\_MASK', 'MOD\_CLOTH', 'MOD\_EXPLODE', 'MOD\_FLUIDSIM', 'MOD\_MULTIRES', 'MOD\_SMOKE', 'MOD\_SOLIDIFY', 'MOD\_SCREW', 'MOD\_VERTEX\_WEIGHT', 'REC', 'PLAY', 'FF', 'REW', 'PAUSE', 'PREV\_KEYFRAME', 'NEXT\_KEYFRAME', 'PLAY\_AUDIO', 'PLAY\_REVERSE', 'PREVIEW\_RANGE', 'PMARKER\_ACT', 'PMARKER\_SEL', 'PMARKER', 'MARKER\_HLT', 'MARKER', 'SPACE2', 'SPACE3', 'KEYINGSET', 'KEY\_DEHLT', 'KEY\_HLT', 'MUTE\_IPO\_OFF', 'MUTE\_IPO\_ON', 'VISIBLE\_IPO\_OFF', 'VISIBLE\_IPO\_ON', 'DRIVER', 'SOLO\_OFF', 'SOLO\_ON', 'VERTEXSEL', 'EDGESEL', 'FACESEL', 'ROTATE', 'CURSOR', 'ROTATECOLLECTION', 'ROTATECENTER', 'ROACTIVE', 'ALIGN', 'SMOOTHCURVE', 'SPHERECURVE', 'ROOTCURVE', 'SHARPCURVE', 'LINCURVE', 'NOCURVE', 'RNDCURVE', 'PROP\_OFF', 'PROP\_ON', 'PROP\_CON', 'PARTICLE\_POINT', 'PARTICLE\_TIP', 'PARTICLE\_PATH', 'MAN\_TRANS', 'MAN\_ROT', 'MAN\_SCALE', 'MANIPUL', 'SNAP\_OFF', 'SNAP\_ON', 'SNAP\_NORMAL', 'SNAP\_INCREMENT', 'SNAP\_VERTEX', 'SNAP\_EDGE', 'SNAP\_FACE', 'SNAP\_VOLUME', 'STICKY\_UVS\_LOC', 'STICKY\_UVS\_DISABLE', 'STICKY\_UVS\_VERT', 'CLIPUV\_DEHLT', 'CLIPUV\_HLT', 'SNAP\_PEEL\_OBJECT', 'GRID', 'PASTEDOWN', 'COPYDOWN', 'PASTEFLIPUP', 'PASTEFLIPDOWN', 'SNAP\_SURFACE', 'RETOPO', 'UV\_VERTEXSEL', 'UV\_EDGESEL', 'UV\_FACESEL', 'UV\_ISLANDSEL', 'UV\_SYNC\_SELECT', 'BBOX', 'WIRE', 'SOLID', 'SMOOTH', 'POTATO', 'ORTHO', 'LOCKVIEW\_OFF', 'LOCKVIEW\_ON', 'AXIS\_SIDE', 'AXIS\_FRONT', 'AXIS\_TOP', 'NDOF\_DOM', 'NDOF\_TURN', 'NDOF\_FLY', 'NDOF\_TRANS', 'LAYER\_USED', 'LAYER\_ACTIVE', 'SORTALPHA', 'SORTBYEXT', 'SORTTIME', 'SORTSIZE', 'LONGDISPLAY', 'SHORTDISPLAY', 'GHOST', 'IMGDISPLAY', 'BOOKMARKS', 'FONTPREVIEW', 'FILTER', 'NEWFOLDER', 'FILE\_PARENT', 'FILE\_REFRESH', 'FILE\_FOLDER', 'FILE\_BLANK', 'FILE\_BLEND', 'FILE\_IMAGE', 'FILE\_MOVIE', 'FILE\_SCRIPT', 'FILE\_SOUND', 'FILE\_FONT', 'BACK', 'FORWARD', 'DISK\_DRIVE', 'MATPLANE', 'MATSPHERE', 'MATCUBE', 'MONKEY', 'HAIR', 'ALIASED', 'ANTIALIASED', 'MAT\_SPHERE\_SKY', 'WORDWRAP\_OFF', 'WORDWRAP\_ON', 'SYNTAX\_OFF', 'SYNTAX\_ON', 'LINENUMBERS\_OFF', 'LINENUMBERS\_ON', 'SCRIPTPLUGINS', 'SEQ\_SEQUENCER', 'SEQ\_PREVIEW', 'SEQ\_LUMA\_WAVEFORM', 'SEQ\_CHROMA\_SCOPE', 'SEQ\_HISTOGRAM', 'SEQ\_SPLITVIEW', 'IMAGE\_RGB', 'IMAGE\_RGB\_ALPHA', 'IMAGE\_ALPHA', 'IMAGE\_ZDEPTH', 'IMAGEFILE', 'BRUSH\_ADD', 'BRUSH\_BLOB', 'BRUSH\_BLUR', 'BRUSH\_CLAY', 'BRUSH\_CLONE', 'BRUSH\_CREASE', 'BRUSH\_DARKEN', 'BRUSH\_FILL', 'BRUSH\_FLATTEN', 'BRUSH\_GRAB', 'BRUSH\_INFLATE', 'BRUSH\_LAYER', 'BRUSH\_LIGHTEN', 'BRUSH\_MIX', 'BRUSH\_MULTIPLY', 'BRUSH\_NUDGE', 'BRUSH\_PINCH', 'BRUSH\_SCRAPE', 'BRUSH\_SCULPT\_DRAW', 'BRUSH\_SMEAR', 'BRUSH\_SMOOTH', 'BRUSH\_SNAKE\_HOOK', 'BRUSH\_SOFTEN', 'BRUSH\_SUBTRACT', 'BRUSH\_TEXDRAW', 'BRUSH\_THUMB', 'BRUSH\_ROTATE', 'BRUSH\_VERTEXDRAW', 'VIEW3D\_VEC', 'EDIT\_VEC', 'EDITMODE\_DEHLT', 'EDITMODE\_HLT', 'DISCLOSURE\_TRI\_RIGHT\_VEC', 'DISCLOSURE\_TRI\_DOWN\_VEC', 'MOVE\_UP\_VEC', 'MOVE\_DOWN\_VEC', 'X\_VEC', 'SMALL\_TRI\_RIGHT\_VEC'], (*optional*) – Icon, Override automatic icon of the item

**prop\_enum** (*data*, *property*, *value*, *text*=" ", *icon*='NONE')

prop\_enum

#### Parameters

- **data** (*AnyType*, (never None)) – Data from which to take property
- **property** (*string*) – Identifier of property in data
- **value** (*string*) – Enum property value
- **text** (*string*, (*optional*)) – Override automatic text of the item

- **icon** (enum in [`'NONE'`, `'QUESTION'`, `'ERROR'`, `'CANCEL'`, `'TRIA_RIGHT'`, `'TRIA_DOWN'`, `'TRIA_LEFT'`, `'TRIA_UP'`, `'ARROW_LEFTRIGHT'`, `'PLUS'`, `'DISCLOSURE_TRI_DOWN'`, `'DISCLOSURE_TRI_RIGHT'`, `'RADIOBUT_OFF'`, `'RADIOBUT_ON'`, `'MENU_PANEL'`, `'BLENDER'`, `'DOT'`, `'X'`, `'GO_LEFT'`, `'PLUG'`, `'UI'`, `'NODE'`, `'NODE_SEL'`, `'FULLSCREEN'`, `'SPLITSCREEN'`, `'RIGHTARROW_THIN'`, `'BORDERMOVE'`, `'VIEWZOOM'`, `'ZOOMIN'`, `'ZOOMOUT'`, `'PANEL_CLOSE'`, `'COPY_ID'`, `'EYEDROPPER'`, `'LINK_AREA'`, `'AUTO'`, `'CHECKBOX_DEHLT'`, `'CHECKBOX_HLT'`, `'UNLOCKED'`, `'LOCKED'`, `'UNPINNED'`, `'PINNED'`, `'SCREEN_BACK'`, `'RIGHTARROW'`, `'DOWNARROW_HLT'`, `'DOTSUP'`, `'DOTSDOWN'`, `'LINK'`, `'INLINK'`, `'PLUGIN'`, `'HELP'`, `'GHOST_ENABLED'`, `'COLOR'`, `'LINKED'`, `'UNLINKED'`, `'HAND'`, `'ZOOM_ALL'`, `'ZOOM_SELECTED'`, `'ZOOM_PREVIOUS'`, `'ZOOM_IN'`, `'ZOOM_OUT'`, `'RENDER_REGION'`, `'BORDER_RECT'`, `'BORDER_LASSO'`, `'FREEZE'`, `'STYLUS_PRESSURE'`, `'GHOST_DISABLED'`, `'NEW'`, `'FILE_TICK'`, `'QUIT'`, `'URL'`, `'RECOVER_LAST'`, `'FULLSCREEN_ENTER'`, `'FULLSCREEN_EXIT'`, `'BLANK1'`, `'LAMP'`, `'MATERIAL'`, `'TEXTURE'`, `'ANIM'`, `'WORLD'`, `'SCENE'`, `'EDIT'`, `'GAME'`, `'RADIO'`, `'SCRIPT'`, `'PARTICLES'`, `'PHYSICS'`, `'SPEAKER'`, `'TEXTURE_SHADED'`, `'VIEW3D'`, `'IPO'`, `'OOPS'`, `'BUTS'`, `'FILESEL'`, `'IMAGE_COL'`, `'INFO'`, `'SEQUENCE'`, `'TEXT'`, `'IMASEL'`, `'SOUND'`, `'ACTION'`, `'NLA'`, `'SCRIPTWIN'`, `'TIME'`, `'NODETREE'`, `'LOGIC'`, `'CONSOLE'`, `'PREFERENCES'`, `'ASSET_MANAGER'`, `'OBJECT_DATAMODE'`, `'EDIT_MODE_HLT'`, `'FACESEL_HLT'`, `'VPAIN_T_HLT'`, `'TPAIN_T_HLT'`, `'WPAIN_T_HLT'`, `'SCULPTMODE_HLT'`, `'POSE_HLT'`, `'PARTICLEMODE'`, `'LIGHTPAINT'`, `'SCENE_DATA'`, `'RENDERLAYERS'`, `'WORLD_DATA'`, `'OBJECT_DATA'`, `'MESH_DATA'`, `'CURVE_DATA'`, `'META_DATA'`, `'LATTICE_DATA'`, `'LAMP_DATA'`, `'MATERIAL_DATA'`, `'TEXTURE_DATA'`, `'ANIM_DATA'`, `'CAMERA_DATA'`, `'PARTICLE_DATA'`, `'LIBRARY_DATA_DIRECT'`, `'GROUP'`, `'ARMATURE_DATA'`, `'POSE_DATA'`, `'BONE_DATA'`, `'CONSTRAINT'`, `'SHAPEKEY_DATA'`, `'CONSTRAINT_BONE'`, `'PACKAGE'`, `'UGLYPACKAGE'`, `'BRUSH_DATA'`, `'IMAGE_DATA'`, `'FILE'`, `'FCURVE'`, `'FONT_DATA'`, `'RENDER_RESULT'`, `'SURFACE_DATA'`, `'EMPTY_DATA'`, `'SETTINGS'`, `'RENDER_ANIMATION'`, `'RENDER_STILL'`, `'BOIDS'`, `'STRANDS'`, `'LIBRARY_DATA_INDIRECT'`, `'GREASEPENCIL'`, `'GROUP_BONE'`, `'GROUP_VERTEX'`, `'GROUP_VCOL'`, `'GROUP_UVS'`, `'RNA'`, `'RNA_ADD'`, `'OUTLINER_OB_EMPTY'`, `'OUTLINER_OB_MESH'`, `'OUTLINER_OB_CURVE'`, `'OUTLINER_OB_LATTICE'`, `'OUTLINER_OB_META'`, `'OUTLINER_OB_LAMP'`, `'OUTLINER_OB_CAMERA'`, `'OUTLINER_OB_ARMATURE'`, `'OUTLINER_OB_FONT'`, `'OUTLINER_OB_SURFACE'`, `'OUTLINER_OB_SPEAKER'`, `'RESTRICT_VIEW_OFF'`, `'RESTRICT_VIEW_ON'`, `'RESTRICT_SELECT_OFF'`, `'RESTRICT_SELECT_ON'`, `'RESTRICT_RENDER_OFF'`, `'RESTRICT_RENDER_ON'`, `'OUTLINER_DATA_EMPTY'`, `'OUTLINER_DATA_MESH'`, `'OUTLINER_DATA_CURVE'`, `'OUTLINER_DATA_LATTICE'`, `'OUTLINER_DATA_META'`, `'OUTLINER_DATA_LAMP'`, `'OUTLINER_DATA_CAMERA'`, `'OUTLINER_DATA_ARMATURE'`, `'OUTLINER_DATA_FONT'`, `'OUTLINER_DATA_SURFACE'`, `'OUTLINER_DATA_SPEAKER'`, `'OUTLINER_DATA_POSE'`, `'MESH_PLANE'`, `'MESH_CUBE'`, `'MESH_CIRCLE'`, `'MESH_UVSPHERE'`, `'MESH_ICOSPHERE'`, `'MESH_GRID'`, `'MESH_MONKEY'`, `'MESH_CYLINDER'`, `'MESH_TORUS'`, `'MESH_CONE'`, `'LAMP_POINT'`, `'LAMP_SUN'`, `'LAMP_SPOT'`, `'LAMP_HEMI'`, `'LAMP_AREA'`, `'META_PLANE'`, `'META_CUBE'`, `'META_BALL'`, `'META_ELLIPSOID'`, `'META_CAPSULE'`, `'SURFACE_NCURVE'`, `'SURFACE_NCIRCLE'`, `'SURFACE_NSURFACE'`, `'SURFACE_NCYLINDER'`, `'SURFACE_NSPPHERE'`, `'SURFACE_NTORUS'`, `'CURVE_BEZCURVE'`, `'CURVE_BEZCIRCLE'`, `'CURVE_NCURVE'`, `'CURVE_NCIRCLE'`, `'CURVE_PATH'`, `'FORCE_FORCE'`, `'FORCE_WIND'`, `'FORCE_VORTEX'`, `'FORCE_MAGNETIC'`, `'FORCE_HARMONIC'`, `'FORCE_CHARGE'`, `'FORCE_LENNARDJONES'`, `'FORCE_TEXTURE'`, `'FORCE_CURVE'`, `'FORCE_BOID'`, `'FORCE_TURBULENCE'`,

'FORCE\_DRAG', 'MODIFIER', 'MOD\_WAVE', 'MOD\_BUILD', 'MOD\_DECIM',  
 'MOD\_MIRROR', 'MOD\_SOFT', 'MOD\_SUBSURF', 'HOOK', 'MOD\_PHYSICS',  
 'MOD\_PARTICLES', 'MOD\_BOOLEAN', 'MOD\_EDGESPLIT', 'MOD\_ARRAY',  
 'MOD\_UVPROJECT', 'MOD\_DISPLACE', 'MOD\_CURVE', 'MOD\_LATTICE',  
 'CONSTRAINT\_DATA', 'MOD\_ARMATURE', 'MOD\_SHRINKWRAP',  
 'MOD\_CAST', 'MOD\_MESHDEFORM', 'MOD\_BEVEL', 'MOD\_SMOOTH',  
 'MOD\_SIMPLEDEFORM', 'MOD\_MASK', 'MOD\_CLOTH', 'MOD\_EXPLODE',  
 'MOD\_FLUIDSIM', 'MOD\_MULTIRES', 'MOD\_SMOKE', 'MOD\_SOLIDIFY',  
 'MOD\_SCREW', 'MOD\_VERTEX\_WEIGHT', 'REC', 'PLAY', 'FF', 'REW', 'PAUSE',  
 'PREV\_KEYFRAME', 'NEXT\_KEYFRAME', 'PLAY\_AUDIO', 'PLAY\_REVERSE', 'PRE-  
 VIEW\_RANGE', 'PMARKER\_ACT', 'PMARKER\_SEL', 'PMARKER', 'MARKER\_HLT',  
 'MARKER', 'SPACE2', 'SPACE3', 'KEYINGSET', 'KEY\_DEHLT', 'KEY\_HLT',  
 'MUTE\_IPO\_OFF', 'MUTE\_IPO\_ON', 'VISIBLE\_IPO\_OFF', 'VISIBLE\_IPO\_ON',  
 'DRIVER', 'SOLO\_OFF', 'SOLO\_ON', 'VERTEXSEL', 'EDGESEL', 'FACE-  
 SEL', 'ROTATE', 'CURSOR', 'ROTATECOLLECTION', 'ROTATECENTER', 'RO-  
 TACTIVE', 'ALIGN', 'SMOOTHCURVE', 'SPHERECURVE', 'ROOTCURVE',  
 'SHARPCURVE', 'LINCURVE', 'NOCURVE', 'RNDCURVE', 'PROP\_OFF',  
 'PROP\_ON', 'PROP\_CON', 'PARTICLE\_POINT', 'PARTICLE\_TIP', 'PARTI-  
 CLE\_PATH', 'MAN\_TRANS', 'MAN\_ROT', 'MAN\_SCALE', 'MANIPUL', 'SNAP\_OFF',  
 'SNAP\_ON', 'SNAP\_NORMAL', 'SNAP\_INCREMENT', 'SNAP\_VERTEX',  
 'SNAP\_EDGE', 'SNAP\_FACE', 'SNAP\_VOLUME', 'STICKY\_UVS\_LOC',  
 'STICKY\_UVS\_DISABLE', 'STICKY\_UVS\_VERT', 'CLIPUV\_DEHLT', 'CLIPUV\_HLT',  
 'SNAP\_PEEL\_OBJECT', 'GRID', 'PASTEDOWN', 'COPYDOWN', 'PASTE-  
 FLIPUP', 'PASTEFLIPDOWN', 'SNAP\_SURFACE', 'RETOPO', 'UV\_VERTEXSEL',  
 'UV\_EDGESEL', 'UV\_FACESEL', 'UV\_ISLANDSEL', 'UV\_SYNC\_SELECT',  
 'BBOX', 'WIRE', 'SOLID', 'SMOOTH', 'POTATO', 'ORTHO', 'LOCKVIEW\_OFF',  
 'LOCKVIEW\_ON', 'AXIS\_SIDE', 'AXIS\_FRONT', 'AXIS\_TOP', 'NDOF\_DOM',  
 'NDOF\_TURN', 'NDOF\_FLY', 'NDOF\_TRANS', 'LAYER\_USED', 'LAYER\_ACTIVE',  
 'SORTALPHA', 'SORTBYEXT', 'SORTTIME', 'SORTSIZE', 'LONGDISPLAY',  
 'SHORTDISPLAY', 'GHOST', 'IMGDISPLAY', 'BOOKMARKS', 'FONTPRE-  
 VIEW', 'FILTER', 'NEWFOLDER', 'FILE\_PARENT', 'FILE\_REFRESH',  
 'FILE\_FOLDER', 'FILE\_BLANK', 'FILE\_BLEND', 'FILE\_IMAGE', 'FILE\_MOVIE',  
 'FILE\_SCRIPT', 'FILE\_SOUND', 'FILE\_FONT', 'BACK', 'FORWARD',  
 'DISK\_DRIVE', 'MATPLANE', 'MATSPHERE', 'MATCUBE', 'MONKEY', 'HAIR',  
 'ALIASED', 'ANTIALIASED', 'MAT\_SPHERE\_SKY', 'WORDWRAP\_OFF',  
 'WORDWRAP\_ON', 'SYNTAX\_OFF', 'SYNTAX\_ON', 'LINENUMBERS\_OFF',  
 'LINENUMBERS\_ON', 'SCRIPTPLUGINS', 'SEQ\_SEQUENCER', 'SEQ\_PREVIEW',  
 'SEQ\_LUMA\_WAVEFORM', 'SEQ\_CHROMA\_SCOPE', 'SEQ\_HISTOGRAM',  
 'SEQ\_SPLITVIEW', 'IMAGE\_RGB', 'IMAGE\_RGB\_ALPHA', 'IMAGE\_ALPHA', 'IM-  
 AGE\_ZDEPTH', 'IMAGEFILE', 'BRUSH\_ADD', 'BRUSH\_BLOB', 'BRUSH\_BLUR',  
 'BRUSH\_CLAY', 'BRUSH\_CLONE', 'BRUSH\_CREASE', 'BRUSH\_DARKEN',  
 'BRUSH\_FILL', 'BRUSH\_FLATTEN', 'BRUSH\_GRAB', 'BRUSH\_INFLATE',  
 'BRUSH\_LAYER', 'BRUSH\_LIGHTEN', 'BRUSH\_MIX', 'BRUSH\_MULTIPLY',  
 'BRUSH\_NUDGE', 'BRUSH\_PINCH', 'BRUSH\_SCRAPE', 'BRUSH\_SCULPT\_DRAW',  
 'BRUSH\_SMEAR', 'BRUSH\_SMOOTH', 'BRUSH\_SNAKE\_HOOK',  
 'BRUSH\_SOFTEN', 'BRUSH\_SUBTRACT', 'BRUSH\_TEXDRAW', 'BRUSH\_THUMB',  
 'BRUSH\_ROTATE', 'BRUSH\_VERTEXDRAW', 'VIEW3D\_VEC', 'EDIT\_VEC', 'ED-  
 ITMODE\_DEHLT', 'EDITMODE\_HLT', 'DISCLOSURE\_TRI\_RIGHT\_VEC', 'DIS-  
 CLOSURE\_TRI\_DOWN\_VEC', 'MOVE\_UP\_VEC', 'MOVE\_DOWN\_VEC', 'X\_VEC',  
 'SMALL\_TRI\_RIGHT\_VEC'], (optional) – Icon, Override automatic icon of the item

**prop\_search** (*data, property, search\_data, search\_property, text=""*, *icon='NONE'*)

prop\_search

#### Parameters

- **data** (*AnyType*, (never None)) – Data from which to take property
- **property** (*string*) – Identifier of property in data
- **search\_data** (*AnyType*, (never None)) – Data from which to take collection to search in
- **search\_property** (*string*) – Identifier of search collection property
- **text** (*string*, (*optional*)) – Override automatic text of the item
- **icon** (*enum in* ['NONE', 'QUESTION', 'ERROR', 'CANCEL', 'TRIA\_RIGHT', 'TRIA\_DOWN', 'TRIA\_LEFT', 'TRIA\_UP', 'ARROW\_LEFTRIGHT', 'PLUS', 'DISCLOSURE\_TRI\_DOWN', 'DISCLOSURE\_TRI\_RIGHT', 'RADIOBUT\_OFF', 'RADIOBUT\_ON', 'MENU\_PANEL', 'BLENDER', 'DOT', 'X', 'GO\_LEFT', 'PLUG', 'UI', 'NODE', 'NODE\_SEL', 'FULLSCREEN', 'SPLITSCREEN', 'RIGHTARROW\_THIN', 'BORDERMOVE', 'VIEWZOOM', 'ZOOMIN', 'ZOOMOUT', 'PANEL\_CLOSE', 'COPY\_ID', 'EYEDROPPER', 'LINK\_AREA', 'AUTO', 'CHECKBOX\_DEHLT', 'CHECKBOX\_HLT', 'UNLOCKED', 'LOCKED', 'UNPINNED', 'PINNED', 'SCREEN\_BACK', 'RIGHTARROW', 'DOWNARROW\_HLT', 'DOTSUP', 'DOTSDOWN', 'LINK', 'INLINK', 'PLUGIN', 'HELP', 'GHOST\_ENABLED', 'COLOR', 'LINKED', 'UNLINKED', 'HAND', 'ZOOM\_ALL', 'ZOOM\_SELECTED', 'ZOOM\_PREVIOUS', 'ZOOM\_IN', 'ZOOM\_OUT', 'RENDER\_REGION', 'BORDER\_RECT', 'BORDER\_LASSO', 'FREEZE', 'STYLUS\_PRESSURE', 'GHOST\_DISABLED', 'NEW', 'FILE\_TICK', 'QUIT', 'URL', 'RECOVER\_LAST', 'FULLSCREEN\_ENTER', 'FULLSCREEN\_EXIT', 'BLANK1', 'LAMP', 'MATERIAL', 'TEXTURE', 'ANIM', 'WORLD', 'SCENE', 'EDIT', 'GAME', 'RADIO', 'SCRIPT', 'PARTICLES', 'PHYSICS', 'SPEAKER', 'TEXTURE\_SHADED', 'VIEW3D', 'IPO', 'OOPS', 'BUTS', 'FILESEL', 'IMAGE\_COL', 'INFO', 'SEQUENCE', 'TEXT', 'IMASEL', 'SOUND', 'ACTION', 'NLA', 'SCRIPTWIN', 'TIME', 'NODETREE', 'LOGIC', 'CONSOLE', 'PREFERENCES', 'ASSET\_MANAGER', 'OBJECT\_DATAMODE', 'EDIT\_MODE\_HLT', 'FACESEL\_HLT', 'VPAINT\_HLT', 'TPAINT\_HLT', 'WPAINT\_HLT', 'SCULPTMODE\_HLT', 'POSE\_HLT', 'PARTICLEMODE', 'LIGHTPAINT', 'SCENE\_DATA', 'RENDERLAYERS', 'WORLD\_DATA', 'OBJECT\_DATA', 'MESH\_DATA', 'CURVE\_DATA', 'META\_DATA', 'LATTICE\_DATA', 'LAMP\_DATA', 'MATERIAL\_DATA', 'TEXTURE\_DATA', 'ANIM\_DATA', 'CAMERA\_DATA', 'PARTICLE\_DATA', 'LIBRARY\_DATA\_DIRECT', 'GROUP', 'ARMATURE\_DATA', 'POSE\_DATA', 'BONE\_DATA', 'CONSTRAINT', 'SHAPEKEY\_DATA', 'CONSTRAINT\_BONE', 'PACKAGE', 'UGLYPACKAGE', 'BRUSH\_DATA', 'IMAGE\_DATA', 'FILE', 'FCURVE', 'FONT\_DATA', 'RENDER\_RESULT', 'SURFACE\_DATA', 'EMPTY\_DATA', 'SETTINGS', 'RENDER\_ANIMATION', 'RENDER\_STILL', 'BOIDS', 'STRANDS', 'LIBRARY\_DATA\_INDIRECT', 'GREASEPENCIL', 'GROUP\_BONE', 'GROUP\_VERTEX', 'GROUP\_VCOL', 'GROUP\_UVS', 'RNA', 'RNA\_ADD', 'OUTLINER\_OB\_EMPTY', 'OUTLINER\_OB\_MESH', 'OUTLINER\_OB\_CURVE', 'OUTLINER\_OB\_LATTICE', 'OUTLINER\_OB\_META', 'OUTLINER\_OB\_LAMP', 'OUTLINER\_OB\_CAMERA', 'OUTLINER\_OB\_ARMATURE', 'OUTLINER\_OB\_FONT', 'OUTLINER\_OB\_SURFACE', 'OUTLINER\_OB\_SPEAKER', 'RESTRICT\_VIEW\_OFF', 'RESTRICT\_VIEW\_ON', 'RESTRICT\_SELECT\_OFF', 'RESTRICT\_SELECT\_ON', 'RESTRICT\_RENDER\_OFF', 'RESTRICT\_RENDER\_ON', 'OUTLINER\_DATA\_EMPTY', 'OUTLINER\_DATA\_MESH', 'OUTLINER\_DATA\_CURVE', 'OUTLINER\_DATA\_LATTICE', 'OUTLINER\_DATA\_META', 'OUTLINER\_DATA\_LAMP', 'OUTLINER\_DATA\_CAMERA', 'OUTLINER\_DATA\_ARMATURE', 'OUTLINER\_DATA\_FONT', 'OUTLINER\_DATA\_SURFACE', 'OUTLINER\_DATA\_SPEAKER', 'OUTLINER\_DATA\_POSE', 'MESH\_PLANE', 'MESH\_CUBE', 'MESH\_CIRCLE', 'MESH\_UVSPHERE', 'MESH\_ICOSPHERE', 'MESH\_GRID', 'MESH\_MONKEY', 'MESH\_CYLINDER', 'MESH\_TORUS', 'MESH\_CONE', 'LAMP\_POINT', 'LAMP\_SUN', 'LAMP\_SPOT', 'LAMP\_HEMI',



'LAMP\_AREA', 'META\_PLANE', 'META\_CUBE', 'META\_BALL',  
 'META\_ELLIPSOID', 'META\_CAPSULE', 'SURFACE\_NCURVE', 'SUR-  
 FACE\_NCIRCLE', 'SURFACE\_NSURFACE', 'SURFACE\_NCYLINDER',  
 'SURFACE\_NSHERE', 'SURFACE\_NTORUS', 'CURVE\_BEZCURVE',  
 'CURVE\_BEZCIRCLE', 'CURVE\_NCURVE', 'CURVE\_NCIRCLE', 'CURVE\_PATH',  
 'FORCE\_FORCE', 'FORCE\_WIND', 'FORCE\_VORTEX', 'FORCE\_MAGNETIC',  
 'FORCE\_HARMONIC', 'FORCE\_CHARGE', 'FORCE\_LENNARDJONES',  
 'FORCE\_TEXTURE', 'FORCE\_CURVE', 'FORCE\_BOID', 'FORCE\_TURBULENCE',  
 'FORCE\_DRAG', 'MODIFIER', 'MOD\_WAVE', 'MOD\_BUILD', 'MOD\_DECIM',  
 'MOD\_MIRROR', 'MOD\_SOFT', 'MOD\_SUBSURF', 'HOOK', 'MOD\_PHYSICS',  
 'MOD\_PARTICLES', 'MOD\_BOOLEAN', 'MOD\_EDGESPLIT', 'MOD\_ARRAY',  
 'MOD\_UVPROJECT', 'MOD\_DISPLACE', 'MOD\_CURVE', 'MOD\_LATTICE',  
 'CONSTRAINT\_DATA', 'MOD\_ARMATURE', 'MOD\_SHRINKWRAP',  
 'MOD\_CAST', 'MOD\_MESHDEFORM', 'MOD\_BEVEL', 'MOD\_SMOOTH',  
 'MOD\_SIMPLEDEFORM', 'MOD\_MASK', 'MOD\_CLOTH', 'MOD\_EXPLODE',  
 'MOD\_FLUIDSIM', 'MOD\_MULTIRES', 'MOD\_SMOKE', 'MOD\_SOLIDIFY',  
 'MOD\_SCREW', 'MOD\_VERTEX\_WEIGHT', 'REC', 'PLAY', 'FF', 'REW', 'PAUSE',  
 'PREV\_KEYFRAME', 'NEXT\_KEYFRAME', 'PLAY\_AUDIO', 'PLAY\_REVERSE', 'PRE-  
 VIEW\_RANGE', 'PMARKER\_ACT', 'PMARKER\_SEL', 'PMARKER', 'MARKER\_HLT',  
 'MARKER', 'SPACE2', 'SPACE3', 'KEYINGSET', 'KEY\_DEHLT', 'KEY\_HLT',  
 'MUTE\_IPO\_OFF', 'MUTE\_IPO\_ON', 'VISIBLE\_IPO\_OFF', 'VISIBLE\_IPO\_ON',  
 'DRIVER', 'SOLO\_OFF', 'SOLO\_ON', 'VERTEXSEL', 'EDGESEL', 'FACE-  
 SEL', 'ROTATE', 'CURSOR', 'ROTATECOLLECTION', 'ROTATECENTER', 'RO-  
 TACTIVE', 'ALIGN', 'SMOOTHCURVE', 'SPHERECURVE', 'ROOTCURVE',  
 'SHARPCURVE', 'LINCURVE', 'NOCURVE', 'RNDCURVE', 'PROP\_OFF',  
 'PROP\_ON', 'PROP\_CON', 'PARTICLE\_POINT', 'PARTICLE\_TIP', 'PARTI-  
 CLE\_PATH', 'MAN\_TRANS', 'MAN\_ROT', 'MAN\_SCALE', 'MANIPUL', 'SNAP\_OFF',  
 'SNAP\_ON', 'SNAP\_NORMAL', 'SNAP\_INCREMENT', 'SNAP\_VERTEX',  
 'SNAP\_EDGE', 'SNAP\_FACE', 'SNAP\_VOLUME', 'STICKY\_UVS\_LOC',  
 'STICKY\_UVS\_DISABLE', 'STICKY\_UVS\_VERT', 'CLIPUV\_DEHLT', 'CLIPUV\_HLT',  
 'SNAP\_PEEL\_OBJECT', 'GRID', 'PASTEDOWN', 'COPYDOWN', 'PASTE-  
 FLIPUP', 'PASTEFLIPDOWN', 'SNAP\_SURFACE', 'RETOPO', 'UV\_VERTEXSEL',  
 'UV\_EDGESEL', 'UV\_FACESEL', 'UV\_ISLANDSEL', 'UV\_SYNC\_SELECT',  
 'BBOX', 'WIRE', 'SOLID', 'SMOOTH', 'POTATO', 'ORTHO', 'LOCKVIEW\_OFF',  
 'LOCKVIEW\_ON', 'AXIS\_SIDE', 'AXIS\_FRONT', 'AXIS\_TOP', 'NDOF\_DOM',  
 'NDOF\_TURN', 'NDOF\_FLY', 'NDOF\_TRANS', 'LAYER\_USED', 'LAYER\_ACTIVE',  
 'SORTALPHA', 'SORTBYEXT', 'SORTTIME', 'SORTSIZE', 'LONGDISPLAY',  
 'SHORTDISPLAY', 'GHOST', 'IMGDISPLAY', 'BOOKMARKS', 'FONTPRE-  
 VIEW', 'FILTER', 'NEWFOLDER', 'FILE\_PARENT', 'FILE\_REFRESH',  
 'FILE\_FOLDER', 'FILE\_BLANK', 'FILE\_BLEND', 'FILE\_IMAGE', 'FILE\_MOVIE',  
 'FILE\_SCRIPT', 'FILE\_SOUND', 'FILE\_FONT', 'BACK', 'FORWARD',  
 'DISK\_DRIVE', 'MATPLANE', 'MATSPHERE', 'MATCUBE', 'MONKEY', 'HAIR',  
 'ALIASED', 'ANTIALIASED', 'MAT\_SPHERE\_SKY', 'WORDWRAP\_OFF',  
 'WORDWRAP\_ON', 'SYNTAX\_OFF', 'SYNTAX\_ON', 'LINENUMBERS\_OFF',  
 'LINENUMBERS\_ON', 'SCRIPTPLUGINS', 'SEQ\_SEQUENCER', 'SEQ\_PREVIEW',  
 'SEQ\_LUMA\_WAVEFORM', 'SEQ\_CHROMA\_SCOPE', 'SEQ\_HISTOGRAM',  
 'SEQ\_SPLITVIEW', 'IMAGE\_RGB', 'IMAGE\_RGB\_ALPHA', 'IMAGE\_ALPHA', 'IM-  
 AGE\_ZDEPTH', 'IMAGEFILE', 'BRUSH\_ADD', 'BRUSH\_BLOB', 'BRUSH\_BLUR',  
 'BRUSH\_CLAY', 'BRUSH\_CLONE', 'BRUSH\_CREASE', 'BRUSH\_DARKEN',  
 'BRUSH\_FILL', 'BRUSH\_FLATTEN', 'BRUSH\_GRAB', 'BRUSH\_INFLATE',  
 'BRUSH\_LAYER', 'BRUSH\_LIGHTEN', 'BRUSH\_MIX', 'BRUSH\_MULTIPLY',  
 'BRUSH\_NUDGE', 'BRUSH\_PINCH', 'BRUSH\_SCRAPE', 'BRUSH\_SCULPT\_DRAW',  
 'BRUSH\_SMEAR', 'BRUSH\_SMOOTH', 'BRUSH\_SNAKE\_HOOK',  
 'BRUSH\_SOFTEN', 'BRUSH\_SUBTRACT', 'BRUSH\_TEXDRAW', 'BRUSH\_THUMB',

‘BRUSH\_ROTATE’, ‘BRUSH\_VERTEXDRAW’, ‘VIEW3D\_VEC’, ‘EDIT\_VEC’, ‘EDITMODE\_DEHLT’, ‘EDITMODE\_HLT’, ‘DISCLOSURE\_TRI\_RIGHT\_VEC’, ‘DISCLOSURE\_TRI\_DOWN\_VEC’, ‘MOVE\_UP\_VEC’, ‘MOVE\_DOWN\_VEC’, ‘X\_VEC’, ‘SMALL\_TRI\_RIGHT\_VEC’], (optional)) – Icon, Override automatic icon of the item

**operator** (*operator*, *text*=”“, *icon*=‘NONE’, *emboss*=True)

Item. Places a button into the layout to call an Operator

#### Parameters

- **operator** (*string*) – Identifier of the operator
- **text** (*string*, (optional)) – Override automatic text of the item
- **icon** (*enum in* [‘NONE’, ‘QUESTION’, ‘ERROR’, ‘CANCEL’, ‘TRIA\_RIGHT’, ‘TRIA\_DOWN’, ‘TRIA\_LEFT’, ‘TRIA\_UP’, ‘ARROW\_LEFTRIGHT’, ‘PLUS’, ‘DISCLOSURE\_TRI\_DOWN’, ‘DISCLOSURE\_TRI\_RIGHT’, ‘RADIOBUT\_OFF’, ‘RADIOBUT\_ON’, ‘MENU\_PANEL’, ‘BLENDER’, ‘DOT’, ‘X’, ‘GO\_LEFT’, ‘PLUG’, ‘UI’, ‘NODE’, ‘NODE\_SEL’, ‘FULLSCREEN’, ‘SPLITSCREEN’, ‘RIGHTARROW\_THIN’, ‘BORDERMOVE’, ‘VIEWZOOM’, ‘ZOOMIN’, ‘ZOOMOUT’, ‘PANEL\_CLOSE’, ‘COPY\_ID’, ‘EYEDROPPER’, ‘LINK\_AREA’, ‘AUTO’, ‘CHECKBOX\_DEHLT’, ‘CHECKBOX\_HLT’, ‘UNLOCKED’, ‘LOCKED’, ‘UNPINNED’, ‘PINNED’, ‘SCREEN\_BACK’, ‘RIGHTARROW’, ‘DOWNARROW\_HLT’, ‘DOTSUP’, ‘DOTSDOWN’, ‘LINK’, ‘INLINK’, ‘PLUGIN’, ‘HELP’, ‘GHOST\_ENABLED’, ‘COLOR’, ‘LINKED’, ‘UNLINKED’, ‘HAND’, ‘ZOOM\_ALL’, ‘ZOOM\_SELECTED’, ‘ZOOM\_PREVIOUS’, ‘ZOOM\_IN’, ‘ZOOM\_OUT’, ‘RENDER\_REGION’, ‘BORDER\_RECT’, ‘BORDER\_LASSO’, ‘FREEZE’, ‘STYLUS\_PRESSURE’, ‘GHOST\_DISABLED’, ‘NEW’, ‘FILE\_TICK’, ‘QUIT’, ‘URL’, ‘RECOVER\_LAST’, ‘FULLSCREEN\_ENTER’, ‘FULLSCREEN\_EXIT’, ‘BLANK1’, ‘LAMP’, ‘MATERIAL’, ‘TEXTURE’, ‘ANIM’, ‘WORLD’, ‘SCENE’, ‘EDIT’, ‘GAME’, ‘RADIO’, ‘SCRIPT’, ‘PARTICLES’, ‘PHYSICS’, ‘SPEAKER’, ‘TEXTURE\_SHADED’, ‘VIEW3D’, ‘IPO’, ‘OOPS’, ‘BUTS’, ‘FILESEL’, ‘IMAGE\_COL’, ‘INFO’, ‘SEQUENCE’, ‘TEXT’, ‘IMASEL’, ‘SOUND’, ‘ACTION’, ‘NLA’, ‘SCRIPTWIN’, ‘TIME’, ‘NODETREE’, ‘LOGIC’, ‘CONSOLE’, ‘PREFERENCES’, ‘ASSET\_MANAGER’, ‘OBJECT\_DATAMODE’, ‘EDITMODE\_HLT’, ‘FACESEL\_HLT’, ‘VPAINT\_HLT’, ‘TPAINT\_HLT’, ‘WPAINT\_HLT’, ‘SCULPTMODE\_HLT’, ‘POSE\_HLT’, ‘PARTICLEMODE’, ‘LIGHTPAINT’, ‘SCENE\_DATA’, ‘RENDERLAYERS’, ‘WORLD\_DATA’, ‘OBJECT\_DATA’, ‘MESH\_DATA’, ‘CURVE\_DATA’, ‘META\_DATA’, ‘LATTICE\_DATA’, ‘LAMP\_DATA’, ‘MATERIAL\_DATA’, ‘TEXTURE\_DATA’, ‘ANIM\_DATA’, ‘CAMERA\_DATA’, ‘PARTICLE\_DATA’, ‘LIBRARY\_DATA\_DIRECT’, ‘GROUP’, ‘ARMATURE\_DATA’, ‘POSE\_DATA’, ‘BONE\_DATA’, ‘CONSTRAINT’, ‘SHAPEKEY\_DATA’, ‘CONSTRAINT\_BONE’, ‘PACKAGE’, ‘UGLYPACKAGE’, ‘BRUSH\_DATA’, ‘IMAGE\_DATA’, ‘FILE’, ‘FCURVE’, ‘FONT\_DATA’, ‘RENDER\_RESULT’, ‘SURFACE\_DATA’, ‘EMPTY\_DATA’, ‘SETTINGS’, ‘RENDER\_ANIMATION’, ‘RENDER\_STILL’, ‘BOIDS’, ‘STRANDS’, ‘LIBRARY\_DATA\_INDIRECT’, ‘GREASEPENCIL’, ‘GROUP\_BONE’, ‘GROUP\_VERTEX’, ‘GROUP\_VCOL’, ‘GROUP\_UVS’, ‘RNA’, ‘RNA\_ADD’, ‘OUTLINER\_OB\_EMPTY’, ‘OUTLINER\_OB\_MESH’, ‘OUTLINER\_OB\_CURVE’, ‘OUTLINER\_OB\_LATTICE’, ‘OUTLINER\_OB\_META’, ‘OUTLINER\_OB\_LAMP’, ‘OUTLINER\_OB\_CAMERA’, ‘OUTLINER\_OB\_ARMATURE’, ‘OUTLINER\_OB\_FONT’, ‘OUTLINER\_OB\_SURFACE’, ‘OUTLINER\_OB\_SPEAKER’, ‘RESTRICT\_VIEW\_OFF’, ‘RESTRICT\_VIEW\_ON’, ‘RESTRICT\_SELECT\_OFF’, ‘RESTRICT\_SELECT\_ON’, ‘RESTRICT\_RENDER\_OFF’, ‘RESTRICT\_RENDER\_ON’, ‘OUTLINER\_DATA\_EMPTY’, ‘OUTLINER\_DATA\_MESH’, ‘OUTLINER\_DATA\_CURVE’, ‘OUTLINER\_DATA\_LATTICE’, ‘OUTLINER\_DATA\_META’, ‘OUTLINER\_DATA\_LAMP’, ‘OUTLINER\_DATA\_CAMERA’, ‘OUTLINER\_DATA\_ARMATURE’, ‘OUTLINER\_DATA\_FONT’, ‘OUTLINER\_DATA\_SURFACE’, ‘OUT-

LINER\_DATA\_SPEAKER', 'OUTLINER\_DATA\_POSE', 'MESH\_PLANE',  
 'MESH\_CUBE', 'MESH\_CIRCLE', 'MESH\_UVSPHERE', 'MESH\_ICOSPHERE',  
 'MESH\_GRID', 'MESH\_MONKEY', 'MESH\_CYLINDER', 'MESH\_TORUS',  
 'MESH\_CONE', 'LAMP\_POINT', 'LAMP\_SUN', 'LAMP\_SPOT', 'LAMP\_HEMI',  
 'LAMP\_AREA', 'META\_PLANE', 'META\_CUBE', 'META\_BALL',  
 'META\_ELLIPSOID', 'META\_CAPSULE', 'SURFACE\_NCURVE', 'SUR-  
 FACE\_NCIRCLE', 'SURFACE\_NSURFACE', 'SURFACE\_NCYLINDER',  
 'SURFACE\_NSPPHERE', 'SURFACE\_NTORUS', 'CURVE\_BEZCURVE',  
 'CURVE\_BEZCIRCLE', 'CURVE\_NCURVE', 'CURVE\_NCIRCLE', 'CURVE\_PATH',  
 'FORCE\_FORCE', 'FORCE\_WIND', 'FORCE\_VORTEX', 'FORCE\_MAGNETIC',  
 'FORCE\_HARMONIC', 'FORCE\_CHARGE', 'FORCE\_LENNARDJONES',  
 'FORCE\_TEXTURE', 'FORCE\_CURVE', 'FORCE\_BOID', 'FORCE\_TURBULENCE',  
 'FORCE\_DRAG', 'MODIFIER', 'MOD\_WAVE', 'MOD\_BUILD', 'MOD\_DECIM',  
 'MOD\_MIRROR', 'MOD\_SOFT', 'MOD\_SUBSURF', 'HOOK', 'MOD\_PHYSICS',  
 'MOD\_PARTICLES', 'MOD\_BOOLEAN', 'MOD\_EDGESPLIT', 'MOD\_ARRAY',  
 'MOD\_UVPROJECT', 'MOD\_DISPLACE', 'MOD\_CURVE', 'MOD\_LATTICE',  
 'CONSTRAINT\_DATA', 'MOD\_ARMATURE', 'MOD\_SHRINKWRAP',  
 'MOD\_CAST', 'MOD\_MESHDEFORM', 'MOD\_BEVEL', 'MOD\_SMOOTH',  
 'MOD\_SIMPLEDEFORM', 'MOD\_MASK', 'MOD\_CLOTH', 'MOD\_EXPLODE',  
 'MOD\_FLUIDSIM', 'MOD\_MULTIRES', 'MOD\_SMOKE', 'MOD\_SOLIDIFY',  
 'MOD\_SCREW', 'MOD\_VERTEX\_WEIGHT', 'REC', 'PLAY', 'FF', 'REW', 'PAUSE',  
 'PREV\_KEYFRAME', 'NEXT\_KEYFRAME', 'PLAY\_AUDIO', 'PLAY\_REVERSE', 'PRE-  
 VIEW\_RANGE', 'PMARKER\_ACT', 'PMARKER\_SEL', 'PMARKER', 'MARKER\_HLT',  
 'MARKER', 'SPACE2', 'SPACE3', 'KEYINGSET', 'KEY\_DEHLT', 'KEY\_HLT',  
 'MUTE\_IPO\_OFF', 'MUTE\_IPO\_ON', 'VISIBLE\_IPO\_OFF', 'VISIBLE\_IPO\_ON',  
 'DRIVER', 'SOLO\_OFF', 'SOLO\_ON', 'VERTEXSEL', 'EDGESEL', 'FACE-  
 SEL', 'ROTATE', 'CURSOR', 'ROTATECOLLECTION', 'ROTATECENTER', 'RO-  
 TACTIVE', 'ALIGN', 'SMOOTHCURVE', 'SPHERECURVE', 'ROOTCURVE',  
 'SHARPCURVE', 'LINCURVE', 'NOCURVE', 'RNDCURVE', 'PROP\_OFF',  
 'PROP\_ON', 'PROP\_CON', 'PARTICLE\_POINT', 'PARTICLE\_TIP', 'PARTI-  
 CLE\_PATH', 'MAN\_TRANS', 'MAN\_ROT', 'MAN\_SCALE', 'MANIPUL', 'SNAP\_OFF',  
 'SNAP\_ON', 'SNAP\_NORMAL', 'SNAP\_INCREMENT', 'SNAP\_VERTEX',  
 'SNAP\_EDGE', 'SNAP\_FACE', 'SNAP\_VOLUME', 'STICKY\_UVS\_LOC',  
 'STICKY\_UVS\_DISABLE', 'STICKY\_UVS\_VERT', 'CLIPUV\_DEHLT', 'CLIPUV\_HLT',  
 'SNAP\_PEEL\_OBJECT', 'GRID', 'PASTEDOWN', 'COPYDOWN', 'PASTE-  
 FLIPUP', 'PASTEFLIPDOWN', 'SNAP\_SURFACE', 'RETOPO', 'UV\_VERTEXSEL',  
 'UV\_EDGESEL', 'UV\_FACESEL', 'UV\_ISLANDSEL', 'UV\_SYNC\_SELECT',  
 'BBOX', 'WIRE', 'SOLID', 'SMOOTH', 'POTATO', 'ORTHO', 'LOCKVIEW\_OFF',  
 'LOCKVIEW\_ON', 'AXIS\_SIDE', 'AXIS\_FRONT', 'AXIS\_TOP', 'NDOF\_DOM',  
 'NDOF\_TURN', 'NDOF\_FLY', 'NDOF\_TRANS', 'LAYER\_USED', 'LAYER\_ACTIVE',  
 'SORTALPHA', 'SORTBYEXT', 'SORTTIME', 'SORTSIZE', 'LONGDISPLAY',  
 'SHORTDISPLAY', 'GHOST', 'IMGDISPLAY', 'BOOKMARKS', 'FONTPRE-  
 VIEW', 'FILTER', 'NEWFOLDER', 'FILE\_PARENT', 'FILE\_REFRESH',  
 'FILE\_FOLDER', 'FILE\_BLANK', 'FILE\_BLEND', 'FILE\_IMAGE', 'FILE\_MOVIE',  
 'FILE\_SCRIPT', 'FILE\_SOUND', 'FILE\_FONT', 'BACK', 'FORWARD',  
 'DISK\_DRIVE', 'MATPLANE', 'MATSPHERE', 'MATCUBE', 'MONKEY', 'HAIR',  
 'ALIASED', 'ANTIALIASED', 'MAT\_SPHERE\_SKY', 'WORDWRAP\_OFF',  
 'WORDWRAP\_ON', 'SYNTAX\_OFF', 'SYNTAX\_ON', 'LINENUMBERS\_OFF',  
 'LINENUMBERS\_ON', 'SCRIPTPLUGINS', 'SEQ\_SEQUENCER', 'SEQ\_PREVIEW',  
 'SEQ\_LUMA\_WAVEFORM', 'SEQ\_CHROMA\_SCOPE', 'SEQ\_HISTOGRAM',  
 'SEQ\_SPLITVIEW', 'IMAGE\_RGB', 'IMAGE\_RGB\_ALPHA', 'IMAGE\_ALPHA', 'IM-  
 AGE\_ZDEPTH', 'IMAGEFILE', 'BRUSH\_ADD', 'BRUSH\_BLOB', 'BRUSH\_BLUR',  
 'BRUSH\_CLAY', 'BRUSH\_CLONE', 'BRUSH\_CREASE', 'BRUSH\_DARKEN',  
 'BRUSH\_FILL', 'BRUSH\_FLATTEN', 'BRUSH\_GRAB', 'BRUSH\_INFLATE',

`'BRUSH_LAYER', 'BRUSH_LIGHTEN', 'BRUSH_MIX', 'BRUSH_MULTIPLY', 'BRUSH_NUDGE', 'BRUSH_PINCH', 'BRUSH_SCRAPPE', 'BRUSH_SCULPT_DRAW', 'BRUSH_SMEAR', 'BRUSH_SMOOTH', 'BRUSH_SNAKE_HOOK', 'BRUSH_SOFTEN', 'BRUSH_SUBTRACT', 'BRUSH_TEXDRAW', 'BRUSH_THUMB', 'BRUSH_ROTATE', 'BRUSH_VERTEXDRAW', 'VIEW3D_VEC', 'EDIT_VEC', 'EDITMODE_DEHLT', 'EDITMODE_HLT', 'DISCLOSURE_TRI_RIGHT_VEC', 'DISCLOSURE_TRI_DOWN_VEC', 'MOVE_UP_VEC', 'MOVE_DOWN_VEC', 'X_VEC', 'SMALL_TRI_RIGHT_VEC']`, (*optional*) – Icon, Override automatic icon of the item

- **emboss** (*boolean, (optional)*) – Draw the button itself, just the icon/text

**Returns** Operator properties to fill in, return when ‘properties’ is set to true

**Return type** `OperatorProperties`

**operator\_enum** (*operator, property*)

`operator_enum`

#### Parameters

- **operator** (*string*) – Identifier of the operator
- **property** (*string*) – Identifier of property in operator

**operator\_menu\_enum** (*operator, property, text=" ", icon='NONE'*)

`operator_menu_enum`

#### Parameters

- **operator** (*string*) – Identifier of the operator
- **property** (*string*) – Identifier of property in operator
- **text** (*string, (optional)*) – Override automatic text of the item
- **icon** (*enum in ['NONE', 'QUESTION', 'ERROR', 'CANCEL', 'TRIA\_RIGHT', 'TRIA\_DOWN', 'TRIA\_LEFT', 'TRIA\_UP', 'ARROW\_LEFTRIGHT', 'PLUS', 'DISCLOSURE\_TRI\_DOWN', 'DISCLOSURE\_TRI\_RIGHT', 'RADIOBUT\_OFF', 'RADIOBUT\_ON', 'MENU\_PANEL', 'BLENDER', 'DOT', 'X', 'GO\_LEFT', 'PLUG', 'UP', 'NODE', 'NODE\_SEL', 'FULLSCREEN', 'SPLITSCREEN', 'RIGHTARROW\_THIN', 'BORDERMOVE', 'VIEWZOOM', 'ZOOMIN', 'ZOOMOUT', 'PANEL\_CLOSE', 'COPY\_ID', 'EYEDROPPER', 'LINK\_AREA', 'AUTO', 'CHECKBOX\_DEHLT', 'CHECKBOX\_HLT', 'UNLOCKED', 'LOCKED', 'UNPINNED', 'PINNED', 'SCREEN\_BACK', 'RIGHTARROW', 'DOWNARROW\_HLT', 'DOTSUP', 'DOTSDOWN', 'LINK', 'INLINK', 'PLUGIN', 'HELP', 'GHOST\_ENABLED', 'COLOR', 'LINKED', 'UNLINKED', 'HAND', 'ZOOM\_ALL', 'ZOOM\_SELECTED', 'ZOOM\_PREVIOUS', 'ZOOM\_IN', 'ZOOM\_OUT', 'RENDER\_REGION', 'BORDER\_RECT', 'BORDER\_LASSO', 'FREEZE', 'STYLUS\_PRESSURE', 'GHOST\_DISABLED', 'NEW', 'FILE\_TICK', 'QUIT', 'URL', 'RECOVER\_LAST', 'FULLSCREEN\_ENTER', 'FULLSCREEN\_EXIT', 'BLANK1', 'LAMP', 'MATERIAL', 'TEXTURE', 'ANIM', 'WORLD', 'SCENE', 'EDIT', 'GAME', 'RADIO', 'SCRIPT', 'PARTICLES', 'PHYSICS', 'SPEAKER', 'TEXTURE\_SHADED', 'VIEW3D', 'IPO', 'OOPS', 'BUTS', 'FILESEL', 'IMAGE\_COL', 'INFO', 'SEQUENCE', 'TEXT', 'IMASEL', 'SOUND', 'ACTION', 'NLA', 'SCRIPTWIN', 'TIME', 'NODETREE', 'LOGIC', 'CONSOLE', 'PREFERENCES', 'ASSET\_MANAGER', 'OBJECT\_DATAMODE', 'EDITMODE\_HLT', 'FACESEL\_HLT', 'VPAINT\_HLT', 'TPAINT\_HLT', 'WPAINT\_HLT', 'SCULPTMODE\_HLT', 'POSE\_HLT', 'PARTICLEMODE', 'LIGHTPAINT', 'SCENE\_DATA', 'RENDERLAYERS', 'WORLD\_DATA', 'OBJECT\_DATA', 'MESH\_DATA', 'CURVE\_DATA', 'META\_DATA', 'LATTICE\_DATA', 'LAMP\_DATA',*

'MATERIAL\_DATA', 'TEXTURE\_DATA', 'ANIM\_DATA', 'CAMERA\_DATA', 'PARTICLE\_DATA', 'LIBRARY\_DATA\_DIRECT', 'GROUP', 'ARMATURE\_DATA', 'POSE\_DATA', 'BONE\_DATA', 'CONSTRAINT', 'SHAPEKEY\_DATA', 'CONSTRAINT\_BONE', 'PACKAGE', 'UGLYPACKAGE', 'BRUSH\_DATA', 'IMAGE\_DATA', 'FILE', 'FCURVE', 'FONT\_DATA', 'RENDER\_RESULT', 'SURFACE\_DATA', 'EMPTY\_DATA', 'SETTINGS', 'RENDER\_ANIMATION', 'RENDER\_STILL', 'BOIDS', 'STRANDS', 'LIBRARY\_DATA\_INDIRECT', 'GREASEPENCIL', 'GROUP\_BONE', 'GROUP\_VERTEX', 'GROUP\_VCOL', 'GROUP\_UVS', 'RNA', 'RNA\_ADD', 'OUTLINER\_OB\_EMPTY', 'OUTLINER\_OB\_MESH', 'OUTLINER\_OB\_CURVE', 'OUTLINER\_OB\_LATTICE', 'OUTLINER\_OB\_META', 'OUTLINER\_OB\_LAMP', 'OUTLINER\_OB\_CAMERA', 'OUTLINER\_OB\_ARMATURE', 'OUTLINER\_OB\_FONT', 'OUTLINER\_OB\_SURFACE', 'OUTLINER\_OB\_SPEAKER', 'RESTRICT\_VIEW\_OFF', 'RESTRICT\_VIEW\_ON', 'RESTRICT\_SELECT\_OFF', 'RESTRICT\_SELECT\_ON', 'RESTRICT\_RENDER\_OFF', 'RESTRICT\_RENDER\_ON', 'OUTLINER\_DATA\_EMPTY', 'OUTLINER\_DATA\_MESH', 'OUTLINER\_DATA\_CURVE', 'OUTLINER\_DATA\_LATTICE', 'OUTLINER\_DATA\_META', 'OUTLINER\_DATA\_LAMP', 'OUTLINER\_DATA\_CAMERA', 'OUTLINER\_DATA\_ARMATURE', 'OUTLINER\_DATA\_FONT', 'OUTLINER\_DATA\_SURFACE', 'OUTLINER\_DATA\_SPEAKER', 'OUTLINER\_DATA\_POSE', 'MESH\_PLANE', 'MESH\_CUBE', 'MESH\_CIRCLE', 'MESH\_UVSPHERE', 'MESH\_ICOSPHERE', 'MESH\_GRID', 'MESH\_MONKEY', 'MESH\_CYLINDER', 'MESH\_TORUS', 'MESH\_CONE', 'LAMP\_POINT', 'LAMP\_SUN', 'LAMP\_SPOT', 'LAMP\_HEMI', 'LAMP\_AREA', 'META\_PLANE', 'META\_CUBE', 'META\_BALL', 'META\_ELLIPSOID', 'META\_CAPSULE', 'SURFACE\_NCURVE', 'SURFACE\_NCIRCLE', 'SURFACE\_NSURFACE', 'SURFACE\_NCYLINDER', 'SURFACE\_NSHERE', 'SURFACE\_NTORUS', 'CURVE\_BEZCURVE', 'CURVE\_BEZCIRCLE', 'CURVE\_NCURVE', 'CURVE\_NCIRCLE', 'CURVE\_PATH', 'FORCE\_FORCE', 'FORCE\_WIND', 'FORCE\_VORTEX', 'FORCE\_MAGNETIC', 'FORCE\_HARMONIC', 'FORCE\_CHARGE', 'FORCE\_LENNARDJONES', 'FORCE\_TEXTURE', 'FORCE\_CURVE', 'FORCE\_BOID', 'FORCE\_TURBULENCE', 'FORCE\_DRAG', 'MODIFIER', 'MOD\_WAVE', 'MOD\_BUILD', 'MOD\_DECIM', 'MOD\_MIRROR', 'MOD\_SOFT', 'MOD\_SUBSURF', 'HOOK', 'MOD\_PHYSICS', 'MOD\_PARTICLES', 'MOD\_BOOLEAN', 'MOD\_EDGESPLIT', 'MOD\_ARRAY', 'MOD\_UVPROJECT', 'MOD\_DISPLACE', 'MOD\_CURVE', 'MOD\_LATTICE', 'CONSTRAINT\_DATA', 'MOD\_ARMATURE', 'MOD\_SHRINKWRAP', 'MOD\_CAST', 'MOD\_MESHDEFORM', 'MOD\_BEVEL', 'MOD\_SMOOTH', 'MOD\_SIMPLEDEFORM', 'MOD\_MASK', 'MOD\_CLOTH', 'MOD\_EXPLODE', 'MOD\_FLUIDSIM', 'MOD\_MULTITRES', 'MOD\_SMOKE', 'MOD\_SOLIDIFY', 'MOD\_SCREW', 'MOD\_VERTEX\_WEIGHT', 'REC', 'PLAY', 'FF', 'REW', 'PAUSE', 'PREV\_KEYFRAME', 'NEXT\_KEYFRAME', 'PLAY\_AUDIO', 'PLAY\_REVERSE', 'PREVIEW\_RANGE', 'PMARKER\_ACT', 'PMARKER\_SEL', 'PMARKER', 'MARKER\_HLT', 'MARKER', 'SPACE2', 'SPACE3', 'KEYINGSET', 'KEY\_DEHLT', 'KEY\_HLT', 'MUTE\_IPO\_OFF', 'MUTE\_IPO\_ON', 'VISIBLE\_IPO\_OFF', 'VISIBLE\_IPO\_ON', 'DRIVER', 'SOLO\_OFF', 'SOLO\_ON', 'VERTEXSEL', 'EDGESEL', 'FACESEL', 'ROTATE', 'CURSOR', 'ROTATECOLLECTION', 'ROTATECENTER', 'ROTTACTIVE', 'ALIGN', 'SMOOTHCURVE', 'SPHERECURVE', 'ROOTCURVE', 'SHARPCURVE', 'LINCURVE', 'NOCURVE', 'RNDCURVE', 'PROP\_OFF', 'PROP\_ON', 'PROP\_CON', 'PARTICLE\_POINT', 'PARTICLE\_TIP', 'PARTICLE\_PATH', 'MAN\_TRANS', 'MAN\_ROT', 'MAN\_SCALE', 'MANIPUL', 'SNAP\_OFF', 'SNAP\_ON', 'SNAP\_NORMAL', 'SNAP\_INCREMENT', 'SNAP\_VERTEX', 'SNAP\_EDGE', 'SNAP\_FACE', 'SNAP\_VOLUME', 'STICKY\_UVS\_LOC', 'STICKY\_UVS\_DISABLE', 'STICKY\_UVS\_VERT', 'CLIPUV\_DEHLT', 'CLIPUV\_HLT', 'SNAP\_PEEL\_OBJECT', 'GRID', 'PASTEDOWN', 'COPYDOWN', 'PASTEFLIPUP', 'PASTEFLIPDOWN', 'SNAP\_SURFACE', 'RETOPO', 'UV\_VERTEXSEL',

‘UV\_EDGESEL’, ‘UV\_FACESEL’, ‘UV\_ISLANDSEL’, ‘UV\_SYNC\_SELECT’, ‘BBOX’, ‘WIRE’, ‘SOLID’, ‘SMOOTH’, ‘POTATO’, ‘ORTHO’, ‘LOCKVIEW\_OFF’, ‘LOCKVIEW\_ON’, ‘AXIS\_SIDE’, ‘AXIS\_FRONT’, ‘AXIS\_TOP’, ‘NDOF\_DOM’, ‘NDOF\_TURN’, ‘NDOF\_FLY’, ‘NDOF\_TRANS’, ‘LAYER\_USED’, ‘LAYER\_ACTIVE’, ‘SORTALPHA’, ‘SORTBYEXT’, ‘SORTTIME’, ‘SORTSIZE’, ‘LONGDISPLAY’, ‘SHORTDISPLAY’, ‘GHOST’, ‘IMGDISPLAY’, ‘BOOKMARKS’, ‘FONTPREVIEW’, ‘FILTER’, ‘NEWFOLDER’, ‘FILE\_PARENT’, ‘FILE\_REFRESH’, ‘FILE\_FOLDER’, ‘FILE\_BLANK’, ‘FILE\_BLEND’, ‘FILE\_IMAGE’, ‘FILE\_MOVIE’, ‘FILE\_SCRIPT’, ‘FILE\_SOUND’, ‘FILE\_FONT’, ‘BACK’, ‘FORWARD’, ‘DISK\_DRIVE’, ‘MATPLANE’, ‘MATSPHERE’, ‘MATCUBE’, ‘MONKEY’, ‘HAIR’, ‘ALIASED’, ‘ANTIALIASED’, ‘MAT\_SPHERE\_SKY’, ‘WORDWRAP\_OFF’, ‘WORDWRAP\_ON’, ‘SYNTAX\_OFF’, ‘SYNTAX\_ON’, ‘LINENUMBERS\_OFF’, ‘LINENUMBERS\_ON’, ‘SCRIPTPLUGINS’, ‘SEQ\_SEQUENCER’, ‘SEQ\_PREVIEW’, ‘SEQ\_LUMA\_WAVEFORM’, ‘SEQ\_CHROMA\_SCOPE’, ‘SEQ\_HISTOGRAM’, ‘SEQ\_SPLITVIEW’, ‘IMAGE\_RGB’, ‘IMAGE\_RGB\_ALPHA’, ‘IMAGE\_ALPHA’, ‘IMAGE\_ZDEPTH’, ‘IMAGEFILE’, ‘BRUSH\_ADD’, ‘BRUSH\_BLOB’, ‘BRUSH\_BLUR’, ‘BRUSH\_CLAY’, ‘BRUSH\_CLONE’, ‘BRUSH\_CREASE’, ‘BRUSH\_DARKEN’, ‘BRUSH\_FILL’, ‘BRUSH\_FLATTEN’, ‘BRUSH\_GRAB’, ‘BRUSH\_INFLATE’, ‘BRUSH\_LAYER’, ‘BRUSH\_LIGHTEN’, ‘BRUSH\_MIX’, ‘BRUSH\_MULTIPLY’, ‘BRUSH\_NUDGE’, ‘BRUSH\_PINCH’, ‘BRUSH\_SCRAPE’, ‘BRUSH\_SCULPT\_DRAW’, ‘BRUSH\_SMEAR’, ‘BRUSH\_SMOOTH’, ‘BRUSH\_SNAKE\_HOOK’, ‘BRUSH\_SOFTEN’, ‘BRUSH\_SUBTRACT’, ‘BRUSH\_TEXDRAW’, ‘BRUSH\_THUMB’, ‘BRUSH\_ROTATE’, ‘BRUSH\_VERTEXDRAW’, ‘VIEW3D\_VEC’, ‘EDIT\_VEC’, ‘EDITMODE\_DEHLT’, ‘EDITMODE\_HLT’, ‘DISCLOSURE\_TRI\_RIGHT\_VEC’, ‘DISCLOSURE\_TRI\_DOWN\_VEC’, ‘MOVE\_UP\_VEC’, ‘MOVE\_DOWN\_VEC’, ‘X\_VEC’, ‘SMALL\_TRI\_RIGHT\_VEC’], (optional)) – Icon, Override automatic icon of the item

**label** (*text*=”“, *icon*=’NONE’)

Item. Display text in the layout

#### Parameters

- **text** (*string*, (optional)) – Override automatic text of the item
- **icon** (*enum in* [’NONE’, ‘QUESTION’, ‘ERROR’, ‘CANCEL’, ‘TRIA\_RIGHT’, ‘TRIA\_DOWN’, ‘TRIA\_LEFT’, ‘TRIA\_UP’, ‘ARROW\_LEFTRIGHT’, ‘PLUS’, ‘DISCLOSURE\_TRI\_DOWN’, ‘DISCLOSURE\_TRI\_RIGHT’, ‘RADIOBUT\_OFF’, ‘RADIOBUT\_ON’, ‘MENU\_PANEL’, ‘BLENDER’, ‘DOT’, ‘X’, ‘GO\_LEFT’, ‘PLUG’, ‘UI’, ‘NODE’, ‘NODE\_SEL’, ‘FULLSCREEN’, ‘SPLITSCREEN’, ‘RIGHTARROW\_THIN’, ‘BORDERMOVE’, ‘VIEWZOOM’, ‘ZOOMIN’, ‘ZOOMOUT’, ‘PANEL\_CLOSE’, ‘COPY\_ID’, ‘EYEDROPPER’, ‘LINK\_AREA’, ‘AUTO’, ‘CHECKBOX\_DEHLT’, ‘CHECKBOX\_HLT’, ‘UNLOCKED’, ‘LOCKED’, ‘UNPINNED’, ‘PINNED’, ‘SCREEN\_BACK’, ‘RIGHTARROW’, ‘DOWNARROW\_HLT’, ‘DOTSUP’, ‘DOTSDOWN’, ‘LINK’, ‘INLINK’, ‘PLUGIN’, ‘HELP’, ‘GHOST\_ENABLED’, ‘COLOR’, ‘LINKED’, ‘UNLINKED’, ‘HAND’, ‘ZOOM\_ALL’, ‘ZOOM\_SELECTED’, ‘ZOOM\_PREVIOUS’, ‘ZOOM\_IN’, ‘ZOOM\_OUT’, ‘RENDER\_REGION’, ‘BORDER\_RECT’, ‘BORDER\_LASSO’, ‘FREEZE’, ‘STYLUS\_PRESSURE’, ‘GHOST\_DISABLED’, ‘NEW’, ‘FILE\_TICK’, ‘QUIT’, ‘URL’, ‘RECOVER\_LAST’, ‘FULLSCREEN\_ENTER’, ‘FULLSCREEN\_EXIT’, ‘BLANK1’, ‘LAMP’, ‘MATERIAL’, ‘TEXTURE’, ‘ANIM’, ‘WORLD’, ‘SCENE’, ‘EDIT’, ‘GAME’, ‘RADIO’, ‘SCRIPT’, ‘PARTICLES’, ‘PHYSICS’, ‘SPEAKER’, ‘TEXTURE\_SHADED’, ‘VIEW3D’, ‘IPO’, ‘OOPS’, ‘BUTS’, ‘FILESEL’, ‘IMAGE\_COL’, ‘INFO’, ‘SEQUENCE’, ‘TEXT’, ‘IMASEL’, ‘SOUND’, ‘ACTION’, ‘NLA’, ‘SCRIPTWIN’, ‘TIME’, ‘NODETREE’, ‘LOGIC’, ‘CONSOLE’, ‘PREFERENCES’, ‘ASSET\_MANAGER’, ‘OBJECT\_DATAMODE’, ‘EDITMODE\_HLT’, ‘FACESEL\_HLT’, ‘VPAINT\_HLT’, ‘TPAINT\_HLT’, ‘WPAINT\_HLT’, ‘SCULPTMODE\_HLT’, ‘POSE\_HLT’, ‘PARTICLEMODE’, ‘LIGHTPAINT’,

'SCENE\_DATA', 'RENDERLAYERS', 'WORLD\_DATA', 'OBJECT\_DATA',  
 'MESH\_DATA', 'CURVE\_DATA', 'META\_DATA', 'LATTICE\_DATA', 'LAMP\_DATA',  
 'MATERIAL\_DATA', 'TEXTURE\_DATA', 'ANIM\_DATA', 'CAMERA\_DATA', 'PARTICLE\_DATA',  
 'LIBRARY\_DATA\_DIRECT', 'GROUP', 'ARMATURE\_DATA',  
 'POSE\_DATA', 'BONE\_DATA', 'CONSTRAINT', 'SHAPEKEY\_DATA', 'CONSTRAINT\_BONE',  
 'PACKAGE', 'UGLYPACKAGE', 'BRUSH\_DATA', 'IMAGE\_DATA',  
 'FILE', 'FCURVE', 'FONT\_DATA', 'RENDER\_RESULT', 'SURFACE\_DATA',  
 'EMPTY\_DATA', 'SETTINGS', 'RENDER\_ANIMATION', 'RENDER\_STILL', 'BOIDS',  
 'STRANDS', 'LIBRARY\_DATA\_INDIRECT', 'GREASEPENCIL', 'GROUP\_BONE',  
 'GROUP\_VERTEX', 'GROUP\_VCOL', 'GROUP\_UVS', 'RNA', 'RNA\_ADD', 'OUTLINER\_OB\_EMPTY',  
 'OUTLINER\_OB\_MESH', 'OUTLINER\_OB\_CURVE', 'OUTLINER\_OB\_LATTICE', 'OUTLINER\_OB\_META',  
 'OUTLINER\_OB\_LAMP', 'OUTLINER\_OB\_CAMERA', 'OUTLINER\_OB\_ARMATURE', 'OUTLINER\_OB\_FONT',  
 'OUTLINER\_OB\_SURFACE', 'OUTLINER\_OB\_SPEAKER', 'RESTRICT\_VIEW\_OFF',  
 'RESTRICT\_VIEW\_ON', 'RESTRICT\_SELECT\_OFF', 'RESTRICT\_SELECT\_ON', 'RESTRICT\_RENDER\_OFF',  
 'RESTRICT\_RENDER\_ON', 'OUTLINER\_DATA\_EMPTY', 'OUTLINER\_DATA\_MESH',  
 'OUTLINER\_DATA\_CURVE', 'OUTLINER\_DATA\_LATTICE', 'OUTLINER\_DATA\_META',  
 'OUTLINER\_DATA\_LAMP', 'OUTLINER\_DATA\_CAMERA', 'OUTLINER\_DATA\_ARMATURE',  
 'OUTLINER\_DATA\_FONT', 'OUTLINER\_DATA\_SURFACE', 'OUTLINER\_DATA\_SPEAKER',  
 'OUTLINER\_DATA\_POSE', 'MESH\_PLANE', 'MESH\_CUBE', 'MESH\_CIRCLE', 'MESH\_UVSPHERE',  
 'MESH\_ICOSPHERE', 'MESH\_GRID', 'MESH\_MONKEY', 'MESH\_CYLINDER', 'MESH\_TORUS',  
 'MESH\_CONE', 'LAMP\_POINT', 'LAMP\_SUN', 'LAMP\_SPOT', 'LAMP\_HEMI', 'LAMP\_AREA',  
 'META\_PLANE', 'META\_CUBE', 'META\_BALL', 'META\_ELLIPSOID', 'META\_CAPSULE',  
 'SURFACE\_NCURVE', 'SURFACE\_NCIRCLE', 'SURFACE\_NSURFACE', 'SURFACE\_NCYLINDER',  
 'SURFACE\_NSPPHERE', 'SURFACE\_NTORUS', 'CURVE\_BEZCURVE', 'CURVE\_BEZCIRCLE',  
 'CURVE\_NCURVE', 'CURVE\_NCIRCLE', 'CURVE\_PATH', 'FORCE\_FORCE', 'FORCE\_WIND',  
 'FORCE\_VORTEX', 'FORCE\_MAGNETIC', 'FORCE\_HARMONIC', 'FORCE\_CHARGE',  
 'FORCE\_LENNARDJONES', 'FORCE\_TEXTURE', 'FORCE\_CURVE', 'FORCE\_BOID', 'FORCE\_TURBULENCE',  
 'FORCE\_DRAG', 'MODIFIER', 'MOD\_WAVE', 'MOD\_BUILD', 'MOD\_DECIM', 'MOD\_MIRROR',  
 'MOD\_SOFT', 'MOD\_SUBSURF', 'HOOK', 'MOD\_PHYSICS', 'MOD\_PARTICLES', 'MOD\_BOOLEAN',  
 'MOD\_EDGESPLIT', 'MOD\_ARRAY', 'MOD\_UVPROJECT', 'MOD\_DISPLACE', 'MOD\_CURVE',  
 'MOD\_LATTICE', 'CONSTRAINT\_DATA', 'MOD\_ARMATURE', 'MOD\_SHRINKWRAP', 'MOD\_CAST',  
 'MOD\_MESHDEFORM', 'MOD\_BEVEL', 'MOD\_SMOOTH', 'MOD\_SIMPLEDEFORM', 'MOD\_MASK',  
 'MOD\_CLOTH', 'MOD\_EXPLODE', 'MOD\_FLUIDSIM', 'MOD\_MULTIRES', 'MOD\_SMOKE',  
 'MOD\_SOLIDIFY', 'MOD\_SCREW', 'MOD\_VERTEX\_WEIGHT', 'REC', 'PLAY', 'FF', 'REW', 'PAUSE',  
 'PREV\_KEYFRAME', 'NEXT\_KEYFRAME', 'PLAY\_AUDIO', 'PLAY\_REVERSE', 'PREVIEW\_RANGE',  
 'PMARKER\_ACT', 'PMARKER\_SEL', 'PMARKER', 'MARKER\_HLT', 'MARKER', 'SPACE2',  
 'SPACE3', 'KEYINGSET', 'KEY\_DEHLT', 'KEY\_HLT', 'MUTE\_IPO\_OFF', 'MUTE\_IPO\_ON',  
 'VISIBLE\_IPO\_OFF', 'VISIBLE\_IPO\_ON', 'DRIVER', 'SOLO\_OFF', 'SOLO\_ON', 'VERTEXSEL',  
 'EDGESEL', 'FACESEL', 'ROTATE', 'CURSOR', 'ROTATECOLLECTION', 'ROTATECENTER', 'RO-TACTIVE',  
 'ALIGN', 'SMOOTHCURVE', 'SPHERECURVE', 'ROOTCURVE', 'SHARPCURVE', 'LINCURVE',  
 'NOCURVE', 'RNDCURVE', 'PROP\_OFF', 'PROP\_ON', 'PROP\_CON', 'PARTICLE\_POINT',  
 'PARTICLE\_TIP', 'PARTICLE\_PATH', 'MAN\_TRANS', 'MAN\_ROT', 'MAN\_SCALE', 'MANIPUL',  
 'SNAP\_OFF', 'SNAP\_ON', 'SNAP\_NORMAL', 'SNAP\_INCREMENT', 'SNAP\_VERTEX', 'SNAP\_EDGE',  
 'SNAP\_FACE', 'SNAP\_VOLUME', 'STICKY\_UVS\_LOC', 'STICKY\_UVS\_DISABLE', 'STICKY\_UVS\_VERT',  
 'CLIPUV\_DEHLT', 'CLIPUV\_HLT'

'SNAP\_PEEL\_OBJECT', 'GRID', 'PASTEDOWN', 'COPYDOWN', 'PASTE-FLIPUP', 'PASTEFLIPDOWN', 'SNAP\_SURFACE', 'RETOPO', 'UV\_VERTEXSEL', 'UV\_EDGESEL', 'UV\_FACESEL', 'UV\_ISLANDSEL', 'UV\_SYNC\_SELECT', 'BBOX', 'WIRE', 'SOLID', 'SMOOTH', 'POTATO', 'ORTHO', 'LOCKVIEW\_OFF', 'LOCKVIEW\_ON', 'AXIS\_SIDE', 'AXIS\_FRONT', 'AXIS\_TOP', 'NDOF\_DOM', 'NDOF\_TURN', 'NDOF\_FLY', 'NDOF\_TRANS', 'LAYER\_USED', 'LAYER\_ACTIVE', 'SORTALPHA', 'SORTBYEXT', 'SORTTIME', 'SORTSIZE', 'LONGDISPLAY', 'SHORTDISPLAY', 'GHOST', 'IMGDISPLAY', 'BOOKMARKS', 'FONTPREVIEW', 'FILTER', 'NEWFOLDER', 'FILE\_PARENT', 'FILE\_REFRESH', 'FILE\_FOLDER', 'FILE\_BLANK', 'FILE\_BLEND', 'FILE\_IMAGE', 'FILE\_MOVIE', 'FILE\_SCRIPT', 'FILE\_SOUND', 'FILE\_FONT', 'BACK', 'FORWARD', 'DISK\_DRIVE', 'MATPLANE', 'MATSPHERE', 'MATCUBE', 'MONKEY', 'HAIR', 'ALIASED', 'ANTIALIASED', 'MAT\_SPHERE\_SKY', 'WORDWRAP\_OFF', 'WORDWRAP\_ON', 'SYNTAX\_OFF', 'SYNTAX\_ON', 'LINENUMBERS\_OFF', 'LINENUMBERS\_ON', 'SCRIPTPLUGINS', 'SEQ\_SEQUENCER', 'SEQ\_PREVIEW', 'SEQ\_LUMA\_WAVEFORM', 'SEQ\_CHROMA\_SCOPE', 'SEQ\_HISTOGRAM', 'SEQ\_SPLITVIEW', 'IMAGE\_RGB', 'IMAGE\_RGB\_ALPHA', 'IMAGE\_ALPHA', 'IMAGE\_ZDEPTH', 'IMAGEFILE', 'BRUSH\_ADD', 'BRUSH\_BLOB', 'BRUSH\_BLUR', 'BRUSH\_CLAY', 'BRUSH\_CLONE', 'BRUSH\_CREASE', 'BRUSH\_DARKEN', 'BRUSH\_FILL', 'BRUSH\_FLATTEN', 'BRUSH\_GRAB', 'BRUSH\_INFLATE', 'BRUSH\_LAYER', 'BRUSH\_LIGHTEN', 'BRUSH\_MIX', 'BRUSH\_MULTIPLY', 'BRUSH\_NUDGE', 'BRUSH\_PINCH', 'BRUSH\_SCRAPE', 'BRUSH\_SCULPT\_DRAW', 'BRUSH\_SMEAR', 'BRUSH\_SMOOTH', 'BRUSH\_SNAKE\_HOOK', 'BRUSH\_SOFTEN', 'BRUSH\_SUBTRACT', 'BRUSH\_TEXDRAW', 'BRUSH\_THUMB', 'BRUSH\_ROTATE', 'BRUSH\_VERTEXDRAW', 'VIEW3D\_VEC', 'EDIT\_VEC', 'EDITMODE\_DEHLT', 'EDITMODE\_HLT', 'DISCLOSURE\_TRI\_RIGHT\_VEC', 'DISCLOSURE\_TRI\_DOWN\_VEC', 'MOVE\_UP\_VEC', 'MOVE\_DOWN\_VEC', 'X\_VEC', 'SMALL\_TRI\_RIGHT\_VEC'], (optional)) – Icon, Override automatic icon of the item

**menu** (*menu*, *text*=" ", *icon*='NONE')

menu

### Parameters

- **menu** (*string*) – Identifier of the menu
- **text** (*string*, (optional)) – Override automatic text of the item
- **icon** (*enum in* ['NONE', 'QUESTION', 'ERROR', 'CANCEL', 'TRIA\_RIGHT', 'TRIA\_DOWN', 'TRIA\_LEFT', 'TRIA\_UP', 'ARROW\_LEFTRIGHT', 'PLUS', 'DISCLOSURE\_TRI\_DOWN', 'DISCLOSURE\_TRI\_RIGHT', 'RADIOBUT\_OFF', 'RADIOBUT\_ON', 'MENU\_PANEL', 'BLENDER', 'DOT', 'X', 'GO\_LEFT', 'PLUG', 'UI', 'NODE', 'NODE\_SEL', 'FULLSCREEN', 'SPLITSCREEN', 'RIGHTARROW\_THIN', 'BORDERMOVE', 'VIEWZOOM', 'ZOOMIN', 'ZOOMOUT', 'PANEL\_CLOSE', 'COPY\_ID', 'EYEDROPPER', 'LINK\_AREA', 'AUTO', 'CHECKBOX\_DEHLT', 'CHECKBOX\_HLT', 'UNLOCKED', 'LOCKED', 'UNPINNED', 'PINNED', 'SCREEN\_BACK', 'RIGHTARROW', 'DOWNARROW\_HLT', 'DOTSUP', 'DOTSDOWN', 'LINK', 'INLINK', 'PLUGIN', 'HELP', 'GHOST\_ENABLED', 'COLOR', 'LINKED', 'UNLINKED', 'HAND', 'ZOOM\_ALL', 'ZOOM\_SELECTED', 'ZOOM\_PREVIOUS', 'ZOOM\_IN', 'ZOOM\_OUT', 'RENDER\_REGION', 'BORDER\_RECT', 'BORDER\_LASSO', 'FREEZE', 'STYLUS\_PRESSURE', 'GHOST\_DISABLED', 'NEW', 'FILE\_TICK', 'QUIT', 'URL', 'RECOVER\_LAST', 'FULLSCREEN\_ENTER', 'FULLSCREEN\_EXIT', 'BLANK1', 'LAMP', 'MATERIAL', 'TEXTURE', 'ANIM', 'WORLD', 'SCENE', 'EDIT', 'GAME', 'RADIO', 'SCRIPT', 'PARTICLES', 'PHYSICS', 'SPEAKER', 'TEXTURE\_SHADED', 'VIEW3D', 'IPO', 'OOPS', 'BUTS', 'FILESEL', 'IMAGE\_COL', 'INFO', 'SEQUENCE', 'TEXT', 'IMASEL',



'SOUND', 'ACTION', 'NLA', 'SCRIPTWIN', 'TIME', 'NODETREE', 'LOGIC',  
 'CONSOLE', 'PREFERENCES', 'ASSET\_MANAGER', 'OBJECT\_DATAMODE', 'EDIT-  
 MODE\_HLT', 'FACESEL\_HLT', 'VPAINT\_HLT', 'TPAINT\_HLT', 'WPAINT\_HLT',  
 'SCULPTMODE\_HLT', 'POSE\_HLT', 'PARTICLEMODE', 'LIGHTPAINT',  
 'SCENE\_DATA', 'RENDERLAYERS', 'WORLD\_DATA', 'OBJECT\_DATA',  
 'MESH\_DATA', 'CURVE\_DATA', 'META\_DATA', 'LATTICE\_DATA', 'LAMP\_DATA',  
 'MATERIAL\_DATA', 'TEXTURE\_DATA', 'ANIM\_DATA', 'CAMERA\_DATA', 'PAR-  
 TICLE\_DATA', 'LIBRARY\_DATA\_DIRECT', 'GROUP', 'ARMATURE\_DATA',  
 'POSE\_DATA', 'BONE\_DATA', 'CONSTRAINT', 'SHAPEKEY\_DATA', 'CON-  
 STRAINT\_BONE', 'PACKAGE', 'UGLYPACKAGE', 'BRUSH\_DATA', 'IMAGE\_DATA',  
 'FILE', 'FCURVE', 'FONT\_DATA', 'RENDER\_RESULT', 'SURFACE\_DATA',  
 'EMPTY\_DATA', 'SETTINGS', 'RENDER\_ANIMATION', 'RENDER\_STILL', 'BOIDS',  
 'STRANDS', 'LIBRARY\_DATA\_INDIRECT', 'GREASEPENCIL', 'GROUP\_BONE',  
 'GROUP\_VERTEX', 'GROUP\_VCOL', 'GROUP\_UVS', 'RNA', 'RNA\_ADD', 'OUT-  
 LINER\_OB\_EMPTY', 'OUTLINER\_OB\_MESH', 'OUTLINER\_OB\_CURVE', 'OUT-  
 LINER\_OB\_LATTICE', 'OUTLINER\_OB\_META', 'OUTLINER\_OB\_LAMP', 'OUT-  
 LINER\_OB\_CAMERA', 'OUTLINER\_OB\_ARMATURE', 'OUTLINER\_OB\_FONT',  
 'OUTLINER\_OB\_SURFACE', 'OUTLINER\_OB\_SPEAKER', 'RESTRICT\_VIEW\_OFF',  
 'RESTRICT\_VIEW\_ON', 'RESTRICT\_SELECT\_OFF', 'RESTRICT\_SELECT\_ON', 'RE-  
 STRICT\_RENDER\_OFF', 'RESTRICT\_RENDER\_ON', 'OUTLINER\_DATA\_EMPTY',  
 'OUTLINER\_DATA\_MESH', 'OUTLINER\_DATA\_CURVE', 'OUT-  
 LINER\_DATA\_LATTICE', 'OUTLINER\_DATA\_META', 'OUTLINER\_DATA\_LAMP',  
 'OUTLINER\_DATA\_CAMERA', 'OUTLINER\_DATA\_ARMATURE',  
 'OUTLINER\_DATA\_FONT', 'OUTLINER\_DATA\_SURFACE', 'OUT-  
 LINER\_DATA\_SPEAKER', 'OUTLINER\_DATA\_POSE', 'MESH\_PLANE',  
 'MESH\_CUBE', 'MESH\_CIRCLE', 'MESH\_UVSPHERE', 'MESH\_ICOSPHERE',  
 'MESH\_GRID', 'MESH\_MONKEY', 'MESH\_CYLINDER', 'MESH\_TORUS',  
 'MESH\_CONE', 'LAMP\_POINT', 'LAMP\_SUN', 'LAMP\_SPOT', 'LAMP\_HEMI',  
 'LAMP\_AREA', 'META\_PLANE', 'META\_CUBE', 'META\_BALL',  
 'META\_ELLIPSOID', 'META\_CAPSULE', 'SURFACE\_NCURVE', 'SUR-  
 FACE\_NCIRCLE', 'SURFACE\_NSURFACE', 'SURFACE\_NCYLINDER',  
 'SURFACE\_NSPPHERE', 'SURFACE\_NTORUS', 'CURVE\_BEZCURVE',  
 'CURVE\_BEZCIRCLE', 'CURVE\_NCURVE', 'CURVE\_NCIRCLE', 'CURVE\_PATH',  
 'FORCE\_FORCE', 'FORCE\_WIND', 'FORCE\_VORTEX', 'FORCE\_MAGNETIC',  
 'FORCE\_HARMONIC', 'FORCE\_CHARGE', 'FORCE\_LENNARDJONES',  
 'FORCE\_TEXTURE', 'FORCE\_CURVE', 'FORCE\_BOID', 'FORCE\_TURBULENCE',  
 'FORCE\_DRAG', 'MODIFIER', 'MOD\_WAVE', 'MOD\_BUILD', 'MOD\_DECIM',  
 'MOD\_MIRROR', 'MOD\_SOFT', 'MOD\_SUBSURF', 'HOOK', 'MOD\_PHYSICS',  
 'MOD\_PARTICLES', 'MOD\_BOOLEAN', 'MOD\_EDGESPLIT', 'MOD\_ARRAY',  
 'MOD\_UVPROJECT', 'MOD\_DISPLACE', 'MOD\_CURVE', 'MOD\_LATTICE',  
 'CONSTRAINT\_DATA', 'MOD\_ARMATURE', 'MOD\_SHRINKWRAP',  
 'MOD\_CAST', 'MOD\_MESHDEFORM', 'MOD\_BEVEL', 'MOD\_SMOOTH',  
 'MOD\_SIMPLEDEFORM', 'MOD\_MASK', 'MOD\_CLOTH', 'MOD\_EXPLODE',  
 'MOD\_FLUIDSIM', 'MOD\_MULTIRES', 'MOD\_SMOKE', 'MOD\_SOLIDIFY',  
 'MOD\_SCREW', 'MOD\_VERTEX\_WEIGHT', 'REC', 'PLAY', 'FF', 'REW', 'PAUSE',  
 'PREV\_KEYFRAME', 'NEXT\_KEYFRAME', 'PLAY\_AUDIO', 'PLAY\_REVERSE', 'PRE-  
 VIEW\_RANGE', 'PMARKER\_ACT', 'PMARKER\_SEL', 'PMARKER', 'MARKER\_HLT',  
 'MARKER', 'SPACE2', 'SPACE3', 'KEYINGSET', 'KEY\_DEHLT', 'KEY\_HLT',  
 'MUTE\_IPO\_OFF', 'MUTE\_IPO\_ON', 'VISIBLE\_IPO\_OFF', 'VISIBLE\_IPO\_ON',  
 'DRIVER', 'SOLO\_OFF', 'SOLO\_ON', 'VERTEXSEL', 'EDGESEL', 'FACE-  
 SEL', 'ROTATE', 'CURSOR', 'ROTATECOLLECTION', 'ROTATECENTER', 'RO-  
 TACTIVE', 'ALIGN', 'SMOOTHCURVE', 'SPHERECURVE', 'ROOTCURVE',  
 'SHARPCURVE', 'LINCURVE', 'NOCURVE', 'RNDCURVE', 'PROP\_OFF',  
 'PROP\_ON', 'PROP\_CON', 'PARTICLE\_POINT', 'PARTICLE\_TIP', 'PARTI-

`CLE_PATH`, `MAN_TRANS`, `MAN_ROT`, `MAN_SCALE`, `MANIPUL`, `SNAP_OFF`,  
`SNAP_ON`, `SNAP_NORMAL`, `SNAP_INCREMENT`, `SNAP_VERTEX`,  
`SNAP_EDGE`, `SNAP_FACE`, `SNAP_VOLUME`, `STICKY_UVS_LOC`,  
`STICKY_UVS_DISABLE`, `STICKY_UVS_VERT`, `CLIPUV_DEHLT`, `CLIPUV_HLT`,  
`SNAP_PEEL_OBJECT`, `GRID`, `PASTEDOWN`, `COPYDOWN`, `PASTE-  
FLIPUP`, `PASTEFLIPDOWN`, `SNAP_SURFACE`, `RETOPO`, `UV_VERTEXSEL`,  
`UV_EDGESEL`, `UV_FACESEL`, `UV_ISLANDSEL`, `UV_SYNC_SELECT`,  
`BBOX`, `WIRE`, `SOLID`, `SMOOTH`, `POTATO`, `ORTHO`, `LOCKVIEW_OFF`,  
`LOCKVIEW_ON`, `AXIS_SIDE`, `AXIS_FRONT`, `AXIS_TOP`, `NDOF_DOM`,  
`NDOF_TURN`, `NDOF_FLY`, `NDOF_TRANS`, `LAYER_USED`, `LAYER_ACTIVE`,  
`SORTALPHA`, `SORTBYEXT`, `SORTTIME`, `SORTSIZE`, `LONGDISPLAY`,  
`SHORTDISPLAY`, `GHOST`, `IMGDISPLAY`, `BOOKMARKS`, `FONTPRE-  
VIEW`, `FILTER`, `NEWFOLDER`, `FILE_PARENT`, `FILE_REFRESH`,  
`FILE_FOLDER`, `FILE_BLANK`, `FILE_BLEND`, `FILE_IMAGE`, `FILE_MOVIE`,  
`FILE_SCRIPT`, `FILE_SOUND`, `FILE_FONT`, `BACK`, `FORWARD`,  
`DISK_DRIVE`, `MATPLANE`, `MATSPHERE`, `MATCUBE`, `MONKEY`, `HAIR`,  
`ALIASED`, `ANTIALIASED`, `MAT_SPHERE_SKY`, `WORDWRAP_OFF`,  
`WORDWRAP_ON`, `SYNTAX_OFF`, `SYNTAX_ON`, `LINENUMBERS_OFF`,  
`LINENUMBERS_ON`, `SCRIPTPLUGINS`, `SEQ_SEQUENCER`, `SEQ_PREVIEW`,  
`SEQ_LUMA_WAVEFORM`, `SEQ_CHROMA_SCOPE`, `SEQ_HISTOGRAM`,  
`SEQ_SPLITVIEW`, `IMAGE_RGB`, `IMAGE_RGB_ALPHA`, `IMAGE_ALPHA`, `IM-  
AGE_ZDEPTH`, `IMAGEFILE`, `BRUSH_ADD`, `BRUSH_BLOB`, `BRUSH_BLUR`,  
`BRUSH_CLAY`, `BRUSH_CLONE`, `BRUSH_CREASE`, `BRUSH_DARKEN`,  
`BRUSH_FILL`, `BRUSH_FLATTEN`, `BRUSH_GRAB`, `BRUSH_INFLATE`,  
`BRUSH_LAYER`, `BRUSH_LIGHTEN`, `BRUSH_MIX`, `BRUSH_MULTIPLY`,  
`BRUSH_NUDGE`, `BRUSH_PINCH`, `BRUSH_SCRAPPE`, `BRUSH_SCULPT_DRAW`,  
`BRUSH_SMEAR`, `BRUSH_SMOOTH`, `BRUSH_SNAKE_HOOK`,  
`BRUSH_SOFTEN`, `BRUSH_SUBTRACT`, `BRUSH_TEXDRAW`, `BRUSH_THUMB`,  
`BRUSH_ROTATE`, `BRUSH_VERTEXDRAW`, `VIEW3D_VEC`, `EDIT_VEC`, `ED-  
ITMODE_DEHLT`, `EDITMODE_HLT`, `DISCLOSURE_TRI_RIGHT_VEC`, `DIS-  
CLOSURE_TRI_DOWN_VEC`, `MOVE_UP_VEC`, `MOVE_DOWN_VEC`, `X_VEC`,  
`SMALL_TRI_RIGHT_VEC`], (optional) – Icon, Override automatic icon of the item

**separator** ()

Item. Inserts empty space into the layout between items

**context\_pointer\_set** (*name*, *data*)

context\_pointer\_set

**Parameters**

- **name** (*string*) – Name, Name of entry in the context
- **data** (*AnyType*) – Pointer to put in context

**template\_header** (*menus=True*)

template\_header

**Parameters** **menus** (*boolean*, (optional)) – The header has menus, and should show menu ex-  
pander

**template\_ID** (*data*, *property*, *new=""*, *open=""*, *unlink=""*)

template\_ID

**Parameters**

- **data** (*AnyType*, (never None)) – Data from which to take property
- **property** (*string*) – Identifier of property in data

- **new** (*string, (optional)*) – Operator identifier to create a new ID block
- **open** (*string, (optional)*) – Operator identifier to open a file for creating a new ID block
- **unlink** (*string, (optional)*) – Operator identifier to unlink the ID block

**template\_ID\_preview** (*data, property, new=""*, *open=""*, *unlink=""*, *rows=0, cols=0*)  
 template\_ID\_preview

#### Parameters

- **data** (*AnyType*, (never None)) – Data from which to take property
- **property** (*string*) – Identifier of property in data
- **new** (*string, (optional)*) – Operator identifier to create a new ID block
- **open** (*string, (optional)*) – Operator identifier to open a file for creating a new ID block
- **unlink** (*string, (optional)*) – Operator identifier to unlink the ID block
- **rows** (*int in [0, inf], (optional)*) – Number of thumbnail preview rows to display
- **cols** (*int in [0, inf], (optional)*) – Number of thumbnail preview columns to display

**template\_any\_ID** (*data, property, type\_property, text=""*)  
 template\_any\_ID

#### Parameters

- **data** (*AnyType*, (never None)) – Data from which to take property
- **property** (*string*) – Identifier of property in data
- **type\_property** (*string*) – Identifier of property in data giving the type of the ID-blocks to use
- **text** (*string, (optional)*) – Custom label to display in UI

**template\_path\_builder** (*data, property, root, text=""*)  
 template\_path\_builder

#### Parameters

- **data** (*AnyType*, (never None)) – Data from which to take property
- **property** (*string*) – Identifier of property in data
- **root** (*ID*) – ID-block from which path is evaluated from
- **text** (*string, (optional)*) – Custom label to display in UI

**template\_modifier** (*data*)

Layout . Generates the UI layout for modifiers

**Parameters** **data** (*Modifier*, (never None)) – Modifier data

**Returns** Sub-layout to put items in

**Return type** *UILayout*

**template\_constraint** (*data*)

Layout . Generates the UI layout for constraints

**Parameters** **data** (*Constraint*, (never None)) – Constraint data

**Returns** Sub-layout to put items in

**Return type** *UILayout*

**template\_preview** (*id, show\_buttons=True, parent=None, slot=None*)

Item. A preview window for materials, textures, lamps, etc.

**Parameters**

- **id** (*ID*) – ID datablock
- **show\_buttons** (*boolean, (optional)*) – Show preview buttons?
- **parent** (*ID, (optional)*) – ID datablock
- **slot** (*TextureSlot, (optional)*) – Texture slot

**template\_curve\_mapping** (*data, property, type='NONE', levels=False, brush=False*)

Item. A curve mapping widget used for e.g falloff curves for lamps

**Parameters**

- **data** (*AnyType, (never None)*) – Data from which to take property
- **property** (*string*) – Identifier of property in data
- **type** (*enum in ['NONE', 'VECTOR', 'COLOR'], (optional)*) – Type, Type of curves to display
- **levels** (*boolean, (optional)*) – Show black/white levels
- **brush** (*boolean, (optional)*) – Show brush options

**template\_color\_ramp** (*data, property, expand=False*)

Item. A color ramp widget

**Parameters**

- **data** (*AnyType, (never None)*) – Data from which to take property
- **property** (*string*) – Identifier of property in data
- **expand** (*boolean, (optional)*) – Expand button to show more detail

**template\_histogram** (*data, property*)

Item. A histogram widget to analyze image data

**Parameters**

- **data** (*AnyType, (never None)*) – Data from which to take property
- **property** (*string*) – Identifier of property in data

**template\_waveform** (*data, property*)

Item. A waveform widget to analyze image data

**Parameters**

- **data** (*AnyType, (never None)*) – Data from which to take property
- **property** (*string*) – Identifier of property in data

**template\_vectorscope** (*data, property*)

Item. A vectorscope widget to analyze image data

**Parameters**

- **data** (*AnyType, (never None)*) – Data from which to take property
- **property** (*string*) – Identifier of property in data

**template\_layers** (*data, property, used\_layers\_data, used\_layers\_property, active\_layer*)

template\_layers

**Parameters**

- **data** (*AnyType*, (never None)) – Data from which to take property
- **property** (*string*) – Identifier of property in data
- **used\_layers\_data** (*AnyType*) – Data from which to take property
- **used\_layers\_property** (*string*) – Identifier of property in data
- **active\_layer** (*int in [0, inf]*) – Active Layer

**template\_color\_wheel** (*data, property, value\_slider=False, lock=False, lock\_luminosity=False, cubic=True*)

Item. A color wheel widget to pick colors

**Parameters**

- **data** (*AnyType*, (never None)) – Data from which to take property
- **property** (*string*) – Identifier of property in data
- **value\_slider** (*boolean, (optional)*) – Display the value slider to the right of the color wheel
- **lock** (*boolean, (optional)*) – Lock the color wheel display to value 1.0 regardless of actual color
- **lock\_luminosity** (*boolean, (optional)*) – Keep the color at its original vector length
- **cubic** (*boolean, (optional)*) – Cubic saturation for picking values close to white

**template\_image\_layers** (*image, image\_user*)

template\_image\_layers

**template\_image** (*data, property, image\_user, compact=False*)

Item(s). User interface for selecting images and their source paths

**Parameters**

- **data** (*AnyType*, (never None)) – Data from which to take property
- **property** (*string*) – Identifier of property in data
- **compact** (*boolean, (optional)*) – Use more compact layout

**template\_list** (*data, property, active\_data, active\_property, prop\_list=""*, *rows=5, maxrows=5, type='DEFAULT'*)

Item. A list widget to display data. e.g. vertexgroups

**Parameters**

- **data** (*AnyType*) – Data from which to take property
- **property** (*string*) – Identifier of property in data
- **active\_data** (*AnyType*, (never None)) – Data from which to take property for the active element
- **active\_property** (*string*) – Identifier of property in data, for the active element
- **prop\_list** (*string, (optional)*) – Identifier of a string property in each data member, specifying which of its properties should have a widget displayed in its row
- **rows** (*int in [0, inf], (optional)*) – Number of rows to display
- **maxrows** (*int in [0, inf], (optional)*) – Maximum number of rows to display

- **type** (*enum in ['DEFAULT', 'COMPACT', 'ICONS'], (optional)*) – Type, Type of list to use

**template\_running\_jobs** ()  
template\_running\_jobs

**template\_operator\_search** ()  
template\_operator\_search

**template\_header\_3D** ()  
template\_header\_3D

**template\_edit\_mode\_selection** ()  
template\_edit\_mode\_selection

**template\_reports\_banner** ()  
template\_reports\_banner

**template\_keymap\_item\_properties** (*item*)  
template\_keymap\_item\_properties

**introspect** ()  
introspect

**Returns** Descr, DESCR

**Return type** string

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Header.layout`

- `Menu.layout`
- `Operator.layout`
- `Panel.layout`
- `UILayout.box`
- `UILayout.column`
- `UILayout.column_flow`
- `UILayout.row`
- `UILayout.split`
- `UILayout.template_constraint`
- `UILayout.template_modifier`

### 2.4.616 UVProjectModifier(Modifier)

base classes — `bpy_struct`, `Modifier`

**class** `bpy.types.UVProjectModifier` (*Modifier*)

UV projection modifier to sets UVs from a projector

**aspect\_x**

**Type** float in [1, inf], default 0.0

**aspect\_y**

**Type** float in [1, inf], default 0.0

**image**

**Type** `Image`

**projector\_count**

Number of projectors to use

**Type** int in [1, 10], default 0

**projectors**

**Type** `bpy_prop_collection` of `UVProjector`, (readonly)

**scale\_x**

**Type** float in [0, inf], default 0.0

**scale\_y**

**Type** float in [0, inf], default 0.0

**use\_image\_override**

Override faces' current images with the given image

**Type** boolean, default False

**uv\_layer**

UV layer name

**Type** string, default ""

#### Inherited Properties

- `bpy_struct.id_data`
- `Modifier.name`
- `Modifier.use_apply_on_spline`

- `Modifier.show_in_editmode`
- `Modifier.show_expanded`
- `Modifier.show_on_cage`
- `Modifier.show_viewport`
- `Modifier.show_render`
- `Modifier.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.617 UVProjector(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.UVProjector` (*bpy\_struct*)  
UV projector used by the UV project modifier

**object**

Object to use as projector transform

**Type** `Object`

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`



- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `UVProjectModifier.projectors`

## 2.4.618 UVTextures(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.UVTextures` (*bpy\_struct*)

Collection of uv textures

### **active**

Active UV texture

**Type** `MeshTextureFaceLayer`

### **active\_index**

Active UV texture index

**Type** `int` in `[0, inf]`, default `0`

**new** (*name="UVTex"*)

Add a UV texture layer to Mesh

**Parameters** **name** (*string, (optional)*) – UV Texture name

**Returns** The newly created layer

**Return type** `MeshTextureFaceLayer`

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`

- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Mesh.uv_textures`

## 2.4.619 UnitSettings(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.UnitSettings` (*bpy\_struct*)

### **scale\_length**

Scale to use when converting between blender units and dimensions

**Type** float in [1e-05, 100000], default 0.0

### **system**

The unit system to use for button display

**Type** enum in ['NONE', 'METRIC', 'IMPERIAL'], default 'NONE'

### **system\_rotation**

Unit to use for displaying/editing rotation values

- **DEGREES** Degrees, Use degrees for measuring angles and rotations.
- **RADIANS** Radians.

**Type** enum in ['DEGREES', 'RADIANS'], default 'DEGREES'

### **use\_separate**

Display units in pairs

**Type** boolean, default False

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`

- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Scene.unit_settings`

## 2.4.620 UnknownType(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.UnknownType` (*bpy\_struct*)

Stub RNA type used for pointers to unknown or internal data

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `ShapeKey.data`
- `SpaceSequenceEditor.grease_pencil`

## 2.4.621 UserPreferences(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.UserPreferences` (*bpy\_struct*)

Global user preferences

**active\_section**

Active section of the user preferences shown in the user interface

**Type** enum in ['INTERFACE', 'EDITING', 'INPUT', 'ADDONS', 'THEMES', 'FILES', 'SYSTEM'], default 'INTERFACE'

**addons**

**Type** `Addons bpy_prop_collection` of `Addon`, (readonly)

**edit**

Settings for interacting with Blender data

**Type** `UserPreferencesEdit`, (readonly, never None)

**filepaths**

Default paths for external files

**Type** `UserPreferencesFilePaths`, (readonly, never None)

**inputs**

Settings for input devices

**Type** `UserPreferencesInput`, (readonly, never None)

**system**

Graphics driver and operating system settings

**Type** `UserPreferencesSystem`, (readonly, never None)

**themes**

**Type** `bpy_prop_collection` of `Theme`, (readonly)

**ui\_styles**

**Type** `bpy_prop_collection` of `ThemeStyle`, (readonly)

**view**

Preferences related to viewing data

**Type** `UserPreferencesView`, (readonly, never None)

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`

- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Context.user_preferences`

## 2.4.622 UserPreferencesEdit(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.UserPreferencesEdit` (*bpy\_struct*)

Settings for interacting with Blender data

### **auto\_keying\_mode**

Mode of automatic keyframe insertion for Objects and Bones (default setting used for new Scenes)

**Type** enum in ['ADD\_REPLACE\_KEYS', 'REPLACE\_KEYS'], default 'ADD\_REPLACE\_KEYS'

### **grease\_pencil\_eraser\_radius**

Radius of eraser 'brush'

**Type** int in [0, 100], default 0

### **grease\_pencil\_euclidean\_distance**

Distance moved by mouse when drawing stroke (in pixels) to include

**Type** int in [0, 100], default 0

### **grease\_pencil\_manhattan\_distance**

Pixels moved by mouse per axis when drawing stroke

**Type** int in [0, 100], default 0

### **keyframe\_new\_handle\_type**

Handle type for handles of new keyframes

- FREE Free.
- VECTOR Vector.
- ALIGNED Aligned.
- AUTO Automatic.
- AUTO\_CLAMPED Auto Clamped, Auto handles clamped to not overshoot.

**Type** enum in ['FREE', 'VECTOR', 'ALIGNED', 'AUTO', 'AUTO\_CLAMPED'], default 'FREE'

**keyframe\_new\_interpolation\_type**

Interpolation mode used for first keyframe on newly added F-Curves (subsequent keyframes take interpolation from preceding keyframe)

**Type** enum in ['CONSTANT', 'LINEAR', 'BEZIER'], default 'CONSTANT'

**material\_link**

Toggle whether the material is linked to object data or the object block

- OBDATA ObData, Toggle whether the material is linked to object data or the object block.
- OBJECT Object, Toggle whether the material is linked to object data or the object block.

**Type** enum in ['OBDATA', 'OBJECT'], default 'OBDATA'

**object\_align**

When adding objects from a 3D View menu, either align them with that view or with the world

- WORLD World, Align newly added objects to the world coordinate system.
- VIEW View, Align newly added objects facing the active 3D View direction.

**Type** enum in ['WORLD', 'VIEW'], default 'WORLD'

**sculpt\_paint\_overlay\_color**

Color of texture overlay

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**undo\_memory\_limit**

Maximum memory usage in megabytes (0 means unlimited)

**Type** int in [0, 32767], default 0

**undo\_steps**

Number of undo steps available (smaller values conserve memory)

**Type** int in [0, 64], default 0

**use\_auto\_keying**

Automatic keyframe insertion for Objects and Bones (default setting used for new Scenes)

**Type** boolean, default False

**use\_drag\_immediately**

Moving things with a mouse drag confirms when releasing the button

**Type** boolean, default False

**use\_duplicate\_action**

Causes actions to be duplicated with the object

**Type** boolean, default False

**use\_duplicate\_armature**

Causes armature data to be duplicated with the object

**Type** boolean, default False

**use\_duplicate\_curve**

Causes curve data to be duplicated with the object

**Type** boolean, default False

**use\_duplicate\_fcurve**

Causes F-curve data to be duplicated with the object

**Type** boolean, default False

**use\_duplicate\_lamp**

Causes lamp data to be duplicated with the object

**Type** boolean, default False

**use\_duplicate\_material**

Causes material data to be duplicated with the object

**Type** boolean, default False

**use\_duplicate\_mesh**

Causes mesh data to be duplicated with the object

**Type** boolean, default False

**use\_duplicate\_metaball**

Causes metaball data to be duplicated with the object

**Type** boolean, default False

**use\_duplicate\_particle**

Causes particle systems to be duplicated with the object

**Type** boolean, default False

**use\_duplicate\_surface**

Causes surface data to be duplicated with the object

**Type** boolean, default False

**use\_duplicate\_text**

Causes text data to be duplicated with the object

**Type** boolean, default False

**use\_duplicate\_texture**

Causes texture data to be duplicated with the object

**Type** boolean, default False

**use\_enter\_edit\_mode**

Enter Edit Mode automatically after adding a new object

**Type** boolean, default False

**use\_global\_undo**

Global undo works by keeping a full copy of the file itself in memory, so takes extra memory

**Type** boolean, default False

**use\_grease\_pencil\_simplify\_stroke**

Simplify the final stroke

**Type** boolean, default False

**use\_grease\_pencil\_smooth\_stroke**

Smooth the final stroke

**Type** boolean, default False

**use\_insertkey\_xyz\_to\_rgb**

Color for newly added transformation F-Curves (Location, Rotation, Scale) and also Color is based on the transform axis

**Type** boolean, default False

**use\_keyframe\_insert\_available**

Automatic keyframe insertion in available F-Curves

**Type** boolean, default False

**use\_keyframe\_insert\_needed**

Keyframe insertion only when keyframe needed

**Type** boolean, default False

**use\_negative\_frames**

Current frame number can be manually set to a negative value

**Type** boolean, default False

**use\_visual\_keying**

Use Visual keying automatically for constrained objects

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `UserPreferences.edit`



## 2.4.623 UserPreferencesFilePaths(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.UserPreferencesFilePaths` (*bpy\_struct*)

Default paths for external files

### **animation\_player**

Path to a custom animation/frame sequence player

**Type** string, default ""

### **animation\_player\_preset**

Preset configs for external animation players

- **BLENDER24** Blender 2.4, Blender command line animation playback - path to Blender 2.4.
- **DJV** Djv, Open source frame player: <http://djv.sourceforge.net>.
- **FRAMECYCLER** FrameCycler, Frame player from IRIDAS.
- **RV** rv, Frame player from Tweak Software.
- **MPLAYER** MPlayer, Media player for video & png/jpeg/sgi image sequences.
- **CUSTOM** Custom, Custom animation player executable path.

**Type** enum in ['BLENDER24', 'DJV', 'FRAMECYCLER', 'RV', 'MPLAYER', 'CUSTOM'], default 'BLENDER24'

### **auto\_save\_time**

The time (in minutes) to wait between automatic temporary saves

**Type** int in [1, 60], default 0

### **font\_directory**

The default directory to search for loading fonts

**Type** string, default ""

### **hide\_recent\_locations**

Hide recent locations in the file selector

**Type** boolean, default False

### **image\_editor**

Path to an image editor

**Type** string, default ""

### **recent\_files**

Maximum number of recently opened files to remember

**Type** int in [0, 30], default 0

### **render\_output\_directory**

The default directory for rendering output, for new scenes

**Type** string, default ""

### **save\_version**

The number of old versions to maintain in the current directory, when manually saving

**Type** int in [0, 32], default 0

**script\_directory**

Alternate script path, matching the default layout with subdirs: startup, addons & modules (requires restart)

**Type** string, default ""

**sequence\_plugin\_directory**

The default directory to search for sequence plugins

**Type** string, default ""

**show\_hidden\_files\_datablocks**

Hide files/datablocks that start with a dot (.\*)

**Type** boolean, default False

**show\_thumbnails**

Open in thumbnail view for images and movies

**Type** boolean, default False

**sound\_directory**

The default directory to search for sounds

**Type** string, default ""

**temporary\_directory**

The directory for storing temporary save files

**Type** string, default ""

**texture\_directory**

The default directory to search for textures

**Type** string, default ""

**texture\_plugin\_directory**

The default directory to search for texture plugins

**Type** string, default ""

**use\_auto\_save\_temporary\_files**

Automatic saving of temporary files in temp directory, uses process ID

**Type** boolean, default False

**use\_file\_compression**

Enable file compression when saving .blend files

**Type** boolean, default False

**use\_filter\_files**

Display only files with extensions in the image select window

**Type** boolean, default False

**use\_load\_ui**

Load user interface setup when loading .blend files

**Type** boolean, default False

**use\_relative\_paths**

Default relative path option for the file selector

**Type** boolean, default False

**use\_save\_preview\_images**

Enables automatic saving of preview images in the .blend file

**Type** boolean, default False

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `UserPreferences.filepaths`

## 2.4.624 UserPreferencesInput(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.UserPreferencesInput` (*bpy\_struct*)  
Settings for input devices

**active\_keyconfig**

The name of the active key configuration

**Type** string, default ""

**drag\_threshold**

Amount of pixels you have to drag before dragging UI items happens

**Type** int in [3, 40], default 0

**invert\_mouse\_zoom**

Invert the axis of mouse movement for zooming

**Type** boolean, default False

**invert\_zoom\_wheel**

Swap the Mouse Wheel zoom direction

**Type** boolean, default False

**mouse\_double\_click\_time**

Time/delay (in ms) for a double click

**Type** int in [1, 1000], default 0

**ndof\_fly\_helicopter**

Device up/down directly controls your Z position

**Type** boolean, default False

**ndof\_lock\_horizon**

Keep horizon level while flying with 3D Mouse

**Type** boolean, default False

**ndof\_orbit\_invert\_axes**

Toggle between moving the viewpoint or moving the scene being viewed

**Type** boolean, default False

**ndof\_panz\_invert\_axis**

Invert x axis

**Type** boolean, default False

**ndof\_pany\_invert\_axis**

Invert y axis

**Type** boolean, default False

**ndof\_panz\_invert\_axis**

Invert z axis

**Type** boolean, default False

**ndof\_roll\_invert\_axis**

Invert roll axis

**Type** boolean, default False

**ndof\_rotate\_invert\_axis**

Invert rotation axis

**Type** boolean, default False

**ndof\_sensitivity**

Overall sensitivity of the 3D Mouse

**Type** float in [0.25, 4], default 0.0

**ndof\_show\_guide**

Display the center and axis during rotation

**Type** boolean, default False

**ndof\_tilt\_invert\_axis**

Invert tilt axis

**Type** boolean, default False

**ndof\_zoom\_invert**

Zoom using opposite direction

**Type** boolean, default False

**ndof\_zoom\_updown**

Zoom using up/down on the device (otherwise forward/backward)

**Type** boolean, default False

#### **select\_mouse**

Mouse button used for selection

- LEFT Left, Use left Mouse Button for selection.
- RIGHT Right, Use Right Mouse Button for selection.

**Type** enum in ['LEFT', 'RIGHT'], default 'RIGHT'

#### **use\_emulate\_numpad**

Main 1 to 0 keys act as the numpad ones (useful for laptops)

**Type** boolean, default False

#### **use\_mouse\_continuous**

Allow moving the mouse outside the view on some manipulations (transform, ui control drag)

**Type** boolean, default False

#### **use\_mouse\_emulate\_3\_button**

Emulate Middle Mouse with Alt+Left Mouse (doesn't work with Left Mouse Select option)

**Type** boolean, default False

#### **use\_mouse\_mmb\_paste**

In text window, paste with middle mouse button instead of panning

**Type** boolean, default False

#### **view\_rotate\_method**

Rotation style in the viewport

- TURNTABLE Turntable, Use turntable style rotation in the viewport.
- TRACKBALL Trackball, Use trackball style rotation in the viewport.

**Type** enum in ['TURNTABLE', 'TRACKBALL'], default 'TURNTABLE'

#### **view\_zoom\_axis**

Axis of mouse movement to zoom in or out on

- VERTICAL Vertical, Zoom in and out based on vertical mouse movement.
- HORIZONTAL Horizontal, Zoom in and out based on horizontal mouse movement.

**Type** enum in ['VERTICAL', 'HORIZONTAL'], default 'VERTICAL'

#### **view\_zoom\_method**

Which style to use for viewport scaling

- CONTINUE Continue, Old style zoom, continues while moving mouse up or down.
- DOLLY Dolly, Zoom in and out based on vertical mouse movement.
- SCALE Scale, Zoom in and out like scaling the view, mouse movements relative to center.

**Type** enum in ['CONTINUE', 'DOLLY', 'SCALE'], default 'CONTINUE'

#### **wheel\_scroll\_lines**

Number of lines scrolled at a time with the mouse wheel

**Type** int in [0, 32], default 0

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `UserPreferences.inputs`

## 2.4.625 UserPreferencesSystem(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.UserPreferencesSystem` (*bpy\_struct*)  
Graphics driver and operating system settings

#### **anisotropic\_filter**

Quality of the anisotropic filtering (values greater than 1.0 enable anisotropic filtering)

**Type** enum in ['FILTER\_0', 'FILTER\_2', 'FILTER\_4', 'FILTER\_8', 'FILTER\_16'], default 'FILTER\_0'

#### **audio\_channels**

Audio channel count

- **MONO Mono**, Set audio channels to mono.
- **STEREO Stereo**, Set audio channels to stereo.
- **SURROUND4 4 Channels**, Set audio channels to 4 channels.
- **SURROUND51 5.1 Surround**, Set audio channels to 5.1 surround sound.
- **SURROUND71 7.1 Surround**, Set audio channels to 7.1 surround sound.

**Type** enum in ['MONO', 'STEREO', 'SURROUND4', 'SURROUND51', 'SURROUND71'], default 'MONO'

#### **audio\_device**

Audio output device

- NONE None, Null device - there will be no audio output.

**Type** enum in ['NONE'], default 'NONE'

#### **audio\_mixing\_buffer**

Number of samples used by the audio mixing buffer

- SAMPLES\_256 256, Set audio mixing buffer size to 256 samples.
- SAMPLES\_512 512, Set audio mixing buffer size to 512 samples.
- SAMPLES\_1024 1024, Set audio mixing buffer size to 1024 samples.
- SAMPLES\_2048 2048, Set audio mixing buffer size to 2048 samples.
- SAMPLES\_4096 4096, Set audio mixing buffer size to 4096 samples.
- SAMPLES\_8192 8192, Set audio mixing buffer size to 8192 samples.
- SAMPLES\_16384 16384, Set audio mixing buffer size to 16384 samples.
- SAMPLES\_32768 32768, Set audio mixing buffer size to 32768 samples.

**Type** enum in ['SAMPLES\_256', 'SAMPLES\_512', 'SAMPLES\_1024', 'SAMPLES\_2048', 'SAMPLES\_4096', 'SAMPLES\_8192', 'SAMPLES\_16384', 'SAMPLES\_32768'], default 'SAMPLES\_256'

#### **audio\_sample\_format**

Audio sample format

- U8 8-bit Unsigned, Set audio sample format to 8 bit unsigned integer.
- S16 16-bit Signed, Set audio sample format to 16 bit signed integer.
- S24 24-bit Signed, Set audio sample format to 24 bit signed integer.
- S32 32-bit Signed, Set audio sample format to 32 bit signed integer.
- FLOAT 32-bit Float, Set audio sample format to 32 bit float.
- DOUBLE 64-bit Float, Set audio sample format to 64 bit float.

**Type** enum in ['U8', 'S16', 'S24', 'S32', 'FLOAT', 'DOUBLE'], default 'U8'

#### **audio\_sample\_rate**

Audio sample rate

- RATE\_44100 44.1 kHz, Set audio sampling rate to 44100 samples per second.
- RATE\_48000 48 kHz, Set audio sampling rate to 48000 samples per second.
- RATE\_96000 96 kHz, Set audio sampling rate to 96000 samples per second.
- RATE\_192000 192 kHz, Set audio sampling rate to 192000 samples per second.

**Type** enum in ['RATE\_44100', 'RATE\_48000', 'RATE\_96000', 'RATE\_192000'], default 'RATE\_44100'

**author**

Name that will be used in exported files when format supports such feature

**Type** string, default ""

**color\_picker\_type**

Different styles of displaying the color picker widget

- CIRCLE Circle, A circular Hue/Saturation color wheel, with Value slider.
- SQUARE\_SV Square (SV + H), A square showing Saturation/Value, with Hue slider.
- SQUARE\_HS Square (HS + V), A square showing Hue/Saturation, with Value slider.
- SQUARE\_HV Square (HV + S), A square showing Hue/Value, with Saturation slider.

**Type** enum in ['CIRCLE', 'SQUARE\_SV', 'SQUARE\_HS', 'SQUARE\_HV'], default 'CIRCLE'

**dpi**

Font size and resolution for display

**Type** int in [48, 128], default 0

**frame\_server\_port**

Frameserver Port for Frameserver Rendering

**Type** int in [0, 32727], default 0

**gl\_clip\_alpha**

Clip alpha below this threshold in the 3D textured view

**Type** float in [0, 1], default 0.0

**gl\_texture\_limit**

Limit the texture size to save graphics memory

**Type** enum in ['CLAMP\_OFF', 'CLAMP\_8192', 'CLAMP\_4096', 'CLAMP\_2048', 'CLAMP\_1024', 'CLAMP\_512', 'CLAMP\_256', 'CLAMP\_128'], default 'CLAMP\_OFF'

**language**

Language used for translation

- DEFAULT Default (Default).
- ENGLISH English (English), en\_US.
- FRENCH French (Français), fr\_FR.
- SPANISH Spanish (Español), es\_ES.
- SIMPLIFIED\_CHINESE Simplified Chinese (), zh\_CN.
- JAPANESE Japanese (), ja\_JP.
- DUTCH Dutch (Nederlandse taal), nl\_NL.
- ITALIAN Italian (Italiano), it\_IT.
- GERMAN German (Deutsch), de\_DE.
- FINNISH Finnish (Suomi), fi\_FI.
- SWEDISH Swedish (Svenska), sv\_SE.
- CATALAN Catalan (Català), ca\_AD.



- CZECH Czech (Český), cs\_CZ.
- BRAZILIAN\_PORTUGUESE Brazilian Portuguese (Português do Brasil), pt\_BR.
- TRADITIONAL\_CHINESE Traditional Chinese (), zh\_TW.
- RUSSIAN Russian (), ru\_RU.
- CROATIAN Croatian (Hrvatski), hr\_HR.
- SERBIAN Serbian (), sr\_RS.
- UKRAINIAN Ukrainian (), uk\_UA.
- POLISH Polish (Polski), pl\_PL.
- ROMANIAN Romanian (Român), ro\_RO.
- ARABIC Arabic (), ar\_EG.
- BULGARIAN Bulgarian (), bg\_BG.
- GREEK Greek (Ελληνικ), el\_GR.
- KOREAN Korean (), ko\_KR.
- PERSIAN Persian (), fa\_PE.
- INDONESIAN Indonesian (Bahasa indonesia), id\_ID.

**Type** enum in ['DEFAULT', 'ENGLISH', 'FRENCH', 'SPANISH', 'SIMPLIFIED\_CHINESE', 'JAPANESE', 'DUTCH', 'ITALIAN', 'GERMAN', 'FINNISH', 'SWEDISH', 'CATALAN', 'CZECH', 'BRAZILIAN\_PORTUGUESE', 'TRADITIONAL\_CHINESE', 'RUSSIAN', 'CROATIAN', 'SERBIAN', 'UKRAINIAN', 'POLISH', 'ROMANIAN', 'ARABIC', 'BULGARIAN', 'GREEK', 'KOREAN', 'PERSIAN', 'INDONESIAN'], default 'DEFAULT'

#### **memory\_cache\_limit**

Memory cache limit in sequencer (megabytes)

**Type** int in [0, 16384], default 0

#### **prefetch\_frames**

Number of frames to render ahead during playback

**Type** int in [0, 500], default 0

#### **screencast\_fps**

Frame rate for the screencast to be played back

**Type** int in [10, 50], default 0

#### **screencast\_wait\_time**

Time in milliseconds between each frame recorded for screencast

**Type** int in [50, 1000], default 0

#### **scrollback**

Maximum number of lines to store for the console buffer

**Type** int in [32, 32768], default 0

#### **solid\_lights**

Lights user to display objects in solid draw mode

**Type** bpy\_prop\_collection of UserSolidLight, (readonly)

**texture\_collection\_rate**

Number of seconds between each run of the GL texture garbage collector

**Type** int in [1, 3600], default 0

**texture\_time\_out**

Time since last access of a GL texture in seconds after which it is freed (set to 0 to keep textures allocated)

**Type** int in [0, 3600], default 0

**use\_antialiasing**

Use anti-aliasing for the 3D view (may impact redraw performance)

**Type** boolean, default False

**use\_international\_fonts**

Use international fonts

**Type** boolean, default False

**use\_mipmaps**

Scale textures for the 3D View (looks nicer but uses more memory and slows image reloading)

**Type** boolean, default False

**use\_preview\_images**

Allow user to choose any codec (Windows only, might generate instability)

**Type** boolean, default False

**use\_scripts\_auto\_execute**

Allow any .blend file to run scripts automatically (unsafe with blend files from an untrusted source)

**Type** boolean, default False

**use\_tabs\_as\_spaces**

Automatically convert all new tabs into spaces for new and loaded text files

**Type** boolean, default False

**use\_text\_antialiasing**

Draw user interface text anti-aliased

**Type** boolean, default False

**use\_textured\_fonts**

Use textures for drawing international fonts

**Type** boolean, default False

**use\_translate\_interface**

Translate Interface

**Type** boolean, default False

**use\_translate\_tooltips**

Translate Tooltips

**Type** boolean, default False

**use\_vertex\_buffer\_objects**

Use Vertex Buffer Objects (or Vertex Arrays, if unsupported) for viewport rendering

**Type** boolean, default False

**use\_weight\_color\_range**

Enable color range used for weight visualization in weight painting mode

**Type** boolean, default False

#### **weight\_color\_range**

Color range used for weight visualization in weight painting mode

**Type** `ColorRamp`, (readonly, never None)

#### **window\_draw\_method**

Drawing method used by the window manager

- **AUTOMATIC** Automatic, Automatically set based on graphics card and driver.
- **TRIPLE\_BUFFER** Triple Buffer, Use a third buffer for minimal redraws at the cost of more memory.
- **OVERLAP** Overlap, Redraw all overlapping regions, minimal memory usage but more redraws.
- **OVERLAP\_FLIP** Overlap Flip, Redraw all overlapping regions, minimal memory usage but more redraws (for graphics drivers that do flipping).
- **FULL** Full, Do a full redraw each time, slow, only use for reference or when everything else fails.

**Type** enum in ['AUTOMATIC', 'TRIPLE\_BUFFER', 'OVERLAP', 'OVERLAP\_FLIP', 'FULL'], default 'TRIPLE\_BUFFER'

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `UserPreferences.system`

## 2.4.626 `UserPreferencesView(bpy_struct)`

base class — `bpy_struct`

**class** `bpy.types.UserPreferencesView` (*bpy\_struct*)

Preferences related to viewing data

**manipulator\_handle\_size**

Size of widget handles as percentage of widget radius

**Type** int in [2, 40], default 25

**manipulator\_hotspot**

Pixel distance around the handles to accept mouse clicks

**Type** int in [4, 40], default 14

**manipulator\_size**

Diameter of widget, in 10 pixel units

**Type** int in [2, 40], default 15

**mini\_axis\_brightness**

Brightness of the icon

**Type** int in [0, 10], default 0

**mini\_axis\_size**

The axes icon's size

**Type** int in [10, 64], default 0

**object\_origin\_size**

Diameter in Pixels for Object/Lamp origin display

**Type** int in [4, 10], default 0

**open\_left\_mouse\_delay**

Time in 1/10 seconds to hold the Left Mouse Button before opening the toolbox

**Type** int in [1, 40], default 0

**open\_right\_mouse\_delay**

Time in 1/10 seconds to hold the Right Mouse Button before opening the toolbox

**Type** int in [1, 40], default 0

**open\_sublevel\_delay**

Time delay in 1/10 seconds before automatically opening sub level menus

**Type** int in [1, 40], default 0

**open\_toplevel\_delay**

Time delay in 1/10 seconds before automatically opening top level menus

**Type** int in [1, 40], default 0

**rotation\_angle**

Rotation step for numerical pad keys (2 4 6 8)

**Type** int in [0, 90], default 0

**show\_column\_layout**

Use a column layout for toolbox

**Type** boolean, default False

**show\_large\_cursors**

Use large mouse cursors when available

**Type** boolean, default False

**show\_manipulator**

Use 3D transform manipulator

**Type** boolean, default False

**show\_mini\_axis**

Show a small rotating 3D axes in the bottom left corner of the 3D View

**Type** boolean, default False

**show\_object\_info**

Display objects name and frame number in 3D view

**Type** boolean, default False

**show\_playback\_fps**

Show the frames per second screen refresh rate, while animation is played back

**Type** boolean, default False

**show\_splash**

Display splash screen on startup

**Type** boolean, default False

**show\_tooltips**

Display tooltips

**Type** boolean, default False

**show\_tooltips\_python**

Show Python references in tooltips

**Type** boolean, default False

**show\_view\_name**

Show the name of the view's direction in each 3D View

**Type** boolean, default False

**smooth\_view**

Time to animate the view in milliseconds, zero to disable

**Type** int in [0, 1000], default 0

**timecode\_style**

Format of Time Codes displayed when not displaying timing in terms of frames

- **MINIMAL** Minimal Info, Most compact representation, uses '+' as separator for sub-second frame numbers, with left and right truncation of the timecode as necessary.
- **SMPTE** SMPTE (Full), Full SMPTE timecode (format is HH:MM:SS:FF).
- **SMPTE\_COMPACT** SMPTE (Compact), SMPTE timecode showing minutes, seconds, and frames only - hours are also shown if necessary, but not by default.
- **MILLISECONDS** Compact with Milliseconds, Similar to SMPTE (Compact), except that instead of frames, milliseconds are shown instead.
- **SECONDS\_ONLY** Only Seconds, Direct conversion of frame numbers to seconds.

**Type** enum in ['MINIMAL', 'SMPTE', 'SMPTE\_COMPACT', 'MILLISECONDS', 'SECONDS\_ONLY'], default 'MINIMAL'

**use\_auto\_perspective**

Automatically switch between orthographic and perspective when changing from top/front/side views

**Type** boolean, default False

**use\_camera\_lock\_parent**

When the camera is locked to the view and in fly mode, transform the parent rather than the camera

**Type** boolean, default False

**use\_directional\_menus**

Otherwise menus, etc will always be top to bottom, left to right, no matter opening direction

**Type** boolean, default False

**use\_global\_pivot**

Lock the same rotation/scaling pivot in all 3D Views

**Type** boolean, default False

**use\_global\_scene**

Force the current Scene to be displayed in all Screens

**Type** boolean, default False

**use\_mouse\_auto\_depth**

Use the depth under the mouse to improve view pan/rotate/zoom functionality

**Type** boolean, default False

**use\_mouse\_over\_open**

Open menu buttons and pulldowns automatically when the mouse is hovering

**Type** boolean, default False

**use\_rotate\_around\_active**

Use selection as the pivot point

**Type** boolean, default False

**use\_zoom\_to\_mouse**

Zoom in towards the mouse pointer's position in the 3D view, rather than the 2D window center

**Type** boolean, default False

**view2d\_grid\_spacing\_min**

Minimum number of pixels between each gridline in 2D Viewports

**Type** int in [1, 500], default 0

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`

- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `UserPreferences.view`

## 2.4.627 UserSolidLight(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.UserSolidLight` (*bpy\_struct*)

Light used for OpenGL lighting in solid draw mode

### **diffuse\_color**

Diffuse color of the OpenGL light

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

### **direction**

Direction that the OpenGL light is shining

**Type** float array of 3 items in [-inf, inf], default (0.0, 1.0, 0.0)

### **specular\_color**

Color of the light's specular highlight

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

### **use**

Enable this OpenGL light in solid draw mode

**Type** boolean, default False

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`

- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `UserPreferencesSystem.solid_lights`

## 2.4.628 VectorFont(ID)

base classes — `bpy_struct`, `ID`

**class** `bpy.types.VectorFont` (*ID*)  
Vector font for Text objects

### **filepath**

**Type** string, default "", (readonly)

### **packed\_file**

**Type** `PackedFile`, (readonly)

## Inherited Properties

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`



- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`

## References

- `BlendData.fonts`
- `BlendDataFonts.load`
- `BlendDataFonts.remove`
- `TextCurve.font`
- `TextCurve.font_bold`
- `TextCurve.font_bold_italic`
- `TextCurve.font_italic`

## 2.4.629 VertexColors(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.VertexColors` (*bpy\_struct*)  
Collection of vertex colors

### **active**

Active vertex color layer

**Type** `MeshColorLayer`

### **active\_index**

Active vertex color index

**Type** `int` in `[0, inf]`, default `0`

### **new** (*name="Col"*)

Add a vertex color layer to Mesh

**Parameters** *name* (*string, (optional)*) – Vertex color name

**Returns** The newly created layer

**Return type** `MeshColorLayer`

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`

- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `Mesh.vertex_colors`

## 2.4.630 VertexGroup(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.VertexGroup` (*bpy\_struct*)

Group of vertices, used for armature deform and other purposes

### **index**

Index number of the vertex group

**Type** `int` in `[0, inf]`, default `0`, (readonly)

### **lock\_weight**

Maintain the relative weights for the group

**Type** `boolean`, default `False`

### **name**

Vertex group name

**Type** `string`, default `""`

**add** (*index, weight, type*)

Add vertices to the group

### **Parameters**

- **index** (*int array of 1 items in `[-inf, inf]`*) – Index List
- **weight** (*float in `[0, 1]`*) – Vertex weight
- **type** (*enum in `['REPLACE', 'ADD', 'SUBTRACT']`*) – Vertex assign mode
  - `REPLACE` Replace, Replace.
  - `ADD` Add, Add.
  - `SUBTRACT` Subtract, Subtract.

**remove** (*index*)

Remove a vertex from the group

**Parameters** `index` (*int array of 1 items in [-inf, inf]*) – Index List

**weight** (*index*)

Get a vertex weight from the group

**Parameters** `index` (*int in [0, inf]*) – Index, The index of the vertex

**Returns** Vertex weight

**Return type** float in [0, 1]

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Object.vertex_groups`
- `VertexGroups.active`
- `VertexGroups.new`
- `VertexGroups.remove`

## 2.4.631 VertexGroupElement(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.VertexGroupElement` (*bpy\_struct*)

Weight value of a vertex in a vertex group

**group**

**Type** int in [0, inf], default 0, (readonly)

**weight**

Vertex Weight

**Type** float in [0, 1], default 0.0

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `LatticePoint.groups`
- `MeshVertex.groups`

## 2.4.632 VertexGroups(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.VertexGroups` (`bpy_struct`)  
Collection of vertex groups

**active**

Vertex groups of the object

**Type** `VertexGroup`, (readonly)

**active\_index**

Active index in vertex group array

**Type** int in [-32768, 32767], default 0

**new** (`name="Group"`)

Add vertex group to object

**Parameters** `name` (`string`, *optional*) – Vertex group name

**Returns** New vertex group

**Return type** `VertexGroup`

**remove** (*group*)

Delete vertex group from object

**Parameters** *group* (*VertexGroup*, (never *None*)) – Vertex group to remove

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `Object.vertex_groups`

## 2.4.633 VertexPaint(Paint)

base classes — `bpy_struct`, `Paint`

**class** `bpy.types.VertexPaint` (*Paint*)

Properties of vertex and weight paint mode

**use\_all\_faces**

Paint on all faces inside brush

**Type** boolean, default `False`

**use\_normal**

Applies the vertex normal before painting

**Type** boolean, default `False`

**use\_spray**

Keep applying paint effect while holding mouse

**Type** boolean, default `False`

### Inherited Properties

- `bpy_struct.id_data`
- `Paint.brush`
- `Paint.show_low_resolution`
- `Paint.show_brush`
- `Paint.show_brush_on_surface`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `ToolSettings.vertex_paint`
- `ToolSettings.weight_paint`

## 2.4.634 VertexWeightEditModifier(Modifier)

base classes — `bpy_struct`, `Modifier`

**class** `bpy.types.VertexWeightEditModifier` (*Modifier*)

Edit the weights of vertices in a group

**add\_threshold**

Lower bound for a vertex's weight to be added to the vgroup

**Type** float in [0, 1], default 0.0

**default\_weight**

Default weight a vertex will have if it is not in the vgroup

**Type** float in [0, 1], default 0.0

**falloff\_type**

How weights are mapped to there new values

**Type** enum in ['LINEAR', 'CURVE', 'SHARP', 'SMOOTH', 'ROOT', 'ICON\_SPHERECURVE', 'RANDOM', 'STEP'], default 'LINEAR'

**map\_curve**

Custom mapping curve

**Type** `CurveMapping`, (readonly)

**mask\_constant**

Global influence of current modifications on vgroup

**Type** float in [-inf, inf], default 0.0

**mask\_tex\_map\_object**

Which object to take texture coordinates from

**Type** `Object`

**mask\_tex\_mapping**

Which texture coordinates to use for mapping

**Type** enum in ['LOCAL', 'GLOBAL', 'OBJECT', 'UV'], default 'LOCAL'

**mask\_tex\_use\_channel**

Which texture channel to use for masking

**Type** enum in ['INT', 'RED', 'GREEN', 'BLUE', 'HUE', 'SAT', 'VAL', 'ALPHA'], default 'INT'

**mask\_tex\_uv\_layer**

UV layer name

**Type** string, default ""

**mask\_texture**

Masking texture

**Type** `Texture`

**mask\_vertex\_group**

Masking vertex group name

**Type** string, default ""

**remove\_threshold**

Upper bound for a vertex's weight to be removed from the vgroup

**Type** float in [0, 1], default 0.0

**use\_add**

Add vertices with weight over threshold to vgroup

**Type** boolean, default False

**use\_remove**

Remove vertices with weight below threshold from vgroup

**Type** boolean, default False

**vertex\_group**

Vertex group name

**Type** string, default ""

**Inherited Properties**

- `bpy_struct.id_data`

- `Modifier.name`
- `Modifier.use_apply_on_spline`
- `Modifier.show_in_editmode`
- `Modifier.show_expanded`
- `Modifier.show_on_cage`
- `Modifier.show_viewport`
- `Modifier.show_render`
- `Modifier.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## 2.4.635 VertexWeightMixModifier(Modifier)

base classes — `bpy_struct`, `Modifier`

**class** `bpy.types.VertexWeightMixModifier` (*Modifier*)

Mix the weights of two vertex groups

**default\_weight\_a**

Default weight a vertex will have if it is not in the first vgroup

**Type** float in [0, 1], default 0.0

**default\_weight\_b**

Default weight a vertex will have if it is not in the second vgroup

**Type** float in [0, 1], default 0.0

**mask\_constant**

Global influence of current modifications on vgroup

**Type** float in [-inf, inf], default 0.0

**mask\_tex\_map\_object**

Which object to take texture coordinates from

**Type** `Object`

**mask\_tex\_mapping**

Which texture coordinates to use for mapping



**Type** enum in ['LOCAL', 'GLOBAL', 'OBJECT', 'UV'], default 'LOCAL'

**mask\_tex\_use\_channel**

Which texture channel to use for masking

**Type** enum in ['INT', 'RED', 'GREEN', 'BLUE', 'HUE', 'SAT', 'VAL', 'ALPHA'], default 'INT'

**mask\_tex\_uv\_layer**

UV layer name

**Type** string, default ""

**mask\_texture**

Masking texture

**Type** `Texture`

**mask\_vertex\_group**

Masking vertex group name

**Type** string, default ""

**mix\_mode**

How weights from vgroup 2 affect weights of vgroup 1

**Type** enum in ['SET', 'ADD', 'SUB', 'MUL', 'DIV', 'DIF', 'AVG'], default 'SET'

**mix\_set**

Which vertices should be affected

**Type** enum in ['ALL', 'A', 'B', 'OR', 'AND'], default 'ALL'

**vertex\_group\_a**

First vertex group name

**Type** string, default ""

**vertex\_group\_b**

Second vertex group name

**Type** string, default ""

### Inherited Properties

- `bpy_struct.id_data`
- `Modifier.name`
- `Modifier.use_apply_on_spline`
- `Modifier.show_in_editmode`
- `Modifier.show_expanded`
- `Modifier.show_on_cage`
- `Modifier.show_viewport`
- `Modifier.show_render`
- `Modifier.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`

- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.636 VertexWeightProximityModifier(Modifier)

base classes — `bpy_struct`, `Modifier`

**class** `bpy.types.VertexWeightProximityModifier` (*Modifier*)

Set the weights of vertices in a group from a target object's distance

**falloff\_type**

How weights are mapped to there new values

**Type** enum in ['LINEAR', 'SHARP', 'SMOOTH', 'ROOT', 'ICON\_SPHERECURVE', 'RANDOM', 'STEP'], default 'LINEAR'

**mask\_constant**

Global influence of current modifications on vgroup

**Type** float in [-inf, inf], default 0.0

**mask\_tex\_map\_object**

Which object to take texture coordinates from

**Type** `Object`

**mask\_tex\_mapping**

Which texture coordinates to use for mapping

**Type** enum in ['LOCAL', 'GLOBAL', 'OBJECT', 'UV'], default 'LOCAL'

**mask\_tex\_use\_channel**

Which texture channel to use for masking

**Type** enum in ['INT', 'RED', 'GREEN', 'BLUE', 'HUE', 'SAT', 'VAL', 'ALPHA'], default 'INT'

**mask\_tex\_uv\_layer**

UV layer name

**Type** string, default ""

**mask\_texture**

Masking texture

**Type** `Texture`

**mask\_vertex\_group**

Masking vertex group name

**Type** string, default ""

**max\_dist**

Distance mapping to weight 1.0 (or weight 0.0 if below Lowest Dist)

**Type** float in [0, inf], default 0.0

**min\_dist**

Distance mapping to weight 0.0 (or weight 1.0 if above Highest Dist)

**Type** float in [0, inf], default 0.0

**proximity\_geometry**

Use the shortest computed distance to target object's geometry as weight

- VERTEX Vertex, Compute distance to nearest vertex.
- EDGE Edge, Compute distance to nearest edge.
- FACE Face, Compute distance to nearest face.

**Type** enum set in {'VERTEX', 'EDGE', 'FACE'}, default {'VERTEX'}

**proximity\_mode**

Which distances to target object to use

- OBJECT Object Distance, Use distance between affected and target objects.
- GEOMETRY Geometry Distance, Use distance between affected object's vertices and target object, or target object's geometry.

**Type** enum in ['OBJECT', 'GEOMETRY'], default 'OBJECT'

**target**

Object to calculate vertices distances from

**Type** Object

**vertex\_group**

Vertex group name

**Type** string, default ""

**Inherited Properties**

- bpy\_struct.id\_data
- Modifier.name
- Modifier.use\_apply\_on\_spline
- Modifier.show\_in\_editmode
- Modifier.show\_expanded
- Modifier.show\_on\_cage
- Modifier.show\_viewport
- Modifier.show\_render
- Modifier.type

**Inherited Functions**

- bpy\_struct.as\_pointer
- bpy\_struct.callback\_add
- bpy\_struct.callback\_remove

- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.637 VisibilityActuator(Actuator)

base classes — `bpy_struct`, `Actuator`

**class** `bpy.types.VisibilityActuator` (*Actuator*)

Actuator to set visibility and occlusion of the object

**apply\_to\_children**

Set all the children of this object to the same visibility/occlusion recursively

**Type** boolean, default False

**use\_occlusion**

Set the object to occlude objects behind it. Initialized from the object type in physics button

**Type** boolean, default False

**use\_visible**

Set the objects visible. Initialized from the object render restriction toggle in physics button

**Type** boolean, default False

#### Inherited Properties

- `bpy_struct.id_data`
- `Actuator.name`
- `Actuator.show_expanded`
- `Actuator.pin`
- `Actuator.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`

- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Actuator.link`
- `Actuator.unlink`

## 2.4.638 VoronoiTexture(Texture)

base classes — `bpy_struct`, `ID`, `Texture`

**class** `bpy.types.VoronoiTexture` (*Texture*)

Procedural voronoi texture

### **color\_mode**

- **INTENSITY** Intensity, Only calculate intensity.
- **POSITION** Position, Color cells by position.
- **POSITION\_OUTLINE** Position and Outline, Use position plus an outline based on F2-F.1.
- **POSITION\_OUTLINE\_INTENSITY** Position, Outline, and Intensity, Multiply position and outline by intensity.

**Type** enum in ['INTENSITY', 'POSITION', 'POSITION\_OUTLINE', 'POSITION\_OUTLINE\_INTENSITY'], default 'INTENSITY'

### **distance\_metric**

- **DISTANCE** Actual Distance, Algorithm used to calculate distance of sample points to feature points; Actual Distance:  $\sqrt{x*x+y*y+z*z}$ .
- **DISTANCE\_SQUARED** Distance Squared, Algorithm used to calculate distance of sample points to feature points; Distance squared:  $(x*x+y*y+z*z)$ .
- **MANHATTAN** Manhattan, Algorithm used to calculate distance of sample points to feature points; Manhattan: The length of the distance in axial directions.
- **CHEBYCHEV** Chebychev, Algorithm used to calculate distance of sample points to feature points; Chebychev: The length of the longest Axial journey.
- **MINKOVSKY\_HALF** Minkovsky 1/2, Algorithm used to calculate distance of sample points to feature points; Minovsky 1/2: Sets Minkovsky variable to 0.5.
- **MINKOVSKY\_FOUR** Minkovsky 4, Algorithm used to calculate distance of sample points to feature points; Minkovsky 4: Sets Minkovsky variable to 4.
- **MINKOVSKY** Minkovsky, Algorithm used to calculate distance of sample points to feature points; Minkovsky: Uses the Minkovsky function to calculate distance (exponent value determines the shape of the boundaries).

**Type** enum in ['DISTANCE', 'DISTANCE\_SQUARED', 'MANHATTAN', 'CHEBYCHEV', 'MINKOVSKY\_HALF', 'MINKOVSKY\_FOUR', 'MINKOVSKY'], default 'DISTANCE'

### **minkovsky\_exponent**

Minkovsky exponent

**Type** float in [0.01, 10], default 0.0

**nabla**

Size of derivative offset used for calculating normal

**Type** float in [0.001, 0.1], default 0.0

**noise\_intensity**

Scales the intensity of the noise

**Type** float in [0.01, 10], default 0.0

**noise\_scale**

Sets scaling for noise input

**Type** float in [0.0001, inf], default 0.0

**weight\_1**

Voronoi feature weight 1

**Type** float in [-2, 2], default 0.0

**weight\_2**

Voronoi feature weight 2

**Type** float in [-2, 2], default 0.0

**weight\_3**

Voronoi feature weight 3

**Type** float in [-2, 2], default 0.0

**weight\_4**

Voronoi feature weight 4

**Type** float in [-2, 2], default 0.0

**users\_material**

Materials that use this texture (readonly)

**users\_object\_modifier**

Object modifiers that use this texture (readonly)

### Inherited Properties

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`
- `Texture.animation_data`
- `Texture.intensity`
- `Texture.color_ramp`
- `Texture.contrast`
- `Texture.factor_blue`
- `Texture.factor_green`
- `Texture.factor_red`
- `Texture.node_tree`

- `Texture.saturation`
- `Texture.use_preview_alpha`
- `Texture.type`
- `Texture.use_color_ramp`
- `Texture.use_nodes`
- `Texture.users_material`
- `Texture.users_object_modifier`
- `Texture.users_material`
- `Texture.users_object_modifier`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`
- `Texture.evaluate`

### 2.4.639 VoxelData(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.VoxelData` (*bpy\_struct*)

Voxel data settings

#### **domain\_object**

Object used as the smoke simulation domain

**Type** `Object`

#### **extension**

Sets how the texture is extrapolated past its original bounds

- **EXTEND** Extend, Extend by repeating edge pixels of the image.
- **CLIP** Clip, Clip to image size and set exterior pixels as transparent.
- **REPEAT** Repeat, Cause the image to repeat horizontally and vertically.

**Type** enum in ['EXTEND', 'CLIP', 'REPEAT'], default 'EXTEND'

**file\_format**

Format of the source data set to render

- BLENDER\_VOXEL Blender Voxel, Default binary voxel file format.
- RAW\_8BIT 8 bit RAW, 8 bit greyscale binary data.
- IMAGE\_SEQUENCE Image Sequence, Generate voxels from a sequence of image slices.
- SMOKE Smoke, Render voxels from a Blender smoke simulation.

**Type** enum in ['BLENDER\_VOXEL', 'RAW\_8BIT', 'IMAGE\_SEQUENCE', 'SMOKE'], default 'BLENDER\_VOXEL'

**filepath**

The external source data file to use

**Type** string, default ""

**intensity**

Multiplier for intensity values

**Type** float in [0.01, inf], default 0.0

**interpolation**

Method to interpolate/smooth values between voxel cells

- NEREASTNEIGHBOR Nearest Neighbor, No interpolation, fast but blocky and low quality.
- TRILINEAR Linear, Good smoothness and speed.
- QUADRATIC Quadratic, Mid-range quality and speed.
- TRICUBIC\_CATROM Cubic Catmull-Rom, High quality interpolation, but slower.
- TRICUBIC\_BSPLINE Cubic B-Spline, Smoothed high quality interpolation, but slower.

**Type** enum in ['NEREASTNEIGHBOR', 'TRILINEAR', 'QUADRATIC', 'TRICUBIC\_CATROM', 'TRICUBIC\_BSPLINE'], default 'NEREASTNEIGHBOR'

**resolution**

Resolution of the voxel grid

**Type** int array of 3 items in [1, 100000], default (0, 0, 0)

**smoke\_data\_type**

Simulation value to be used as a texture

- SMOKEDENSITY Density, Use smoke density as texture data.
- SMOKEHEAT Heat, Use smoke heat as texture data. Values from -2.0 to 2.0 are used.
- SMOKELEVEL Velocity, Use smoke velocity as texture data.

**Type** enum in ['SMOKEDENSITY', 'SMOKEHEAT', 'SMOKELEVEL'], default 'SMOKEDENSITY'

**still\_frame**

The frame number to always use

**Type** int in [-300000, 300000], default 0



**use\_still\_frame**

Always render a still frame from the voxel data sequence

**Type** boolean, default False

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `VoxelDataTexture.voxel_data`

**2.4.640 VoxelDataTexture(Texture)**

base classes — `bpy_struct`, `ID`, `Texture`

**class** `bpy.types.VoxelDataTexture` (*Texture*)

Settings for the Voxel Data texture

**image**

**Type** `Image`

**image\_user**

Parameters defining which layer, pass and frame of the image is displayed

**Type** `ImageUser`, (readonly)

**voxel\_data**

The voxel data associated with this texture

**Type** `VoxelData`, (readonly)

**users\_material**

Materials that use this texture (readonly)

### **users\_object\_modifier**

Object modifiers that use this texture (readonly)

#### **Inherited Properties**

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`
- `Texture.animation_data`
- `Texture.intensity`
- `Texture.color_ramp`
- `Texture.contrast`
- `Texture.factor_blue`
- `Texture.factor_green`
- `Texture.factor_red`
- `Texture.node_tree`
- `Texture.saturation`
- `Texture.use_preview_alpha`
- `Texture.type`
- `Texture.use_color_ramp`
- `Texture.use_nodes`
- `Texture.users_material`
- `Texture.users_object_modifier`
- `Texture.users_material`
- `Texture.users_object_modifier`

#### **Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`

- `ID.animation_data_clear`
- `ID.update_tag`
- `Texture.evaluate`

## 2.4.641 WarpModifier(Modifier)

base classes — `bpy_struct`, `Modifier`

**class** `bpy.types.WarpModifier` (*Modifier*)

Warp modifier

**falloff\_curve**

Custom Lamp Falloff Curve

**Type** `CurveMapping`, (readonly)

**falloff\_radius**

Radius to apply

**Type** float in [-inf, inf], default 0.0

**falloff\_type**

**Type** enum in ['NONE', 'CURVE', 'SMOOTH', 'SPHERE', 'ROOT', 'SHARP', 'LINEAR', 'CONSTANT'], default 'NONE'

**object\_from**

Object to transform from

**Type** `Object`

**object\_to**

Object to transform to

**Type** `Object`

**strength**

**Type** float in [-inf, inf], default 0.0

**texture**

**Type** `Texture`

**texture\_coords**

- `LOCAL` Map, Uses the local coordinate system for the texture coordinates.
- `GLOBAL` Global, Uses the global coordinate system for the texture coordinates.
- `OBJECT` Object, Uses the linked object's local coordinate system for the texture coordinates.
- `UV` UV, Uses UV coordinates for the texture coordinates.

**Type** enum in ['LOCAL', 'GLOBAL', 'OBJECT', 'UV'], default 'LOCAL'

**texture\_coords\_object**

Object to set the texture coordinates

**Type** `Object`

**use\_volume\_preserve**

Preserve volume when rotations are used

**Type** boolean, default False

**uv\_layer**

UV layer name

**Type** string, default ""

**vertex\_group**

Vertex group name for modulating the deform

**Type** string, default ""

**Inherited Properties**

- `bpy_struct.id_data`
- `Modifier.name`
- `Modifier.use_apply_on_spline`
- `Modifier.show_in_editmode`
- `Modifier.show_expanded`
- `Modifier.show_on_cage`
- `Modifier.show_viewport`
- `Modifier.show_render`
- `Modifier.type`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**2.4.642 WaveModifier(Modifier)**

base classes — `bpy_struct`, `Modifier`

**class** `bpy.types.WaveModifier` (*Modifier*)

Wave effect modifier

**damping\_time**

Number of frames in which the waves damps out after it dies

**Type** float in [-300000, 300000], default 0.0

**falloff\_radius**

Distance after which it fades out

**Type** float in [0, inf], default 0.0

#### **height**

Height of the wave

**Type** float in [-inf, inf], default 0.0

#### **lifetime**

Lifetime of the wave in frames, zero means infinite

**Type** float in [-300000, 300000], default 0.0

#### **narrowness**

Distance between the top and the base of a wave, the higher the value, the more narrow the wave

**Type** float in [0, inf], default 0.0

#### **speed**

Speed of the wave, towards the starting point when negative

**Type** float in [-inf, inf], default 0.0

#### **start\_position\_object**

Object which defines the wave center

**Type** *Object*

#### **start\_position\_x**

X coordinate of the start position

**Type** float in [-inf, inf], default 0.0

#### **start\_position\_y**

Z coordinate of the start position

**Type** float in [-inf, inf], default 0.0

#### **texture**

Texture for modulating the wave

**Type** *Texture*

#### **texture\_coords**

Texture coordinates used for modulating input

**Type** enum in ['LOCAL', 'GLOBAL', 'OBJECT', 'MAP\_UV'], default 'LOCAL'

#### **texture\_coords\_object**

**Type** *Object*

#### **time\_offset**

Either the starting frame (for positive speed) or ending frame (for negative speed.)

**Type** float in [-300000, 300000], default 0.0

#### **use\_cyclic**

Cyclic wave effect

**Type** boolean, default False

#### **use\_normal**

Displace along normals

**Type** boolean, default False

**use\_normal\_x**

Enable displacement along the X normal

**Type** boolean, default False

**use\_normal\_y**

Enable displacement along the Y normal

**Type** boolean, default False

**use\_normal\_z**

Enable displacement along the Z normal

**Type** boolean, default False

**use\_x**

X axis motion

**Type** boolean, default False

**use\_y**

Y axis motion

**Type** boolean, default False

**uv\_layer**

UV layer name

**Type** string, default ""

**vertex\_group**

Vertex group name for modulating the wave

**Type** string, default ""

**width**

Distance between the waves

**Type** float in [0, inf], default 0.0

### Inherited Properties

- `bpy_struct.id_data`
- `Modifier.name`
- `Modifier.use_apply_on_spline`
- `Modifier.show_in_editmode`
- `Modifier.show_expanded`
- `Modifier.show_on_cage`
- `Modifier.show_viewport`
- `Modifier.show_render`
- `Modifier.type`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`

- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### 2.4.643 Window(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.Window` (*bpy\_struct*)  
Open window

**screen**

Active screen showing in the window

**Type** `Screen`, (never `None`)

#### Inherited Properties

- `bpy_struct.id_data`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

#### References

- `Context.window`
- `WindowManager.event_timer_add`
- `WindowManager.windows`

## 2.4.644 WindowManager(ID)

base classes — `bpy_struct`, `ID`

**class** `bpy.types.WindowManager` (*ID*)

Window manager datablock defining open windows and other user interface data

**addon\_filter**

Filter addons by category

**Type** enum in [], default ""

**addon\_search**

Search within the selected filter

**Type** string, default ""

**addon\_support**

Display support level

**Type** enum set in {'OFFICIAL', 'COMMUNITY'}, default {'OFFICIAL', 'COMMUNITY'}

**clipboard**

**Type** string, default ""

**keyconfigs**

Registered key configurations

**Type** `KeyConfigurations` `bpy_prop_collection` of `KeyConfig`, (readonly)

**operators**

Operator registry

**Type** `bpy_prop_collection` of `Operator`, (readonly)

**windows**

Open windows

**Type** `bpy_prop_collection` of `Window`, (readonly)

**classmethod** `fileselect_add` (*operator*)

Show up the file selector

**Parameters** `operator` (`Operator`) – Operator to call

**classmethod** `modal_handler_add` (*operator*)

`modal_handler_add`

**Parameters** `operator` (`Operator`) – Operator to call

**Return type** boolean

**event\_timer\_add** (*time\_step*, *window=None*)

`event_timer_add`

**Parameters**

- **time\_step** (*float in [0, inf]*) – Time Step, Interval in seconds between timer events
- **window** (`Window`, (optional)) – Window to attach the timer to or None

**Return type** `Timer`

**event\_timer\_remove** (*timer*)

`event_timer_remove`



**classmethod `invoke_props_popup`** (*operator*, *event*)

Operator popup invoke

**Parameters**

- **operator** (`Operator`) – Operator to call
- **event** (`Event`) – Event

**Returns**

result

- `RUNNING_MODAL` Running Modal, Keep the operator running with blender.
- `CANCELLED` Cancelled, When no action has been taken, operator exits.
- `FINISHED` Finished, When the operator is complete, operator exits.
- `PASS_THROUGH` Pass Through, Do nothing and pass the event on.

**Return type** enum set in {`'RUNNING_MODAL'`, `'CANCELLED'`, `'FINISHED'`, `'PASS_THROUGH'`}

**classmethod `invoke_props_dialog`** (*operator*, *width=300*, *height=20*)

Operator dialog (non-autoexec popup) invoke

**Parameters**

- **operator** (`Operator`) – Operator to call
- **width** (*int in [0, inf], (optional)*) – Width of the popup
- **height** (*int in [0, inf], (optional)*) – Height of the popup

**Returns**

result

- `RUNNING_MODAL` Running Modal, Keep the operator running with blender.
- `CANCELLED` Cancelled, When no action has been taken, operator exits.
- `FINISHED` Finished, When the operator is complete, operator exits.
- `PASS_THROUGH` Pass Through, Do nothing and pass the event on.

**Return type** enum set in {`'RUNNING_MODAL'`, `'CANCELLED'`, `'FINISHED'`, `'PASS_THROUGH'`}

**classmethod `invoke_search_popup`** (*operator*)

invoke\_search\_popup

**Parameters** **operator** (`Operator`) – Operator to call

**classmethod `invoke_popup`** (*operator*, *width=300*, *height=20*)

Operator popup invoke

**Parameters**

- **operator** (`Operator`) – Operator to call
- **width** (*int in [0, inf], (optional)*) – Width of the popup
- **height** (*int in [0, inf], (optional)*) – Height of the popup

### Returns

result

- `RUNNING_MODAL` Running Modal, Keep the operator running with blender.
- `CANCELLED` Cancelled, When no action has been taken, operator exits.
- `FINISHED` Finished, When the operator is complete, operator exits.
- `PASS_THROUGH` Pass Through, Do nothing and pass the event on.

**Return type** enum set in {`'RUNNING_MODAL'`, `'CANCELLED'`, `'FINISHED'`, `'PASS_THROUGH'`}

**classmethod** `invoke_confirm` (*operator*, *event*)

Operator confirmation

### Parameters

- **operator** (`Operator`) – Operator to call
- **event** (`Event`) – Event

### Returns

result

- `RUNNING_MODAL` Running Modal, Keep the operator running with blender.
- `CANCELLED` Cancelled, When no action has been taken, operator exits.
- `FINISHED` Finished, When the operator is complete, operator exits.
- `PASS_THROUGH` Pass Through, Do nothing and pass the event on.

**Return type** enum set in {`'RUNNING_MODAL'`, `'CANCELLED'`, `'FINISHED'`, `'PASS_THROUGH'`}

## Inherited Properties

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`

- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`

## References

- `BlendData.window_managers`
- `Context.window_manager`

## 2.4.645 WipeSequence(EffectSequence)

base classes — `bpy_struct`, `Sequence`, `EffectSequence`

**class** `bpy.types.WipeSequence` (*EffectSequence*)

Sequence strip creating a wipe transition

### **angle**

Edge angle

**Type** float in [-1.5708, 1.5708], default 0.0

### **blur\_width**

Width of the blur edge, in percentage relative to the image size

**Type** float in [0, 1], default 0.0

### **direction**

Wipe direction

**Type** enum in ['OUT', 'IN'], default 'OUT'

### **transition\_type**

**Type** enum in ['SINGLE', 'DOUBLE', 'IRIS', 'CLOCK'], default 'SINGLE'

## Inherited Properties

- `bpy_struct.id_data`
- `Sequence.name`
- `Sequence.blend_type`
- `Sequence.blend_alpha`
- `Sequence.channel`
- `Sequence.waveform`
- `Sequence.effect_fader`
- `Sequence.frame_final_end`
- `Sequence.frame_offset_end`
- `Sequence.frame_still_end`

- `Sequence.input_1`
- `Sequence.input_2`
- `Sequence.input_3`
- `Sequence.select_left_handle`
- `Sequence.frame_final_duration`
- `Sequence.frame_duration`
- `Sequence.lock`
- `Sequence.mute`
- `Sequence.select_right_handle`
- `Sequence.select`
- `Sequence.speed_factor`
- `Sequence.frame_start`
- `Sequence.frame_final_start`
- `Sequence.frame_offset_start`
- `Sequence.frame_still_start`
- `Sequence.type`
- `Sequence.use_default_fade`
- `Sequence.input_count`
- `EffectSequence.color_balance`
- `EffectSequence.use_float`
- `EffectSequence.crop`
- `EffectSequence.use_deinterlace`
- `EffectSequence.use_reverse_frames`
- `EffectSequence.use_flip_x`
- `EffectSequence.use_flip_y`
- `EffectSequence.color_multiply`
- `EffectSequence.use_premultiply`
- `EffectSequence.proxy`
- `EffectSequence.use_proxy_custom_directory`
- `EffectSequence.use_proxy_custom_file`
- `EffectSequence.color_saturation`
- `EffectSequence.strobe`
- `EffectSequence.ttransform`
- `EffectSequence.use_color_balance`
- `EffectSequence.use_crop`
- `EffectSequence.use_proxy`
- `EffectSequence.use_translation`

### **Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`

- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Sequence.getStripElem`
- `Sequence.swap`

## 2.4.646 WoodTexture(Texture)

base classes — `bpy_struct`, `ID`, `Texture`

**class** `bpy.types.WoodTexture` (*Texture*)

Procedural noise texture

### **nabla**

Size of derivative offset used for calculating normal

**Type** float in [0.001, 0.1], default 0.0

### **noise\_basis**

Sets the noise basis used for turbulence

- `BLENDER_ORIGINAL` Blender Original, Noise algorithm - Blender original: Smooth interpolated noise.
- `ORIGINAL_PERLIN` Original Perlin, Noise algorithm - Original Perlin: Smooth interpolated noise.
- `IMPROVED_PERLIN` Improved Perlin, Noise algorithm - Improved Perlin: Smooth interpolated noise.
- `VORONOI_F1` Voronoi F1, Noise algorithm - Voronoi F1: Returns distance to the closest feature point.
- `VORONOI_F2` Voronoi F2, Noise algorithm - Voronoi F2: Returns distance to the 2nd closest feature point.
- `VORONOI_F3` Voronoi F3, Noise algorithm - Voronoi F3: Returns distance to the 3rd closest feature point.
- `VORONOI_F4` Voronoi F4, Noise algorithm - Voronoi F4: Returns distance to the 4th closest feature point.
- `VORONOI_F2_F1` Voronoi F2-F1, Noise algorithm - Voronoi F1-F2.
- `VORONOI_CRACKLE` Voronoi Crackle, Noise algorithm - Voronoi Crackle: Voronoi tessellation with sharp edges.
- `CELL_NOISE` Cell Noise, Noise algorithm - Cell Noise: Square cell tessellation.

**Type** enum in ['BLENDER\_ORIGINAL', 'ORIGINAL\_PERLIN', 'IMPROVED\_PERLIN', 'VORONOI\_F1', 'VORONOI\_F2', 'VORONOI\_F3', 'VORONOI\_F4', 'VORONOI\_F2\_F1', 'VORONOI\_CRACKLE', 'CELL\_NOISE'], default 'BLENDER\_ORIGINAL'

### **noise\_basis\_2**

- `SIN` Sine, Uses a sine wave to produce bands.
- `SAW` Saw, Uses a saw wave to produce bands.
- `TRI` Tri, Uses a triangle wave to produce bands.

**Type** enum in ['SIN', 'SAW', 'TRI'], default 'SIN'

**noise\_scale**

Sets scaling for noise input

**Type** float in [0.0001, inf], default 0.0

**noise\_type**

- SOFT\_NOISE Soft, Generate soft noise (smooth transitions).
- HARD\_NOISE Hard, Generate hard noise (sharp transitions).

**Type** enum in ['SOFT\_NOISE', 'HARD\_NOISE'], default 'SOFT\_NOISE'

**turbulence**

Sets the turbulence of the bandnoise and ringnoise types

**Type** float in [0.0001, inf], default 0.0

**wood\_type**

- BANDS Bands, Uses standard wood texture in bands.
- RINGS Rings, Uses wood texture in rings.
- BANDNOISE Band Noise, Adds noise to standard wood.
- RINGNOISE Ring Noise, Adds noise to rings.

**Type** enum in ['BANDS', 'RINGS', 'BANDNOISE', 'RINGNOISE'], default 'BANDS'

**users\_material**

Materials that use this texture (readonly)

**users\_object\_modifier**

Object modifiers that use this texture (readonly)

**Inherited Properties**

- bpy\_struct.id\_data
- ID.name
- ID.use\_fake\_user
- ID.is\_updated
- ID.is\_updated\_data
- ID.library
- ID.tag
- ID.users
- Texture.animation\_data
- Texture.intensity
- Texture.color\_ramp
- Texture.contrast
- Texture.factor\_blue
- Texture.factor\_green
- Texture.factor\_red
- Texture.node\_tree
- Texture.saturation
- Texture.use\_preview\_alpha
- Texture.type
- Texture.use\_color\_ramp

- `Texture.use_nodes`
- `Texture.users_material`
- `Texture.users_object_modifier`
- `Texture.users_material`
- `Texture.users_object_modifier`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`
- `Texture.evaluate`

## 2.4.647 World(ID)

base classes — `bpy_struct`, `ID`

**class** `bpy.types.World` (*ID*)

World datablock describing the environment and ambient lighting of a scene

**active\_texture**

Active texture slot being displayed

**Type** `Texture`

**active\_texture\_index**

Index of active texture slot

**Type** `int` in `[0, 17]`, default `0`

**ambient\_color**

Ambient color of the world

**Type** `float` array of 3 items in `[-inf, inf]`, default `(0.0, 0.0, 0.0)`

**animation\_data**

Animation data for this datablock

**Type** `AnimData`, (readonly)

**color\_range**

The color range that will be mapped to 0-1

**Type** float in [0.2, 5], default 0.0

**exposure**

Amount of exponential color correction for light

**Type** float in [0, 1], default 0.0

**horizon\_color**

Color at the horizon

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)

**light\_settings**

World lighting settings

**Type** `WorldLighting`, (readonly, never None)

**mist\_settings**

World mist settings

**Type** `WorldMistSettings`, (readonly, never None)

**node\_tree**

Node tree for node based worlds

**Type** `NodeTree`, (readonly)

**star\_settings**

World stars settings

**Type** `WorldStarsSettings`, (readonly, never None)

**texture\_slots**

Texture slots defining the mapping and influence of textures

**Type** `WorldTextureSlots` `bpy_prop_collection` of `WorldTextureSlot`, (readonly)

**use\_nodes**

Use shader nodes to render the world

**Type** boolean, default False

**use\_sky\_blend**

Render background with natural progression from horizon to zenith

**Type** boolean, default False

**use\_sky\_paper**

Flatten blend or texture coordinates

**Type** boolean, default False

**use\_sky\_real**

Render background with a real horizon, relative to the camera angle

**Type** boolean, default False

**zenith\_color**

Color at the zenith

**Type** float array of 3 items in [-inf, inf], default (0.0, 0.0, 0.0)



## Inherited Properties

- `bpy_struct.id_data`
- `ID.name`
- `ID.use_fake_user`
- `ID.is_updated`
- `ID.is_updated_data`
- `ID.library`
- `ID.tag`
- `ID.users`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `ID.copy`
- `ID.user_clear`
- `ID.animation_data_create`
- `ID.animation_data_clear`
- `ID.update_tag`

## References

- `BlendData.worlds`
- `BlendDataWorlds.new`
- `BlendDataWorlds.remove`
- `Scene.world`

### 2.4.648 WorldLighting(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.WorldLighting` (*bpy\_struct*)  
Lighting for a World datablock

#### **adapt\_to\_speed**

Use the speed vector pass to reduce AO samples in fast moving pixels. Higher values result in more aggressive sample reduction. Requires Vec pass enabled (for Raytrace Adaptive QMC)

**Type** float in [0, 1], default 0.0

**ao\_blend\_type**

Defines how AO mixes with material shading

- MULTIPLY Multiply, Multiply direct lighting with ambient occlusion, darkening the result.
- ADD Add, Add light and shadow.

**Type** enum in ['MULTIPLY', 'ADD'], default 'ADD'

**ao\_factor**

Factor for ambient occlusion blending

**Type** float in [0, inf], default 0.0

**bias**

Bias (in radians) to prevent smoothed faces from showing banding (for Raytrace Constant Jittered)

**Type** float in [0, 0.5], default 0.0

**correction**

Ad-hoc correction for over-occlusion due to the approximation

**Type** float in [0, 1], default 0.0

**distance**

Length of rays, defines how far away other faces give occlusion effect

**Type** float in [-inf, inf], default 0.0

**environment\_color**

Defines where the color of the environment light comes from

- PLAIN White, Plain diffuse energy (white.).
- SKY\_COLOR Sky Color, Use horizon and zenith color for diffuse energy.
- SKY\_TEXTURE Sky Texture, Does full Sky texture render for diffuse energy.

**Type** enum in ['PLAIN', 'SKY\_COLOR', 'SKY\_TEXTURE'], default 'PLAIN'

**environment\_energy**

Defines the strength of environment light

**Type** float in [-inf, inf], default 0.0

**error\_threshold**

Low values are slower and higher quality

**Type** float in [0.0001, 10], default 0.0

**falloff\_strength**

Attenuation falloff strength, the higher, the less influence distant objects have

**Type** float in [-inf, inf], default 0.0

**gather\_method**

- RAYTRACE Raytrace, Accurate, but slow when noise-free results are required.
- APPROXIMATE Approximate, Inaccurate, but faster and without noise.

**Type** enum in ['RAYTRACE', 'APPROXIMATE'], default 'RAYTRACE'

**indirect\_bounces**

Number of indirect diffuse light bounces

**Type** int in [1, 32767], default 0

**indirect\_factor**

Factor for how much surrounding objects contribute to light

**Type** float in [0, inf], default 0.0

**passes**

Number of preprocessing passes to reduce overocclusion

**Type** int in [0, 10], default 0

**sample\_method**

Method for generating shadow samples (for Raytrace)

- **CONSTANT\_JITTERED** Constant Jittered, Fastest and gives the most noise.
- **ADAPTIVE\_QMC** Adaptive QMC, Fast in high-contrast areas.
- **CONSTANT\_QMC** Constant QMC, Best quality.

**Type** enum in ['CONSTANT\_JITTERED', 'ADAPTIVE\_QMC', 'CONSTANT\_QMC'], default 'CONSTANT\_JITTERED'

**samples**

Amount of ray samples. Higher values give smoother results and longer rendering times

**Type** int in [1, 128], default 0

**threshold**

Samples below this threshold will be considered fully shadowed/unshadowed and skipped (for Raytrace Adaptive QMC)

**Type** float in [0, 1], default 0.0

**use\_ambient\_occlusion**

Use Ambient Occlusion to add shadowing based on distance between objects

**Type** boolean, default False

**use\_cache**

Cache AO results in pixels and interpolate over neighbouring pixels for speedup

**Type** boolean, default False

**use\_environment\_light**

Add light coming from the environment

**Type** boolean, default False

**use\_falloff**

Distance will be used to attenuate shadows

**Type** boolean, default False

**use\_indirect\_light**

Add indirect light bouncing of surrounding objects

**Type** boolean, default False

## Inherited Properties

- `bpy_struct.id_data`

## Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

## References

- `World.light_settings`

## 2.4.649 WorldMistSettings(`bpy_struct`)

base class — `bpy_struct`

**class** `bpy.types.WorldMistSettings` (*`bpy_struct`*)  
Mist settings for a World data-block

### **depth**

Distance over which the mist effect fades in

**Type** float in [0, inf], default 0.0

### **falloff**

Type of transition used to fade mist

- `QUADRATIC` Quadratic, Mist uses quadratic progression.
- `LINEAR` Linear, Mist uses linear progression.
- `INVERSE_QUADRATIC` Inverse Quadratic, Mist uses inverse quadratic progression.

**Type** enum in ['QUADRATIC', 'LINEAR', 'INVERSE\_QUADRATIC'], default 'QUADRATIC'

### **height**

Control how much mist density decreases with height

**Type** float in [0, 100], default 0.0

**intensity**

Overall minimum intensity of the mist effect

**Type** float in [0, 1], default 0.0

**start**

Starting distance of the mist, measured from the camera

**Type** float in [0, inf], default 0.0

**use\_mist**

Occlude objects with the environment color as they are further away

**Type** boolean, default False

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `World.mist_settings`

**2.4.650 WorldStarsSettings(`bpy_struct`)**

base class — `bpy_struct`

**class** `bpy.types.WorldStarsSettings` (*`bpy_struct`*)

Stars settings for a World data-block

**average\_separation**

Average distance between any two stars

**Type** float in [2, 1000], default 0.0

**color\_random**

Randomize star colors

**Type** float in [0, 1], default 0.0

**distance\_min**

Minimum distance to the camera for stars

**Type** float in [0, 1000], default 0.0

**size**

Average screen dimension of stars

**Type** float in [0, 10], default 0.0

**use\_stars**

Enable starfield generation

**Type** boolean, default False

**Inherited Properties**

- `bpy_struct.id_data`

**Inherited Functions**

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

**References**

- `World.star_settings`

**2.4.651 WorldTextureSlot(TextureSlot)**

base classes — `bpy_struct`, `TextureSlot`

**class** `bpy.types.WorldTextureSlot` (*TextureSlot*)  
Texture slot for textures in a World datablock

**blend\_factor**

Amount texture affects color progression of the background

**Type** float in [-inf, inf], default 0.0

**horizon\_factor**

Amount texture affects color of the horizon

**Type** float in [-inf, inf], default 0.0

**object**

Object to use for mapping with Object texture coordinates

**Type** `Object`

**texture\_coords**

Texture coordinates used to map the texture onto the background

- VIEW View, Uses view vector for the texture coordinates.
- GLOBAL Global, Uses global coordinates for the texture coordinates (interior mist).
- ANGMAP AngMap, Uses 360 degree angular coordinates, e.g. for spherical light probes.
- SPHERE Sphere, For 360 degree panorama sky, spherical mapped, only top half.
- TUBE Tube, For 360 degree panorama sky, cylindrical mapped, only top half.
- OBJECT Object, Uses linked object's coordinates for texture coordinates.

**Type** enum in ['VIEW', 'GLOBAL', 'ANGMAP', 'SPHERE', 'TUBE', 'OBJECT'], default 'VIEW'

**use\_map\_blend**

Affect the color progression of the background

**Type** boolean, default False

**use\_map\_horizon**

Affect the color of the horizon

**Type** boolean, default False

**use\_map\_zenith\_down**

Affect the color of the zenith below

**Type** boolean, default False

**use\_map\_zenith\_up**

Affect the color of the zenith above

**Type** boolean, default False

**zenith\_down\_factor**

Amount texture affects color of the zenith below

**Type** float in [-inf, inf], default 0.0

**zenith\_up\_factor**

Amount texture affects color of the zenith above

**Type** float in [-inf, inf], default 0.0

### Inherited Properties

- `bpy_struct.id_data`
- `TextureSlot.name`
- `TextureSlot.blend_type`
- `TextureSlot.color`
- `TextureSlot.default_value`
- `TextureSlot.invert`
- `TextureSlot.offset`
- `TextureSlot.output_node`
- `TextureSlot.use_rgb_to_intensity`
- `TextureSlot.scale`
- `TextureSlot.use_stencil`
- `TextureSlot.texture`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `World.texture_slots`
- `WorldTextureSlots.add`
- `WorldTextureSlots.create`

## 2.4.652 WorldTextureSlots(bpy\_struct)

base class — `bpy_struct`

**class** `bpy.types.WorldTextureSlots` (*bpy\_struct*)

Collection of texture slots

**classmethod** `add()`

add

**Returns** The newly initialized `mtx`

**Return type** `WorldTextureSlot`



**classmethod** `create` (*index*)

`create`

**Parameters** `index` (*int in [0, inf]*) – Index, Slot index to initialize

**Returns** The newly initialized `mtx`

**Return type** `WorldTextureSlot`

**classmethod** `clear` (*index*)

`clear`

**Parameters** `index` (*int in [0, inf]*) – Index, Slot index to clear

### Inherited Properties

- `bpy_struct.id_data`

### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`

### References

- `World.texture_slots`

## 2.4.653 XnorController(Controller)

base classes — `bpy_struct`, `Controller`

**class** `bpy.types.XnorController` (*Controller*)

Controller passing on events based on a logical XNOR operation

### Inherited Properties

- `bpy_struct.id_data`
- `Controller.name`

- `Controller.states`
- `Controller.show_expanded`
- `Controller.use_priority`
- `Controller.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Controller.link`
- `Controller.unlink`

### 2.4.654 XorController(Controller)

base classes — `bpy_struct`, `Controller`

**class** `bpy.types.XorController` (*Controller*)  
Controller passing on events based on a logical XOR operation

#### Inherited Properties

- `bpy_struct.id_data`
- `Controller.name`
- `Controller.states`
- `Controller.show_expanded`
- `Controller.use_priority`
- `Controller.type`

#### Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.callback_add`
- `bpy_struct.callback_remove`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.is_property_hidden`

- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.type_recast`
- `bpy_struct.values`
- `Controller.link`
- `Controller.unlink`

## 2.4.655 bpy\_prop\_collection

**class** `bpy.types.bpy_prop_collection`  
built-in class used for all collections.

---

**Note:** Note that `bpy.types.bpy_prop_collection` is not actually available from within blender, it only exists for the purpose of documentation.

---

**foreach\_get** (*attr, seq*)

This is a function to give fast access to attributes within a collection.

```
collection.foreach_get(someseq, attr)
```

```
# Python equivalent
```

```
for i in range(len(seq)): someseq[i]= getattr(collection, attr)
```

**foreach\_set** (*attr, seq*)

This is a function to give fast access to attributes within a collection.

```
collection.foreach_set(seq, attr)
```

```
# Python equivalent
```

```
for i in range(len(seq)): setattr(collection[i], attr, seq[i])
```

**get** (*key, default=None*)

Returns the value of the item assigned to *key* or *default* when not found (matches python's dictionary function of the same name).

### Parameters

- **key** (*string*) – The identifier for the collection member.
- **default** (*Undefined*) – Optional argument for the value to return if *key* is not found.

**items** ()

Return the identifiers of collection members (matching python's `dict.items()` functionality).

**Returns** (*key, value*) pairs for each member of this collection.

**Return type** list of tuples

**keys** ()

Return the identifiers of collection members (matching python's `dict.keys()` functionality).

**Returns** the identifiers for each member of this collection.

**Return type** list of strings

**values ()**

Return the values of collection (matching python's dict.values() functionality).

**Returns** the members of this collection.

**Return type** list

**2.4.656 bpy\_struct**

subclasses — ActionFCurves, ActionGroup, ActionGroups, ActionPoseMarkers, Actuator, Addon, Addons, AnimData, AnimDataDrivers, AnimViz, AnimVizMotionPaths, AnimVizOnionSkinning, AnyType, Area, AreaSpaces, ArmatureBones, ArmatureEditBones, BackgroundImage, BezierSplinePoint, BlendData, BlendDataActions, BlendDataArmatures, BlendDataBrushes, BlendDataCameras, BlendDataCurves, BlendDataFonts, BlendDataGreasePencils, BlendDataGroups, BlendDataImages, BlendDataLamps, BlendDataLattices, BlendDataLibraries, BlendDataMaterials, BlendDataMeshes, BlendDataMetaBalls, BlendDataNodeTrees, BlendDataObjects, BlendDataParticles, BlendDataScenes, BlendDataScreens, BlendDataSounds, BlendDataSpeakers, BlendDataTexts, BlendDataTextures, BlendDataWindowManagers, BlendDataWorlds, BlenderRNA, BoidRule, BoidSettings, BoidState, Bone, BoneGroup, BoneGroups, ChannelDriverVariables, ChildParticle, ClothCollisionSettings, ClothSettings, CollisionSettings, ColorRamp, ColorRampElement, ColorRampElements, CompositorNodes, ConsoleLine, Constraint, ConstraintTarget, Context, Controller, CurveMap, CurveMapPoint, CurveMapping, CurveSplines, DopeSheet, Driver, DriverTarget, DriverVariable, DupliObject, EditBone, EffectorWeights, EnumPropertyItem, EnvironmentMap, Event, FCurve, FCurveKeyframePoints, FCurveModifiers, FCurveSample, FModifier, FModifierEnvelopeControlPoint, FieldSettings, FileSelectParams, FluidMeshVertex, FluidSettings, Function, GPencilFrame, GPencilLayer, GPencilStroke, GPencilStrokePoint, GameObjectSettings, GameProperty, GameSoftBodySettings, GreasePencilLayers, GroupInputs, GroupObjects, GroupOutputs, Header, Histogram, ID, IDMaterials, IKParam, ImageUser, KeyConfig, KeyConfigurations, KeyMap, KeyMapItem, KeyMapItems, KeyMaps, Keyframe, KeyingSet, KeyingSetInfo, KeyingSetPath, KeyingSetPaths, KeyingSets, KeyingSetsAll, LampSkySettings, LampTextureSlots, LatticePoint, Macro, MaterialGameSettings, MaterialHalo, MaterialPhysics, MaterialRaytraceMirror, MaterialRaytraceTransparency, MaterialSlot, MaterialStrand, MaterialSubsurfaceScattering, MaterialTextureSlots, MaterialVolume, Menu, MeshColor, MeshColorLayer, MeshEdge, MeshEdges, MeshFace, MeshFaces, MeshFloatProperty, MeshFloatPropertyLayer, MeshIntProperty, MeshIntPropertyLayer, MeshSticky, MeshStringProperty, MeshStringPropertyLayer, MeshTextureFace, MeshTextureFaceLayer, MeshVertex, MeshVertices, MetaBallElements, MetaElement, Modifier, MotionPath, MotionPathVert, NlaStrip, NlaStrips, NlaTrack, NlaTracks, Node, NodeLink, NodeLinks, NodeSocket, ObjectBase, ObjectConstraints, ObjectModifiers, Operator, OperatorMacro, OperatorProperties, PackedFile, Paint, Panel, Particle, ParticleBrush, ParticleDupliWeight, ParticleEdit, ParticleHairKey, ParticleKey, ParticleSettingsTextureSlots, ParticleSystem, ParticleSystems, ParticleTarget, PointCache, PointCaches, PointDensity, Pose, PoseBone, PoseBoneConstraints, Property, PropertyGroup, PropertyGroupItem, Region, RegionView3D, RenderEngine, RenderLayer, RenderLayers, RenderPass, RenderResult, RenderSettings, SPHFluidSettings, SceneBases, SceneGameData, SceneGameRecastData, SceneObjects, SceneRenderLayer, Scopes, Sensor, Sequence, SequenceColorBalance, SequenceCrop, SequenceEditor, SequenceElement, SequenceProxy, SequenceTransform, ShaderNodes, ShapeKey, ShapeKeyBezierPoint, ShapeKeyCurvePoint, ShapeKeyPoint, SmokeCollSettings, SmokeDomainSettings, SmokeFlowSettings, SoftBodySettings, Space, SpaceUVEditor, Spline, SplineBezierPoints, SplinePoint,

SplinePoints, Struct, TexMapping, TextBox, TextCharacterFormat, TextLine, TextMarker, TextureNodes, TextureSlot, Theme, ThemeBoneColorSet, ThemeConsole, ThemeDopeSheet, ThemeFileBrowser, ThemeFontStyle, ThemeGraphEditor, ThemeImageEditor, ThemeInfo, ThemeLogicEditor, ThemeNLAEEditor, ThemeNodeEditor, ThemeOutliner, ThemeProperties, ThemeSequenceEditor, ThemeStyle, ThemeTextEditor, ThemeTimeline, ThemeUserInterface, ThemeUserPreferences, ThemeView3D, ThemeWidgetColors, ThemeWidgetStateColors, TimelineMarker, TimelineMarkers, Timer, ToolSettings, TransformOrientation, UILayout, UVProjector, UVTextures, UnitSettings, UnknownType, UserPreferences, UserPreferencesEdit, UserPreferencesFilePaths, UserPreferencesInput, UserPreferencesSystem, UserPreferencesView, UserSolidLight, VertexColors, VertexGroup, VertexGroupElement, VertexGroups, VoxelData, Window, WorldLighting, WorldMistSettings, WorldStarsSettings, WorldTextureSlots

**class** `bpy.types.bpy_struct`  
built-in base class for all classes in `bpy.types`.

---

**Note:** Note that `bpy.types.bpy_struct` is not actually available from within blender, it only exists for the purpose of documentation.

---

**as\_pointer** ()  
Returns the memory address which holds a pointer to blenders internal data

**Returns** int (memory address).

**Return type** int

---

**Note:** This is intended only for advanced script writers who need to pass blender data to their own C/Python modules.

---

Undocumented ([contribute](#))

Undocumented ([contribute](#))

**driver\_add** (*path*, *index=-1*)  
Adds driver(s) to the given property

**Parameters**

- **path** (*string*) – path to the property to drive, analogous to the fcurve’s data path.
- **index** (*int*) – array index of the property drive. Defaults to -1 for all indices or a single channel if the property is not an array.

**Returns** The driver(s) added.

**Return type** `bpy.types.FCurve` or list if index is -1 with an array property.

**driver\_remove** (*path*, *index=-1*)  
Remove driver(s) from the given property

**Parameters**

- **path** (*string*) – path to the property to drive, analogous to the fcurve’s data path.
- **index** (*int*) – array index of the property drive. Defaults to -1 for all indices or a single channel if the property is not an array.

**Returns** Success of driver removal.

**Return type** boolean

**get** (*key*, *default=None*)

Returns the value of the custom property assigned to *key* or *default* when not found (matches python's dictionary function of the same name).

**Parameters**

- **key** (*string*) – The key associated with the custom property.
  - **default** (*Undefined*) – Optional argument for the value to return if *key* is not found.
- 

**Note:** Only `bpy.types.ID`, `bpy.types.Bone` and `bpy.types.PoseBone` classes support custom properties.

---

**is\_property\_hidden** (*property*)

Check if a property is hidden.

**Returns** True when the property is hidden.

**Return type** boolean

**is\_property\_set** (*property*)

Check if a property is set, use for testing operator properties.

**Returns** True when the property has been set.

**Return type** boolean

**items** ()

Returns the items of this object's custom properties (matches python's dictionary function of the same name).

**Returns** custom property key, value pairs.

**Return type** list of key, value tuples

---

**Note:** Only `bpy.types.ID`, `bpy.types.Bone` and `bpy.types.PoseBone` classes support custom properties.

---

**keyframe\_delete** (*data\_path*, *index=-1*, *frame=bpy.context.scene.frame\_current*, *group=""*)

Remove a keyframe from this property's fcurve.

**Parameters**

- **data\_path** (*string*) – path to the property to remove a key, analogous to the fcurve's data path.
- **index** (*int*) – array index of the property to remove a key. Defaults to -1 removing all indices or a single channel if the property is not an array.
- **frame** (*float*) – The frame on which the keyframe is deleted, defaulting to the current frame.
- **group** (*str*) – The name of the group the F-Curve should be added to if it doesn't exist yet.

**Returns** Success of keyframe deletion.

**Return type** boolean

**keyframe\_insert** (*data\_path*, *index=-1*, *frame=bpy.context.scene.frame\_current*, *group=""*)

Insert a keyframe on the property given, adding fcurves and animation data when necessary.

**Parameters**

- **data\_path** (*string*) – path to the property to key, analogous to the fcurve’s data path.
- **index** (*int*) – array index of the property to key. Defaults to -1 which will key all indices or a single channel if the property is not an array.
- **frame** (*float*) – The frame on which the keyframe is inserted, defaulting to the current frame.
- **group** (*str*) – The name of the group the F-Curve should be added to if it doesn’t exist yet.

**Returns** Success of keyframe insertion.

**Return type** boolean

This is the most simple example of inserting a keyframe from python.

```
import bpy

obj = bpy.context.object

# set the keyframe at frame 1
obj.location = 3.0, 4.0, 10.0
obj.keyframe_insert(data_path="location", frame=1)
```

Note that when keying data paths which contain nested properties this must be done from the `ID` subclass, in this case the `Armature` rather than the bone.

```
import bpy
from bpy.props import PointerProperty

# define a nested property
class MyPropGroup(bpy.types.PropertyGroup):
    nested = bpy.props.FloatProperty(name="Nested", default=0.0)

# register it so its available for all bones
bpy.utils.register_class(MyPropGroup)
bpy.types.Bone.my_prop = PointerProperty(type=MyPropGroup,
                                         name="MyProp")

# get a bone
obj = bpy.data.objects["Armature"]
arm = obj.data

# set the keyframe at frame 1
arm.bones["Bone"].my_prop_group.nested = 10
arm.keyframe_insert(data_path='bones["Bone"].my_prop.nested',
                    frame=1,
                    group="Nested Group")
```

**keys()**

Returns the keys of this objects custom properties (matches python’s dictionary function of the same name).

**Returns** custom property keys.

**Return type** list of strings

---

**Note:** Only `bpy.types.ID`, `bpy.types.Bone` and `bpy.types.PoseBone` classes support custom properties.

---

**path\_from\_id** (*property=""*)

Returns the data path from the ID to this object (string).

**Parameters** **property** (*string*) – Optional property name which can be used if the path is to a property of this object.

**Returns** The path from `bpy.types.bpy_struct.id_data` to this struct and property (when given).

**Return type** str

**path\_resolve** (*path, coerce=True*)

Returns the property from the path, raise an exception when not found.

**Parameters**

- **path** (*string*) – path which this property resolves.
- **coerce** (*boolean*) – optional argument, when True, the property will be converted into its python representation.

**type\_recast** ()

Return a new instance, this is needed because types such as textures can be changed at runtime.

**Returns** a new instance of this object with the type initialized again.

**Return type** subclass of `bpy.types.bpy_struct`

**values** ()

Returns the values of this objects custom properties (matches python's dictionary function of the same name).

**Returns** custom property values.

**Return type** list

---

**Note:** Only `bpy.types.ID`, `bpy.types.Bone` and `bpy.types.PoseBone` classes support custom properties.

---

**id\_data**

The `bpy.types.ID` object this datablock is from or None, (not available for all data types)

## 2.5 Utilities (bpy.utils)

This module contains utility functions specific to blender but not associated with blenders internal data.

`bpy.utils.blend_paths` (*absolute=False, packed=False, local=False*)

Returns a list of paths to external files referenced by the loaded .blend file.

**Parameters**

- **absolute** (*boolean*) – When true the paths returned are made absolute.
- **packed** (*boolean*) – When true skip file paths for packed data.
- **local** (*boolean*) – When true skip linked library paths.

**Returns** path list.

**Return type** list of strings



`bpy.utils.register_class` (*cls*)

Register a subclass of a blender type in (`bpy.types.Panel`, `bpy.types.Menu`, `bpy.types.Header`, `bpy.types.Operator`, `bpy.types.KeyingSetInfo`, `bpy.types.RenderEngine`).

If the class has a `register` class method it will be called before registration.

---

**Note:** `ValueError` exception is raised if the class is not a subclass of a registerable blender class.

---

`bpy.utils.resource_path` (*type*, *major=2*, *minor=57*)

Return the base path for storing system files.

**Parameters**

- **type** (*string*) – string in ['USER', 'LOCAL', 'SYSTEM'].
- **major** (*int*) – major version, defaults to current.
- **minor** (*string*) – minor version, defaults to current.

**Returns** the resource path (not necessarily existing).

**Return type** string

`bpy.utils.unregister_class` (*cls*)

Unload the python class from blender.

If the class has an `unregister` class method it will be called before unregistering.

`bpy.utils.keyconfig_set` (*filepath*)

`bpy.utils.load_scripts` (*reload\_scripts=False*, *refresh\_scripts=False*)

Load scripts and run each modules register function.

**Parameters**

- **reload\_scripts** (*bool*) – Causes all scripts to have their unregister method called before loading.
- **refresh\_scripts** (*bool*) – only load scripts which are not already loaded as modules.

`bpy.utils.modules_from_path` (*path*, *loaded\_modules*)

Load all modules in a path and return them as a list.

**Parameters**

- **path** (*string*) – this path is scanned for scripts and packages.
- **loaded\_modules** (*set*) – already loaded module names, files matching these names will be ignored.

**Returns** all loaded modules.

**Return type** list

`bpy.utils.preset_find` (*name*, *preset\_path*, *display\_name=False*)

`bpy.utils.preset_paths` (*subdir*)

Returns a list of paths for a specific preset.

**Parameters** **subdir** (*string*) – preset subdirectory (must not be an absolute path).

**Returns** script paths.

**Return type** list

`bpy.utils.refresh_script_paths()`  
Run this after creating new script paths to update `sys.path`

`bpy.utils.register_module(module, verbose=False)`

`bpy.utils.script_paths(subdir=None, user_pref=True, all=False)`  
Returns a list of valid script paths.

#### Parameters

- **subdir** (*string*) – Optional subdir.
- **user\_pref** (*bool*) – Include the user preference script path.
- **all** (*bool*) – Include local, user and system paths rather just the paths blender uses.

**Returns** script paths.

**Return type** list

`bpy.utils.smpte_from_frame(frame, fps=None, fps_base=None)`  
Returns an SMPTE formatted string from the frame: “HH:MM:SS:FF”.

If *fps* and *fps\_base* are not given the current scene is used.

`bpy.utils.smpte_from_seconds(time, fps=None)`  
Returns an SMPTE formatted string from the time in seconds: “HH:MM:SS:FF”.

If the *fps* is not given the current scene is used.

`bpy.utils.unregister_module(module, verbose=False)`

`bpy.utils.user_resource(type, path='', create=False)`  
Return a user resource path (normally from the users home directory).

#### Parameters

- **type** (*string*) – Resource type in ['DATAFILES', 'CONFIG', 'SCRIPTS', 'AUTOSAVE'].
- **subdir** (*string*) – Optional subdirectory.
- **create** (*boolean*) – Treat the path as a directory and create it if its not existing.

**Returns** a path.

**Return type** string

`bpy.utils.user_script_path()`

## 2.6 Path Utilities (bpy.path)

This module has a similar scope to `os.path`, containing utility functions for dealing with paths in Blender.

`bpy.path.abspath(path, start=None, library=None)`  
Returns the absolute path relative to the current blend file using the “//” prefix.

#### Parameters

- **start** (*string*) – Relative to this path, when not set the current filename is used.
- **library** (`bpy.types.Library`) – The library this path is from. This is only included for convenience, when the library is not None its path replaces *start*.

`bpy.path.basename` (*path*)

Equivalent to `os.path.basename`, but skips a “//” prefix.

Use for Windows compatibility.

`bpy.path.clean_name` (*name*, *replace*='\_')

**Returns a name with characters replaced that** may cause problems under various circumstances,

such as writing to a file. All characters besides A-Z/a-z, 0-9 are replaced with “\_” or the *replace* argument if defined.

`bpy.path.display_name` (*name*)

Creates a display string from *name* to be used menus and the user interface. Capitalize the first letter in all lowercase names, mixed case names are kept as is. Intended for use with filenames and module names.

`bpy.path.display_name_from_filepath` (*name*)

Returns the path stripped of directory and extension, ensured to be utf8 compatible.

`bpy.path.ensure_ext` (*filepath*, *ext*, *case\_sensitive*=*False*)

Return the path with the extension added if it is not already set.

#### Parameters

- **ext** (*string*) – The extension to check for.
- **case\_sensitive** (*bool*) – Check for matching case when comparing extensions.

`bpy.path.is_subdir` (*path*, *directory*)

Returns true if *path* in a subdirectory of *directory*. Both paths must be absolute.

`bpy.path.module_names` (*path*, *recursive*=*False*)

Return a list of modules which can be imported from *path*.

#### Parameters

- **path** (*string*) – a directory to scan.
- **recursive** (*bool*) – Also return submodule names for packages.

**Returns** a list of string pairs (*module\_name*, *module\_file*).

**Return type** list

`bpy.path.relpath` (*path*, *start*=*None*)

Returns the path relative to the current blend file using the “//” prefix.

**Parameters** **start** (*string*) – Relative to this path, when not set the current filename is used.

`bpy.path.resolve_ncase` (*path*)

Resolve a case insensitive path on a case sensitive system, returning a string with the path if found else return the original path.

## 2.7 Application Data (bpy.app)

This module contains application values that remain unchanged during runtime.

`bpy.app.debug`

Boolean, set when blender is running in debug mode (started with -d)

`bpy.app.debug_value`

Int, number which can be set to non-zero values for testing purposes.

- `bpy.app.driver_namespace`  
Dictionary for drivers namespace, editable in-place, reset on file load (read-only)
- `bpy.app.tempdir`  
String, the temp directory used by blender (read-only)
- `bpy.app.background`  
Boolean, True when blender is running without a user interface (started with -b)
- `bpy.app.handlers`  
Application handler callbacks  
*[bpy.app.handlers submodule details](#)*
- `bpy.app.binary_path`  
The location of blenders executable, useful for utilities that spawn new instances
- `bpy.app.build_cflags`  
C compiler flags
- `bpy.app.build_cxxflags`  
C++ compiler flags
- `bpy.app.build_date`  
The date this blender instance was built
- `bpy.app.build_linkflags`  
Binary linking flags
- `bpy.app.build_platform`  
The platform this blender instance was built for
- `bpy.app.build_revision`  
The subversion revision this blender instance was built with
- `bpy.app.build_system`  
Build system used
- `bpy.app.build_time`  
The time this blender instance was built
- `bpy.app.build_type`  
The type of build (Release, Debug)
- `bpy.app.version_char`  
The Blender version character (for minor releases)
- `bpy.app.version_cycle`  
The release status of this build alpha/beta/rc/release
- `bpy.app.version_string`  
The Blender version formatted as a string
- `bpy.app.version`  
The Blender version as a tuple of 3 numbers. eg. (2, 50, 11)

## 2.8 Application Handlers (`bpy.app.handlers`)

This module contains callbacks

## 2.8.1 Basic Handler Example

This script shows the most simple example of adding a handler.

```
import bpy

def my_handler(scene):
    print("Frame Change", scene.frame_current)

bpy.app.handlers.frame_change_pre.append(my_handler)
```

## 2.8.2 Persistent Handler Example

By default handlers are freed when loading new files, in some cases you may want the handler stay running across multiple files (when the handler is part of an addon for example).

For this the `bpy.app.handlers.persistent` decorator needs to be used.

```
import bpy
from bpy.app.handlers import persistent

@persistent
def load_handler(dummy):
    print("Load Handler:", bpy.data.filepath)

bpy.app.handlers.load_post.append(load_handler)

bpy.app.handlers.frame_change_post
    Callback list - on frame change for playback and rendering (after)

bpy.app.handlers.frame_change_pre
    Callback list - on frame change for playback and rendering (before)

bpy.app.handlers.load_post
    Callback list - on loading a new blend file (after)

bpy.app.handlers.load_pre
    Callback list - on loading a new blend file (before)

bpy.app.handlers.render_post
    Callback list - on render (after)

bpy.app.handlers.render_pre
    Callback list - on render (before)

bpy.app.handlers.render_stats
    Callback list - on printing render statistics

bpy.app.handlers.save_post
    Callback list - on saving a blend file (after)

bpy.app.handlers.save_pre
    Callback list - on saving a blend file (before)

bpy.app.handlers.scene_update_post
    Callback list - on updating the scenes data (after)

bpy.app.handlers.scene_update_pre
    Callback list - on updating the scenes data (before)
```

`bpy.app.handlers.persistent`

Function decorator for callback functions not to be removed when loading new files

## 2.9 Property Definitions (`bpy.props`)

This module defines properties to extend blenders internal data, the result of these functions is used to assign properties to classes registered with blender and can't be used directly.

### 2.9.1 Assigning to Existing Classes

Custom properties can be added to any subclass of an ID, Bone and PoseBone.

These properties can be animated, accessed by the user interface and python like blenders existing properties.

```
import bpy

# Assign a custom property to an existing type.
bpy.types.Material.custom_float = bpy.props.FloatProperty(name="Test Prob")

# Test the property is there.
bpy.data.materials[0].custom_float = 5.0
```

### 2.9.2 Operator Example

A common use of custom properties is for python based Operator classes.

```
import bpy

class DialogOperator(bpy.types.Operator):
    bl_idname = "object.dialog_operator"
    bl_label = "Property Example"

    my_float = bpy.props.FloatProperty(name="Some Floating Point")
    my_bool = bpy.props.BoolProperty(name="Toggle Option")
    my_string = bpy.props.StringProperty(name="String Value")

    def execute(self, context):
        print("Dialog Runs")
        return {'FINISHED'}

    def invoke(self, context, event):
        wm = context.window_manager
        return wm.invoke_props_dialog(self)

bpy.utils.register_class(DialogOperator)

# test call
bpy.ops.object.dialog_operator('INVOKE_DEFAULT')
```

### 2.9.3 PropertyGroup Example

PropertyGroups can be used for collecting custom settings into one value to avoid many individual settings mixed in together.

```
import bpy

class MaterialSettings(bpy.types.PropertyGroup):
    my_int = bpy.props.IntProperty()
    my_float = bpy.props.FloatProperty()
    my_string = bpy.props.StringProperty()

bpy.utils.register_class(MaterialSettings)

bpy.types.Material.my_settings = \
    bpy.props.PointerProperty(type=MaterialSettings)

# test the new settings work
material = bpy.data.materials[0]

material.my_settings.my_int = 5
material.my_settings.my_float = 3.0
material.my_settings.my_string = "Foo"
```

### 2.9.4 Collection Example

Custom properties can be added to any subclass of an ID, Bone and PoseBone.

```
import bpy

# Assign a collection
class SceneSettingItem(bpy.types.PropertyGroup):
    name = bpy.props.StringProperty(name="Test Prop", default="Unknown")
    value = bpy.props.IntProperty(name="Test Prop", default=22)

bpy.utils.register_class(SceneSettingItem)

bpy.types.Scene.my_settings = \
    bpy.props.CollectionProperty(type=SceneSettingItem)

# Assume an armature object selected
print("Adding 3 values!")

my_item = bpy.context.scene.my_settings.add()
my_item.name = "Spam"
my_item.value = 1000

my_item = bpy.context.scene.my_settings.add()
my_item.name = "Eggs"
my_item.value = 30

for my_item in bpy.context.scene.my_settings:
    print(my_item.name, my_item.value)
```

## 2.9.5 Update Example

It can be useful to perform an action when a property is changed and can be used to update other properties or synchronize with external data.

All properties define update functions except for `CollectionProperty`.

```
import bpy
```

```
def update_func(self, context):
    print("my test function", self)
```

```
bpy.types.Scene.testprop = bpy.props.FloatProperty(update=update_func)
```

```
bpy.context.scene.testprop = 11.0
```

```
# >>> my test function <bpy_struct, Scene("Scene")>
```

```
bpy.props.BoolProperty(name="", description="", default=False, options={'ANIMATABLE'}, subtype='NONE', update=None)
```

Returns a new boolean property definition.

### Parameters

- **name** (*string*) – Name used in the user interface.
- **description** (*string*) – Text used for the tooltip and api documentation.
- **options** (*set*) – Enumerator in ['HIDDEN', 'SKIP\_SAVE', 'ANIMATABLE'].
- **subtype** (*string*) – Enumerator in ['UNSIGNED', 'PERCENTAGE', 'FACTOR', 'ANGLE', 'TIME', 'DISTANCE', 'NONE'].
- **update** (*function*) – function to be called when this value is modified, This function must take 2 values (self, context) and return None.

```
bpy.props.BoolVectorProperty(name="", description="", default=(False, False, False), options={'ANIMATABLE'}, subtype='NONE', size=3, update=None)
```

Returns a new vector boolean property definition.

### Parameters

- **name** (*string*) – Name used in the user interface.
- **description** (*string*) – Text used for the tooltip and api documentation.
- **default** (*sequence*) – sequence of booleans the length of *size*.
- **options** (*set*) – Enumerator in ['HIDDEN', 'SKIP\_SAVE', 'ANIMATABLE'].
- **subtype** (*string*) – Enumerator in ['COLOR', 'TRANSLATION', 'DIRECTION', 'VELOCITY', 'ACCELERATION', 'MATRIX', 'EULER', 'QUATERNION', 'AXISANGLE', 'XYZ', 'COLOR\_GAMMA', 'LAYER', 'NONE'].
- **size** (*int*) – Vector dimensions in [1, and 32].
- **update** (*function*) – function to be called when this value is modified, This function must take 2 values (self, context) and return None.

```
bpy.props.CollectionProperty(items, type="", description="", default="", options={'ANIMATABLE'})
```

Returns a new collection property definition.



**Parameters**

- **type** (*class*) – A subclass of `bpy.types.PropertyGroup`.
- **name** (*string*) – Name used in the user interface.
- **description** (*string*) – Text used for the tooltip and api documentation.
- **options** (*set*) – Enumerator in ['HIDDEN', 'SKIP\_SAVE', 'ANIMATABLE'].

`bpy.props.EnumProperty` (*items*, *name*="", *description*="", *default*="", *options*={'ANIMATABLE'}, *update*=None)

Returns a new enumerator property definition.

**Parameters**

- **name** (*string*) – Name used in the user interface.
- **description** (*string*) – Text used for the tooltip and api documentation.
- **default** (*string or set*) – The default value for this enum, A string when `ENUM_FLAG` is disabled otherwise a set which may only contain string identifiers used in *items*.
- **options** (*set*) – Enumerator in ['HIDDEN', 'SKIP\_SAVE', 'ANIMATABLE', 'ENUM\_FLAG'].
- **items** (*sequence of string triplets or a function*) – sequence of enum items formatted: [(identifier, name, description, number), ...] where the identifier is used for python access and other values are used for the interface. Note the item is optional. For dynamic values a callback can be passed which returns a list in the same format as the static list. This function must take 2 arguments (self, context)
- **update** (*function*) – function to be called when this value is modified, This function must take 2 values (self, context) and return None.

`bpy.props.FloatProperty` (*name*="", *description*="", *default*=0.0, *min*=`sys.float_info.min`, *max*=`sys.float_info.max`, *soft\_min*=`sys.float_info.min`, *soft\_max*=`sys.float_info.max`, *step*=3, *precision*=2, *options*={'ANIMATABLE'}, *subtype*='NONE', *unit*='NONE', *update*=None)

Returns a new float property definition.

**Parameters**

- **name** (*string*) – Name used in the user interface.
- **description** (*string*) – Text used for the tooltip and api documentation.
- **options** (*set*) – Enumerator in ['HIDDEN', 'SKIP\_SAVE', 'ANIMATABLE'].
- **subtype** (*string*) – Enumerator in ['UNSIGNED', 'PERCENTAGE', 'FACTOR', 'ANGLE', 'TIME', 'DISTANCE', 'NONE'].
- **unit** (*string*) – Enumerator in ['NONE', 'LENGTH', 'AREA', 'VOLUME', 'ROTATION', 'TIME', 'VELOCITY', 'ACCELERATION'].
- **update** (*function*) – function to be called when this value is modified, This function must take 2 values (self, context) and return None.

`bpy.props.FloatVectorProperty` (*name*="", *description*="", *default*=(0.0, 0.0, 0.0), *min*=`sys.float_info.min`, *max*=`sys.float_info.max`, *soft\_min*=`sys.float_info.min`, *soft\_max*=`sys.float_info.max`, *step*=3, *precision*=2, *options*={'ANIMATABLE'}, *subtype*='NONE', *size*=3, *update*=None)

Returns a new vector float property definition.

**Parameters**

- **name** (*string*) – Name used in the user interface.
- **description** (*string*) – Text used for the tooltip and api documentation.
- **default** (*sequence*) – sequence of floats the length of *size*.
- **options** (*set*) – Enumerator in ['HIDDEN', 'SKIP\_SAVE', 'ANIMATABLE'].
- **subtype** (*string*) – Enumerator in ['COLOR', 'TRANSLATION', 'DIRECTION', 'VELOCITY', 'ACCELERATION', 'MATRIX', 'EULER', 'QUATERNION', 'AXISANGLE', 'XYZ', 'COLOR\_GAMMA', 'LAYER', 'NONE'].
- **unit** (*string*) – Enumerator in ['NONE', 'LENGTH', 'AREA', 'VOLUME', 'ROTATION', 'TIME', 'VELOCITY', 'ACCELERATION'].
- **size** (*int*) – Vector dimensions in [1, and 32].
- **update** (*function*) – function to be called when this value is modified, This function must take 2 values (self, context) and return None.

```
bpy.props.IntProperty(name="", description="", default=0, min=-sys.maxint, max=sys.maxint,
                    soft_min=-sys.maxint, soft_max=sys.maxint, step=1, options={'ANIMATABLE'},
                    subtype='NONE', update=None)
```

Returns a new int property definition.

**Parameters**

- **name** (*string*) – Name used in the user interface.
- **description** (*string*) – Text used for the tooltip and api documentation.
- **options** (*set*) – Enumerator in ['HIDDEN', 'SKIP\_SAVE', 'ANIMATABLE'].
- **subtype** (*string*) – Enumerator in ['UNSIGNED', 'PERCENTAGE', 'FACTOR', 'ANGLE', 'TIME', 'DISTANCE', 'NONE'].
- **update** (*function*) – function to be called when this value is modified, This function must take 2 values (self, context) and return None.

```
bpy.props.IntVectorProperty(name="", description="", default=(0, 0, 0), min=-sys.maxint,
                          max=sys.maxint, soft_min=-sys.maxint, soft_max=sys.maxint,
                          options={'ANIMATABLE'}, subtype='NONE', size=3, update=None)
```

Returns a new vector int property definition.

**Parameters**

- **name** (*string*) – Name used in the user interface.
- **description** (*string*) – Text used for the tooltip and api documentation.
- **default** (*sequence*) – sequence of ints the length of *size*.
- **options** (*set*) – Enumerator in ['HIDDEN', 'SKIP\_SAVE', 'ANIMATABLE'].
- **subtype** (*string*) – Enumerator in ['COLOR', 'TRANSLATION', 'DIRECTION', 'VELOCITY', 'ACCELERATION', 'MATRIX', 'EULER', 'QUATERNION', 'AXISANGLE', 'XYZ', 'COLOR\_GAMMA', 'LAYER', 'NONE'].
- **size** (*int*) – Vector dimensions in [1, and 32].
- **update** (*function*) – function to be called when this value is modified, This function must take 2 values (self, context) and return None.

```
bpy.props.PointerProperty(type="", description="", options={'ANIMATABLE'}, update=None)
```

Returns a new pointer property definition.

**Parameters**

- **type** (*class*) – A subclass of `bpy.types.PropertyGroup`.
- **name** (*string*) – Name used in the user interface.
- **description** (*string*) – Text used for the tooltip and api documentation.
- **options** (*set*) – Enumerator in ['HIDDEN', 'SKIP\_SAVE', 'ANIMATABLE'].
- **update** (*function*) – function to be called when this value is modified, This function must take 2 values (self, context) and return None.

`bpy.props.RemoveProperty` (*attr*)

Removes a dynamically defined property.

**Parameters** *attr* (*string*) – Property name.

`bpy.props.StringProperty` (*name=""*, *description=""*, *default=""*, *maxlen=0*, *options={'ANIMATABLE'}*, *subtype='NONE'*, *update=None*)

Returns a new string property definition.

**Parameters**

- **name** (*string*) – Name used in the user interface.
- **description** (*string*) – Text used for the tooltip and api documentation.
- **options** (*set*) – Enumerator in ['HIDDEN', 'SKIP\_SAVE', 'ANIMATABLE'].
- **subtype** (*string*) – Enumerator in ['FILE\_PATH', 'DIR\_PATH', 'FILENAME', 'NONE'].
- **update** (*function*) – function to be called when this value is modified, This function must take 2 values (self, context) and return None.



# STANDALONE MODULES

## 3.1 Math Types & Utilities (mathutils)

This module provides access to matrices, eulers, quaternions and vectors.

```
import mathutils
from math import radians

vec = mathutils.Vector((1.0, 2.0, 3.0))

mat_rot = mathutils.Matrix.Rotation(radians(90.0), 4, 'X')
mat_trans = mathutils.Matrix.Translation(vec)

mat = mat_trans * mat_rot
mat.invert()

mat3 = mat.to_3x3()
quat1 = mat.to_quaternion()
quat2 = mat3.to_quaternion()

quat_diff = quat1.rotation_difference(quat2)

print(quat_diff.angle)
```

**class** mathutils.Color

This object gives access to Colors in Blender.

```
import mathutils

# color values are represented as RGB values from 0 - 1, this is blue
col = mathutils.Color((0.0, 0.0, 1.0))

# as well as r/g/b attribute access you can adjust them by h/s/v
col.s *= 0.5

# you can access its components by attribute or index
print("Color R:", col.r)
print("Color G:", col[1])
print("Color B:", col[-1])
print("Color HSV: %.2f, %.2f, %.2f", col[:])

# components of an existing color can be set
col[:] = 0.0, 0.5, 1.0
```

```
# components of an existing color can use slice notation to get a tuple
print("Values: %f, %f, %f" % col[:])

# colors can be added and subtracted
col += mathutils.Color((0.25, 0.0, 0.0))

# Color can be multiplied, in this example color is scaled to 0-255
# can printed as integers
print("Color: %d, %d, %d" % (col * 255.0)[:])

# This example prints the color as hexadecimal
print("Hexidecimal: %.2x%.2x%.2x" % (col * 255.0)[:])
```

**copy()**

Returns a copy of this color.

**Returns** A copy of the color.

**Return type** Color

---

**Note:** use this to get a copy of a wrapped color with no reference to the original data.

---

**b**

Blue color channel.

**Type** float

**g**

Green color channel.

**Type** float

**h**

HSV Hue component in [0, 1].

**Type** float

**hsv**

HSV Values in [0, 1].

**Type** float triplet

**is\_wrapped**

True when this object wraps external data (readonly).

**Type** boolean

**owner**

The item this is wrapping or None (readonly).

**r**

Red color channel.

**Type** float

**s**

HSV Saturation component in [0, 1].

**Type** float

**v**

HSV Value component in [0, 1].

**Type** float

**class** `mathutils.Euler`

This object gives access to Eulers in Blender.

```
import mathutils
import math

# create a new euler with default axis rotation order
eul = mathutils.Euler((0.0, math.radians(45.0), 0.0), 'XYZ')

# rotate the euler
eul.rotate_axis(math.radians(10.0), 'Z')

# you can access its components by attribute or index
print("Euler X", eul.x)
print("Euler Y", eul[1])
print("Euler Z", eul[-1])

# components of an existing euler can be set
eul[:] = 1.0, 2.0, 3.0

# components of an existing euler can use slice notation to get a tuple
print("Values: %f, %f, %f" % eul[:])

# the order can be set at any time too
eul.order = 'ZYX'

# eulers can be used to rotate vectors
vec = mathutils.Vector((0.0, 0.0, 1.0))
vec.rotate(eul)

# often its useful to convert the euler into a matrix so it can be used as
# transformations with more flexibility
mat_rot = eul.to_matrix()
mat_loc = mathutils.Matrix.Translation((2.0, 3.0, 4.0))
mat = mat_loc * mat_rot.to_4x4()
```

**copy()**

Returns a copy of this euler.

**Returns** A copy of the euler.

**Return type** Euler

---

**Note:** use this to get a copy of a wrapped euler with no reference to the original data.

---

**make\_compatible** (*other*)

Make this euler compatible with another, so interpolating between them works as intended.

---

**Note:** the rotation order is not taken into account for this function.

---

**rotate** (*other*)

Rotates the euler a by another mathutils value.

**Parameters** *other* (Euler, Quaternion or Matrix) – rotation component of mathutils value

**rotate\_axis** (*axis, angle*)

Rotates the euler a certain amount and returning a unique euler rotation (no 720 degree pitches).

**Parameters**

- **axis** (*string*) – single character in ['X', 'Y', 'Z'].
- **angle** (*float*) – angle in radians.

**to\_matrix** ()

Return a matrix representation of the euler.

**Returns** A 3x3 rotation matrix representation of the euler.

**Return type** `Matrix`

**to\_quaternion** ()

Return a quaternion representation of the euler.

**Returns** Quaternion representation of the euler.

**Return type** `Quaternion`

**zero** ()

Set all values to zero.

**is\_wrapped**

True when this object wraps external data (readonly).

**Type** `boolean`

**order**

Euler rotation order.

**Type** string in ['XYZ', 'XZY', 'YXZ', 'YZX', 'ZXY', 'ZYX']

**owner**

The item this is wrapping or None (readonly).

**x**

Euler X axis in radians.

**Type** `float`

**y**

Euler Y axis in radians.

**Type** `float`

**z**

Euler Z axis in radians.

**Type** `float`

**class** `mathutils.Matrix`

This object gives access to Matrices in Blender.

```
import mathutils
import math
```

```
# create a location matrix
mat_loc = mathutils.Matrix.Translation((2.0, 3.0, 4.0))
```

```
# create an identity matrix
mat_sca = mathutils.Matrix.Scale(0.5, 4, (0.0, 0.0, 1.0))
```



```

# create a rotation matrix
mat_rot = mathutils.Matrix.Rotation(math.radians(45.0), 4, 'X')

# combine transformations
mat_out = mat_loc * mat_rot * mat_sca
print(mat_out)

# extract components back out of the matrix
loc, rot, sca = mat_out.decompose()
print(loc, rot, sca)

# it can also be useful to access components of a matrix directly
mat = mathutils.Matrix()
mat[0][0], mat[1][0], mat[2][0] = 0.0, 1.0, 2.0

mat[0][0:3] = 0.0, 1.0, 2.0

# each item in a matrix is a vector so vector utility functions can be used
mat[0].xyz = 0.0, 1.0, 2.0

```

**classmethod `OrthoProjection`** (*axis, size*)

Create a matrix to represent an orthographic projection.

**Parameters**

- **axis** (string or `Vector`) – Can be any of the following: ['X', 'Y', 'XY', 'XZ', 'YZ'], where a single axis is for a 2D matrix. Or a vector for an arbitrary axis
- **size** (*int*) – The size of the projection matrix to construct [2, 4].

**Returns** A new projection matrix.

**Return type** `Matrix`

**classmethod `Rotation`** (*angle, size, axis*)

Create a matrix representing a rotation.

**Parameters**

- **angle** (*float*) – The angle of rotation desired, in radians.
- **size** (*int*) – The size of the rotation matrix to construct [2, 4].
- **axis** (string or `Vector`) – a string in ['X', 'Y', 'Z'] or a 3D Vector Object (optional when size is 2).

**Returns** A new rotation matrix.

**Return type** `Matrix`

**classmethod `Scale`** (*factor, size, axis*)

Create a matrix representing a scaling.

**Parameters**

- **factor** (*float*) – The factor of scaling to apply.
- **size** (*int*) – The size of the scale matrix to construct [2, 4].
- **axis** (`Vector`) – Direction to influence scale. (optional).

**Returns** A new scale matrix.

**Return type** `Matrix`

**classmethod Shear** (*plane, size, factor*)

Create a matrix to represent an shear transformation.

**Parameters**

- **plane** (*string*) – Can be any of the following: ['X', 'Y', 'XY', 'XZ', 'YZ'], where a single axis is for a 2D matrix only.
- **size** (*int*) – The size of the shear matrix to construct [2, 4].
- **factor** (*float or float pair*) – The factor of shear to apply. For a 3 or 4 *size* matrix pass a pair of floats corresponding with the *plane* axis.

**Returns** A new shear matrix.

**Return type** `Matrix`

**classmethod Translation** (*vector*)

Create a matrix representing a translation.

**Parameters** **vector** (`Vector`) – The translation vector.

**Returns** An identity matrix with a translation.

**Return type** `Matrix`

**copy** ()

Returns a copy of this matrix.

**Returns** an instance of itself

**Return type** `Matrix`

**decompose** ()

Return the location, rotaion and scale components of this matrix.

**Returns** loc, rot, scale triple.

**Return type** (`Vector, Quaternion, Vector`)

**determinant** ()

Return the determinant of a matrix.

**Returns** Return a the determinant of a matrix.

**Return type** float

**See Also:**

[<http://en.wikipedia.org/wiki/Determinant>](http://en.wikipedia.org/wiki/Determinant)

**identity** ()

Set the matrix to the identity matrix.

---

**Note:** An object with zero location and rotation, a scale of one, will have an identity matrix.

---

**See Also:**

[<http://en.wikipedia.org/wiki/Identity\\_matrix>](http://en.wikipedia.org/wiki/Identity_matrix)

**invert** ()

Set the matrix to its inverse.

**See Also:**

[<http://en.wikipedia.org/wiki/Inverse\\_matrix>](http://en.wikipedia.org/wiki/Inverse_matrix)

**inverted()**

Return an inverted copy of the matrix.

**Returns** the inverted matrix.

**Return type** `Matrix`

**lerp** (*other*, *factor*)

Returns the interpolation of two matrices.

**Parameters**

- **other** (`Matrix`) – value to interpolate with.
- **factor** (*float*) – The interpolation value in [0.0, 1.0].

**Returns** The interpolated rotation.

**Return type** `Matrix`

**resize\_4x4()**

Resize the matrix to 4x4.

**rotate** (*other*)

Rotates the matrix a by another mathutils value.

**Parameters** **other** (`Euler`, `Quaternion` or `Matrix`) – rotation component of mathutils value

---

**Note:** If any of the columns are not unit length this may not have desired results.

---

**to\_3x3()**

Return a 3x3 copy of this matrix.

**Returns** a new matrix.

**Return type** `Matrix`

**to\_4x4()**

Return a 4x4 copy of this matrix.

**Returns** a new matrix.

**Return type** `Matrix`

**to\_euler** (*order*, *euler\_compat*)

Return an Euler representation of the rotation matrix (3x3 or 4x4 matrix only).

**Parameters**

- **order** (*string*) – Optional rotation order argument in ['XYZ', 'XZY', 'YXZ', 'YZX', 'ZXY', 'ZYX'].
- **euler\_compat** (`Euler`) – Optional euler argument the new euler will be made compatible with (no axis flipping between them). Useful for converting a series of matrices to animation curves.

**Returns** Euler representation of the matrix.

**Return type** `Euler`

**to\_quaternion()**

Return a quaternion representation of the rotation matrix.

**Returns** Quaternion representation of the rotation matrix.

**Return type** `Quaternion`

**to\_scale()**

Return a the scale part of a 3x3 or 4x4 matrix.

**Returns** Return a the scale of a matrix.

**Return type** `Vector`

---

**Note:** This method does not return negative a scale on any axis because it is not possible to obtain this data from the matrix alone.

---

**to\_translation()**

Return a the translation part of a 4 row matrix.

**Returns** Return a the translation of a matrix.

**Return type** `Vector`

**transpose()**

Set the matrix to its transpose.

**See Also:**

[<http://en.wikipedia.org/wiki/Transpose>](http://en.wikipedia.org/wiki/Transpose)

**transposed()**

Return a new, transposed matrix.

**Returns** a transposed matrix

**Return type** `Matrix`

**zero()**

Set all the matrix values to zero.

**Returns** an instance of itself

**Return type** `Matrix`

**col\_size**

The column size of the matrix (readonly).

**Type** `int`

**is\_negative**

True if this matrix results in a negative scale, 3x3 and 4x4 only, (readonly).

**Type** `bool`

**is\_orthogonal**

True if this matrix is orthogonal, 3x3 and 4x4 only, (readonly).

**Type** `bool`

**is\_wrapped**

True when this object wraps external data (readonly).

**Type** `boolean`

**median\_scale**

The average scale applied to each axis (readonly).

**Type** `float`

**owner**

The item this is wrapping or None (readonly).

**row\_size**

The row size of the matrix (readonly).

**Type** int

**class** mathutils.Quaternion

This object gives access to Quaternions in Blender.

```
import mathutils
import math

# a new rotation 90 degrees about the Y axis
quat_a = mathutils.Quaternion((0.7071068, 0.0, 0.7071068, 0.0))

# passing values to Quaternion's directly can be confusing so axis, angle
# is supported for initializing too
quat_b = mathutils.Quaternion((0.0, 1.0, 0.0), math.radians(90.0))

print("Check quaternions match", quat_a == quat_b)

# like matrices, quaternions can be multiplied to accumulate rotational values
quat_a = mathutils.Quaternion((0.0, 1.0, 0.0), math.radians(90.0))
quat_b = mathutils.Quaternion((0.0, 0.0, 1.0), math.radians(45.0))
quat_out = quat_a * quat_b

# print the quat, euler degrees for near mortals and (axis, angle)
print("Final Rotation:")
print(quat_out)
print("%.2f, %.2f, %.2f" % tuple(math.degrees(a) for a in quat_out.to_euler()))
print("%.2f, %.2f, %.2f), %.2f" % (quat_out.axis[:] +
                                (math.degrees(quat_out.angle), )))
```

**conjugate()**

Set the quaternion to its conjugate (negate x, y, z).

**conjugated()**

Return a new conjugated quaternion.

**Returns** a new quaternion.

**Return type** Quaternion

**copy()**

Returns a copy of this quaternion.

**Returns** A copy of the quaternion.

**Return type** Quaternion

---

**Note:** use this to get a copy of a wrapped quaternion with no reference to the original data.

---

**cross** (*other*)

Return the cross product of this quaternion and another.

**Parameters** *other* (Quaternion) – The other quaternion to perform the cross product with.

**Returns** The cross product.

**Return type** Quaternion

**dot** (*other*)

Return the dot product of this quaternion and another.

**Parameters** **other** (`Quaternion`) – The other quaternion to perform the dot product with.

**Returns** The dot product.

**Return type** `Quaternion`

**identity** ()

Set the quaternion to an identity quaternion.

**Returns** an instance of itself.

**Return type** `Quaternion`

**invert** ()

Set the quaternion to its inverse.

**inverted** ()

Return a new, inverted quaternion.

**Returns** the inverted value.

**Return type** `Quaternion`

**negate** ()

Set the quaternion to its negative.

**Returns** an instance of itself.

**Return type** `Quaternion`

**normalize** ()

Normalize the quaternion.

**normalized** ()

Return a new normalized quaternion.

**Returns** a normalized copy.

**Return type** `Quaternion`

**rotate** (*other*)

Rotates the quaternion a by another mathutils value.

**Parameters** **other** (`Euler`, `Quaternion` or `Matrix`) – rotation component of mathutils value

**rotation\_difference** (*other*)

Returns a quaternion representing the rotational difference.

**Parameters** **other** (`Quaternion`) – second quaternion.

**Returns** the rotational difference between the two quat rotations.

**Return type** `Quaternion`

**slerp** (*other, factor*)

Returns the interpolation of two quaternions.

**Parameters**

- **other** (`Quaternion`) – value to interpolate with.
- **factor** (`float`) – The interpolation value in [0.0, 1.0].

**Returns** The interpolated rotation.

**Return type** `Quaternion`

**to\_axis\_angle** ()

Return the axis, angle representation of the quaternion.

**Returns** axis, angle.

**Return type** (`Vector`, float) pair

**to\_euler** (*order*, *euler\_compat*)

Return Euler representation of the quaternion.

**Parameters**

- **order** (*string*) – Optional rotation order argument in ['XYZ', 'XZY', 'YXZ', 'YZX', 'ZXY', 'ZYX'].
- **euler\_compat** (`Euler`) – Optional euler argument the new euler will be made compatible with (no axis flipping between them). Useful for converting a series of matrices to animation curves.

**Returns** Euler representation of the quaternion.

**Return type** `Euler`

**to\_matrix** ()

Return a matrix representation of the quaternion.

**Returns** A 3x3 rotation matrix representation of the quaternion.

**Return type** `Matrix`

**angle**

angle of the quaternion.

**Type** float

**axis**

quaternion axis as a vector.

**Type** `Vector`

**is\_wrapped**

True when this object wraps external data (readonly).

**Type** boolean

**magnitude**

Size of the quaternion (readonly).

**Type** float

**owner**

The item this is wrapping or None (readonly).

**w**

Quaternion W value.

**Type** float

**x**

Quaternion X axis.

**Type** float

**y**

Quaternion Y axis.

**Type** float

**z**

Quaternion Z axis.

**Type** float

**class** `mathutils.Vector`

This object gives access to Vectors in Blender.

```
import mathutils

# zero length vector
vec = mathutils.Vector((0.0, 0.0, 1.0))

# unit length vector
vec_a = vec.copy().normalize()

vec_b = mathutils.Vector((0.0, 1.0, 2.0))

vec2d = mathutils.Vector((1.0, 2.0))
vec3d = mathutils.Vector((1.0, 0.0, 0.0))
vec4d = vec_a.to_4d()

# other mathutils types
quat = mathutils.Quaternion()
matrix = mathutils.Matrix()

# Comparison operators can be done on Vector classes:

# greater and less then test vector length.
vec_a > vec_b
vec_a >= vec_b
vec_a < vec_b
vec_a <= vec_b

# ==, != test vector values e.g. 1,2,3 != 3,2,1 even if they are the same length
vec_a == vec_b
vec_a != vec_b

# Math can be performed on Vector classes
vec_a + vec_b
vec_a - vec_b
vec_a * vec_b
vec_a * 10.0
matrix * vec_a
quat * vec_a
vec_a * vec_b
-vec_a

# You can access a vector object like a sequence
x = vec_a[0]
len(vec)
vec_a[:] = vec_b
vec_a[:] = 1.0, 2.0, 3.0
vec2d[:] = vec3d[:2]
```



```
# Vectors support 'swizzle' operations
# See http://en.wikipedia.org/wiki/Swizzling_(computer_graphics)
vec.xyz = vec.zyx
vec.xy = vec4d.zw
vec.xyz = vec4d.wzz
vec4d.wxyz = vec.yxyx
```

**angle** (*other, fallback*)

Return the angle between two vectors.

**Parameters**

- **other** (`Vector`) – another vector to compare the angle with
- **fallback** (*any*) – return this value when the angle cant be calculated (zero length vector)

**Returns** angle in radians or fallback when given

**Return type** float

---

**Note:** Zero length vectors raise an `AttributeError`.

---

**copy** ()

Returns a copy of this vector.

**Returns** A copy of the vector.

**Return type** `Vector`

---

**Note:** use this to get a copy of a wrapped vector with no reference to the original data.

---

**cross** (*other*)

Return the cross product of this vector and another.

**Parameters** **other** (`Vector`) – The other vector to perform the cross product with.

**Returns** The cross product.

**Return type** `Vector`

---

**Note:** both vectors must be 3D

---

**dot** (*other*)

Return the dot product of this vector and another.

**Parameters** **other** (`Vector`) – The other vector to perform the dot product with.

**Returns** The dot product.

**Return type** `Vector`

**lerp** (*other, factor*)

Returns the interpolation of two vectors.

**Parameters**

- **other** (`Vector`) – value to interpolate with.
- **factor** (*float*) – The interpolation value in [0.0, 1.0].

**Returns** The interpolated rotation.

**Return type** `Vector`

**negate** ()

Set all values to their negative.

**Returns** an instance of itself

**Return type** `Vector`

**normalize** ()

Normalize the vector, making the length of the vector always 1.0.

|  |
|--|
| <p><b>Warning:</b> Normalizing a vector where all values are zero results in all axis having a nan value (not a number).</p> |
|--|

---

**Note:** Normalize works for vectors of all sizes, however 4D Vectors w axis is left untouched.

---

**normalized** ()

Return a new, normalized vector.

**Returns** a normalized copy of the vector

**Return type** `Vector`

**project** (*other*)

Return the projection of this vector onto the *other*.

**Parameters** **other** (`Vector`) – second vector.

**Returns** the parallel projection vector

**Return type** `Vector`

**reflect** (*mirror*)

Return the reflection vector from the *mirror* argument.

**Parameters** **mirror** (`Vector`) – This vector could be a normal from the reflecting surface.

**Returns** The reflected vector matching the size of this vector.

**Return type** `Vector`

**resize\_2d** ()

Resize the vector to 2D (x, y).

**Returns** an instance of itself

**Return type** `Vector`

**resize\_3d** ()

Resize the vector to 3D (x, y, z).

**Returns** an instance of itself

**Return type** `Vector`

**resize\_4d** ()

Resize the vector to 4D (x, y, z, w).

**Returns** an instance of itself

**Return type** `Vector`

**rotate** (*other*)

Return vector by a rotation value.

**Parameters other** (*Euler*, *Quaternion* or *Matrix*) – rotation component of mathutils value

**difference** (*other*)

Returns a quaternion representing the rotational difference between this vector and another.

**Parameters other** (*Vector*) – second vector.

**Returns** the rotational difference between the two vectors.

**Return type** *Quaternion*

**Note:** 2D vectors raise an *AttributeError*.

**to\_2d**()

Return a 2d copy of the vector.

**Returns** a new vector

**Return type** *Vector*

**to\_3d**()

Return a 3d copy of the vector.

**Returns** a new vector

**Return type** *Vector*

**to\_4d**()

Return a 4d copy of the vector.

**Returns** a new vector

**Return type** *Vector*

**to\_track\_quat** (*track*, *up*)

Return a quaternion rotation from the vector and the track and up axis.

**Parameters**

- **track** (*string*) – Track axis in ['X', 'Y', 'Z', '-X', '-Y', '-Z'].
- **up** (*string*) – Up axis in ['X', 'Y', 'Z'].

**Returns** rotation from the vector and the track and up axis.

**Return type** *Quaternion*

**to\_tuple** (*precision=-1*)

Return this vector as a tuple with.

**Parameters precision** (*int*) – The number to round the value to in [-1, 21].

**Returns** the values of the vector rounded by *precision*

**Return type** tuple

**zero**()

Set all values to zero.

**is\_wrapped**

True when this object wraps external data (readonly).

**Type** boolean

**length**

Vector Length.

**Type** float

**length\_squared**

Vector length squared ( $v \cdot v$ ).

**Type** float

**magnitude**

Vector Length.

**Type** float

**owner**

The item this is wrapping or None (readonly).

**w**

Vector W axis (4D Vectors only).

**Type** float

**ww**

Undocumented ([contribute](#))

**www**

Undocumented ([contribute](#))

**wwww**

Undocumented ([contribute](#))

**wwwx**

Undocumented ([contribute](#))

**wwwy**

Undocumented ([contribute](#))

**wwwz**

Undocumented ([contribute](#))

**wwx**

Undocumented ([contribute](#))

**wwxw**

Undocumented ([contribute](#))

**wwxx**

Undocumented ([contribute](#))

**wwxy**

Undocumented ([contribute](#))

**wwxz**

Undocumented ([contribute](#))

**wwy**

Undocumented ([contribute](#))

**wwyw**

Undocumented ([contribute](#))

**wwyx**

Undocumented ([contribute](#))



**wxyy**  
Undocumented ([contribute](#))

**wxyz**  
Undocumented ([contribute](#))

**wxz**  
Undocumented ([contribute](#))

**wxzw**  
Undocumented ([contribute](#))

**wxzx**  
Undocumented ([contribute](#))

**wxzy**  
Undocumented ([contribute](#))

**wxzz**  
Undocumented ([contribute](#))

**wy**  
Undocumented ([contribute](#))

**wyw**  
Undocumented ([contribute](#))

**wyww**  
Undocumented ([contribute](#))

**wywx**  
Undocumented ([contribute](#))

**wywy**  
Undocumented ([contribute](#))

**wywz**  
Undocumented ([contribute](#))

**wyx**  
Undocumented ([contribute](#))

**wyxx**  
Undocumented ([contribute](#))

**wyxy**  
Undocumented ([contribute](#))

**wyxz**  
Undocumented ([contribute](#))

**wyy**  
Undocumented ([contribute](#))

**wyyw**  
Undocumented ([contribute](#))

**wyyx**  
Undocumented ([contribute](#))

**wyyy**  
Undocumented (contribute)

**wyyz**  
Undocumented (contribute)

**wyz**  
Undocumented (contribute)

**wyzw**  
Undocumented (contribute)

**wyzz**  
Undocumented (contribute)

**wyzy**  
Undocumented (contribute)

**wyzz**  
Undocumented (contribute)

**wz**  
Undocumented (contribute)

**wzw**  
Undocumented (contribute)

**wzww**  
Undocumented (contribute)

**wzwx**  
Undocumented (contribute)

**wzwy**  
Undocumented (contribute)

**wzwz**  
Undocumented (contribute)

**wzx**  
Undocumented (contribute)

**wzxw**  
Undocumented (contribute)

**wzxx**  
Undocumented (contribute)

**wzxy**  
Undocumented (contribute)

**wzxx**  
Undocumented (contribute)

**wzy**  
Undocumented (contribute)

**wzyw**  
Undocumented (contribute)

**wzyx**  
Undocumented (contribute)

**wzyy**  
Undocumented (contribute)

**wzyz**  
Undocumented (contribute)

**wzz**  
Undocumented (contribute)

**wzzw**  
Undocumented (contribute)

**wzzx**  
Undocumented (contribute)

**wzzy**  
Undocumented (contribute)

**wzzz**  
Undocumented (contribute)

**x**  
Vector X axis.  
**Type** float

**xw**  
Undocumented (contribute)

**xww**  
Undocumented (contribute)

**xwww**  
Undocumented (contribute)

**xwwx**  
Undocumented (contribute)

**xwwy**  
Undocumented (contribute)

**xwwz**  
Undocumented (contribute)

**xwx**  
Undocumented (contribute)

**xwxw**  
Undocumented (contribute)

**xwxx**  
Undocumented (contribute)

**xwxy**  
Undocumented (contribute)

**xwxz**  
Undocumented (contribute)

**xwy**  
Undocumented (contribute)

**xwyw**  
Undocumented (contribute)



**xwyx**  
Undocumented (contribute)

**xwyy**  
Undocumented (contribute)

**xwyz**  
Undocumented (contribute)

**xwz**  
Undocumented (contribute)

**xwzw**  
Undocumented (contribute)

**xwzx**  
Undocumented (contribute)

**xwzy**  
Undocumented (contribute)

**xwzz**  
Undocumented (contribute)

**xx**  
Undocumented (contribute)

**xxw**  
Undocumented (contribute)

**xxww**  
Undocumented (contribute)

**xxwx**  
Undocumented (contribute)

**xxwy**  
Undocumented (contribute)

**xxwz**  
Undocumented (contribute)

**xxx**  
Undocumented (contribute)

**xxxw**  
Undocumented (contribute)

**xxxx**  
Undocumented (contribute)

**xxxxy**  
Undocumented (contribute)

**xxxz**  
Undocumented (contribute)

**xyy**  
Undocumented (contribute)

**xyyw**  
Undocumented (contribute)

**xxyx**  
Undocumented (contribute)

**xyyy**  
Undocumented (contribute)

**xyyz**  
Undocumented (contribute)

**xxz**  
Undocumented (contribute)

**xxzw**  
Undocumented (contribute)

**xxzx**  
Undocumented (contribute)

**xxzy**  
Undocumented (contribute)

**xxzz**  
Undocumented (contribute)

**xy**  
Undocumented (contribute)

**xyw**  
Undocumented (contribute)

**xyww**  
Undocumented (contribute)

**xywx**  
Undocumented (contribute)

**xywy**  
Undocumented (contribute)

**xywz**  
Undocumented (contribute)

**xyx**  
Undocumented (contribute)

**xyxw**  
Undocumented (contribute)

**xyxx**  
Undocumented (contribute)

**xyxy**  
Undocumented (contribute)

**xyxz**  
Undocumented (contribute)

**xyy**  
Undocumented (contribute)

**xyyw**  
Undocumented (contribute)

**xyyx**  
Undocumented (contribute)

**xyyy**  
Undocumented (contribute)

**xyyz**  
Undocumented (contribute)

**xyz**  
Undocumented (contribute)

**xyzw**  
Undocumented (contribute)

**xyzx**  
Undocumented (contribute)

**xyzy**  
Undocumented (contribute)

**xyzz**  
Undocumented (contribute)

**xz**  
Undocumented (contribute)

**xzw**  
Undocumented (contribute)

**xzww**  
Undocumented (contribute)

**xzwx**  
Undocumented (contribute)

**xzwy**  
Undocumented (contribute)

**xzww**  
Undocumented (contribute)

**xzx**  
Undocumented (contribute)

**xzxw**  
Undocumented (contribute)

**xzxx**  
Undocumented (contribute)

**xzxy**  
Undocumented (contribute)

**xzxx**  
Undocumented (contribute)

**xzy**  
Undocumented (contribute)

**xzyw**  
Undocumented (contribute)

**xzyx**  
Undocumented (contribute)

**xzyy**  
Undocumented (contribute)

**xzyz**  
Undocumented (contribute)

**xzz**  
Undocumented (contribute)

**xzzw**  
Undocumented (contribute)

**xzzx**  
Undocumented (contribute)

**xzzy**  
Undocumented (contribute)

**xzzz**  
Undocumented (contribute)

**Y**  
Vector Y axis.  
**Type** float

**yw**  
Undocumented (contribute)

**yww**  
Undocumented (contribute)

**ywww**  
Undocumented (contribute)

**ywwx**  
Undocumented (contribute)

**ywwy**  
Undocumented (contribute)

**ywwz**  
Undocumented (contribute)

**ywx**  
Undocumented (contribute)

**ywxw**  
Undocumented (contribute)

**ywx x**  
Undocumented (contribute)

**ywxy**  
Undocumented (contribute)

**ywxz**  
Undocumented (contribute)

**ywy**  
Undocumented (contribute)

**ywyw**  
Undocumented (contribute)

**ywyx**  
Undocumented (contribute)

**ywyy**  
Undocumented (contribute)

**ywyz**  
Undocumented (contribute)

**ywz**  
Undocumented (contribute)

**ywzw**  
Undocumented (contribute)

**ywzx**  
Undocumented (contribute)

**ywzy**  
Undocumented (contribute)

**ywzz**  
Undocumented (contribute)

**yx**  
Undocumented (contribute)

**yxw**  
Undocumented (contribute)

**yxww**  
Undocumented (contribute)

**yxwx**  
Undocumented (contribute)

**yxwy**  
Undocumented (contribute)

**yxwz**  
Undocumented (contribute)

**yxx**  
Undocumented (contribute)

**yxxw**  
Undocumented (contribute)

**yxxx**  
Undocumented (contribute)

**yxxy**  
Undocumented (contribute)

**yxxz**  
Undocumented (contribute)

**xyy**  
Undocumented (contribute)

**yxyw**  
Undocumented ([contribute](#))

**yxyx**  
Undocumented ([contribute](#))

**yxyy**  
Undocumented ([contribute](#))

**yxyz**  
Undocumented ([contribute](#))

**yxz**  
Undocumented ([contribute](#))

**yxzw**  
Undocumented ([contribute](#))

**yxzx**  
Undocumented ([contribute](#))

**yxzy**  
Undocumented ([contribute](#))

**yxzz**  
Undocumented ([contribute](#))

**yy**  
Undocumented ([contribute](#))

**yyw**  
Undocumented ([contribute](#))

**yyww**  
Undocumented ([contribute](#))

**yywx**  
Undocumented ([contribute](#))

**yywy**  
Undocumented ([contribute](#))

**yywz**  
Undocumented ([contribute](#))

**yyx**  
Undocumented ([contribute](#))

**yyxw**  
Undocumented ([contribute](#))

**yyxx**  
Undocumented ([contribute](#))

**yyxy**  
Undocumented ([contribute](#))

**yyxz**  
Undocumented ([contribute](#))

**yyy**  
Undocumented ([contribute](#))

**YYYw**  
Undocumented (contribute)

**YYYx**  
Undocumented (contribute)

**YYYY**  
Undocumented (contribute)

**YYYz**  
Undocumented (contribute)

**YYz**  
Undocumented (contribute)

**YYzw**  
Undocumented (contribute)

**YYzx**  
Undocumented (contribute)

**YYzy**  
Undocumented (contribute)

**YYzz**  
Undocumented (contribute)

**yz**  
Undocumented (contribute)

**yzw**  
Undocumented (contribute)

**yzww**  
Undocumented (contribute)

**yzwx**  
Undocumented (contribute)

**yzwy**  
Undocumented (contribute)

**yzwz**  
Undocumented (contribute)

**yzx**  
Undocumented (contribute)

**yzxw**  
Undocumented (contribute)

**yzxx**  
Undocumented (contribute)

**yzxy**  
Undocumented (contribute)

**yzxz**  
Undocumented (contribute)

**zyz**  
Undocumented (contribute)

- zyzw**  
Undocumented (contribute)
- zyzx**  
Undocumented (contribute)
- zyyy**  
Undocumented (contribute)
- zyyz**  
Undocumented (contribute)
- zyzz**  
Undocumented (contribute)
- zyzw**  
Undocumented (contribute)
- zyzx**  
Undocumented (contribute)
- zyzy**  
Undocumented (contribute)
- zyzz**  
Undocumented (contribute)
- z**  
Vector Z axis (3D Vectors only).  
**Type** float
- zw**  
Undocumented (contribute)
- zww**  
Undocumented (contribute)
- zwww**  
Undocumented (contribute)
- zwwx**  
Undocumented (contribute)
- zwwy**  
Undocumented (contribute)
- zwwz**  
Undocumented (contribute)
- zwx**  
Undocumented (contribute)
- zwxw**  
Undocumented (contribute)
- zwxw**  
Undocumented (contribute)
- zwxxy**  
Undocumented (contribute)
- zwxz**  
Undocumented (contribute)



**zwy**  
Undocumented (contribute)

**zwyw**  
Undocumented (contribute)

**zwyx**  
Undocumented (contribute)

**zwy y**  
Undocumented (contribute)

**zwyz**  
Undocumented (contribute)

**zwz**  
Undocumented (contribute)

**zwzw**  
Undocumented (contribute)

**zwzx**  
Undocumented (contribute)

**zwzy**  
Undocumented (contribute)

**zwzz**  
Undocumented (contribute)

**zx**  
Undocumented (contribute)

**zxw**  
Undocumented (contribute)

**zxww**  
Undocumented (contribute)

**zxwx**  
Undocumented (contribute)

**zxwy**  
Undocumented (contribute)

**zxwz**  
Undocumented (contribute)

**zxx**  
Undocumented (contribute)

**zxxw**  
Undocumented (contribute)

**zxxx**  
Undocumented (contribute)

**zxx y**  
Undocumented (contribute)

**zxxxz**  
Undocumented (contribute)

**zxy**  
Undocumented ([contribute](#))

**zxyw**  
Undocumented ([contribute](#))

**zxyx**  
Undocumented ([contribute](#))

**zxyy**  
Undocumented ([contribute](#))

**zxyz**  
Undocumented ([contribute](#))

**zxxz**  
Undocumented ([contribute](#))

**zxxw**  
Undocumented ([contribute](#))

**zxxx**  
Undocumented ([contribute](#))

**zxzy**  
Undocumented ([contribute](#))

**zxzz**  
Undocumented ([contribute](#))

**zy**  
Undocumented ([contribute](#))

**zyw**  
Undocumented ([contribute](#))

**zyww**  
Undocumented ([contribute](#))

**zywx**  
Undocumented ([contribute](#))

**zywy**  
Undocumented ([contribute](#))

**zywz**  
Undocumented ([contribute](#))

**zyx**  
Undocumented ([contribute](#))

**zyxw**  
Undocumented ([contribute](#))

**zyxx**  
Undocumented ([contribute](#))

**zyxy**  
Undocumented ([contribute](#))

**zyxz**  
Undocumented ([contribute](#))

**zYY**  
Undocumented (contribute)

**zYYw**  
Undocumented (contribute)

**zYYx**  
Undocumented (contribute)

**zYYY**  
Undocumented (contribute)

**zYYz**  
Undocumented (contribute)

**zyz**  
Undocumented (contribute)

**zyzw**  
Undocumented (contribute)

**zyzx**  
Undocumented (contribute)

**zyzy**  
Undocumented (contribute)

**zyzz**  
Undocumented (contribute)

**zz**  
Undocumented (contribute)

**zzw**  
Undocumented (contribute)

**zzww**  
Undocumented (contribute)

**zzwx**  
Undocumented (contribute)

**zzwy**  
Undocumented (contribute)

**zzwz**  
Undocumented (contribute)

**zzx**  
Undocumented (contribute)

**zzxw**  
Undocumented (contribute)

**zzxx**  
Undocumented (contribute)

**zzxy**  
Undocumented (contribute)

**zzxz**  
Undocumented (contribute)

**zzy**  
Undocumented (contribute)

**zzyw**  
Undocumented (contribute)

**zzyx**  
Undocumented (contribute)

**zzyy**  
Undocumented (contribute)

**zzyz**  
Undocumented (contribute)

**zzz**  
Undocumented (contribute)

**zzzw**  
Undocumented (contribute)

**zzzx**  
Undocumented (contribute)

**zzzy**  
Undocumented (contribute)

**zzzz**  
Undocumented (contribute)

## 3.2 Geometry Utilities (mathutils.geometry)

The Blender geometry module

`mathutils.geometry.area_tri` (*v1*, *v2*, *v3*)

Returns the area size of the 2D or 3D triangle defined.

### Parameters

- **v1** (`mathutils.Vector`) – Point1
- **v2** (`mathutils.Vector`) – Point2
- **v3** (`mathutils.Vector`) – Point3

**Return type** float

`mathutils.geometry.barycentric_transform` (*point*, *tri\_a1*, *tri\_a2*, *tri\_a3*, *tri\_b1*, *tri\_b2*,  
*tri\_b3*)

Return a transformed point, the transformation is defined by 2 triangles.

### Parameters

- **point** (`mathutils.Vector`) – The point to transform.
- **tri\_a1** (`mathutils.Vector`) – source triangle vertex.
- **tri\_a2** (`mathutils.Vector`) – source triangle vertex.
- **tri\_a3** (`mathutils.Vector`) – source triangle vertex.
- **tri\_b1** – target triangle vertex.
- **tri\_a2** – target triangle vertex.

- **tri\_a3** – target triangle vertex.

**Returns** The transformed point

**Return type** `mathutils.Vector`'s

`mathutils.geometry.box_pack_2d` (*boxes*)

Returns the normal of the 3D tri or quad.

**Parameters** **boxes** (*list*) – list of boxes, each box is a list where the first 4 items are [x, y, width, height, ...] other items are ignored.

**Returns** the width and height of the packed bounding box

**Return type** tuple, pair of floats

`mathutils.geometry.interpolate_bezier` (*knot1, handle1, handle2, knot2, resolution*)

Interpolate a bezier spline segment.

**Parameters**

- **knot1** (`mathutils.Vector`) – First bezier spline point.
- **handle1** (`mathutils.Vector`) – First bezier spline handle.
- **handle2** (`mathutils.Vector`) – Second bezier spline handle.
- **knot2** (`mathutils.Vector`) – Second bezier spline point.
- **resolution** (*int*) – Number of points to return.

**Returns** The interpolated points

**Return type** list of `mathutils.Vector`'s

`mathutils.geometry.intersect_line_line` (*v1, v2, v3, v4*)

Returns a tuple with the points on each line respectively closest to the other.

**Parameters**

- **v1** (`mathutils.Vector`) – First point of the first line
- **v2** (`mathutils.Vector`) – Second point of the first line
- **v3** (`mathutils.Vector`) – First point of the second line
- **v4** (`mathutils.Vector`) – Second point of the second line

**Return type** tuple of `mathutils.Vector`'s

`mathutils.geometry.intersect_line_line_2d` (*lineA\_p1, lineA\_p2, lineB\_p1, lineB\_p2*)

Takes 2 lines (as 4 vectors) and returns a vector for their point of intersection or None.

**Parameters**

- **lineA\_p1** (`mathutils.Vector`) – First point of the first line
- **lineA\_p2** (`mathutils.Vector`) – Second point of the first line
- **lineB\_p1** (`mathutils.Vector`) – First point of the second line
- **lineB\_p2** (`mathutils.Vector`) – Second point of the second line

**Returns** The point of intersection or None when not found

**Return type** `mathutils.Vector` or None

`mathutils.geometry.intersect_line_plane` (*line\_a, line\_b, plane\_co, plane\_no, no\_flip=False*)

Takes 2 lines (as 4 vectors) and returns a vector for their point of intersection or None.

**Parameters**

- **line\_a** (`mathutils.Vector`) – First point of the first line
- **line\_b** (`mathutils.Vector`) – Second point of the first line
- **plane\_co** (`mathutils.Vector`) – A point on the plane
- **plane\_no** (`mathutils.Vector`) – The direction the plane is facing
- **no\_flip** (`:boolean`) – Always return an intersection on the direction defined by line\_a -> line\_b

**Returns** The point of intersection or None when not found

**Return type** `mathutils.Vector` or None

`mathutils.geometry.intersect_line_sphere` (*line\_a*, *line\_b*, *sphere\_co*, *sphere\_radius*,  
*clip=True*)

Takes a lines (as 2 vectors), a sphere as a point and a radius and returns the intersection

**Parameters**

- **line\_a** (`mathutils.Vector`) – First point of the first line
- **line\_b** (`mathutils.Vector`) – Second point of the first line
- **sphere\_co** (`mathutils.Vector`) – The center of the sphere
- **sphere\_radius** (*sphere\_radius*) – Radius of the sphere

**Returns** The intersection points as a pair of vectors or None when there is no intersection

**Return type** A tuple pair containing `mathutils.Vector` or None

`mathutils.geometry.intersect_line_sphere_2d` (*line\_a*, *line\_b*, *sphere\_co*, *sphere\_radius*,  
*clip=True*)

Takes a lines (as 2 vectors), a sphere as a point and a radius and returns the intersection

**Parameters**

- **line\_a** (`mathutils.Vector`) – First point of the first line
- **line\_b** (`mathutils.Vector`) – Second point of the first line
- **sphere\_co** (`mathutils.Vector`) – The center of the sphere
- **sphere\_radius** (*sphere\_radius*) – Radius of the sphere

**Returns** The intersection points as a pair of vectors or None when there is no intersection

**Return type** A tuple pair containing `mathutils.Vector` or None

`mathutils.geometry.intersect_point_line` (*pt*, *line\_p1*, *line\_p2*)

Takes a point and a line and returns a tuple with the closest point on the line and its distance from the first point of the line as a percentage of the length of the line.

**Parameters**

- **pt** (`mathutils.Vector`) – Point
- **line\_p1** (`mathutils.Vector`) – First point of the line
- **line\_p2** – Second point of the line

**Return type** (`mathutils.Vector`, float)

`mathutils.geometry.intersect_point_quad_2d` (*pt*, *quad\_p1*, *quad\_p2*, *quad\_p3*, *quad\_p4*)

Takes 5 vectors (using only the x and y coordinates): one is the point and the next 4 define the quad, only the x and y are used from the vectors. Returns 1 if the point is within the quad, otherwise 0.

**Parameters**

- **pt** – Point
- **quad\_p1** (`mathutils.Vector`) – First point of the quad
- **quad\_p2** (`mathutils.Vector`) – Second point of the quad
- **quad\_p3** (`mathutils.Vector`) – Third point of the quad
- **quad\_p4** (`mathutils.Vector`) – Forth point of the quad

**Return type** `int`

`mathutils.geometry.intersect_point_tri_2d` (*pt, tri\_p1, tri\_p2, tri\_p3*)

Takes 4 vectors (using only the x and y coordinates): one is the point and the next 3 define the triangle. Returns 1 if the point is within the triangle, otherwise 0.

**Parameters**

- **pt** – Point
- **tri\_p1** (`mathutils.Vector`) – First point of the triangle
- **tri\_p2** (`mathutils.Vector`) – Second point of the triangle
- **tri\_p3** (`mathutils.Vector`) – Third point of the triangle

**Return type** `int`

`mathutils.geometry.intersect_ray_tri` (*v1, v2, v3, ray, orig, clip=True*)

Returns the intersection between a ray and a triangle, if possible, returns `None` otherwise.

**Parameters**

- **v1** (`mathutils.Vector`) – Point1
- **v2** (`mathutils.Vector`) – Point2
- **v3** (`mathutils.Vector`) – Point3
- **ray** (`mathutils.Vector`) – Direction of the projection
- **orig** (`mathutils.Vector`) – Origin
- **clip** (*boolean*) – When `False`, don't restrict the intersection to the area of the triangle, use the infinite plane defined by the triangle.

**Returns** The point of intersection or `None` if no intersection is found

**Return type** `mathutils.Vector` or `None`

`mathutils.geometry.normal` (*v1, v2, v3, v4=None*)

Returns the normal of the 3D tri or quad.

**Parameters**

- **v1** (`mathutils.Vector`) – Point1
- **v2** (`mathutils.Vector`) – Point2
- **v3** (`mathutils.Vector`) – Point3
- **v4** (`mathutils.Vector`) – Point4 (optional)

**Return type** `mathutils.Vector`

`mathutils.geometry.tessellate_polygon` (*veclist\_list*)

Takes a list of polylines (each point a vector) and returns the point indices for a polyline filled with triangles.

**Parameters** `veclist_list` – list of polylines

**Return type** `list`

### 3.3 OpenGL Wrapper (bgl)

This module wraps OpenGL constants and functions, making them available from within Blender Python.

The complete list can be retrieved from the module itself, by listing its contents: `dir(bgl)`. A simple search on the net can point to more than enough material to teach OpenGL programming, from books to many collections of tutorials.

The “red book”: “I{OpenGL Programming Guide: The Official Guide to Learning OpenGL}” and the online NeHe tutorials are two of the best resources.

---

**Note:** You can use the `Image` type to load and set textures. See `Image.gl_load` and `Image.gl_load`, for example. [OpenGL.org NeHe GameDev](#)

---

**glAccum(op, value):**

Operate on the accumulation buffer.

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **op** (*Enumerated constant*) – The accumulation buffer operation.
- **value** (*float*) – a value used in the accumulation buffer operation.

**glAlphaFunc(func, ref):**

Specify the alpha test function.

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **func** (*Enumerated constant*) – Specifies the alpha comparison function.
- **ref** (*float*) – The reference value that incoming alpha values are compared to. Clamped between 0 and 1.

**glAreTexturesResident(n, textures, residences):**

Determine if textures are loaded in texture memory

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **n** (*int*) – Specifies the number of textures to be queried.
- **textures** (`bgl.Buffer` object `I{type GL_INT}`) – Specifies an array containing the names of the textures to be queried
- **residences** (`bgl.Buffer` object `I{type GL_INT}(boolean)`) – An array in which the texture residence status is returned. The residence status of a texture named by an element of textures is returned in the corresponding element of residences.



**glBegin(mode) :**

Delimit the vertices of a primitive or a group of like primitives

**See Also:**

[OpenGL Docs](#)

**Parameters mode** (*Enumerated constant*) – Specifies the primitive that will be create from vertices between glBegin and glEnd.

**glBindTexture(target, texture) :**

Bind a named texture to a texturing target

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **target** (*Enumerated constant*) – Specifies the target to which the texture is bound.
- **texture** (*unsigned int*) – Specifies the name of a texture.

**glBitmap(width, height, xorig, yorig, xmove, ymove, bitmap) :**

Draw a bitmap

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **height** (*width,*) – Specify the pixel width and height of the bitmap image.
- **yorig** (*xorig,*) – Specify the location of the origin in the bitmap image. The origin is measured from the lower left corner of the bitmap, with right and up being the positive axes.
- **ymove** (*xmove,*) – Specify the x and y offsets to be added to the current raster position after the bitmap is drawn.
- **bitmap** (*bgl.Buffer object I{type GL\_BYTE}*) – Specifies the address of the bitmap image.

**glBlendFunc(sfactor, dfactor) :**

Specify pixel arithmetic

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **sfactor** (*Enumerated constant*) – Specifies how the red, green, blue, and alpha source blending factors are computed.
- **dfactor** (*Enumerated constant*) – Specifies how the red, green, blue, and alpha destination blending factors are computed.

**glCallList(list) :**

Execute a display list

**See Also:**

[OpenGL Docs](#)

**Parameters** **list** (*unsigned int*) – Specifies the integer name of the display list to be executed.

**glCallLists**(**n**, **type**, **lists**):

Execute a list of display lists

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **n** (*int*) – Specifies the number of display lists to be executed.
- **type** (*Enumerated constant*) – Specifies the type of values in lists.
- **lists** (*ogl.Buffer object*) – Specifies the address of an array of name offsets in the display list. The pointer type is void because the offsets can be bytes, shorts, ints, or floats, depending on the value of type.

**glClear**(**mask**):

Clear buffers to preset values

**See Also:**

[OpenGL Docs](#)

**Parameters** **mask** (*Enumerated constant(s)*) – Bitwise OR of masks that indicate the buffers to be cleared.

**glClearAccum**(**red**, **green**, **blue**, **alpha**):

Specify clear values for the accumulation buffer

**See Also:**

[OpenGL Docs](#)

**Parameters** **green**, **blue**, **alpha** (*red*,) – Specify the red, green, blue, and alpha values used when the accumulation buffer is cleared. The initial values are all 0.

**glClearColor**(**red**, **green**, **blue**, **alpha**):

Specify clear values for the color buffers

**See Also:**

[OpenGL Docs](#)

**Parameters** **green**, **blue**, **alpha** (*red*,) – Specify the red, green, blue, and alpha values used when the color buffers are cleared. The initial values are all 0.

**glClearDepth**(**depth**):

Specify the clear value for the depth buffer

**See Also:**

[OpenGL Docs](#)

**Parameters** **depth** (*int*) – Specifies the depth value used when the depth buffer is cleared. The initial value is 1.

**glClearIndex**(**c**):

Specify the clear value for the color index buffers

**See Also:**

[OpenGL Docs](#)

**Parameters** *c* (*float*) – Specifies the index used when the color index buffers are cleared. The initial value is 0.

**glClearStencil (s) :**

Specify the clear value for the stencil buffer

**See Also:**[OpenGL Docs](#)

**Parameters** *s* (*int*) – Specifies the index used when the stencil buffer is cleared. The initial value is 0.

**glClipPlane (plane, equation) :**

Specify a plane against which all geometry is clipped

**See Also:**[OpenGL Docs](#)**Parameters**

- **plane** (*Enumerated constant*) – Specifies which clipping plane is being positioned.
- **equation** (`bg1.Buffer object I{type GL_FLOAT}(double)`) – Specifies the address of an array of four double-precision floating-point values. These values are interpreted as a plane equation.

**glColor (red, green, blue, alpha) :**

B{glColor3b, glColor3d, glColor3f, glColor3i, glColor3s, glColor3ub, glColor3ui, glColor3us, glColor4b, glColor4d, glColor4f, glColor4i, glColor4s, glColor4ub, glColor4ui, glColor4us, glColor3bv, glColor3dv, glColor3fv, glColor3iv, glColor3sv, glColor3ubv, glColor3uiv, glColor3usv, glColor4bv, glColor4dv, glColor4fv, glColor4iv, glColor4sv, glColor4ubv, glColor4uiv, glColor4usv }

Set a new color.

**See Also:**[OpenGL Docs](#)**Parameters**

- **green, blue** (*red,*) – Specify new red, green, and blue values for the current color.
- **alpha** – Specifies a new alpha value for the current color. Included only in the four-argument glColor4 commands. (With '4' colors only)

**glColorMask (red, green, blue, alpha) :**

Enable and disable writing of frame buffer color components

**See Also:**[OpenGL Docs](#)

**Parameters** **green, blue, alpha** (*red,*) – Specify whether red, green, blue, and alpha can or cannot be written into the frame buffer. The initial values are all GL\_TRUE, indicating that the color components can be written.

**glColorMaterial (face, mode) :**

Cause a material color to track the current color

See Also:

[OpenGL Docs](#)

**Parameters**

- **face** (*Enumerated constant*) – Specifies whether front, back, or both front and back material parameters should track the current color.
- **mode** (*Enumerated constant*) – Specifies which of several material parameters track the current color.

**glCopyPixels(x, y, width, height, type):**

Copy pixels in the frame buffer

See Also:

[OpenGL Docs](#)

**Parameters**

- **y (x)** – Specify the window coordinates of the lower left corner of the rectangular region of pixels to be copied.
- **width,height** – Specify the dimensions of the rectangular region of pixels to be copied. Both must be non-negative.
- **type** (*Enumerated constant*) – Specifies whether color values, depth values, or stencil values are to be copied.

def glCopyTexImage2D(target, level, internalformat, x, y, width, height, border):

Copy pixels into a 2D texture image

See Also:

[OpenGL Docs](#)

**Parameters**

- **target** (*Enumerated constant*) – Specifies the target texture.
- **level** (*int*) – Specifies the level-of-detail number. Level 0 is the base image level. Level n is the nth mipmap reduction image.
- **internalformat** (*int*) – Specifies the number of color components in the texture.
- **y (x)** – Specify the window coordinates of the first pixel that is copied from the frame buffer. This location is the lower left corner of a rectangular block of pixels.
- **width** (*int*) – Specifies the width of the texture image. Must be  $2n+2(\text{border})$  for some integer n. All implementations support texture images that are at least 64 texels wide.
- **height** (*int*) – Specifies the height of the texture image. Must be  $2m+2(\text{border})$  for some integer m. All implementations support texture images that are at least 64 texels high.
- **border** (*int*) – Specifies the width of the border. Must be either 0 or 1.

**glCullFace(mode) :**

Specify whether front- or back-facing facets can be culled

See Also:

[OpenGL Docs](#)

**Parameters** **mode** (*Enumerated constant*) – Specifies whether front- or back-facing facets are candidates for culling.

**glDeleteLists(list, range) :**

Delete a contiguous group of display lists

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **list** (*unsigned int*) – Specifies the integer name of the first display list to delete
- **range** (*int*) – Specifies the number of display lists to delete

**glDeleteTextures(n, textures) :**

Delete named textures

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **n** (*int*) – Specifies the number of textures to be deleted
- **textures** (`ogl.Buffer I{GL_INT}`) – Specifies an array of textures to be deleted

**glDepthFunc(func) :**

Specify the value used for depth buffer comparisons

**See Also:**

[OpenGL Docs](#)

**Parameters** **func** (*Enumerated constant*) – Specifies the depth comparison function.

**glDepthMask(flag) :**

Enable or disable writing into the depth buffer

**See Also:**

[OpenGL Docs](#)

**Parameters** **flag** (*int (boolean)*) – Specifies whether the depth buffer is enabled for writing. If flag is `GL_FALSE`, depth buffer writing is disabled. Otherwise, it is enabled. Initially, depth buffer writing is enabled.

**glDepthRange(zNear, zFar) :**

Specify mapping of depth values from normalized device coordinates to window coordinates

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **zNear** (*int*) – Specifies the mapping of the near clipping plane to window coordinates. The initial value is 0.
- **zFar** (*int*) – Specifies the mapping of the far clipping plane to window coordinates. The initial value is 1.

**glDisable (cap) :**

Disable server-side GL capabilities

**See Also:**

[OpenGL Docs](#)

**Parameters cap** (*Enumerated constant*) – Specifies a symbolic constant indicating a GL capability.

**glDrawBuffer (mode) :**

Specify which color buffers are to be drawn into

**See Also:**

[OpenGL Docs](#)

**Parameters mode** (*Enumerated constant*) – Specifies up to four color buffers to be drawn into.

**glDrawPixels (width, height, format, type, pixels) :**

Write a block of pixels to the frame buffer

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **height** (*width*,) – Specify the dimensions of the pixel rectangle to be written into the frame buffer.
- **format** (*Enumerated constant*) – Specifies the format of the pixel data.
- **type** (*Enumerated constant*) – Specifies the data type for pixels.
- **pixels** (`bg1.Buffer` object) – Specifies a pointer to the pixel data.

**glEdgeFlag (flag) :**

B{glEdgeFlag, glEdgeFlagv}

Flag edges as either boundary or non-boundary

**See Also:**

[OpenGL Docs](#)

**Parameters flag** (*Depends of function prototype*) – Specifies the current edge flag value. The initial value is `GL_TRUE`.

**glEnable (cap) :**

Enable server-side GL capabilities

**See Also:**

[OpenGL Docs](#)

**Parameters cap** (*Enumerated constant*) – Specifies a symbolic constant indicating a GL capability.

**glEnd () :**

Delimit the vertices of a primitive or group of like primitives

**See Also:**

[OpenGL Docs](#)

**glEndList () :**

Create or replace a display list

**See Also:**

[OpenGL Docs](#)

**glEvalCoord (u, v) :**

B{glEvalCoord1d, glEvalCoord1f, glEvalCoord2d, glEvalCoord2f, glEvalCoord1dv, glEvalCoord1fv, glEvalCoord2dv, glEvalCoord2fv}

Evaluate enabled one- and two-dimensional maps

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **u** (*Depends on function prototype.*) – Specifies a value that is the domain coordinate u to the basis function defined in a previous glMap1 or glMap2 command. If the function prototype ends in ‘v’ then u specifies a pointer to an array containing either one or two domain coordinates. The first coordinate is u. The second coordinate is v, which is present only in glEvalCoord2 versions.
- **v** (*Depends on function prototype. (only with ‘2’ prototypes)*) – Specifies a value that is the domain coordinate v to the basis function defined in a previous glMap2 command. This argument is not present in a glEvalCoord1 command.

**glEvalMesh (mode, i1, i2) :**

B{glEvalMesh1 or glEvalMesh2}

Compute a one- or two-dimensional grid of points or lines

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **mode** (*Enumerated constant*) – In glEvalMesh1, specifies whether to compute a one-dimensional mesh of points or lines.
- **i2** (*i1*) – Specify the first and last integer values for the grid domain variable i.

**glEvalPoint (i, j) :**

B{glEvalPoint1 and glEvalPoint2}

Generate and evaluate a single point in a mesh

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **i** (*int*) – Specifies the integer value for grid domain variable i.
- **j** (*int (only with ‘2’ prototypes)*) – Specifies the integer value for grid domain variable j (glEvalPoint2 only).

**glFeedbackBuffer (size, type, buffer) :**

Controls feedback mode

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **size** (*int*) – Specifies the maximum number of values that can be written into buffer.
- **type** (*Enumerated constant*) – Specifies a symbolic constant that describes the information that will be returned for each vertex.
- **buffer** (`ogl.Buffer` object I{GL\_FLOAT}) – Returns the feedback data.

**glFinish()** :

Block until all GL execution is complete

**See Also:**

[OpenGL Docs](#)

**glFlush()** :

Force Execution of GL commands in finite time

**See Also:**

[OpenGL Docs](#)

**glFog (pname, param) :**

B{glFogf, glFogi, glFogfv, glFogiv}

Specify fog parameters

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **pname** (*Enumerated constant*) – Specifies a single-valued fog parameter. If the function prototype ends in ‘v’ specifies a fog parameter.
- **param** (*Depends on function prototype.*) – Specifies the value or values to be assigned to pname. GL\_FOG\_COLOR requires an array of four values. All other parameters accept an array containing only a single value.

**glFrontFace(mode) :**

Define front- and back-facing polygons

**See Also:**

[OpenGL Docs](#)

**Parameters mode** (*Enumerated constant*) – Specifies the orientation of front-facing polygons.

**glFrustum(left, right, bottom, top, zNear, zFar) :**

Multiply the current matrix by a perspective matrix

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **right** (*left,*) – Specify the coordinates for the left and right vertical clipping planes.
- **bottom** (*top,*) – Specify the coordinates for the bottom and top horizontal clipping planes.



- **zFar** (*zNear*) – Specify the distances to the near and far depth clipping planes. Both distances must be positive.

**glGenLists (range) :**

Generate a contiguous set of empty display lists

**See Also:**

[OpenGL Docs](#)

**Parameters** **range** (*int*) – Specifies the number of contiguous empty display lists to be generated.

**glGenTextures (n, textures) :**

Generate texture names

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **n** (*int*) – Specifies the number of textures name to be generated.
- **textures** (`bgl.Buffer` object I{type `GL_INT`}) – Specifies an array in which the generated textures names are stored.

**glGet (pname, param) :**

B{`glGetBooleanv`, `glGetfloatv`, `glGetFloatv`, `glGetIntegerv`}

Return the value or values of a selected parameter

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **pname** (*Enumerated constant*) – Specifies the parameter value to be returned.
- **param** (*Depends on function prototype.*) – Returns the value or values of the specified parameter.

**glGetClipPlane (plane, equation) :**

Return the coefficients of the specified clipping plane

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **plane** (*Enumerated constant*) – Specifies a clipping plane. The number of clipping planes depends on the implementation, but at least six clipping planes are supported. They are identified by symbolic names of the form `GL_CLIP_PLANEi` where  $0 < i < \text{GL\_MAX\_CLIP\_PLANES}$ .
- **equation** (`bgl.Buffer` object I{type `GL_FLOAT`}) – Returns four float (double)-precision values that are the coefficients of the plane equation of plane in eye coordinates. The initial value is (0, 0, 0, 0).

**glGetError () :**

Return error information

**See Also:**

[OpenGL Docs](#)

**glGetLight (light, pname, params):**

B{glGetLightfv and glGetLightiv}

Return light source parameter values

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **light** (*Enumerated constant*) – Specifies a light source. The number of possible lights depends on the implementation, but at least eight lights are supported. They are identified by symbolic names of the form `GL_LIGHTi` where  $0 < i < GL\_MAX\_LIGHTS$ .
- **pname** (*Enumerated constant*) – Specifies a light source parameter for light.
- **params** (`bgl.Buffer` object. Depends on function prototype.) – Returns the requested data.

**glGetMap (target, query, v):**

B{glGetMapdv, glGetMapfv, glGetMapiv}

Return evaluator parameters

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **target** (*Enumerated constant*) – Specifies the symbolic name of a map.
- **query** (*Enumerated constant*) – Specifies which parameter to return.
- **v** (`bgl.Buffer` object. Depends on function prototype.) – Returns the requested data.

**glGetMaterial (face, pname, params):**

B{glGetMaterialfv, glGetMaterialiv}

Return material parameters

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **face** (*Enumerated constant*) – Specifies which of the two materials is being queried. representing the front and back materials, respectively.
- **pname** (*Enumerated constant*) – Specifies the material parameter to return.
- **params** (`bgl.Buffer` object. Depends on function prototype.) – Returns the requested data.

**glGetPixelMap (map, values):**

B{glGetPixelMapfv, glGetPixelMapuiv, glGetPixelMapusv}

Return the specified pixel map

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **map** (*Enumerated constant*) – Specifies the name of the pixel map to return.
- **values** (`bgl.Buffer` object. Depends on function prototype.) – Returns the pixel map contents.

**glGetPolygonStipple (mask) :**

Return the polygon stipple pattern

**See Also:**

[OpenGL Docs](#)

**Parameters** **mask** (`bgl.Buffer` object I{type `GL_BYTE`}) – Returns the stipple pattern. The initial value is all 1's.

**glGetString (name) :**

Return a string describing the current GL connection

**See Also:**

[OpenGL Docs](#)

**Parameters** **name** (*Enumerated constant*) – Specifies a symbolic constant.

**glGetTexEnv (target, pname, params) :**

`B{glGetTexEnvfv, glGetTexEnviv}`

Return texture environment parameters

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **target** (*Enumerated constant*) – Specifies a texture environment. Must be `GL_TEXTURE_ENV`.
- **pname** (*Enumerated constant*) – Specifies the symbolic name of a texture environment parameter.
- **params** (`bgl.Buffer` object. Depends on function prototype.) – Returns the requested data.

**glGetTexGen (coord, pname, params) :**

`B{glGetTexGendv, glGetTexGenfv, glGetTexGeniv}`

Return texture coordinate generation parameters

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **coord** (*Enumerated constant*) – Specifies a texture coordinate.
- **pname** (*Enumerated constant*) – Specifies the symbolic name of the value(s) to be returned.
- **params** (`bgl.Buffer` object. Depends on function prototype.) – Returns the requested data.

**glGetTexImage(target, level, format, type, pixels):**

Return a texture image

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **target** (*Enumerated constant*) – Specifies which texture is to be obtained.
- **level** (*int*) – Specifies the level-of-detail number of the desired image. Level 0 is the base image level. Level n is the nth mipmap reduction image.
- **format** (*Enumerated constant*) – Specifies a pixel format for the returned data.
- **type** (*Enumerated constant*) – Specifies a pixel type for the returned data.
- **pixels** (*OpenGL.Buffer object.*) – Returns the texture image. Should be a pointer to an array of the type specified by type

**glGetTexLevelParameter(target, level, pname, params):**

B{glGetTexLevelParameterfv, glGetTexLevelParameteriv}

return texture parameter values for a specific level of detail

**See Also:**

U{[opengl.org/developers/documentation/man\\_pages/hardcopy/GL/html/gl/gettexlevelparameter.html](http://opengl.org/developers/documentation/man_pages/hardcopy/GL/html/gl/gettexlevelparameter.html)>‘\_

**Parameters**

- **target** (*Enumerated constant*) – Specifies the symbolic name of the target texture.
- **level** (*int*) – Specifies the level-of-detail number of the desired image. Level 0 is the base image level. Level n is the nth mipmap reduction image.
- **pname** (*Enumerated constant*) – Specifies the symbolic name of a texture parameter.
- **params** (*OpenGL.Buffer object. Depends on function prototype.*) – Returns the requested data.

**glGetTexParameter(target, pname, params):**

B{glGetTexParameterfv, glGetTexParameteriv}

Return texture parameter values

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **target** (*Enumerated constant*) – Specifies the symbolic name of the target texture.
- **pname** (*Enumerated constant*) – Specifies the symbolic name the target texture.
- **params** (*OpenGL.Buffer object. Depends on function prototype.*) – Returns the texture parameters.

**glHint(target, mode):**

Specify implementation-specific hints

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **target** (*Enumerated constant*) – Specifies a symbolic constant indicating the behavior to be controlled.
- **mode** (*Enumerated constant*) – Specifies a symbolic constant indicating the desired behavior.

**glIndex (c) :**

B{glIndexd, glIndexf, glIndexi, glIndexs, glIndexdv, glIndexfv, glIndexiv, glIndexsv}

Set the current color index

**See Also:**

[OpenGL Docs](#)

**Parameters** **c** (*bgl.Buffer* object. Depends on function prototype.) – Specifies a pointer to a one element array that contains the new value for the current color index.

**glInitNames () :**

Initialize the name stack

**See Also:**

[OpenGL Docs](#)

**glIsEnabled (cap) :**

Test whether a capability is enabled

**See Also:**

[OpenGL Docs](#)

**Parameters** **cap** (*Enumerated constant*) – Specifies a constant representing a GL capability.

**glIsList (list) :**

Determine if a name corresponds to a display-list

**See Also:**

[OpenGL Docs](#)

**Parameters** **list** (*unsigned int*) – Specifies a potential display-list name.

**glIsTexture (texture) :**

Determine if a name corresponds to a texture

**See Also:**

[OpenGL Docs](#)

**Parameters** **texture** (*unsigned int*) – Specifies a value that may be the name of a texture.

**glLight (light, pname, param) :**

B{glLightf, glLighti, glLightfv, glLightiv}

Set the light source parameters

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **light** (*Enumerated constant*) – Specifies a light. The number of lights depends on the implementation, but at least eight lights are supported. They are identified by symbolic names of the form `GL_LIGHTi` where  $0 < i < \text{GL\_MAX\_LIGHTS}$ .
- **pname** (*Enumerated constant*) – Specifies a single-valued light source parameter for light.
- **param** (*Depends on function prototype.*) – Specifies the value that parameter `pname` of light source light will be set to. If function prototype ends in ‘v’ specifies a pointer to the value or values that parameter `pname` of light source light will be set to.

**glLightModel (pname, param) :**

B{glLightModelf, glLightModeli, glLightModelfv, glLightModeliv}

Set the lighting model parameters

**See Also:**[OpenGL Docs](#)**Parameters**

- **pname** (*Enumerated constant*) – Specifies a single-value light model parameter.
- **param** (*Depends on function prototype.*) – Specifies the value that `param` will be set to. If function prototype ends in ‘v’ specifies a pointer to the value or values that `param` will be set to.

**glLineStipple(factor, pattern) :**

Specify the line stipple pattern

**See Also:**[OpenGL Docs](#)**Parameters**

- **factor** (*int*) – Specifies a multiplier for each bit in the line stipple pattern. If `factor` is 3, for example, each bit in the pattern is used three times before the next bit in the pattern is used. `factor` is clamped to the range `[1, 256]` and defaults to 1.
- **pattern** (*unsigned short int*) – Specifies a 16-bit integer whose bit pattern determines which fragments of a line will be drawn when the line is rasterized. Bit zero is used first; the default pattern is all 1’s.

**glLineWidth(width) :**

Specify the width of rasterized lines.

**See Also:**[OpenGL Docs](#)**Parameters** `width` (*float*) – Specifies the width of rasterized lines. The initial value is 1.**glListBase(base) :**Set the display-list base for `glCallLists`**See Also:**[OpenGL Docs](#)**Parameters** `base` (*unsigned int*) – Specifies an integer offset that will be added to `glCallLists` offsets to generate display-list names. The initial value is 0.

**glLoadIdentity() :**

Replace the current matrix with the identity matrix

**See Also:**

[OpenGL Docs](#)

**glLoadMatrix (m) :**

B{glLoadMatrixd, glLoadMatixf}

Replace the current matrix with the specified matrix

**See Also:**

[OpenGL Docs](#)

**Parameters** *m* (*bg1.Buffer* object. Depends on function prototype.) – Specifies a pointer to 16 consecutive values, which are used as the elements of a 4x4 column-major matrix.

**glLoadName (name) :**

Load a name onto the name stack.

**See Also:**

[OpenGL Docs](#)

**Parameters** *name* (*unsigned int*) – Specifies a name that will replace the top value on the name stack.

**glLogicOp (opcode) :**

Specify a logical pixel operation for color index rendering

**See Also:**

[OpenGL Docs](#)

**Parameters** *opcode* (*Enumerated constant*) – Specifies a symbolic constant that selects a logical operation.

**glMap1 (target, u1, u2, stride, order, points) :**

B{glMap1d, glMap1f}

Define a one-dimensional evaluator

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **target** (*Enumerated constant*) – Specifies the kind of values that are generated by the evaluator.
- **u1,u2** – Specify a linear mapping of u, as presented to glEvalCoord1, to ^, t he variable that is evaluated by the equations specified by this command.
- **stride** (*int*) – Specifies the number of floats or float (double)s between the beginning of one control point and the beginning of the next one in the data structure referenced in points. This allows control points to be embedded in arbitrary data structures. The only constraint is that the values for a particular control point must occupy contiguous memory locations.
- **order** (*int*) – Specifies the number of control points. Must be positive.

- **points** (`bg1.Buffer` object. Depends on function prototype.) – Specifies a pointer to the array of control points.

**glMap2** (*target*, *u1*, *u2*, *ustride*, *uorder*, *v1*, *v2*, *vstride*, *vorder*, *points*):  
B{glMap2d, glMap2f}

Define a two-dimensional evaluator

**See Also:**

[OpenGL Docs](#)

#### Parameters

- **target** (*Enumerated constant*) – Specifies the kind of values that are generated by the evaluator.
- **u1,u2** – Specify a linear mapping of *u*, as presented to `glEvalCoord2`, to  $\wedge$ , the variable that is evaluated by the equations specified by this command. Initially *u1* is 0 and *u2* is 1.
- **ustride** (*int*) – Specifies the number of floats or float (double)s between the beginning of control point *R* and the beginning of control point *R<sub>ij</sub>*, where *i* and *j* are the *u* and *v* control point indices, respectively. This allows control points to be embedded in arbitrary data structures. The only constraint is that the values for a particular control point must occupy contiguous memory locations. The initial value of *ustride* is 0.
- **uorder** (*int*) – Specifies the dimension of the control point array in the *u* axis. Must be positive. The initial value is 1.
- **v2** (*v1*,) – Specify a linear mapping of *v*, as presented to `glEvalCoord2`, to  $\wedge$ , one of the two variables that are evaluated by the equations specified by this command. Initially, *v1* is 0 and *v2* is 1.
- **vstride** (*int*) – Specifies the number of floats or float (double)s between the beginning of control point *R* and the beginning of control point *R<sub>ij</sub>*, where *i* and *j* are the *u* and *v* control point (indices, respectively). This allows control points to be embedded in arbitrary data structures. The only constraint is that the values for a particular control point must occupy contiguous memory locations. The initial value of *vstride* is 0.
- **vorder** (*int*) – Specifies the dimension of the control point array in the *v* axis. Must be positive. The initial value is 1.
- **points** (`bg1.Buffer` object. Depends on function prototype.) – Specifies a pointer to the array of control points.

**glMapGrid** (*un*, *u1*, *u2*, *vn*, *v1*, *v2*):  
B{glMapGrid1d, glMapGrid1f, glMapGrid2d, glMapGrid2f}

Define a one- or two-dimensional mesh

**See Also:**

[OpenGL Docs](#)

#### Parameters

- **un** (*int*) – Specifies the number of partitions in the grid range interval [*u1*, *u2*]. Must be positive.
- **u2** (*u1*,) – Specify the mappings for integer grid domain values *i*=0 and *i*=*un*.
- **vn** (*int*) – Specifies the number of partitions in the grid range interval [*v1*, *v2*] (`glMapGrid2` only).



- **v2** (*v1*,) – Specify the mappings for integer grid domain values  $j=0$  and  $j=vn$  (`glMapGrid2` only).

**glMaterial (face, pname, params) :**

Specify material parameters for the lighting model.

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **face** (*Enumerated constant*) – Specifies which face or faces are being updated. Must be one of:
- **pname** (*Enumerated constant*) – Specifies the single-valued material parameter of the face or faces that is being updated. Must be `GL_SHININESS`.
- **params** (*int*) – Specifies the value that parameter `GL_SHININESS` will be set to. If function prototype ends in ‘v’ specifies a pointer to the value or values that `pname` will be set to.

**glMatrixMode (mode) :**

Specify which matrix is the current matrix.

**See Also:**

[OpenGL Docs](#)

**Parameters mode** (*Enumerated constant*) – Specifies which matrix stack is the target for subsequent matrix operations.

**glMultMatrix (m) :**

`B{glMultMatrixd, glMultMatrixf}`

Multiply the current matrix with the specified matrix

**See Also:**

[OpenGL Docs](#)

**Parameters m** (`bg1.Buffer` object. Depends on function prototype.) – Points to 16 consecutive values that are used as the elements of a 4x4 column major matrix.

**glNewList (list, mode) :**

Create or replace a display list

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **list** (*unsigned int*) – Specifies the display list name
- **mode** (*Enumerated constant*) – Specifies the compilation mode.

**glNormal3 (nx, ny, nz, v) :**

`B{Normal3b, Normal3bv, Normal3d, Normal3dv, Normal3f, Normal3fv, Normal3i, Normal3iv, Normal3s, Normal3sv}`

Set the current normal vector

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **ny, nz** (*nx*) – Specify the x, y, and z coordinates of the new current normal. The initial value of the current normal is the unit vector, (0, 0, 1).
- **v** (`ogl.Buffer` object. Depends on function prototype. ('v' prototypes)) – Specifies a pointer to an array of three elements: the x, y, and z coordinates of the new current normal.

**glOrtho(left, right, bottom, top, zNear, zFar):**

Multiply the current matrix with an orthographic matrix

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **right** (*left*) – Specify the coordinates for the left and right vertical clipping planes.
- **top** (*bottom*) – Specify the coordinates for the bottom and top horizontal clipping planes.
- **zFar** (*zNear*) – Specify the distances to the nearer and farther depth clipping planes. These values are negative if the plane is to be behind the viewer.

**glPassThrough(token) :**

Place a marker in the feedback buffer

**See Also:**

[OpenGL Docs](#)

**Parameters** **token** (*float*) – Specifies a marker value to be placed in the feedback buffer following a `GL_PASS_THROUGH_TOKEN`.

**glPixelMap(map, mapsize, values):**

`B{glPixelMapfv, glPixelMapuiv, glPixelMapusv}`

Set up pixel transfer maps

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **map** (*Enumerated constant*) – Specifies a symbolic map name.
- **mapsize** (*int*) – Specifies the size of the map being defined.
- **values** (`ogl.Buffer` object. Depends on function prototype.) – Specifies an array of `mapsize` values.

**glPixelStore(pname, param):**

`B{glPixelStoref, glPixelStorei}`

Set pixel storage modes

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **pname** (*Enumerated constant*) – Specifies the symbolic name of the parameter to be set. Six values affect the packing of pixel data into memory. Six more affect the unpacking of pixel data from memory.
- **param** (*Depends on function prototype.*) – Specifies the value that pname is set to.

**glPixelTransfer (pname, param) :**

B{glPixelTransferf, glPixelTransferi}

Set pixel transfer modes

**See Also:**[OpenGL Docs](#)**Parameters**

- **pname** (*Enumerated constant*) – Specifies the symbolic name of the pixel transfer parameter to be set.
- **param** (*Depends on function prototype.*) – Specifies the value that pname is set to.

**glPixelZoom(xfactor, yfactor) :**

Specify the pixel zoom factors

**See Also:**[OpenGL Docs](#)**Parameters** **yfactor** (*xfactor*,) – Specify the x and y zoom factors for pixel write operations.**glPointSize(size) :**

Specify the diameter of rasterized points

**See Also:**[OpenGL Docs](#)**Parameters** **size** (*float*) – Specifies the diameter of rasterized points. The initial value is 1.**glPolygonMode(face, mode) :**

Select a polygon rasterization mode

**See Also:**[OpenGL Docs](#)**Parameters**

- **face** (*Enumerated constant*) – Specifies the polygons that mode applies to. Must be GL\_FRONT for front-facing polygons, GL\_BACK for back-facing polygons, or GL\_FRONT\_AND\_BACK for front- and back-facing polygons.
- **mode** (*Enumerated constant*) – Specifies how polygons will be rasterized. The initial value is GL\_FILL for both front- and back-facing polygons.

**glPolygonOffset(factor, units) :**

Set the scale and units used to calculate depth values

**See Also:**[OpenGL Docs](#)**Parameters**

- **factor** (*float*) – Specifies a scale factor that is used to create a variable depth offset for each polygon. The initial value is 0.
- **units** (*float*) – Is multiplied by an implementation-specific value to create a constant depth offset. The initial value is 0.

**glPolygonStipple (mask) :**

Set the polygon stippling pattern

**See Also:**

[OpenGL Docs](#)

**Parameters** **mask** (`ogl.Buffer` object I{type `GL_BYTE`}) – Specifies a pointer to a 32x32 stipple pattern that will be unpacked from memory in the same way that `glDrawPixels` unpacks pixels.

**glPopAttrib () :**

Pop the server attribute stack

**See Also:**

[OpenGL Docs](#)

**glPopClientAttrib () :**

Pop the client attribute stack

**See Also:**

[OpenGL Docs](#)

**glPopMatrix () :**

Pop the current matrix stack

**See Also:**

[OpenGL Docs](#)

**glPopName () :**

Pop the name stack

**See Also:**

[OpenGL Docs](#)

**glPrioritizeTextures (n, textures, priorities) :**

Set texture residence priority

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **n** (*int*) – Specifies the number of textures to be prioritized.
- **textures** (`ogl.Buffer` I{type `GL_INT`}) – Specifies an array containing the names of the textures to be prioritized.
- **priorities** (`ogl.Buffer` I{type `GL_FLOAT`}) – Specifies an array containing the texture priorities. A priority given in an element of `priorities` applies to the texture named by the corresponding element of `textures`.

**glPushAttrib (mask) :**

Push the server attribute stack

**See Also:**

[OpenGL Docs](#)

**Parameters mask** (*Enumerated constant(s)*) – Specifies a mask that indicates which attributes to save.

**glPushClientAttrib (mask) :**

Push the client attribute stack

**See Also:**

[OpenGL Docs](#)

**Parameters mask** (*Enumerated constant(s)*) – Specifies a mask that indicates which attributes to save.

**glPushMatrix () :**

Push the current matrix stack

**See Also:**

[OpenGL Docs](#)

**glPushName (name) :**

Push the name stack

**See Also:**

[OpenGL Docs](#)

**Parameters name** (*unsigned int*) – Specifies a name that will be pushed onto the name stack.

**glRasterPos (x, y, z, w) :**

B{glRasterPos2d, glRasterPos2f, glRasterPos2i, glRasterPos2s, glRasterPos3d, glRasterPos3f, glRasterPos3i, glRasterPos3s, glRasterPos4d, glRasterPos4f, glRasterPos4i, glRasterPos4s, glRasterPos2dv, glRasterPos2fv, glRasterPos2iv, glRasterPos2sv, glRasterPos3dv, glRasterPos3fv, glRasterPos3iv, glRasterPos3sv, glRasterPos4dv, glRasterPos4fv, glRasterPos4iv, glRasterPos4sv }

Specify the raster position for pixel operations

**See Also:**

[OpenGL Docs](#)

**Parameters y, z, w (x,)** – Specify the x,y,z, and w object coordinates (if present) for the raster position. If function prototype ends in ‘v’ specifies a pointer to an array of two, three, or four elements, specifying x, y, z, and w coordinates, respectively.

---

**Note:** If you are drawing to the 3d view with a Scriptlink of a space handler the zoom level of the panels will scale the glRasterPos by the view matrix. so a X of 10 will not always offset 10 pixels as you would expect.

To work around this get the scale value of the view matrix and use it to scale your pixel values.

```
import bgl
xval, yval= 100, 40
# Get the scale of the view matrix
view_matrix = bgl.Buffer(bgl.GL_FLOAT, 16)
bgl.glGetFloatv(bgl.GL_MODELVIEW_MATRIX, view_matrix)
```

```
f = 1.0 / view_matrix[0]

# Instead of the usual glRasterPos2i(xval, yval)
bgl.glRasterPos2f(xval * f, yval * f)
```

---

**glReadBuffer(mode) :**

Select a color buffer source for pixels.

**See Also:**

[OpenGL Docs](#)

**Parameters mode** (*Enumerated constant*) – Specifies a color buffer.

**glReadPixels(x, y, width, height, format, type, pixels) :**

Read a block of pixels from the frame buffer

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **y** (*x*,) – Specify the window coordinates of the first pixel that is read from the frame buffer. This location is the lower left corner of a rectangular block of pixels.
- **height** (*width*,) – Specify the dimensions of the pixel rectangle. width and height of one correspond to a single pixel.
- **format** (*Enumerated constant*) – Specifies the format of the pixel data.
- **type** (*Enumerated constant*) – Specifies the data type of the pixel data.
- **pixels** (`bgl.Buffer` object) – Returns the pixel data.

**glRect(x1, y1, x2, y2, v1, v2) :**

B{glRectd, glRectf, glRecti, glRects, glRectdv, glRectfv, glRectiv, glRectsv}

Draw a rectangle

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **y1** (*x1*,) – Specify one vertex of a rectangle
- **y2** (*x2*,) – Specify the opposite vertex of the rectangle
- **v2** (*v1*,) – Specifies a pointer to one vertex of a rectangle and the pointer to the opposite vertex of the rectangle

**glRenderMode(mode) :**

Set rasterization mode

**See Also:**

[OpenGL Docs](#)

**Parameters mode** (*Enumerated constant*) – Specifies the rasterization mode.

**glRotate (angle, x, y, z) :**

B{glRotated, glRotatef}

Multiply the current matrix by a rotation matrix

**See Also:**[OpenGL Docs](#)**Parameters**

- **angle** (*Depends on function prototype.*) – Specifies the angle of rotation in degrees.
- **y, z** (*x*) – Specify the x, y, and z coordinates of a vector respectively.

**glScale (x, y, z) :**

B{glScaled, glScalef}

Multiply the current matrix by a general scaling matrix

**See Also:**[OpenGL Docs](#)**Parameters** **y, z** (*x*) – Specify scale factors along the x, y, and z axes, respectively.**glScissor (x, y, width, height) :**

Define the scissor box

**See Also:**[OpenGL Docs](#)**Parameters**

- **y** (*x*) – Specify the lower left corner of the scissor box. Initially (0, 0).
- **height** (*width*) – Specify the width and height of the scissor box. When a GL context is first attached to a window, width and height are set to the dimensions of that window.

**glSelectBuffer (size, buffer) :**

Establish a buffer for selection mode values

**See Also:**[OpenGL Docs](#)**Parameters**

- **size** (*int*) – Specifies the size of buffer
- **buffer** (`bg1.Buffer I{type GL_INT}`) – Returns the selection data

**glShadeModel (mode) :**

Select flat or smooth shading

**See Also:**[OpenGL Docs](#)**Parameters** **mode** (*Enumerated constant*) – Specifies a symbolic value representing a shading technique.

**glStencilFunc(func, ref, mask) :**

Set function and reference value for stencil testing

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **func** (*Enumerated constant*) – Specifies the test function.
- **ref** (*int*) – Specifies the reference value for the stencil test. ref is clamped to the range  $[0, 2^n - 1]$ , where n is the number of bitplanes in the stencil buffer. The initial value is 0.
- **mask** (*unsigned int*) – Specifies a mask that is ANDed with both the reference value and the stored stencil value when the test is done. The initial value is all 1's.

**glStencilMask(mask) :**

Control the writing of individual bits in the stencil planes

**See Also:**

[OpenGL Docs](#)

**Parameters** **mask** (*unsigned int*) – Specifies a bit mask to enable and disable writing of individual bits in the stencil planes. Initially, the mask is all 1's.

**glStencilOp(fail, zfail, zpass) :**

Set stencil test actions

**See Also:**

[OpenGL Docs](#)

**Parameters**

- **fail** (*Enumerated constant*) – Specifies the action to take when the stencil test fails. The initial value is GL\_KEEP.
- **zfail** (*Enumerated constant*) – Specifies the stencil action when the stencil test passes, but the depth test fails. zfail accepts the same symbolic constants as fail. The initial value is GL\_KEEP.
- **zpass** (*Enumerated constant*) – Specifies the stencil action when both the stencil test and the depth test pass, or when the stencil test passes and either there is no depth buffer or depth testing is not enabled. zpass accepts the same symbolic constants as fail. The initial value is GL\_KEEP.

**glTexCoord (s, t, r, q, v) :**

B{glTexCoord1d, glTexCoord1f, glTexCoord1i, glTexCoord1s, glTexCoord2d, glTexCoord2f, glTexCoord2i, glTexCoord2s, glTexCoord3d, glTexCoord3f, glTexCoord3i, glTexCoord3s, glTexCoord4d, glTexCoord4f, glTexCoord4i, glTexCoord4s, glTexCoord1dv, glTexCoord1fv, glTexCoord1iv, glTexCoord1sv, glTexCoord2dv, glTexCoord2fv, glTexCoord2iv, glTexCoord2sv, glTexCoord3dv, glTexCoord3fv, glTexCoord3iv, glTexCoord3sv, glTexCoord4dv, glTexCoord4fv, glTexCoord4iv, glTexCoord4sv}

Set the current texture coordinates

**See Also:**

[OpenGL Docs](#)

**Parameters**



- **t, r, q** (*s*,) – Specify s, t, r, and q texture coordinates. Not all parameters are present in all forms of the command.
- **v** (`ogl.Buffer` object. Depends on function prototype. (for ‘v’ prototypes only)) – Specifies a pointer to an array of one, two, three, or four elements, which in turn specify the s, t, r, and q texture coordinates.

**glTexEnv (target, pname, param):**

B{glTexEnvf, glTexEnvi, glTexEnvfv, glTexEnviv}

Set texture environment parameters

**See Also:**[OpenGL Docs](#)**Parameters**

- **target** (*Enumerated constant*) – Specifies a texture environment. Must be `GL_TEXTURE_ENV`.
- **pname** (*Enumerated constant*) – Specifies the symbolic name of a single-valued texture environment parameter. Must be `GL_TEXTURE_ENV_MODE`.
- **param** (*Depends on function prototype.*) – Specifies a single symbolic constant. If function prototype ends in ‘v’ specifies a pointer to a parameter array that contains either a single symbolic constant or an RGBA color

**glTexGen (coord, pname, param):**

B{glTexGend, glTexGenf, glTexGeni, glTexGendv, glTexGenfv, glTexGeniv}

Control the generation of texture coordinates

**See Also:**[OpenGL Docs](#)**Parameters**

- **coord** (*Enumerated constant*) – Specifies a texture coordinate.
- **pname** (*Enumerated constant*) – Specifies the symbolic name of the texture- coordinate generation function.
- **param** (*Depends on function prototype.*) – Specifies a single-valued texture generation parameter. If function prototype ends in ‘v’ specifies a pointer to an array of texture generation parameters. If pname is `GL_TEXTURE_GEN_MODE`, then the array must contain a single symbolic constant. Otherwise, param holds the coefficients for the texture-coordinate generation function specified by pname.

**glTexImage1D(target, level, internalformat, width, border, format, type, pixels):**

Specify a one-dimensional texture image

**See Also:**[OpenGL Docs](#)**Parameters**

- **target** (*Enumerated constant*) – Specifies the target texture.
- **level** (*int*) – Specifies the level-of-detail number. Level 0 is the base image level. Level n is the nth mipmap reduction image.

- **internalformat** (*int*) – Specifies the number of color components in the texture.
- **width** (*int*) – Specifies the width of the texture image. Must be  $2n+2(\text{border})$  for some integer  $n$ . All implementations support texture images that are at least 64 texels wide. The height of the 1D texture image is 1.
- **border** (*int*) – Specifies the width of the border. Must be either 0 or 1.
- **format** (*Enumerated constant*) – Specifies the format of the pixel data.
- **type** (*Enumerated constant*) – Specifies the data type of the pixel data.
- **pixels** (`ogl.Buffer` object.) – Specifies a pointer to the image data in memory.

**glTexImage2D(target, level, internalformat, width, height, border, format, type, pixels):**  
Specify a two-dimensional texture image

**See Also:**

[OpenGL Docs](#)

#### Parameters

- **target** (*Enumerated constant*) – Specifies the target texture.
- **level** (*int*) – Specifies the level-of-detail number. Level 0 is the base image level. Level  $n$  is the  $n$ th mipmap reduction image.
- **internalformat** (*int*) – Specifies the number of color components in the texture.
- **width** (*int*) – Specifies the width of the texture image. Must be  $2n+2(\text{border})$  for some integer  $n$ . All implementations support texture images that are at least 64 texels wide.
- **height** (*int*) – Specifies the height of the texture image. Must be  $2m+2(\text{border})$  for some integer  $m$ . All implementations support texture images that are at least 64 texels high.
- **border** (*int*) – Specifies the width of the border. Must be either 0 or 1.
- **format** (*Enumerated constant*) – Specifies the format of the pixel data.
- **type** (*Enumerated constant*) – Specifies the data type of the pixel data.
- **pixels** (`ogl.Buffer` object.) – Specifies a pointer to the image data in memory.

**glTexParameter(target, pname, param):**  
`B{glTexParameterf, glTexParameteri, glTexParameterfv, glTexParameteriv}`

Set texture parameters

**See Also:**

[OpenGL Docs](#)

#### Parameters

- **target** (*Enumerated constant*) – Specifies the target texture.
- **pname** (*Enumerated constant*) – Specifies the symbolic name of a single-valued texture parameter.
- **param** (*Depends on function prototype.*) – Specifies the value of `pname`. If function prototype ends in ‘v’ specifies a pointer to an array where the value or values of `pname` are stored.

**glTranslate (x, y, z) :**

B{glTranslatef, glTranslated}

Multiply the current matrix by a translation matrix

**See Also:**[OpenGL Docs](#)**Parameters** *y, z (x,)* – Specify the x, y, and z coordinates of a translation vector.**glVertex (x, y, z, w, v) :**

B{glVertex2d, glVertex2f, glVertex2i, glVertex2s, glVertex3d, glVertex3f, glVertex3i, glVertex3s, glVertex4d, glVertex4f, glVertex4i, glVertex4s, glVertex2dv, glVertex2fv, glVertex2iv, glVertex2sv, glVertex3dv, glVertex3fv, glVertex3iv, glVertex3sv, glVertex4dv, glVertex4fv, glVertex4iv, glVertex4sv}

Specify a vertex

**See Also:**[OpenGL Docs](#)**Parameters**

- **y, z, w (x,)** – Specify x, y, z, and w coordinates of a vertex. Not all parameters are present in all forms of the command.
- **v** (`bgl.Buffer` object. Depends of function prototype (for ‘v’ prototypes only)) – Specifies a pointer to an array of two, three, or four elements. The elements of a two-element array are x and y; of a three-element array, x, y, and z; and of a four-element array, x, y, z, and w.

**glViewport (x, y, width, height) :**

Set the viewport

**See Also:**[OpenGL Docs](#)**Parameters**

- **y (x,)** – Specify the lower left corner of the viewport rectangle, in pixels. The initial value is (0,0).
- **height (width,)** – Specify the width and height of the viewport. When a GL context is first attached to a window, width and height are set to the dimensions of that window.

**gluPerspective (fovY, aspect, zNear, zFar) :**

Set up a perspective projection matrix.

**See Also:**U{[http://biology.ncsa.uiuc.edu/cgi-bin/infosrch.cgi?cmd=getdoc&coll=0650&db=bks&fname=/SGI\\_Developer/OpenGL\\_RM/ch](http://biology.ncsa.uiuc.edu/cgi-bin/infosrch.cgi?cmd=getdoc&coll=0650&db=bks&fname=/SGI_Developer/OpenGL_RM/ch)**Parameters**

- **fovY (double)** – Specifies the field of view angle, in degrees, in the y direction.
- **aspect (double)** – Specifies the aspect ratio that determines the field of view in the x direction. The aspect ratio is the ratio of x (width) to y (height).
- **zNear (double)** – Specifies the distance from the viewer to the near clipping plane (always positive).

- **zFar** (*double*) – Specifies the distance from the viewer to the far clipping plane (always positive).

**gluLookAt**(*eyex, eyey, eyez, centerx, centery, centerz, upx, upy, upz*):

Define a viewing transformation.

**See Also:**

U{[http://biology.ncsa.uiuc.edu/cgi-bin/infosrch.cgi?cmd=getdoc&coll=0650&db=bks&fname=/SGI\\_Developer/OpenGL\\_RM/ch](http://biology.ncsa.uiuc.edu/cgi-bin/infosrch.cgi?cmd=getdoc&coll=0650&db=bks&fname=/SGI_Developer/OpenGL_RM/ch)}

#### Parameters

- **eyey, eyez** (*eyex,*) – Specifies the position of the eye point.
- **centery, centerz** (*centerx,*) – Specifies the position of the reference point.
- **upy, upz** (*upx,*) – Specifies the direction of the up vector.

**gluOrtho2D**(*left, right, bottom, top*):

Define a 2-D orthographic projection matrix.

**See Also:**

U{[http://biology.ncsa.uiuc.edu/cgi-bin/infosrch.cgi?cmd=getdoc&coll=0650&db=bks&fname=/SGI\\_Developer/OpenGL\\_RM/ch](http://biology.ncsa.uiuc.edu/cgi-bin/infosrch.cgi?cmd=getdoc&coll=0650&db=bks&fname=/SGI_Developer/OpenGL_RM/ch)}

#### Parameters

- **right** (*left,*) – Specify the coordinates for the left and right vertical clipping planes.
- **top** (*bottom,*) – Specify the coordinates for the bottom and top horizontal clipping planes.

**gluPickMatrix**(*x, y, width, height, viewport*):

Define a picking region.

**See Also:**

U{[http://biology.ncsa.uiuc.edu/cgi-bin/infosrch.cgi?cmd=getdoc&coll=0650&db=bks&fname=/SGI\\_Developer/OpenGL\\_RM/ch](http://biology.ncsa.uiuc.edu/cgi-bin/infosrch.cgi?cmd=getdoc&coll=0650&db=bks&fname=/SGI_Developer/OpenGL_RM/ch)}

#### Parameters

- **y** (*x,*) – Specify the center of a picking region in window coordinates.
- **height** (*width,*) – Specify the width and height, respectively, of the picking region in window coordinates.
- **viewport** (*bgl.Buffer* object. [int]) – Specifies the current viewport.

**gluProject**(*objx, objy, objz, modelMatrix, projMatrix, viewport, winx, winy, winz*):

Map object coordinates to window coordinates.

**See Also:**

U{[http://biology.ncsa.uiuc.edu/cgi-bin/infosrch.cgi?cmd=getdoc&coll=0650&db=bks&fname=/SGI\\_Developer/OpenGL\\_RM/ch](http://biology.ncsa.uiuc.edu/cgi-bin/infosrch.cgi?cmd=getdoc&coll=0650&db=bks&fname=/SGI_Developer/OpenGL_RM/ch)}

#### Parameters

- **objy, objz** (*objx,*) – Specify the object coordinates.
- **modelMatrix** (*bgl.Buffer* object. [double]) – Specifies the current modelview matrix (as from a `glGetDoublev` call).
- **projMatrix** (*bgl.Buffer* object. [double]) – Specifies the current projection matrix (as from a `glGetDoublev` call).
- **viewport** (*bgl.Buffer* object. [int]) – Specifies the current viewport (as from a `glGetIntegerv` call).

- **winy, winz** (*winx*,) – Return the computed window coordinates.

**gluUnProject(winx, winy, winz, modelMatrix, projMatrix, viewport, objx, objy, objz):**  
Map object coordinates to window coordinates.

**See Also:**

U{[http://biology.ncsa.uiuc.edu/cgi-bin/infosrch.cgi?cmd=getdoc&coll=0650&db=bks&fname=/SGI\\_Developer/OpenGL\\_RM/ch](http://biology.ncsa.uiuc.edu/cgi-bin/infosrch.cgi?cmd=getdoc&coll=0650&db=bks&fname=/SGI_Developer/OpenGL_RM/ch)}

### Parameters

- **winy, winz** (*winx*,) – Specify the window coordinates to be mapped.
- **modelMatrix** (`bgl.Buffer` object. [double]) – Specifies the current modelview matrix (as from a `glGetDoublev` call).
- **projMatrix** (`bgl.Buffer` object. [double]) – Specifies the current projection matrix (as from a `glGetDoublev` call).
- **viewport** (`bgl.Buffer` object. [int]) – Specifies the current viewport (as from a `glGetIntegerv` call).
- **objx, objy, objz** (*objx*,) – Return the computed object coordinates.

class Buffer:

The Buffer object is simply a block of memory that is delineated and initialized by the user. Many OpenGL functions return data to a C-style pointer, however, because this is not possible in python the Buffer object can be used to this end. Wherever pointer notation is used in the OpenGL functions the Buffer object can be used in it's `bgl` wrapper. In some instances the Buffer object will need to be initialized with the template parameter, while in other instances the user will want to create just a blank buffer which will be zeroed by default.

```
import bgl
```

```
myByteBuffer = bgl.Buffer(bgl.GL_BYTE, [32, 32])
bgl.glGetPolygonStipple(myByteBuffer)
```

```
print(myByteBuffer.dimensions)
print(myByteBuffer.to_list())
```

```
sliceBuffer = myByteBuffer[0:16]
print(sliceBuffer)
```

**bgl.dimensions**

The number of dimensions of the Buffer.

**bgl.to\_list()**

The contents of the Buffer as a python list.

**\_\_init\_\_(type, dimensions, template = None):**

This will create a new Buffer object for use with other `bgl` OpenGL commands. Only the type of argument to store in the buffer and the dimensions of the buffer are necessary. Buffers are zeroed by default unless a template is supplied, in which case the buffer is initialized to the template.

### Parameters

- **type** (*int*) – The format to store data in. The type should be one of `GL_BYTE`, `GL_SHORT`, `GL_INT`, or `GL_FLOAT`.
- **dimensions** (*An int or sequence object specifying the dimensions of the buffer.*) – If the dimensions are specified as an int a linear array will be created for the buffer. If

a sequence is passed for the dimensions, the buffer becomes n-Dimensional, where n is equal to the number of parameters passed in the sequence. Example: [256,2] is a two- dimensional buffer while [256,256,4] creates a three- dimensional buffer. You can think of each additional dimension as a sub-item of the dimension to the left. i.e. [10,2] is a 10 element array each with 2 sub-items. [(0,0), (0,1), (1,0), (1,1), (2,0), ...] etc.

- **template** (*A python sequence object (optional)*) – A sequence of matching dimensions which will be used to initialize the Buffer. If a template is not passed in all fields will be initialized to 0.

**Return type** Buffer object

**Returns** The newly created buffer as a PyObject.

## 3.4 Font Drawing (blf)

This module provides access to blenders text drawing functions.

### 3.4.1 Hello World Text Example

Blender Game Engine example of using the blf module. For this module to work we need to use the OpenGL wrapper `bgl` as well.

```
# import game engine modules
from bge import render
from bge import logic
# import stand alone modules
import bgl
import blf

def init():
    """init function - runs once"""
    # create a new font object, use external ttf file
    font_path = logic.expandPath('//Zeyada.ttf')
    # store the font indice - to use later
    logic.font_id = blf.load(font_path)

    # set the font drawing routine to run every frame
    scene = logic.getCurrentScene()
    scene.post_draw = [write]

def write():
    """write on screen"""
    width = render.getWindowWidth()
    height = render.getWindowHeight()

    # OpenGL setup
    bgl.glMatrixMode(bgl.GL_PROJECTION)
    bgl.glLoadIdentity()
    bgl.gluOrtho2D(0, width, 0, height)
    bgl.glMatrixMode(bgl.GL_MODELVIEW)
    bgl.glLoadIdentity()
```

```
# BLF drawing routine
font_id = logic.font_id
blf.position(font_id, (width * 0.2), (height * 0.3), 0)
blf.size(font_id, 50, 72)
blf.draw(font_id, "Hello World")
```

`blf.aspect` (*fontid*, *aspect*)

Set the aspect for drawing text.

#### Parameters

- **fontid** (*int*) – The id of the typeface as returned by `blf.load()`, for default font use 0.
- **aspect** (*float*) – The aspect ratio for text drawing to use.

`blf.blur` (*fontid*, *radius*)

Set the blur radius for drawing text.

#### Parameters

- **fontid** (*int*) – The id of the typeface as returned by `blf.load()`, for default font use 0.
- **radius** (*int*) – The radius for blurring text (in pixels).

`blf.clipping` (*fontid*, *xmin*, *ymin*, *xmax*, *ymax*)

Set the clipping, enable/disable using CLIPPING.

#### Parameters

- **fontid** (*int*) – The id of the typeface as returned by `blf.load()`, for default font use 0.
- **xmin** (*float*) – Clip the drawing area by these bounds.
- **ymin** (*float*) – Clip the drawing area by these bounds.
- **xmax** (*float*) – Clip the drawing area by these bounds.
- **ymax** (*float*) – Clip the drawing area by these bounds.

`blf.dimensions` (*fontid*, *text*)

Return the width and height of the text.

#### Parameters

- **fontid** (*int*) – The id of the typeface as returned by `blf.load()`, for default font use 0.
- **text** (*string*) – the text to draw.

**Returns** the width and height of the text.

**Return type** tuple of 2 floats

`blf.disable` (*fontid*, *option*)

Disable option.

#### Parameters

- **fontid** (*int*) – The id of the typeface as returned by `blf.load()`, for default font use 0.
- **option** (*int*) – One of ROTATION, CLIPPING, SHADOW or KERNING\_DEFAULT.

`blf.draw` (*fontid*, *text*)

Draw text in the current context.

#### Parameters

- **fontid** (*int*) – The id of the typeface as returned by `blf.load()`, for default font use 0.

- **text** (*string*) – the text to draw.

`blf.enable` (*fontid, option*)

Enable option.

**Parameters**

- **fontid** (*int*) – The id of the typeface as returned by `blf.load()`, for default font use 0.
- **option** (*int*) – One of ROTATION, CLIPPING, SHADOW or KERNING\_DEFAULT.

`blf.load` (*filename*)

Load a new font.

**Parameters** **filename** (*string*) – the filename of the font.

**Returns** the new font's fontid or -1 if there was an error.

**Return type** integer

`blf.position` (*fontid, x, y, z*)

Set the position for drawing text.

**Parameters**

- **fontid** (*int*) – The id of the typeface as returned by `blf.load()`, for default font use 0.
- **x** (*float*) – X axis position to draw the text.
- **y** (*float*) – Y axis position to draw the text.
- **z** (*float*) – Z axis position to draw the text.

`blf.rotation` (*fontid, angle*)

Set the text rotation angle, enable/disable using ROTATION.

**Parameters**

- **fontid** (*int*) – The id of the typeface as returned by `blf.load()`, for default font use 0.
- **angle** (*float*) – The angle for text drawing to use.

`blf.shadow` (*fontid, level, r, g, b, a*)

Shadow options, enable/disable using SHADOW.

**Parameters**

- **fontid** (*int*) – The id of the typeface as returned by `blf.load()`, for default font use 0.
- **level** (*int*) – The blur level, can be 3, 5 or 0.
- **r** (*float*) – Shadow color (red channel 0.0 - 1.0).
- **g** (*float*) – Shadow color (green channel 0.0 - 1.0).
- **b** (*float*) – Shadow color (blue channel 0.0 - 1.0).
- **a** (*float*) – Shadow color (alpha channel 0.0 - 1.0).

`blf.shadow_offset` (*fontid, x, y*)

Set the offset for shadow text.

**Parameters**

- **fontid** (*int*) – The id of the typeface as returned by `blf.load()`, for default font use 0.
- **x** (*float*) – Vertical shadow offset value in pixels.
- **y** (*float*) – Horizontal shadow offset value in pixels.



`blf.size` (*fontid*, *size*, *dpi*)

Set the size and dpi for drawing text.

#### Parameters

- **fontid** (*int*) – The id of the typeface as returned by `blf.load()`, for default font use 0.
- **size** (*int*) – Point size of the font.
- **dpi** (*int*) – dots per inch value to use for drawing.

`blf.unload` (*filename*)

Unload an existing font.

**Parameters** **filename** (*string*) – the filename of the font.

constant value 2

constant value 8

constant value 1

constant value 4

## 3.5 GPU functions (gpu)

This module provides access to materials GLSL shaders.

### 3.5.1 Intro

Module to provide functions concerning the GPU implementation in Blender, in particular the GLSL shaders that blender generates automatically to render materials in the 3D view and in the game engine.

**Warning:** The API provided by this module should be consider unstable. The data exposed by the API are are closely related to Blender's internal GLSL code and may change if the GLSL code is modified (e.g. new uniform type).

### 3.5.2 Constants

#### GLSL data type

Type of GLSL data. For shader uniforms, the data type determines which glUniform function variant to use to send the uniform value to the GPU. For vertex attributes, the data type determines which glVertexAttrib function variant to use to send the vertex attribute to the GPU.

See `export_shader`

`gpu.GPU_DATA_1I`

one integer

**Value** 1

`gpu.GPU_DATA_1F`

one float

**Value** 2

gpu.**GPU\_DATA\_2F**  
two floats

**Value** 3

gpu.**GPU\_DATA\_3F**  
three floats

**Value** 4

gpu.**GPU\_DATA\_4F**  
four floats

**Value** 5

gpu.**GPU\_DATA\_9F**  
matrix 3x3 in column-major order

**Value** 6

gpu.**GPU\_DATA\_16F**  
matrix 4x4 in column-major order

**Value** 7

gpu.**GPU\_DATA\_4UB**  
four unsigned byte

**Value** 8

## GLSL uniform type

Constants that specify the type of uniform used in a GLSL shader. The uniform type determines the data type, origin and method of calculation used by Blender to compute the uniform value.

The calculation of some of the uniforms is based on matrices available in the scene:

***mat4\_cam\_to\_world*** Model matrix of the camera. OpenGL 4x4 matrix that converts camera local coordinates to world coordinates. In blender this is obtained from the ‘matrix\_world’ attribute of the camera object.

Some uniform will need the *mat4\_world\_to\_cam* matrix computed as the inverse of this matrix.

***mat4\_object\_to\_world*** Model matrix of the object that is being rendered. OpenGL 4x4 matrix that converts object local coordinates to world coordinates. In blender this is obtained from the ‘matrix\_world’ attribute of the object.

Some uniform will need the *mat4\_world\_to\_object* matrix, computed as the inverse of this matrix.

***mat4\_lamp\_to\_world*** Model matrix of the lamp lighting the object. OpenGL 4x4 matrix that converts lamp local coordinates to world coordinates. In blender this is obtained from the ‘matrix\_world’ attribute of the lamp object.

Some uniform will need the *mat4\_world\_to\_lamp* matrix computed as the inverse of this matrix.

gpu.**GPU\_DYNAMIC\_OBJECT\_VIEWMAT**

The uniform is a 4x4 GL matrix that converts world coordinates to camera coordinates (see [mat4\\_world\\_to\\_cam](#)). Can be set once per frame. There is at most one uniform of that type per shader.

**Value** 1

gpu.**GPU\_DYNAMIC\_OBJECT\_MAT**

The uniform is a 4x4 GL matrix that converts object coordinates to world coordinates (see

`mat4_object_to_world`). Must be set before drawing the object. There is at most one uniform of that type per shader.

**Value 2**

gpu. **GPU\_DYNAMIC\_OBJECT\_VIEWIMAT**

The uniform is a 4x4 GL matrix that converts coordinates in camera space to world coordinates (see `mat4_cam_to_world`). Can be set once per frame. There is at most one uniform of that type per shader.

**Value 3**

gpu. **GPU\_DYNAMIC\_OBJECT\_IMAT**

The uniform is a 4x4 GL matrix that converts world coordinates to object coordinates (see `mat4_world_to_object`). Must be set before drawing the object. There is at most one uniform of that type per shader.

**Value 4**

gpu. **GPU\_DYNAMIC\_OBJECT\_COLOR**

The uniform is a vector of 4 float representing a RGB color + alpha defined at object level. Each values between 0.0 and 1.0. In blender it corresponds to the 'color' attribute of the object. Must be set before drawing the object. There is at most one uniform of that type per shader.

**Value 5**

gpu. **GPU\_DYNAMIC\_LAMP\_DYNVEC**

The uniform is a vector of 3 float representing the direction of light in camera space. In Blender, this is computed by

`mat4_world_to_cam * (-vec3_lamp_Z_axis)`

as the lamp Z axis points to the opposite direction of light. The norm of the vector should be unity. Can be set once per frame. There is one uniform of that type per lamp lighting the material.

**Value 6**

gpu. **GPU\_DYNAMIC\_LAMP\_DYNCO**

The uniform is a vector of 3 float representing the position of the light in camera space. Computed as

`mat4_world_to_cam * vec3_lamp_pos`

Can be set once per frame. There is one uniform of that type per lamp lighting the material.

**Value 7**

gpu. **GPU\_DYNAMIC\_LAMP\_DYNIMAT**

The uniform is a 4x4 GL matrix that converts vector in camera space to lamp space. Computed as

`mat4_world_to_lamp * mat4_cam_to_world`

Can be set once per frame. There is one uniform of that type per lamp lighting the material.

**Value 8**

gpu. **GPU\_DYNAMIC\_LAMP\_DYNPERSMAT**

The uniform is a 4x4 GL matrix that converts a vector in camera space to shadow buffer depth space. Computed as

`mat4_perspective_to_depth * mat4_lamp_to_perspective * mat4_world_to_lamp * mat4_cam_to_world`.  
`mat4_perspective_to_depth` is a fixed matrix defined as follow:

```
0.5 0.0 0.0 0.5
0.0 0.5 0.0 0.5
0.0 0.0 0.5 0.5
0.0 0.0 0.0 1.0
```

This uniform can be set once per frame. There is one uniform of that type per lamp casting shadow in the scene.

**Value 9**

`gpu.GPU_DYNAMIC_LAMP_DYNENERGY`

The uniform is a single float representing the lamp energy. In blender it corresponds to the 'energy' attribute of the lamp data block. There is one uniform of that type per lamp lighting the material.

**Value 10**

`gpu.GPU_DYNAMIC_LAMP_DYNCOL`

The uniform is a vector of 3 float representing the lamp color. Color elements are between 0.0 and 1.0. In blender it corresponds to the 'color' attribute of the lamp data block. There is one uniform of that type per lamp lighting the material.

**Value 11**

`gpu.GPU_DYNAMIC_SAMPLER_2DBUFFER`

The uniform is an integer representing an internal texture used for certain effect (color band, etc).

**Value 12**

`gpu.GPU_DYNAMIC_SAMPLER_2DIMAGE`

The uniform is an integer representing a texture loaded from an image file.

**Value 13**

`gpu.GPU_DYNAMIC_SAMPLER_2DSHADOW`

The uniform is an integer representing a shadow buffer corresponding to a lamp casting shadow.

**Value 14**

## GLSL attribute type

Type of the vertex attribute used in the GLSL shader. Determines the mesh custom data layer that contains the vertex attribute.

`gpu.CD_MTFACE`

Vertex attribute is a UV layer. Data type is vector of 2 float.

There can be more than one attribute of that type, they are differentiated by name. In blender, you can retrieve the attribute data with:

```
mesh.uv_textures[attribute['name']]
```

**Value 5**

`gpu.CD_MCOL`

Vertex attribute is color layer. Data type is vector 4 unsigned byte (RGBA).

There can be more than one attribute of that type, they are differentiated by name. In blender you can retrieve the attribute data with:

```
mesh.vertex_colors[attribute['name']]
```

**Value 6**

`gpu.CD_ORCO`

Vertex attribute is original coordinates. Data type is vector 3 float.

There can be only 1 attribute of that type per shader. In blender you can retrieve the attribute data with:

`mesh.vertices`

**Value** 14

`gpu.CD_TANGENT`

Vertex attribute is the tangent vector. Data type is vector 4 float.

There can be only 1 attribute of that type per shader. There is currently no way to retrieve this attribute data via the RNA API but a standalone C function to compute the tangent layer from the other layers can be obtained from [blender.org](http://blender.org).

**Value** 18

### 3.5.3 Functions

`gpu.export_shader` (*scene, material*)

Extracts the GLSL shader producing the visual effect of material in scene for the purpose of reusing the shader in an external engine. This function is meant to be used in material exporter so that the GLSL shader can be exported entirely. The return value is a dictionary containing the shader source code and all associated data.

#### Parameters

- **scene** (`bpy.types.Scene`) – the scene in which the material is rendered.
- **material** (`bpy.types.Material`) – the material that you want to export the GLSL shader

**Returns** the shader source code and all associated data in a dictionary

**Return type** dictionary

The dictionary contains the following elements:

- **['fragment']** [string] fragment shader source code.
- **['vertex']** [string] vertex shader source code.
- **['uniforms']** [sequence] list of uniforms used in fragment shader, can be empty list. Each element of the sequence is a dictionary with the following elements:
  - **['varname']** [string] name of the uniform in the fragment shader. Always of the form 'unf<number>'.
  - **['datatype']** [integer] data type of the uniform variable. Can be one of the following:
    - \* `gpu.GPU_DATA_1I` : use `glUniform1i`
    - \* `gpu.GPU_DATA_1F` : use `glUniform1fv`
    - \* `gpu.GPU_DATA_2F` : use `glUniform2fv`
    - \* `gpu.GPU_DATA_3F` : use `glUniform3fv`
    - \* `gpu.GPU_DATA_4F` : use `glUniform4fv`
    - \* `gpu.GPU_DATA_9F` : use `glUniformMatrix3fv`
    - \* `gpu.GPU_DATA_16F` : use `glUniformMatrix4fv`
  - **['type']** [integer] type of uniform, determines the origin and method of calculation. See [uniform-type](#). Depending on the type, more elements will be present.

- ['lamp'] [`bpy.types.Object`] Reference to the lamp object from which the uniforms value are extracted. Set for the following uniforms types:

```
* gpu.GPU_DYNAMIC_LAMP_DYNVEC
* gpu.GPU_DYNAMIC_LAMP_DYNCO
* gpu.GPU_DYNAMIC_LAMP_DYNIMAT
* gpu.GPU_DYNAMIC_LAMP_DYNPERSMAT
* gpu.GPU_DYNAMIC_LAMP_DYNENERGY
* gpu.GPU_DYNAMIC_LAMP_DYNCOL
* gpu.GPU_DYNAMIC_SAMPLER_2DSHADOW
```

Notes:

- \* The uniforms `gpu.GPU_DYNAMIC_LAMP_DYNVEC`, `gpu.GPU_DYNAMIC_LAMP_DYNCO`, `gpu.GPU_DYNAMIC_LAMP_DYNIMAT` and `gpu.GPU_DYNAMIC_LAMP_DYNPERSMAT` refer to the lamp object position and orientation, both of can be derived from the object world matrix:

```
obmat = uniform['lamp'].matrix_world
```

where `obmat` is the `mat4_lamp_to_world` matrix of the lamp as a 2 dimensional array, the lamp world location location is in `obmat[3]`.

- \* The uniform types `gpu.GPU_DYNAMIC_LAMP_DYNENERGY` and `gpu.GPU_DYNAMIC_LAMP_DYNCOL` refer to the lamp data bloc that you get from:

```
la = uniform['lamp'].data
```

from which you get `la.energy` and `la.color`

- \* Lamp duplication is not supported: if you have duplicated lamps in your scene (i.e. lamp that are instantiated by `dupligroup`, etc), this element will only give you a reference to the original lamp and you will not know which instance of the lamp it is referring too. You can still handle that case in the exporter by distributing the uniforms amongst the duplicated lamps.

- ['image'] [`bpy.types.Image`] Reference to the image databloc. Set for uniform type `gpu.GPU_DYNAMIC_SAMPLER_2DIMAGE`. You can get the image data from:

```
# full path to image file
uniform['image'].filepath
# image size as a 2-dimensional array of int
uniform['image'].size
```

- ['texnumber'] [integer] Channel number to which the texture is bound when drawing the object. Set for uniform types `gpu.GPU_DYNAMIC_SAMPLER_2DBUFFER`, `gpu.GPU_DYNAMIC_SAMPLER_2DIMAGE` and `gpu.GPU_DYNAMIC_SAMPLER_2DSHADOW`.

This is provided for information only: when reusing the shader outside blender, you are free to assign the textures to the channel of your choice and to pass that number channel to the GPU in the uniform.

- ['texpixels'] [byte array] texture data for uniform type `gpu.GPU_DYNAMIC_SAMPLER_2DBUFFER`. Although the corresponding uniform is a 2D sampler, the texture is always a 1D texture of `n x 1` pixel. The texture size `n` is provided in ['texsize'] element. These texture are only used for computer generated texture (colorband, etc). The texture data is provided so that you can make a real image out of it in the exporter.

- **['texsize']** [integer] horizontal size of texture for uniform type `gpu.GPU_DYNAMIC_SAMPLER_2DBUFFER`. The texture data is in `['texpixels']`.
- **['attributes']** [sequence] list of attributes used in vertex shader, can be empty. Blender doesn't use standard attributes except for vertex position and normal. All other vertex attributes must be passed using the generic `glVertexAttrib` functions. The attribute data can be found in the derived mesh custom data using RNA. Each element of the sequence is a dictionary containing the following elements:
  - **['varname']** [string] name of the uniform in the vertex shader. Always of the form `'att<number>'`.
  - **['datatype']** [integer] data type of vertex attribute, can be one of the following:
    - \* `gpu.GPU_DATA_2F` : use `glVertexAttrib2fv`
    - \* `gpu.GPU_DATA_3F` : use `glVertexAttrib3fv`
    - \* `gpu.GPU_DATA_4F` : use `glVertexAttrib4fv`
    - \* `gpu.GPU_DATA_4UB` : use `glVertexAttrib4ubv`
  - **['number']** [integer] generic attribute number. This is provided for information only. Blender doesn't use `glBindAttribLocation` to place generic attributes at specific location, it lets the shader compiler place the attributes automatically and query the placement with `glGetAttribLocation`. The result of this placement is returned in this element.
 

When using this shader in a render engine, you should either use `glBindAttribLocation` to force the attribute at this location or use `glGetAttribLocation` to get the placement chosen by the compiler of your GPU.
  - **['type']** [integer] type of the mesh custom data from which the vertex attribute is loaded. See `attribute-type`.
  - **['name']** [string or integer] custom data layer name, used for attribute type `gpu.CD_MTFACE` and `gpu.CD_MCOL`.

Example:

```
import gpu
# get GLSL shader of material Mat.001 in scene Scene.001
scene = bpy.data.scenes['Scene.001']
material = bpy.data.materials['Mat.001']
shader = gpu.export_shader(scene,material)
# scan the uniform list and find the images used in the shader
for uniform in shader['uniforms']:
    if uniform['type'] == gpu.GPU_DYNAMIC_SAMPLER_2DIMAGE:
        print("uniform {0} is using image {1}".format(uniform['varname'], uniform['image'].filepath))
# scan the attribute list and find the UV layer used in the shader
for attribute in shader['attributes']:
    if attribute['type'] == gpu.CD_MTFACE:
        print("attribute {0} is using UV layer {1}".format(attribute['varname'], attribute['name']))
```

### 3.5.4 Notes

1. Calculation of the `mat4_lamp_to_perspective` matrix for a spot lamp.

The following pseudo code shows how the `mat4_lamp_to_perspective` matrix is computed in blender for uniforms of `gpu.GPU_DYNAMIC_LAMP_DYNPERSMAT` type:

```
#Get the lamp datablock with:
lamp=bpy.data.objects[uniform['lamp']].data

#Compute the projection matrix:
# You will need these lamp attributes:
# lamp.clipsta : near clip plane in world unit
# lamp.clipend : far clip plane in world unit
# lamp.spotsize : angle in degree of the spot light

#The size of the projection plane is computed with the usual formula:
wsize = lamp.clista * tan(lamp.spotsize/2)

#And the projection matrix:
mat4_lamp_to_perspective = glFrustum(-wsize,wsize,-wsize,wsize,lamp.clista,lamp.clipend)
```

## 2. Creation of the shadow map for a spot lamp.

The shadow map is the depth buffer of a render performed by placing the camera at the spot light position. The size of the shadow map is given by the attribute `lamp.bufsize` : shadow map size in pixel, same size in both dimensions.

## 3.6 Audio System (aud)

This module provides access to the audaspace audio library.

### 3.6.1 Basic Sound Playback

This script shows how to use the classes: `Device`, `Factory` and `Handle`.

```
import aud

device = aud.device()
# load sound file (it can be a video file with audio)
factory = aud.Factory('music.ogg')

# play the audio, this return a handle to control play/pause
handle = device.play(factory)
# if the audio is not too big and will be used often you can buffer it
factory_buffered = aud.Factory.buffer(factory)
handle_buffered = device.play(factory_buffered)

# stop the sounds (otherwise they play until their ends)
handle.stop()
handle_buffered.stop()
```

`device()`

Returns the application's `Device`.

**return** The application's `Device`.

**rtype** `Device`

constant value 3

constant value 0



constant value 1  
 constant value 2  
 constant value 5  
 constant value 6  
 constant value 0  
 constant value 1  
 constant value 2  
 constant value 3  
 constant value 4  
 constant value 36  
 constant value 40  
 constant value 0  
 constant value 18  
 constant value 19  
 constant value 20  
 constant value 1  
 constant value 0  
 constant value 2  
 constant value 1

**class** `aud.Device`

Device objects represent an audio output backend like OpenAL or SDL, but might also represent a file output or RAM buffer output.

`lock()`

Locks the device so that it's guaranteed, that no samples are read from the streams until `unlock()` is called. This is useful if you want to do start/stop/pause/resume some sounds at the same time.

---

**Note:** The device has to be unlocked as often as locked to be able to continue playback.

---

|  |
|--|
| <p><b>Warning:</b> Make sure the time between locking and unlocking is as short as possible to avoid clicks.</p> |
|--|

`play(factory, keep=False)`

Plays a factory.

**Parameters**

- **factory** (`Factory`) – The factory to play.
- **keep** (`bool`) – See `Handle.keep`.

**Returns** The playback handle with which playback can be controlled with.

**Return type** `Handle`

`stopAll()`

Stops all playing and paused sounds.

`unlock()`

Unlocks the device after a lock call, see `lock()` for details.

**channels**

The channel count of the device.

**distance\_model**

The distance model of the device.

**See Also:**

[http://connect.creativelabs.com/openal/Documentation/OpenAL%201.1%20Specification.htm#\\_Toc199835864](http://connect.creativelabs.com/openal/Documentation/OpenAL%201.1%20Specification.htm#_Toc199835864)

**doppler\_factor**

The doppler factor of the device. This factor is a scaling factor for the velocity vectors in doppler calculation. So a value bigger than 1 will exaggerate the effect as it raises the velocity.

**format**

The native sample format of the device.

**listener\_location**

The listeners's location in 3D space, a 3D tuple of floats.

**listener\_orientation**

The listener's orientation in 3D space as quaternion, a 4 float tuple.

**listener\_velocity**

The listener's velocity in 3D space, a 3D tuple of floats.

**rate**

The sampling rate of the device in Hz.

**speed\_of\_sound**

The speed of sound of the device. The speed of sound in air is typically 343 m/s.

**volume**

The overall volume of the device.

**class** `aud.Factory`

Factory objects are immutable and represent a sound that can be played simultaneously multiple times. They are called factories because they create reader objects internally that are used for playback.

`file(filename)`

Creates a factory object of a sound file.

**Parameters** `filename` (*string*) – Path of the file.

**Returns** The created `Factory` object.

**Return type** `Factory`

|  |
|--|
| <p><b>Warning:</b> If the file doesn't exist or can't be read you will not get an exception immediately, but when you try to start playback of that factory.</p> |
|--|

`sine(frequency, rate=44100)`

Creates a sine factory which plays a sine wave.

**Parameters**

- **frequency** (*float*) – The frequency of the sine wave in Hz.
- **rate** (*int*) – The sampling rate in Hz. It's recommended to set this value to the playback device's sampling rate to avoid resampling.

**Returns** The created `Factory` object.

**Return type** `Factory`

`buffer()`

Buffers a factory into RAM. This saves CPU usage needed for decoding and file access if the underlying factory reads from a file on the harddisk, but it consumes a lot of memory.

**Returns** The created `Factory` object.

**Return type** `Factory`

---

**Note:** Only known-length factories can be buffered.

**Warning:** Raw PCM data needs a lot of space, only buffer short factories.

`delay(time)`

Delays by playing adding silence in front of the other factory's data.

**Parameters** **time** (*float*) – How many seconds of silence should be added before the factory.

**Returns** The created `Factory` object.

**Return type** `Factory`

`fadein(start, length)`

Fades a factory in by raising the volume linearly in the given time interval.

**Parameters**

- **start** (*float*) – Time in seconds when the fading should start.
- **length** (*float*) – Time in seconds how long the fading should last.

**Returns** The created `Factory` object.

**Return type** `Factory`

---

**Note:** Before the fade starts it plays silence.

`fadeout(start, length)`

Fades a factory in by lowering the volume linearly in the given time interval.

**Parameters**

- **start** (*float*) – Time in seconds when the fading should start.
- **length** (*float*) – Time in seconds how long the fading should last.

**Returns** The created `Factory` object.

**Return type** `Factory`

---

**Note:** After the fade this factory plays silence, so that the length of the factory is not altered.

---

`filter(b, a = (1))`

Filters a factory with the supplied IIR filter coefficients. Without the second parameter you'll get a FIR filter. If the first value of the a sequence is 0 it will be set to 1 automatically. If the first value of the a sequence is neither 0 nor 1, all filter coefficients will be scaled by this value so that it is 1 in the end, you don't have to scale yourself.

**Parameters**

- **b** (*sequence of float*) – The nominator filter coefficients.
- **a** (*sequence of float*) – The denominator filter coefficients.

**Returns** The created `Factory` object.

**Return type** `Factory`

`highpass(frequency, Q=0.5)`

Creates a second order highpass filter based on the transfer function  $H(s) = s^2 / (s^2 + s/Q + 1)$

**Parameters**

- **frequency** (*float*) – The cut off frequency of the highpass.
- **Q** (*float*) – Q factor of the lowpass.

**Returns** The created `Factory` object.

**Return type** `Factory`

`join(factory)`

Plays two factories in sequence.

**Parameters** **factory** (`Factory`) – The factory to play second.

**Returns** The created `Factory` object.

**Return type** `Factory`

---

**Note:** The two factories have to have the same specifications (channels and samplerate).

---

`limit(start, end)`

Limits a factory within a specific start and end time.

**Parameters**

- **start** (*float*) – Start time in seconds.
- **end** (*float*) – End time in seconds.

**Returns** The created `Factory` object.

**Return type** `Factory`

`loop(count)`

Loops a factory.

**Parameters** **count** (*integer*) – How often the factory should be looped. Negative values mean endlessly.

**Returns** The created `Factory` object.

**Return type** `Factory`

---

**Note:** This is a filter function, you might consider using `Handle.loop_count` instead.

---

`lowpass(frequency, Q=0.5)`

Creates a second order lowpass filter based on the transfer function  $H(s) = 1 / (s^2 + s/Q + 1)$

**Parameters**

- **frequency** (*float*) – The cut off frequency of the lowpass.
- **Q** (*float*) – Q factor of the lowpass.

**Returns** The created `Factory` object.

**Return type** `Factory`

`mix(factory)`

Mixes two factories.

**Parameters** **factory** (`Factory`) – The factory to mix over the other.

**Returns** The created `Factory` object.

**Return type** `Factory`

---

**Note:** The two factories have to have the same specifications (channels and samplerate).

---

`pingpong()`

Plays a factory forward and then backward. This is like joining a factory with its reverse.

**Returns** The created `Factory` object.

**Return type** `Factory`

`pitch(factor)`

Changes the pitch of a factory with a specific factor.

**Parameters** **factor** (*float*) – The factor to change the pitch with.

**Returns** The created `Factory` object.

**Return type** `Factory`

---

**Note:** This is done by changing the sample rate of the underlying factory, which has to be an integer, so the factor value rounded and the factor may not be 100 % accurate.

---

**Note:** This is a filter function, you might consider using `Handle.pitch` instead.

---

`reverse()`

Plays a factory reversed.

**Returns** The created `Factory` object.

**Return type** `Factory`

---

**Note:** The factory has to have a finite length and has to be seekable. It's recommended to use this only with factories with fast and accurate seeking, which is not true for encoded audio files, such ones should be buffered using `buffer()` before being played reversed.

---

**Warning:** If seeking is not accurate in the underlying factory you'll likely hear skips/jumps/cracks.

---

`square(threshold = 0)`

Makes a square wave out of an audio wave by setting all samples with a amplitude  $\geq$  threshold to 1, all  $\leq$  -threshold to -1 and all between to 0.

**Parameters** **threshold** (*float*) – Threshold value over which an amplitude counts non-zero.

**Returns** The created `Factory` object.

**Return type** `Factory`

`volume(volume)`

Changes the volume of a factory.

**Parameters** **volume** (*float*) – The new volume..

**Returns** The created `Factory` object.

**Return type** `Factory`

---

**Note:** Should be in the range [0, 1] to avoid clipping.

---

**Note:** This is a filter function, you might consider using `Handle.volume` instead.

---

### **class** `aud.Handle`

Handle objects are playback handles that can be used to control playback of a sound. If a sound is played back multiple times then there are as many handles.

`pause()`

Pauses playback.

**Returns** Whether the action succeeded.

**Return type** `bool`

`resume()`

Resumes playback.

**Returns** Whether the action succeeded.

**Return type** `bool`

`stop()`

Stops playback.

**Returns** Whether the action succeeded.

**Return type** `bool`

---

**Note:** This makes the handle invalid.

---

**attenuation**

This factor is used for distance based attenuation of the source.

**See Also:**

`Device.distance_model`

**cone\_angle\_inner**

The opening angle of the inner cone of the source. If the cone values of a source are set there are two (audible) cones with the apex at the `location` of the source and with infinite height, heading in the direction of the source's `orientation`. In the inner cone the volume is normal. Outside the outer cone the volume will be `cone_volume_outer` and in the area between the volume will be interpolated linearly.

**cone\_angle\_outer**

The opening angle of the outer cone of the source.

**See Also:**

`cone_angle_inner`

**cone\_volume\_outer**

The volume outside the outer cone of the source.

**See Also:**

`cone_angle_inner`

**distance\_maximum**

The maximum distance of the source. If the listener is further away the source volume will be 0.

**See Also:**

`Device.distance_model`

**distance\_reference**

The reference distance of the source. At this distance the volume will be exactly `volume`.

**See Also:**

`Device.distance_model`

**keep**

Whether the sound should be kept paused in the device when its end is reached. This can be used to seek the sound to some position and start playback again.

**Warning:** If this is set to true and you forget stopping this equals a memory leak as the handle exists until the device is destroyed.

**location**

The source's location in 3D space, a 3D tuple of floats.

**loop\_count**

The (remaining) loop count of the sound. A negative value indicates infinity.

**orientation**

The source's orientation in 3D space as quaternion, a 4 float tuple.

**pitch**

The pitch of the sound.

**position**

The playback position of the sound in seconds.

**relative**

Whether the source's location, velocity and orientation is relative or absolute to the listener.

**status**

Whether the sound is playing, paused or stopped (=invalid).

**velocity**

The source's velocity in 3D space, a 3D tuple of floats.

**volume**

The volume of the sound.

**volume\_maximum**

The maximum volume of the source.

**See Also:**

`Device.distance_model`

**volume\_minimum**

The minimum volume of the source.

**See Also:**

`Device.distance_model`

**class** `aud.error`

## 3.7 Extra Utilities (bpy\_extras)

Utility modules associated with the bpy module.

### 3.7.1 bpy\_extras submodule (bpy\_extras.anim\_utils)

`bpy_extras.anim_utils.bake_action` (*frame\_start*, *frame\_end*, *frame\_step=1*,  
*only\_selected=False*, *do\_pose=True*, *do\_object=True*,  
*do\_constraint\_clear=False*, *do\_clean=False*, *action=None*)

Return an image from the file path with options to search multiple paths and return a placeholder if its not found.

**Parameters**

- **frame\_start** (*int*) – First frame to bake.
- **frame\_end** (*int*) – Last frame to bake.
- **frame\_step** (*int*) – Frame step.
- **only\_selected** (*bool*) – Only bake selected data.
- **do\_pose** (*bool*) – Bake pose channels.
- **do\_object** (*bool*) – Bake objects.
- **do\_constraint\_clear** (*bool*) – Remove constraints.



- **do\_clean** (*bool*) – Remove redundant keyframes after baking.
- **action** (`bpy.types.Action` or `None`) – An action to bake the data into, or `None` for a new action to be created.

**Returns** an action or `None`

**Return type** `bpy.types.Action`

### 3.7.2 bpy\_extras submodule (bpy\_extras.object\_utils)

`bpy_extras.object_utils.add_object_align_init` (*context, operator*)

Return a matrix using the operator settings and view context.

#### Parameters

- **context** (`bpy.types.Context`) – The context to use.
- **operator** (`bpy.types.Operator`) – The operator, checked for location and rotation properties.

**Returns** the matrix from the context and settings.

**Return type** `mathutils.Matrix`

`bpy_extras.object_utils.object_data_add` (*context, obdata, operator=None*)

Add an object using the view context and preference to initialize the location, rotation and layer.

#### Parameters

- **context** (`bpy.types.Context`) – The context to use.
- **obdata** (*valid object data type or None.*) – the data used for the new object.
- **operator** (`bpy.types.Operator`) – The operator, checked for location and rotation properties.

**Returns** the newly created object in the scene.

**Return type** `bpy.types.ObjectBase`

### 3.7.3 bpy\_extras submodule (bpy\_extras.io\_utils)

`bpy_extras.io_utils.axis_conversion` (*from\_forward='Y', from\_up='Z', to\_forward='Y', to\_up='Z'*)

Each argument us an axis in ['X', 'Y', 'Z', '-X', '-Y', '-Z'] where the first 2 are a source and the second 2 are the target.

`bpy_extras.io_utils.axis_conversion_ensure` (*operator, forward\_attr, up\_attr*)

Function to ensure an operator has valid axis conversion settings, intended to be used from `bpy.types.Operator.check`.

#### Parameters

- **operator** (`bpy.types.Operator`) – the operator to access axis attributes from.
- **forward\_attr** (*string*) – attribute storing the forward axis
- **up\_attr** (*string*) – attribute storing the up axis

**Returns** True if the value was modified.

**Return type** `boolean`

`bpy_extras.io_utils.create_derived_objects` (*scene, ob*)

`bpy_extras.io_utils.free_derived_objects` (*ob*)

`bpy_extras.io_utils.unpack_list` (*list\_of\_tuples*)

`bpy_extras.io_utils.unpack_face_list` (*list\_of\_tuples*)

`bpy_extras.io_utils.path_reference` (*filepath, base\_src, base\_dst, mode='AUTO', copy\_subdir='', copy\_set=None, library=None*)

Return a filepath relative to a destination directory, for use with exporters.

#### Parameters

- **filepath** (*string*) – the file path to return, supporting blenders relative `//` prefix.
- **base\_src** (*string*) – the directory the *filepath* is relative too (normally the blend file).
- **base\_dst** (*string*) – the directory the *filepath* will be referenced from (normally the export path).
- **mode** (*string*) – the method used get the path in [`'AUTO'`, `'ABSOLUTE'`, `'RELATIVE'`, `'MATCH'`, `'STRIP'`, `'COPY'`]
- **copy\_subdir** (*string*) – the subdirectory of *base\_dst* to use when `mode='COPY'`.
- **copy\_set** (*set*) – collect from/to pairs when `mode='COPY'`, pass to `path_reference_copy` when exporting is done.
- **library** (`bpy.types.Library` or `None`) – The library this path is relative to.

**Returns** the new filepath.

**Return type** `string`

`bpy_extras.io_utils.path_reference_copy` (*copy\_set, report=<built-in function print>*)  
Execute copying files of `path_reference`

#### Parameters

- **copy\_set** (*set*) – set of (from, to) pairs to copy.
- **report** (*function*) – function used for reporting warnings, takes a string argument.

`bpy_extras.io_utils.unique_name` (*key, name, name\_dict, name\_max=-1, clean\_func=None, sep='.'*)

Helper function for storing unique names which may have special characters stripped and restricted to a maximum length.

#### Parameters

- **key** (any hashable object associated with the *name*.) – unique item this name belongs to, `name_dict[key]` will be reused when available. This can be the object, mesh, material, etc instance its self.
- **name** (*string*) – The name used to create a unique value in *name\_dict*.
- **name\_dict** (*dict*) – This is used to cache namespace to ensure no collisions occur, this should be an empty dict initially and only modified by this function.
- **clean\_func** (*function*) – Function to call on *name* before creating a unique value.
- **sep** (*string*) – Separator to use when between the name and a number when a duplicate name is found.

constant value (<built-in function EnumProperty>, {'default': 'AUTO', 'items': (('AUTO', 'Auto', 'Use Relative paths with subdirectories only'), ('ABSOLUTE', 'Absolute', 'Always write absolute paths'), ('RELATIVE', 'Relative', 'Always write relative patsh (where possible)'), ('MATCH', 'Match', 'Match Absolute/Relative setting with input path'), ('STRIP', 'Strip Path', 'Filename only'), ('COPY', 'Copy', 'copy the file to the destination path (or subdirectory)'),), 'attr': 'path\_mode', 'description': 'Method used to reference paths', 'name': 'Path Mode'})

**class** bpy\_extras.io\_utils.ExportHelper

**class** bpy\_extras.io\_utils.ImportHelper

### 3.7.4 bpy\_extras submodule (bpy\_extras.image\_utils)

bpy\_extras.image\_utils.load\_image (imagepath, dirname='', place\_holder=False, recursive=False, ncase\_cmp=True, convert\_callback=None, verbose=False)

Return an image from the file path with options to search multiple paths and return a placeholder if its not found.

#### Parameters

- **filepath** (*string*) – The image filename If a path precedes it, this will be searched as well.
- **dirname** (*string*) – is the directory where the image may be located - any file at the end will be ignored.
- **place\_holder** (*bool*) – if True a new place holder image will be created. this is useful so later you can relink the image to its original data.
- **recursive** (*bool*) – If True, directories will be recursively searched. Be careful with this if you have files in your root directory because it may take a long time.
- **ncase\_cmp** (*bool*) – on non windows systems, find the correct case for the file.
- **convert\_callback** (*function*) – a function that takes an existing path and returns a new one. Use this when loading image formats blender may not support, the CONVERT\_CALLBACK can take the path for a GIF (for example), convert it to a PNG and return the PNG's path. For formats blender can read, simply return the path that is given.

**Returns** an image or None

**Return type** `bpy.types.Image`

### 3.7.5 bpy\_extras submodule (bpy\_extras.keyconfig\_utils)

bpy\_extras.keyconfig\_utils.keyconfig\_export (wm, kc, filepath)

bpy\_extras.keyconfig\_utils.keyconfig\_merge (kc1, kc2)  
note: kc1 takes priority over kc2

bpy\_extras.keyconfig\_utils.keyconfig\_test (kc)

bpy\_extras.keyconfig\_utils.km\_exists\_in (km, export\_keymaps)

### 3.7.6 bpy\_extras submodule (bpy\_extras.mesh\_utils)

bpy\_extras.mesh\_utils.mesh\_linked\_faces (mesh)

Splits the mesh into connected faces, use this for seperating cubes from other mesh elements within 1 mesh datablock.

**Parameters** `mesh` (`bpy.types.Mesh`) – the mesh used to group with.

**Returns** lists of lists containing faces.

**Return type** list

`bpy_extras.mesh_utils.edge_face_count_dict` (*mesh*)

**Returns** dict of edge keys with their value set to the number of faces using each edge.

**Return type** dict

`bpy_extras.mesh_utils.edge_face_count` (*mesh*)

**Returns** list face users for each item in `mesh.edges`.

**Return type** list

`bpy_extras.mesh_utils.edge_loops_from_faces` (*mesh, faces=None, seams=()*)

Edge loops defined by faces

Takes `me.faces` or a list of faces and returns the edge loops These edge loops are the edges that sit between quads, so they dont touch 1 quad, note: not connected will make 2 edge loops, both only containing 2 edges.

return a list of edge key lists `[(0, 1), (4, 8), (3, 8), ...]`

**Parameters**

- **mesh** (`bpy.types.Mesh`) – the mesh used to get edge loops from.
- **faces** (`bpy.types.MeshFaces`, sequence or or `NoneType`) – optional face list to only use some of the meshes faces.

**Returns** return a list of edge vertex index lists.

**Return type** list

`bpy_extras.mesh_utils.edge_loops_from_edges` (*mesh, edges=None*)

Edge loops defined by edges

Takes `me.edges` or a list of edges and returns the edge loops

return a list of vertex indices. `[1, 6, 7, 2], ...]`

closed loops have matching start and end values.

`bpy_extras.mesh_utils.ngon_tessellate` (*from\_data, indices, fix\_loops=True*)

Takes a polyline of indices (fgon) and returns a list of face indicie lists. Designed to be used for importers that need indices for an fgon to create from existing verts.

`from_data`: either a mesh, or a list/tuple of vectors. `indices`: a list of indices to use this list is the ordered closed polyline

to fill, and can be a subset of the data given.

**fix\_loops: If this is enabled polylines that use loops to make multiple** polylines are delt with correctly.

`bpy_extras.mesh_utils.face_random_points` (*num\_points, faces*)

Generates a list of random points over mesh faces.

**Parameters**

- **num\_points** – the number of random points to generate on each face.
- **faces** (`bpy.types.MeshFaces`, sequence) – list of the faces to generate points on.

**Returns** list of random points over all faces.

**Return type** list

### 3.7.7 bpy\_extras submodule (bpy\_extras.view3d\_utils)

`bpy_extras.view3d_utils.region_2d_to_vector_3d` (*region, rv3d, coord*)

Return a direction vector from the viewport at the specific 2d region coordinate.

**Parameters**

- **region** (`bpy.types.Region`) – region of the 3D viewport, typically `bpy.context.region`.
- **rv3d** (`bpy.types.RegionView3D`) – 3D region data, typically `bpy.context.space_data.region_3d`.
- **coord** (*2d vector*) – 2d coordinates relative to the region: (`event.mouse_region_x`, `event.mouse_region_y`) for example.

**Returns** normalized 3d vector.

**Return type** `mathutils.Vector`

`bpy_extras.view3d_utils.region_2d_to_location_3d` (*region, rv3d, coord, depth\_location*)

Return a 3d location from the region relative 2d coords, aligned with *depth\_location*.

**Parameters**

- **region** (`bpy.types.Region`) – region of the 3D viewport, typically `bpy.context.region`.
- **rv3d** (`bpy.types.RegionView3D`) – 3D region data, typically `bpy.context.space_data.region_3d`.
- **coord** (*2d vector*) – 2d coordinates relative to the region; (`event.mouse_region_x`, `event.mouse_region_y`) for example.
- **depth\_location** (*3d vector*) – the returned vectors depth is aligned with this since there is no defined depth with a 2d region input.

**Returns** normalized 3d vector.

**Return type** `mathutils.Vector`

`bpy_extras.view3d_utils.location_3d_to_region_2d` (*region, rv3d, coord*)

Return the *region* relative 2d location of a 3d position.

**Parameters**

- **region** (`bpy.types.Region`) – region of the 3D viewport, typically `bpy.context.region`.
- **rv3d** (`bpy.types.RegionView3D`) – 3D region data, typically `bpy.context.space_data.region_3d`.
- **coord** (*3d vector*) – 3d worldspace location.

**Returns** 2d location

**Return type** `mathutils.Vector`



---

# GAME ENGINE MODULES

## 4.1 Game Types (bge.types)

**class** `bge.types.PyObjectPlus`

PyObjectPlus base class of most other types in the Game Engine.

**invalid**

Test if the object has been freed by the game engine and is no longer valid.

Normally this is not a problem but when storing game engine data in the GameLogic module, `KX_Scenes` or other `KX_GameObjects` its possible to hold a reference to invalid data. Calling an attribute or method on an invalid object will raise a `SystemError`.

The invalid attribute allows testing for this case without exception handling.

**Type** boolean

**class** `bge.types.CValue` (*PyObjectPlus*)

This class is a basis for other classes.

**name**

The name of this CValue derived object (read-only).

**Type** string

**class** `bge.types.CPropValue` (*CValue*)

This class has no python functions

**class** `bge.types.SCA_ILogicBrick` (*CValue*)

Base class for all logic bricks.

**executePriority**

This determines the order controllers are evaluated, and actuators are activated (lower priority is executed first).

**Type** executePriority: int

**owner**

The game object this logic brick is attached to (read-only).

**Type** `KX_GameObject` or None in exceptional cases.

**name**

The name of this logic brick (read-only).

**Type** string

**class** `bge.types.SCA_PythonKeyboard` (*PyObjectPlus*)

The current keyboard.

**events**

A dictionary containing the status of each keyboard event or key. (read-only).

**Type** dictionary {*keycode:status*, ...}

**class** `bge.types.SCA_PythonMouse` (*PyObjectPlus*)

The current mouse.

**events**

a dictionary containing the status of each mouse event. (read-only).

**Type** dictionary {*keycode:status*, ...}

**position**

The normalized x and y position of the mouse cursor.

**Type** list [x, y]

**visible**

The visibility of the mouse cursor.

**Type** boolean

**class** `bge.types.SCA_IObject` (*CValue*)

This class has no python functions

**class** `bge.types.SCA_ISensor` (*SCA\_ILogicBrick*)

Base class for all sensor logic bricks.

**usePosPulseMode**

Flag to turn positive pulse mode on and off.

**Type** boolean

**useNegPulseMode**

Flag to turn negative pulse mode on and off.

**Type** boolean

**frequency**

The frequency for pulse mode sensors.

**Type** integer

**level**

level Option whether to detect level or edge transition when entering a state. It makes a difference only in case of logic state transition (state actuator). A level detector will immediately generate a pulse, negative or positive depending on the sensor condition, as soon as the state is activated. A edge detector will wait for a state change before generating a pulse. note: mutually exclusive with `tap`, enabling will disable `tap`.

**Type** boolean

**tap**

When enabled only sensors that are just activated will send a positive event, after this they will be detected as negative by the controllers. This will make a key thats held act as if its only tapped for an instant. note: mutually exclusive with `level`, enabling will disable `level`.

**Type** boolean

**invert**

Flag to set if this sensor activates on positive or negative events.



**Type** boolean

**triggered**

True if this sensor brick is in a positive state. (read-only).

**Type** boolean

**positive**

True if this sensor brick is in a positive state. (read-only).

**Type** boolean

**status**

The status of the sensor (read-only): can be one of *these constants*.

**Type** int

---

**Note:** This convenient attribute combines the values of triggered and positive attributes.

---

**reset** ()

Reset sensor internal state, effect depends on the type of sensor and settings.

The sensor is put in its initial state as if it was just activated.

**class** `bge.types.SCA_IController` (*SCA\_ILogicBrick*)

Base class for all controller logic bricks.

**state**

The controllers state bitmask. This can be used with the `GameObject`'s state to test if the controller is active.

**Type** int bitmask

**sensors**

A list of sensors linked to this controller.

**Type** sequence supporting index/string lookups and iteration.

---

**Note:** The sensors are not necessarily owned by the same object.

---



---

**Note:** When objects are instanced in dupligroups links may be lost from objects outside the dupligroup.

---

**actuators**

A list of actuators linked to this controller.

**Type** sequence supporting index/string lookups and iteration.

---

**Note:** The sensors are not necessarily owned by the same object.

---



---

**Note:** When objects are instanced in dupligroups links may be lost from objects outside the dupligroup.

---

**useHighPriority**

When set the controller executes always before all other controllers that dont have this set.

**Type** boolen

---

**Note:** Order of execution between high priority controllers is not guaranteed.

---

**class** `bge.types.SCA_IActor` (*SCAILogicBrick*)

Base class for all actuator logic bricks.

**class** `bge.types.BL_ActionActor` (*SCA\_IActor*)

Action Actuators apply an action to an actor.

**action**

The name of the action to set as the current action.

**Type** string

**channelNames**

A list of channel names that may be used with `setChannel` and `getChannel`.

**Type** list of strings

**frameStart**

Specifies the starting frame of the animation.

**Type** float

**frameEnd**

Specifies the ending frame of the animation.

**Type** float

**blendIn**

Specifies the number of frames of animation to generate when making transitions between actions.

**Type** float

**priority**

Sets the priority of this actuator. Actuators with lower priority numbers will override actuators with higher numbers.

**Type** integer

**frame**

Sets the current frame for the animation.

**Type** float

**propName**

Sets the property to be used in FromProp playback mode.

**Type** string

**blendTime**

Sets the internal frame timer. This property must be in the range from 0.0 to `blendIn`.

**Type** float

**mode**

The operation mode of the actuator. Can be one of *these constants*.

**Type** integer

**useContinue**

The actions continue option, True or False. When True, the action will always play from where last left off, otherwise negative events to this actuator will reset it to its start frame.

**Type** boolean

**framePropName**

The name of the property that is set to the current frame number.

**Type** string

**setChannel** (*channel, matrix*)

Alternative to the 2 arguments, 4 arguments (channel, matrix, loc, size, quat) are also supported.

**Parameters**

- **channel** (*string*) – A string specifying the name of the bone channel, error raised if not in `channelNames`.
- **matrix** – A 4x4 matrix specifying the overriding transformation as an offset from the bone's rest position.
- **matrix** – list `[[float]]`

---

**Note:** These values are relative to the bones rest position, currently the api has no way to get this info (which is annoying), but can be worked around by using bones with a rest pose that has no translation.

---

**getChannel** (*channel*)

**Parameters** **channel** (*string*) – A string specifying the name of the bone channel. error raised if not in `channelNames`.

**Returns** (loc, size, quat)

**Return type** tuple

**class** `bge.types.BL_Shader` (*PyObjectPlus*)

`BL_Shader` GLSL shaders.

TODO - Description

**setUniformfv** (*name, fList*)

Set a uniform with a list of float values

**Parameters**

- **name** (*string*) – the uniform name
- **fList** (*list[float]*) – a list (2, 3 or 4 elements) of float values

**delSource** ()

Clear the shader. Use this method before the source is changed with `setSource`.

**getFragmentProg** ()

Returns the fragment program.

**Returns** The fragment program.

**Return type** string

**getVertexProg** ()

Get the vertex program.

**Returns** The vertex program.

**Return type** string

**isValid** ()

Check if the shader is valid.

**Returns** True if the shader is valid

**Return type** boolean

**setAttrib** (*enum*)

Set attribute location. (The parameter is ignored a.t.m. and the value of “tangent” is always used.)

**Parameters** **enum** (*integer*) – attribute location value

**setNumberOfPasses** (*max\_pass*)

Set the maximum number of passes. Not used a.t.m.

**Parameters** **max\_pass** (*integer*) – the maximum number of passes

**setSampler** (*name, index*)

Set uniform texture sample index.

**Parameters**

- **name** (*string*) – Uniform name
- **index** (*integer*) – Texture sample index.

**setSource** (*vertexProgram, fragmentProgram*)

Set the vertex and fragment programs

**Parameters**

- **vertexProgram** (*string*) – Vertex program
- **fragmentProgram** (*string*) – Fragment program

**setUniform1f** (*name, fx*)

Set a uniform with 1 float value.

**Parameters**

- **name** (*string*) – the uniform name
- **fx** (*float*) – Uniform value

**setUniform1i** (*name, ix*)

Set a uniform with an integer value.

**Parameters**

- **name** (*string*) – the uniform name
- **ix** (*integer*) – the uniform value

**setUniform2f** (*name, fx, fy*)

Set a uniform with 2 float values

**Parameters**

- **name** (*string*) – the uniform name
- **fx** (*float*) – first float value
- **fy** (*float*) – second float value

**setUniform2i** (*name, ix, iy*)

Set a uniform with 2 integer values

**Parameters**

- **name** (*string*) – the uniform name
- **ix** (*integer*) – first integer value
- **iy** (*integer*) – second integer value

**setUniform3f** (*name, fx, fy, fz*)

Set a uniform with 3 float values.

**Parameters**

- **name** (*string*) – the uniform name
- **fx** (*float*) – first float value
- **fy** (*float*) – second float value
- **fz** (*float*) – third float value

**setUniform3i** (*name, ix, iy, iz*)

Set a uniform with 3 integer values

**Parameters**

- **name** (*string*) – the uniform name
- **ix** (*integer*) – first integer value
- **iy** (*integer*) – second integer value
- **iz** (*integer*) – third integer value

**setUniform4f** (*name, fx, fy, fz, fw*)

Set a uniform with 4 float values.

**Parameters**

- **name** (*string*) – the uniform name
- **fx** (*float*) – first float value
- **fy** (*float*) – second float value
- **fz** (*float*) – third float value
- **fw** (*float*) – fourth float value

**setUniform4i** (*name, ix, iy, iz, iw*)

Set a uniform with 4 integer values

**Parameters**

- **name** (*string*) – the uniform name
- **ix** (*integer*) – first integer value
- **iy** (*integer*) – second integer value
- **iz** (*integer*) – third integer value
- **iw** (*integer*) – fourth integer value

**setUniformDef** (*name, type*)

Define a new uniform

**Parameters**

- **name** (*string*) – the uniform name
- **type** (*UNI\_NONE, UNI\_INT, UNI\_FLOAT, UNI\_INT2, UNI\_FLOAT2, UNI\_INT3, UNI\_FLOAT3, UNI\_INT4, UNI\_FLOAT4, UNI\_MAT3, UNI\_MAT4, UNI\_MAX*) – uniform type

**setUniformMatrix3** (*name, mat, transpose*)

Set a uniform with a 3x3 matrix value

**Parameters**

- **name** (*string*) – the uniform name
- **mat** (*3x3 matrix*) – A 3x3 matrix [[f, f, f], [f, f, f], [f, f, f]]
- **transpose** (*boolean*) – set to True to transpose the matrix

**setUniformMatrix4** (*name, mat, transpose*)

Set a uniform with a 4x4 matrix value

**Parameters**

- **name** (*string*) – the uniform name
- **mat** (*4x4 matrix*) – A 4x4 matrix [[f, f, f, f], [f, f, f, f], [f, f, f, f], [f, f, f, f]]
- **transpose** (*boolean*) – set to True to transpose the matrix

**setUniformiv** (*name, iList*)

Set a uniform with a list of integer values

**Parameters**

- **name** (*string*) – the uniform name
- **iList** (*list[integer]*) – a list (2, 3 or 4 elements) of integer values

**validate** ()

Validate the shader object.

**class** `bge.types.BL_ShapeActionActuator` (*SCA\_IActuator*)

ShapeAction Actuators apply an shape action to an mesh object.

**action**

The name of the action to set as the current shape action.

**Type** string

**frameStart**

Specifies the starting frame of the shape animation.

**Type** float

**frameEnd**

Specifies the ending frame of the shape animation.

**Type** float

**blendIn**

Specifies the number of frames of animation to generate when making transitions between actions.

**Type** float

**priority**

Sets the priority of this actuator. Actuators will lower priority numbers will override actuators with higher numbers.

**Type** integer

**frame**

Sets the current frame for the animation.

**Type** float

**propName**

Sets the property to be used in FromProp playback mode.

**Type** string

**blendTime**

Sets the internal frame timer. This property must be in the range from 0.0 to `blendin`.

**Type** float

**mode**

The operation mode of the actuator. Can be one of *these constants*.

**Type** integer

**framePropName**

The name of the property that is set to the current frame number.

**Type** string

**class** `bge.types.CListValue` (*CPropValue*)

This is a list like object used in the game engine internally that behaves similar to a python list in most ways.

As well as the normal index lookup (`val= clist[i]`), `CListValue` supports string lookups (`val= scene.objects["Cube"]`)

Other operations such as `len(clist)`, `list(clist)`, `clist[0:10]` are also supported.

**append** (*val*)

Add an item to the list (like python's `append`)

**Warning:** Appending values to the list can cause crashes when the list is used internally by the game engine.

**count** (*val*)

Count the number of instances of a value in the list.

**Returns** number of instances

**Return type** integer

**index** (*val*)

Return the index of a value in the list.

**Returns** The index of the value in the list.

**Return type** integer

**reverse** ()

Reverse the order of the list.

**get** (*key, default=None*)

Return the value matching key, or the default value if its not found.

**Returns** The key value or a default.

**from\_id** (*id*)

This is a funtion especially for the game engine to return a value with a specific id.

Since object names are not always unique, the id of an object can be used to get an object from the `CValueList`.

Example:

```
myObID=id(gameObject)
ob= scene.objects.from_id(myObID)
```

Where `myObID` is an int or long from the `id` function.

This has the advantage that you can store the id in places you could not store a `gameObject`.

**Warning:** The id is derived from a memory location and will be different each time the game engine starts.

**class** `bge.types.KX_BlenderMaterial` (*PyObjectPlus*)  
`KX_BlenderMaterial`

**getShader** ()

Returns the material's shader.

**Returns** the material's shader

**Return type** `BL_Shader`

**setBlending** (*src, dest*)

Set the pixel color arithmetic functions.

**Parameters**

- **src** – Specifies how the red, green, blue, and alpha source blending factors are computed.
- **dest** – Specifies how the red, green, blue, and alpha destination blending factors are computed.

**getMaterialIndex** ()

Returns the material's index.

**Returns** the material's index

**Return type** integer

**class** `bge.types.KX_CameraActuator` (*SCA\_IActuator*)  
Applies changes to a camera.

**damping**

strength of of the camera following movement.

**Type** float

**min**

minimum distance to the target object maintained by the actuator.

**Type** float

**max**

maximum distance to stay from the target object.

**Type** float

**height**

height to stay above the target object.

**Type** float

**useXY**

axis this actuator is tracking, True=X, False=Y.

**Type** boolean

**object**

the object this actuator tracks.



**Type** `KX_GameObject` or `None`

**class** `bge.types.KX_ConstraintActuator` (*SCA\_IActuator*)

A constraint actuator limits the position, rotation, distance or orientation of an object.

**damp**

Time constant of the constraint expressed in frame (not use by Force field constraint).

**Type** `integer`

**rotDamp**

Time constant for the rotation expressed in frame (only for the distance constraint), 0 = use damp for rotation as well.

**Type** `integer`

**direction**

The reference direction in world coordinate for the orientation constraint.

**Type** 3-tuple of float: (x, y, z)

**option**

Binary combination of *these constants*

**Type** `integer`

**time**

activation time of the actuator. The actuator disables itself after this many frame. If set to 0, the actuator is not limited in time.

**Type** `integer`

**propName**

the name of the property or material for the ray detection of the distance constraint.

**Type** `string`

**min**

The lower bound of the constraint. For the rotation and orientation constraint, it represents radiant.

**Type** `float`

**distance**

the target distance of the distance constraint.

**Type** `float`

**max**

the upper bound of the constraint. For rotation and orientation constraints, it represents radiant.

**Type** `float`

**rayLength**

the length of the ray of the distance constraint.

**Type** `float`

**limit**

type of constraint. Use one of the *these constants*

**Type** `integer`.

**class** `bge.types.KX_ConstraintWrapper` (*PyObjectPlus*)

`KX_ConstraintWrapper`

**getConstraintId** (*val*)

Returns the constraint's ID

**Returns** the constraint's ID

**Return type** integer

**class** `bge.types.KX_GameActuator` (*SCA\_IActuator*)

The game actuator loads a new .blend file, restarts the current .blend file or quits the game.

**fileName**

the new .blend file to load.

**Type** string

**mode**

The mode of this actuator. Can be on of *these constants*

**Type** Int

**class** `bge.types.KX_GameObject` (*SCA\_IObject*)

All game objects are derived from this class.

Properties assigned to game objects are accessible as attributes of this class.

---

**Note:** Calling ANY method or attribute on an object that has been removed from a scene will raise a SystemError, if an object may have been removed since last accessing it use the `invalid` attribute to check.

---

**name**

The object's name. (read-only).

**Type** string

**mass**

The object's mass

**Type** float

---

**Note:** The object must have a physics controller for the mass to be applied, otherwise the mass value will be returned as 0.0.

---

**linVelocityMin**

Enforces the object keeps moving at a minimum velocity.

**Type** float

---

**Note:** Applies to dynamic and rigid body objects only.

---

**Note:** A value of 0.0 disables this option.

---

**Note:** While objects are stationary the minimum velocity will not be applied.

---

**linVelocityMax**

Clamp the maximum linear velocity to prevent objects moving beyond a set speed.

**Type** float

---

**Note:** Applies to dynamic and rigid body objects only.

---

**Note:** A value of 0.0 disables this option (rather than setting it stationary).

---

**localInertia**

the object's inertia vector in local coordinates. Read only.

**Type** list [ix, iy, iz]

**parent**

The object's parent object. (read-only).

**Type** `KX_GameObject` or None

**visible**

visibility flag.

**Type** boolean

---

**Note:** Game logic will still run for invisible objects.

---

**color**

The object color of the object. [r, g, b, a]

**Type** `mathutils.Vector`

**occlusion**

occlusion capability flag.

**Type** boolean

**position**

The object's position. [x, y, z] On write: local position, on read: world position Deprecated since version use: `localPosition` and `worldPosition`.

**Type** `mathutils.Vector`

**orientation**

The object's orientation. 3x3 Matrix. You can also write a Quaternion or Euler vector. On write: local orientation, on read: world orientation Deprecated since version use: `localOrientation` and `worldOrientation`.

**Type** `mathutils.Matrix`

**scaling**

The object's scaling factor. [sx, sy, sz] On write: local scaling, on read: world scaling Deprecated since version use: `localScale` and `worldScale`.

**Type** `mathutils.Vector`

**localOrientation**

The object's local orientation. 3x3 Matrix. You can also write a Quaternion or Euler vector.

**Type** `mathutils.Matrix`

**worldOrientation**

The object's world orientation. 3x3 Matrix.

**Type** `mathutils.Matrix`

**localScale**

The object's local scaling factor. [sx, sy, sz]

**Type** `mathutils.Vector`

**worldScale**

The object's world scaling factor. Read-only. [sx, sy, sz]

**Type** `mathutils.Vector`

**localPosition**

The object's local position. [x, y, z]

**Type** `mathutils.Vector`

**worldPosition**

The object's world position. [x, y, z]

**Type** `mathutils.Vector`

**localLinearVelocity**

The object's local linear velocity. [x, y, z]

**Type** `mathutils.Vector`

**worldLinearVelocity**

The object's world linear velocity. [x, y, z]

**type** `mathutils.Vector`

**localAngularVelocity**

The object's local angular velocity. [x, y, z]

**type** `mathutils.Vector`

**worldAngularVelocity**

The object's world angular velocity. [x, y, z]

**type** `mathutils.Vector`

**timeOffset**

adjust the slowparent delay at runtime.

**Type** `float`

**state**

the game object's state bitmask, using the first 30 bits, one bit must always be set.

**Type** `int`

**meshes**

a list meshes for this object.

**Type** list of `KX_MeshProxy`

---

**Note:** Most objects use only 1 mesh.

---

---

**Note:** Changes to this list will not update the `KX_GameObject`.

---

**sensors**

a sequence of `SCA_ISensor` objects with string/index lookups and iterator support.

**Type** `list`

---

**Note:** This attribute is experimental and may be removed (but probably won't be).

---

**Note:** Changes to this list will not update the `KX_GameObject`.

---

### controllers

a sequence of `SCA_IController` objects with string/index lookups and iterator support.

**Type** list of `SCA_ISensor`

---

**Note:** This attribute is experimental and may be removed (but probably won't be).

---

**Note:** Changes to this list will not update the `KX_GameObject`.

---

### actuators

a list of `SCA_IActuator` with string/index lookups and iterator support.

**Type** list

---

**Note:** This attribute is experimental and may be removed (but probably won't be).

---

**Note:** Changes to this list will not update the `KX_GameObject`.

---

### attrDict

get the objects internal python attribute dictionary for direct (faster) access.

**Type** dict

### children

direct children of this object, (read-only).

**Type** `CListValue` of `KX_GameObject`'s

### childrenRecursive

all children of this object including childrens children, (read-only).

**Type** `CListValue` of `KX_GameObject`'s

### endObject ()

Delete this object, can be used in place of the EndObject Actuator.

The actual removal of the object from the scene is delayed.

### replaceMesh (*mesh, useDisplayMesh=True, usePhysicsMesh=False*)

Replace the mesh of this object with a new mesh. This works the same was as the actuator.

#### Parameters

- **mesh** (`MeshProxy` or string) – mesh to replace or the meshes name.
- **useDisplayMesh** (*boolean*) – when enabled the display mesh will be replaced (optional argument).
- **usePhysicsMesh** (*boolean*) – when enabled the physics mesh will be replaced (optional argument).

**setVisible** (*visible, recursive*)

Sets the game object's visible flag.

**Parameters**

- **visible** (*boolean*) – the visible state to set.
- **recursive** (*boolean*) – optional argument to set all childrens visibility flag too.

**setOcclusion** (*occlusion, recursive*)

Sets the game object's occlusion capability.

**Parameters**

- **occlusion** (*boolean*) – the state to set the occlusion to.
- **recursive** (*boolean*) – optional argument to set all childrens occlusion flag too.

**alignAxisToVect** (*vect, axis=2, factor=1.0*)

Aligns any of the game object's axis along the given vector.

**Parameters**

- **vect** (*3D vector*) – a vector to align the axis.
- **axis** (*integer*) – The axis you want to align
  - 0: X axis
  - 1: Y axis
  - 2: Z axis
- **factor** (*float*) – Only rotate a feaction of the distance to the target vector (0.0 - 1.0)

**getAxisVect** (*vect*)

Returns the axis vector rotates by the objects worldspace orientation. This is the equivalent of multiplying the vector by the orientation matrix.

**Parameters** **vect** (*3D Vector*) – a vector to align the axis.

**Returns** The vector in relation to the objects rotation.

**Return type** 3d vector.

**applyMovement** (*movement, local=False*)

Sets the game object's movement.

**Parameters**

- **movement** (*3D Vector*) – movement vector.
- **local** –
  - False: you get the “global” movement ie: relative to world orientation.
  - True: you get the “local” movement ie: relative to object orientation.
- **local** – boolean

**applyRotation** (*rotation, local=False*)

Sets the game object's rotation.

**Parameters**

- **rotation** (*3D Vector*) – rotation vector.
- **local** –

- False: you get the “global” rotation ie: relative to world orientation.
- True: you get the “local” rotation ie: relative to object orientation.

- **local** – boolean

**applyForce** (*force*, *local=False*)

Sets the game object’s force.

This requires a dynamic object.

#### Parameters

- **force** (*3D Vector*) – force vector.
- **local** (*boolean*) –
  - False: you get the “global” force ie: relative to world orientation.
  - True: you get the “local” force ie: relative to object orientation.

**applyTorque** (*torque*, *local=False*)

Sets the game object’s torque.

This requires a dynamic object.

#### Parameters

- **torque** (*3D Vector*) – torque vector.
- **local** (*boolean*) –
  - False: you get the “global” torque ie: relative to world orientation.
  - True: you get the “local” torque ie: relative to object orientation.

**getLinearVelocity** (*local=False*)

Gets the game object’s linear velocity.

This method returns the game object’s velocity through it’s centre of mass, ie no angular velocity component.

**Parameters** **local** (*boolean*) –

- False: you get the “global” velocity ie: relative to world orientation.
- True: you get the “local” velocity ie: relative to object orientation.

**Returns** the object’s linear velocity.

**Return type** list [vx, vy, vz]

**setLinearVelocity** (*velocity*, *local=False*)

Sets the game object’s linear velocity.

This method sets game object’s velocity through it’s centre of mass, ie no angular velocity component.

This requires a dynamic object.

#### Parameters

- **velocity** (*3D Vector*) – linear velocity vector.
- **local** (*boolean*) –
  - False: you get the “global” velocity ie: relative to world orientation.
  - True: you get the “local” velocity ie: relative to object orientation.

**getAngularVelocity** (*local=False*)

Gets the game object's angular velocity.

**Parameters** *local* (*boolean*) –

- False: you get the “global” velocity ie: relative to world orientation.
- True: you get the “local” velocity ie: relative to object orientation.

**Returns** the object's angular velocity.

**Return type** list [vx, vy, vz]

**setAngularVelocity** (*velocity, local=False*)

Sets the game object's angular velocity.

This requires a dynamic object.

**Parameters**

- **velocity** (*boolean*) – angular velocity vector.
- **local** –
  - False: you get the “global” velocity ie: relative to world orientation.
  - True: you get the “local” velocity ie: relative to object orientation.

**getVelocity** (*point=(0, 0, 0)*)

Gets the game object's velocity at the specified point.

Gets the game object's velocity at the specified point, including angular components.

**Parameters** *point* (*3D Vector*) – optional point to return the velocity for, in local coordinates.

**Returns** the velocity at the specified point.

**Return type** list [vx, vy, vz]

**getReactionForce** ()

Gets the game object's reaction force.

The reaction force is the force applied to this object over the last simulation timestep. This also includes impulses, eg from collisions.

**Returns** the reaction force of this object.

**Return type** list [fx, fy, fz]

---

**Note:** This is not implimented at the moment.

---

**applyImpulse** (*point, impulse*)

Applies an impulse to the game object.

This will apply the specified impulse to the game object at the specified point. If *point != position*, *applyImpulse* will also change the object's angular momentum. Otherwise, only linear momentum will change.

**Parameters** *point* (*the point to apply the impulse to (in world coordinates)*) – the point to apply the impulse to (in world coordinates)

**suspendDynamics** ()

Suspends physics for this object.



**restoreDynamics ()**

Resumes physics for this object.

---

**Note:** The objects linear velocity will be applied from when the dynamics were suspended.

---

**enableRigidBody ()**

Enables rigid body physics for this object.

Rigid body physics allows the object to roll on collisions.

---

**Note:** This is not working with bullet physics yet.

---

**disableRigidBody ()**

Disables rigid body physics for this object.

---

**Note:** This is not working with bullet physics yet. The angular is removed but rigid body physics can still rotate it later.

---

**setParent (parent, compound=True, ghost=True)**

Sets this object's parent. Control the shape status with the optional compound and ghost parameters:

In that case you can control if it should be ghost or not:

**Parameters**

- **parent** (*KX\_GameObject*) – new parent object.
  - **compound** (*boolean*) – whether the shape should be added to the parent compound shape.
    - True: the object shape should be added to the parent compound shape.
    - False: the object should keep its individual shape.
  - **ghost** (*boolean*) – whether the object should be ghost while parented.
    - True: if the object should be made ghost while parented.
    - False: if the object should be solid while parented.
- 

**Note:** If the object type is sensor, it stays ghost regardless of ghost parameter

---

**removeParent ()**

Removes this objects parent.

**getPhysicsId ()**

Returns the user data object associated with this game object's physics controller.

**getPropertyNames ()**

Gets a list of all property names.

**Returns** All property names for this object.

**Return type** list

**getDistanceTo (other)**

**Parameters other** (*KX\_GameObject* or list [x, y, z]) – a point or another *KX\_GameObject* to measure the distance to.

**Returns** distance to another object or point.

**Return type** float

**getVectTo** (*other*)

Returns the vector and the distance to another object or point. The vector is normalized unless the distance is 0, in which a zero length vector is returned.

**Parameters** **other** (`KX_GameObject` or list [x, y, z]) – a point or another `KX_GameObject` to get the vector and distance to.

**Returns** (distance, `globalVector(3)`, `localVector(3)`)

**Return type** 3-tuple (float, 3-tuple (x, y, z), 3-tuple (x, y, z))

**rayCastTo** (*other, dist, prop*)

Look towards another point/object and find first object hit within dist that matches prop.

The ray is always casted from the center of the object, ignoring the object itself. The ray is casted towards the center of another object or an explicit [x, y, z] point. Use `rayCast()` if you need to retrieve the hit point

**Parameters**

- **other** (`KX_GameObject` or 3-tuple) – [x, y, z] or object towards which the ray is casted
- **dist** (*float*) – max distance to look (can be negative => look behind); 0 or omitted => detect up to other
- **prop** (*string*) – property name that object must have; can be omitted => detect any object

**Returns** the first object hit or None if no object or object does not match prop

**Return type** `KX_GameObject`

**rayCast** (*objto, objfrom, dist, prop, face, xray, poly*)

Look from a point/object to another point/object and find first object hit within dist that matches prop. if poly is 0, returns a 3-tuple with object reference, hit point and hit normal or (None, None, None) if no hit. if poly is 1, returns a 4-tuple with in addition a `KX_PolyProxy` as 4th element. if poly is 2, returns a 5-tuple with in addition a 2D vector with the UV mapping of the hit point as 5th element.

```
# shoot along the axis gun-gunAim (gunAim should be collision-free)
obj, point, normal = gun.rayCast(gunAim, None, 50)
if obj:
    # do something
    pass
```

The face parameter determines the orientation of the normal.

- 0 => hit normal is always oriented towards the ray origin (as if you casted the ray from outside)
- 1 => hit normal is the real face normal (only for mesh object, otherwise face has no effect)

The ray has X-Ray capability if xray parameter is 1, otherwise the first object hit (other than self object) stops the ray. The prop and xray parameters interact as follow.

- prop off, xray off: return closest hit or no hit if there is no object on the full extend of the ray.
- prop off, xray on : idem.
- prop on, xray off: return closest hit if it matches prop, no hit otherwise.
- prop on, xray on : return closest hit matching prop or no hit if there is no object matching prop on the full extend of the ray.

The `KX_PolyProxy` 4th element of the return tuple when `poly=1` allows to retrieve information on the polygon hit by the ray. If there is no hit or the hit object is not a static mesh, `None` is returned as 4th element.

The ray ignores collision-free objects and faces that dont have the collision flag enabled, you can however use ghost objects.

### Parameters

- **objto** (`KX_GameObject` or 3-tuple) – [x, y, z] or object to which the ray is casted
- **objfrom** (`KX_GameObject` or 3-tuple or `None`) – [x, y, z] or object from which the ray is casted; `None` or omitted => use self object center
- **dist** (*float*) – max distance to look (can be negative => look behind); 0 or omitted => detect up to to
- **prop** (*string*) – property name that object must have; can be omitted or "" => detect any object
- **face** (*integer*) – normal option: 1=>return face normal; 0 or omitted => normal is oriented towards origin
- **xray** (*integer*) – X-ray option: 1=>skip objects that don't match prop; 0 or omitted => stop on first object
- **poly** (*integer*) – polygon option: 0, 1 or 2 to return a 3-, 4- or 5-tuple with information on the face hit.
  - 0 or omitted: return value is a 3-tuple (object, hitpoint, hitnormal) or (`None`, `None`, `None`) if no hit
  - 1: return value is a 4-tuple and the 4th element is a `KX_PolyProxy` or `None` if no hit or the object doesn't use a mesh collision shape.
  - 2: return value is a 5-tuple and the 5th element is a 2-tuple (u, v) with the UV mapping of the hit point or `None` if no hit, or the object doesn't use a mesh collision shape, or doesn't have a UV mapping.

### Returns

(object, hitpoint, hitnormal) or (object, hitpoint, hitnormal, polygon) or (object, hitpoint, hitnormal, polygon, hituv).

- object, hitpoint and hitnormal are `None` if no hit.
- polygon is valid only if the object is valid and is a static object, a dynamic object using mesh collision shape or a soft body object, otherwise it is `None`
- hituv is valid only if polygon is valid and the object has a UV mapping, otherwise it is `None`

### Return type

- 3-tuple (`KX_GameObject`, 3-tuple (x, y, z), 3-tuple (nx, ny, nz))
- or 4-tuple (`KX_GameObject`, 3-tuple (x, y, z), 3-tuple (nx, ny, nz), `PolyProxy`)
- or 5-tuple (`KX_GameObject`, 3-tuple (x, y, z), 3-tuple (nx, ny, nz), `PolyProxy`, 2-tuple (u, v))

---

**Note:** The ray ignores the object on which the method is called. It is casted from/to object center or explicit [x, y, z] points.

---

**setCollisionMargin** (*margin*)  
Set the objects collision margin.

**Parameters** *margin* (*float*) – the collision margin distance in blender units.

---

**Note:** If this object has no physics controller (a physics ID of zero), this function will raise RuntimeError.

---

**sendMessage** (*subject*, *body=""*, *to=""*)  
Sends a message.

**Parameters**

- **subject** (*string*) – The subject of the message
- **body** (*string*) – The body of the message (optional)
- **to** (*string*) – The name of the object to send the message to (optional)

**reinstancePhysicsMesh** (*gameObject*, *meshObject*)  
Updates the physics system with the changed mesh.

If no arguments are given the physics mesh will be re-created from the first mesh assigned to the game object.

**Parameters**

- **gameObject** (*string*, *KX\_GameObject* or *None*) – optional argument, set the physics shape from this gameObjects mesh.
- **meshObject** (*string*, *MeshProxy* or *None*) – optional argument, set the physics shape from this mesh.

**Returns** True if reinstance succeeded, False if it failed.

**Return type** boolean

---

**Note:** If this object has instances the other instances will be updated too.

---

**Note:** The gameObject argument has an advantage that it can convert from a mesh with modifiers applied (such as subsurf).

---

**Warning:** Only triangle mesh type objects are supported currently (not convex hull)

**Warning:** If the object is a part of a compound object it will fail (parent or child)

**Warning:** Rebuilding the physics mesh can be slow, running many times per second will give a performance hit.

**get** (*key*, *default=None*)

Return the value matching key, or the default value if its not found. :return: The key value or a default.

**playAction** (*name*, *start\_frame*, *end\_frame*, *layer=0*, *priority=0*, *blendin=0*,  
*play\_mode=ACT\_MODE\_PLAY*, *layer\_weight=0.0*, *ipo\_flags=0*, *speed=1.0*)  
Plays an action.

**Parameters**

- **name** (*string*) – the name of the action
- **start** (*float*) – the start frame of the action
- **end** (*float*) – the end frame of the action
- **layer** (*integer*) – the layer the action will play in (actions in different layers are added/blended together)
- **priority** (*integer*) – only play this action if there isn't an action currently playing in this layer with a higher (lower number) priority
- **blendin** (*float*) – the amount of blending between this animation and the previous one on this layer
- **play\_mode** (*KX\_ACTION\_MODE\_PLAY, KX\_ACTION\_MODE\_LOOP, or KX\_ACTION\_MODE\_PING\_PONG*) – the play mode
- **layer\_weight** (*float*) – how much of the previous layer to use for blending (0 = add)
- **ipo\_flags** (*int bitfield*) – flags for the old IPO behaviors (force, etc)
- **speed** (*float*) – the playback speed of the action as a factor (1.0 = normal speed, 2.0 = 2x speed, etc)

**stopAction** (*layer=0*)

Stop playing the action on the given layer.

**Parameters** **layer** (*integer*) – The layer to stop playing.

**getActionFrame** (*layer=0*)

Gets the current frame of the action playing in the supplied layer.

**Parameters** **layer** (*integer*) – The layer that you want to get the frame from.

**Returns** The current frame of the action

**Return type** float

**setActionFrame** (*frame, layer=0*)

Set the current frame of the action playing in the supplied layer.

**Parameters**

- **layer** (*integer*) – The layer where you want to set the frame
- **frame** (*float*) – The frame to set the action to

**isPlayingAction** (*layer=0*)

Checks to see if there is an action playing in the given layer.

**Parameters** **layer** (*integer*) – The layer to check for a playing action.

**Returns** Whether or not the action is playing

**Return type** boolean

**class** `bge.types.KX_IpoActuator` (*SCA\_IActuator*)

IPO actuator activates an animation.

**frameStart**

Start frame.

**Type** float

**frameEnd**

End frame.

**Type** float

**propName**

Use this property to define the Ipo position.

**Type** string

**framePropName**

Assign this property this action current frame number.

**Type** string

**mode**

Play mode for the ipo. Can be on of *these constants*

**Type** integer

**useIpoAsForce**

Apply Ipo as a global or local force depending on the local option (dynamic objects only).

**Type** boolean

**useIpoAdd**

Ipo is added to the current loc/rot/scale in global or local coordinate according to Local flag.

**Type** boolean

**useIpoLocal**

Let the ipo acts in local coordinates, used in Force and Add mode.

**Type** boolean

**useChildren**

Update IPO on all children Objects as well.

**Type** boolean

**class** `bge.types.KX_LightObject` (*KX\_GameObject*)

A Light object.

```
# Turn on a red alert light.
```

```
import bge
```

```
co = bge.logic.getCurrentController()
```

```
light = co.owner
```

```
light.energy = 1.0
```

```
light.colour = [1.0, 0.0, 0.0]
```

**SPOT**

A spot light source. See attribute `type`

**SUN**

A point light source with no attenuation. See attribute `type`

**NORMAL**

A point light source. See attribute `type`

**type**

The type of light - must be SPOT, SUN or NORMAL

**layer**

The layer mask that this light affects object on.

**Type** bitfield

**energy**

The brightness of this light.

**Type** float

**distance**

The maximum distance this light can illuminate. (SPOT and NORMAL lights only).

**Type** float

**colour**

The colour of this light. Black = [0.0, 0.0, 0.0], White = [1.0, 1.0, 1.0].

**Type** list [r, g, b]

**color**

Synonym for colour.

**lin\_attenuation**

The linear component of this light's attenuation. (SPOT and NORMAL lights only).

**Type** float

**quad\_attenuation**

The quadratic component of this light's attenuation (SPOT and NORMAL lights only).

**Type** float

**spotsize**

The cone angle of the spot light, in degrees (SPOT lights only).

**Type** float in [0 - 180].

**spotblend**

Specifies the intensity distribution of the spot light (SPOT lights only).

**Type** float in [0 - 1]

---

**Note:** Higher values result in a more focused light source.

---

**class** `bge.types.KX_MeshProxy` (*SCA\_IObject*)

A mesh object.

You can only change the vertex properties of a mesh object, not the mesh topology.

To use mesh objects effectively, you should know a bit about how the game engine handles them.

- 1.Mesh Objects are converted from Blender at scene load.
- 2.The Converter groups polygons by Material. This means they can be sent to the renderer efficiently. A material holds:
  - (a)The texture.
  - (b)The Blender material.
  - (c)The Tile properties
  - (d)The face properties - (From the "Texture Face" panel)
  - (e)Transparency & z sorting
  - (f)Light layer
  - (g)Polygon shape (triangle/quad)

(h)Game Object

3.Verticies will be split by face if necessary. Verticies can only be shared between faces if:

(a)They are at the same position

(b)UV coordinates are the same

(c)Their normals are the same (both polygons are “Set Smooth”)

(d)They are the same colour, for example: a cube has 24 verticies: 6 faces with 4 verticies per face.

The correct method of iterating over every `KX_VertexProxy` in a game object

```
import GameLogic

co = GameLogic.getCurrentController()
obj = co.owner

m_i = 0
mesh = obj.getMesh(m_i) # There can be more than one mesh...
while mesh != None:
    for mat in range(mesh.getNumMaterials()):
        for v_index in range(mesh.getVertexArrayLength(mat)):
            vertex = mesh.getVertex(mat, v_index)
            # Do something with vertex here...
            # ... eg: colour the vertex red.
            vertex.colour = [1.0, 0.0, 0.0, 1.0]
        m_i += 1
    mesh = obj.getMesh(m_i)
```

### materials

**Type** list of `KX_BlenderMaterial` or `KX_PolygonMaterial` types

### numPolygons

**Type** integer

### numMaterials

**Type** integer

### getNumMaterials()

**Returns** number of materials associated with this object

**Return type** integer

### getMaterialName(matid)

Gets the name of the specified material.

**Parameters** `matid` (*integer*) – the specified material.

**Returns** the attached material name.

**Return type** string

### getTextureName(matid)

Gets the name of the specified material’s texture.

**Parameters** `matid` (*integer*) – the specified material

**Returns** the attached material’s texture name.

**Return type** string



**getVertexArrayLength** (*matid*)

Gets the length of the vertex array associated with the specified material.

There is one vertex array for each material.

**Parameters** **matid** (*integer*) – the specified material

**Returns** the number of vertices in the vertex array.

**Return type** integer

**getVertex** (*matid, index*)

Gets the specified vertex from the mesh object.

**Parameters**

- **matid** (*integer*) – the specified material
- **index** (*integer*) – the index into the vertex array.

**Returns** a vertex object.

**Return type** `KX_VertexProxy`

**getNumPolygons** ()

**Returns** The number of polygon in the mesh.

**Return type** integer

**getPolygon** (*index*)

Gets the specified polygon from the mesh.

**Parameters** **index** (*integer*) – polygon number

**Returns** a polygon object.

**Return type** `PolyProxy`

**class** `bge.types.SCA_MouseSensor` (*SCA\_ISensor*)

Mouse Sensor logic brick.

**position**

current [x, y] coordinates of the mouse, in frame coordinates (pixels).

**Type** [integer, integer]

**mode**

sensor mode.

**Type** integer

- `KX_MOUSESENSORMODE_LEFTBUTTON(1)`
- `KX_MOUSESENSORMODE_MIDDLEBUTTON(2)`
- `KX_MOUSESENSORMODE_RIGHTBUTTON(3)`
- `KX_MOUSESENSORMODE_WHEELUP(4)`
- `KX_MOUSESENSORMODE_WHEELDOWN(5)`
- `KX_MOUSESENSORMODE_MOVEMENT(6)`

**getButtonStatus** (*button*)

Get the mouse button status.

**Parameters** **button** (*int*) – The code that represents the key you want to get the state of, use one of *these constants*

**Returns** The state of the given key, can be one of *these constants*

**Return type** int

**class** `bge.types.KX_MouseFocusSensor` (*SCA\_MouseSensor*)

The mouse focus sensor detects when the mouse is over the current game object.

The mouse focus sensor works by transforming the mouse coordinates from 2d device space to 3d space then raycasting away from the camera.

**raySource**

The worldspace source of the ray (the view position).

**Type** list (vector of 3 floats)

**rayTarget**

The worldspace target of the ray.

**Type** list (vector of 3 floats)

**rayDirection**

The `rayTarget - raySource` normalized.

**Type** list (normalized vector of 3 floats)

**hitObject**

the last object the mouse was over.

**Type** `KX_GameObject` or None

**hitPosition**

The worldspace position of the ray intersection.

**Type** list (vector of 3 floats)

**hitNormal**

the worldspace normal from the face at point of intersection.

**Type** list (normalized vector of 3 floats)

**hitUV**

the UV coordinates at the point of intersection.

**Type** list (vector of 2 floats)

If the object has no UV mapping, it returns [0, 0].

The UV coordinates are not normalized, they can be < 0 or > 1 depending on the UV mapping.

**usePulseFocus**

When enabled, moving the mouse over a different object generates a pulse. (only used when the 'Mouse Over Any' sensor option is set).

**Type** boolean

**class** `bge.types.KX_TouchSensor` (*SCA\_ISensor*)

Touch sensor detects collisions between objects.

**propName**

The property or material to collide with.

**Type** string

**useMaterial**

Determines if the sensor is looking for a property or material. `KX_True` = Find material; `KX_False` = Find property.

**Type** boolean

**usePulseCollision**

When enabled, changes to the set of colliding objects generate a pulse.

**Type** boolean

**hitObject**

The last collided object. (read-only).

**Type** `KX_GameObject` or None

**hitObjectList**

A list of colliding objects. (read-only).

**Type** `CListValue` of `KX_GameObject`

**class** `bge.types.KX_NearSensor` (*`KX_TouchSensor`*)

A near sensor is a specialised form of touch sensor.

**distance**

The near sensor activates when an object is within this distance.

**Type** float

**resetDistance**

The near sensor deactivates when the object exceeds this distance.

**Type** float

**class** `bge.types.KX_NetworkMessageActuator` (*`SCA_IActuator`*)

Message Actuator

**propName**

Messages will only be sent to objects with the given property name.

**Type** string

**subject**

The subject field of the message.

**Type** string

**body**

The body of the message.

**Type** string

**usePropBody**

Send a property instead of a regular body message.

**Type** boolean

**class** `bge.types.KX_NetworkMessageSensor` (*`SCA_ISensor`*)

The Message Sensor logic brick.

Currently only loopback (local) networks are supported.

**subject**

The subject the sensor is looking for.

**Type** string

**frameMessageCount**

The number of messages received since the last frame. (read-only).

**Type** integer

**subjects**

The list of message subjects received. (read-only).

**Type** list of strings

**bodies**

The list of message bodies received. (read-only).

**Type** list of strings

**class** `bge.types.KX_ObjectActuator` (*SCA\_IActuator*)

The object actuator (“Motion Actuator”) applies force, torque, displacement, angular displacement, velocity, or angular velocity to an object. Servo control allows to regulate force to achieve a certain speed target.

**force**

The force applied by the actuator.

**Type** list [x, y, z]

**useLocalForce**

A flag specifying if the force is local.

**Type** boolean

**torque**

The torque applied by the actuator.

**Type** list [x, y, z]

**useLocalTorque**

A flag specifying if the torque is local.

**Type** boolean

**dLoc**

The displacement vector applied by the actuator.

**Type** list [x, y, z]

**useLocalDLoc**

A flag specifying if the dLoc is local.

**Type** boolean

**dRot**

The angular displacement vector applied by the actuator

**Type** list [x, y, z]

---

**Note:** Since the displacement is applied every frame, you must adjust the displacement based on the frame rate, or you game experience will depend on the player’s computer speed.

---

**useLocalDRot**

A flag specifying if the dRot is local.

**Type** boolean

**linV**

The linear velocity applied by the actuator.

**Type** list [x, y, z]

**useLocalLinV**

A flag specifying if the linear velocity is local.

**Type** boolean

---

**Note:** This is the target speed for servo controllers.

---

**angV**

The angular velocity applied by the actuator.

**Type** list [x, y, z]

**useLocalAngV**

A flag specifying if the angular velocity is local.

**Type** boolean

**damping**

The damping parameter of the servo controller.

**Type** short

**forceLimitX**

The min/max force limit along the X axis and activates or deactivates the limits in the servo controller.

**Type** list [min(float), max(float), bool]

**forceLimitY**

The min/max force limit along the Y axis and activates or deactivates the limits in the servo controller.

**Type** list [min(float), max(float), bool]

**forceLimitZ**

The min/max force limit along the Z axis and activates or deactivates the limits in the servo controller.

**Type** list [min(float), max(float), bool]

**pid**

The PID coefficients of the servo controller.

**Type** list of floats [proportional, integral, derivate]

**reference**

The object that is used as reference to compute the velocity for the servo controller.

**Type** `KX_GameObject` or None

**class** `bge.types.KX_ParentActuator` (*SCA\_IActuator*)

The parent actuator can set or remove an objects parent object.

**object**

the object this actuator sets the parent too.

**Type** `KX_GameObject` or None

**mode**

The mode of this actuator.

**Type** integer from 0 to 1.

**compound**

Whether the object shape should be added to the parent compound shape when parenting.

Effective only if the parent is already a compound shape.

**Type** boolean

**ghost**

Whether the object should be made ghost when parenting Effective only if the shape is not added to the parent compound shape.

**Type** boolean

**class** `bge.types.KX_PhysicsObjectWrapper` (*PyObjectPlus*)

`KX_PhysicsObjectWrapper`

**setActive** (*active*)

Set the object to be active.

**Parameters** *active* (*boolean*) – set to True to be active

**setAngularVelocity** (*x*, *y*, *z*, *local*)

Set the angular velocity of the object.

**Parameters**

- **x** (*float*) – angular velocity for the x-axis
- **y** (*float*) – angular velocity for the y-axis
- **z** (*float*) – angular velocity for the z-axis
- **local** (*boolean*) – set to True for local axis

**setLinearVelocity** (*x*, *y*, *z*, *local*)

Set the linear velocity of the object.

**Parameters**

- **x** (*float*) – linear velocity for the x-axis
- **y** (*float*) – linear velocity for the y-axis
- **z** (*float*) – linear velocity for the z-axis
- **local** (*boolean*) – set to True for local axis

**class** `bge.types.KX_PolyProxy` (*SCA\_IObject*)

A polygon holds the index of the vertex forming the polygon.

Note: The polygon attributes are read-only, you need to retrieve the vertex proxy if you want to change the vertex settings.

**matname**

The name of polygon material, empty if no material.

**Type** string

**material**

The material of the polygon.

**Type** `KX_PolygonMaterial` or `KX_BlenderMaterial`

**texture**

The texture name of the polygon.

**Type** string

**matid**

The material index of the polygon, use this to retrieve vertex proxy from mesh proxy.

**Type** integer

- v1**  
vertex index of the first vertex of the polygon, use this to retrieve vertex proxy from mesh proxy.  
**Type** integer
- v2**  
vertex index of the second vertex of the polygon, use this to retrieve vertex proxy from mesh proxy.  
**Type** integer
- v3**  
vertex index of the third vertex of the polygon, use this to retrieve vertex proxy from mesh proxy.  
**Type** integer
- v4**  
Vertex index of the fourth vertex of the polygon, 0 if polygon has only 3 vertex Use this to retrieve vertex proxy from mesh proxy.  
**Type** integer
- visible**  
visible state of the polygon: 1=visible, 0=invisible.  
**Type** integer
- collide**  
collide state of the polygon: 1=receives collision, 0=collision free.  
**Type** integer
- getMaterialName ()**  
Returns the polygon material name with MA prefix  
**Returns** material name  
**Return type** string
- getMaterial ()**  
**Returns** The polygon material  
**Return type** `KX_PolygonMaterial` or `KX_BlenderMaterial`
- getTextureName ()**  
**Returns** The texture name  
**Return type** string
- getMaterialIndex ()**  
Returns the material bucket index of the polygon. This index and the ones returned by `getVertexIndex()` are needed to retrieve the vertex proxy from `MeshProxy`.  
**Returns** the material index in the mesh  
**Return type** integer
- getNumVertex ()**  
Returns the number of vertex of the polygon.  
**Returns** number of vertex, 3 or 4.  
**Return type** integer
- isVisible ()**  
Returns whether the polygon is visible or not

**Returns** 0=invisible, 1=visible

**Return type** boolean

**isCollider** ()

Returns whether the polygon is receives collision or not

**Returns** 0=collision free, 1=receives collision

**Return type** integer

**getVertexIndex** (*vertex*)

Returns the mesh vertex index of a polygon vertex This index and the one returned by getMaterialIndex() are needed to retrieve the vertex proxy from MeshProxy.

**Parameters**

- **vertex** – index of the vertex in the polygon: 0->3
- **vertex** – integer

**Returns** mesh vertex index

**Return type** integer

**getMesh** ()

Returns a mesh proxy

**Returns** mesh proxy

**Return type** MeshProxy

**class** `bge.types.KX_PolygonMaterial` (*PyObjectPlus*)

This is the interface to materials in the game engine.

Materials define the render state to be applied to mesh objects.

|  |
|--|
| <b>Warning:</b> Some of the methods/variables are CObjects. If you mix these up, you will crash blender. |
|--|

This example requires [PyOpenGL](#) and [GLEWPy](#)

```
import GameLogic
import OpenGL
from OpenGL.GL import *
from OpenGL.GLU import *
import glew
from glew import *

glewInit()

vertex_shader = """

void main(void)
{
    gl_Position = ftransform();
}
"""

fragment_shader = """

void main(void)
{
    gl_FragColor = vec4(1.0, 0.0, 0.0, 1.0);
}
```



```

}
"""

class MyMaterial:
    def __init__(self):
        self.pass_no = 0
        # Create a shader
        self.m_program = glCreateProgramObjectARB()
        # Compile the vertex shader
        self.shader(GL_VERTEX_SHADER_ARB, (vertex_shader))
        # Compile the fragment shader
        self.shader(GL_FRAGMENT_SHADER_ARB, (fragment_shader))
        # Link the shaders together
        self.link()

    def PrintInfoLog(self, tag, object):
        """
        PrintInfoLog prints the GLSL compiler log
        """
        print "Tag:      def PrintGLError(self, tag = ""):

def PrintGLError(self, tag = ""):
    """
    Prints the current GL error status
    """
    if len(tag):
        print tag
    err = glGetError()
    if err != GL_NO_ERROR:
        print "GL Error: %s\\n"%(gluErrorString(err))

def shader(self, type, shaders):
    """
    shader compiles a GLSL shader and attaches it to the current
    program.

    type should be either GL_VERTEX_SHADER_ARB or GL_FRAGMENT_SHADER_ARB
    shaders should be a sequence of shader source to compile.
    """
    # Create a shader object
    shader_object = glCreateShaderObjectARB(type)

    # Add the source code
    glShaderSourceARB(shader_object, len(shaders), shaders)

    # Compile the shader
    glCompileShaderARB(shader_object)

    # Print the compiler log
    self.PrintInfoLog("vertex shader", shader_object)

    # Check if compiled, and attach if it did
    compiled = glGetObjectParameterivARB(shader_object, GL_OBJECT_COMPILE_STATUS_ARB)
    if compiled:
        glAttachObjectARB(self.m_program, shader_object)

    # Delete the object (glAttachObjectARB makes a copy)
    glDeleteObjectARB(shader_object)

```

```
# print the gl error log
self.PrintGLError()

def link(self):
    """
    Links the shaders together.
    """
    # clear error indicator
    glGetError()

    glLinkProgramARB(self.m_program)

    self.PrintInfoLog("link", self.m_program)

    linked = glGetObjectParameterivARB(self.m_program, GL_OBJECT_LINK_STATUS_ARB)
    if not linked:
        print "Shader failed to link"
        return

    glValidateProgramARB(self.m_program)
    valid = glGetObjectParameterivARB(self.m_program, GL_OBJECT_VALIDATE_STATUS_ARB)
    if not valid:
        print "Shader failed to validate"
        return

def activate(self, rasty, cachingInfo, mat):
    self.pass_no+=1
    if (self.pass_no == 1):
        glDisable(GL_COLOR_MATERIAL)
        glUseProgramObjectARB(self.m_program)
        return True

    glEnable(GL_COLOR_MATERIAL)
    glUseProgramObjectARB(0)
    self.pass_no = 0
    return False

obj = GameLogic.getCurrentController().owner

mesh = obj.meshes[0]

for mat in mesh.materials:
    mat.setCustomMaterial(MyMaterial())
    print mat.texture
```

**texture**

Texture name.

**Type** string (read-only)

**gl\_texture**

OpenGL texture handle (eg for glBindTexture(GL\_TEXTURE\_2D, gl\_texture).

**Type** integer (read-only)

**material**

Material name.

**Type** string (read-only)

**tface**

Texture face properties.

**Type** CObject (read-only)

**tile**

Texture is tiling.

**Type** boolean

**tilexrep**

Number of tile repetitions in x direction.

**Type** integer

**tileyrep**

Number of tile repetitions in y direction.

**Type** integer

**drawingmode**

Drawing mode for the material. - 2 (drawingmode & 4) Textured - 4 (drawingmode & 16) Light - 14 (drawingmode & 16384) 3d Polygon Text.

**Type** bitfield

**transparent**

This material is transparent. All meshes with this material will be rendered after non transparent meshes from back to front.

**Type** boolean

**zsort**

Transparent polygons in meshes with this material will be sorted back to front before rendering. Non-Transparent polygons will be sorted front to back before rendering.

**Type** boolean

**lightlayer**

Light layers this material affects.

**Type** bitfield.

**triangle**

Mesh data with this material is triangles. It's probably not safe to change this.

**Type** boolean

**diffuse**

The diffuse colour of the material. black = [0.0, 0.0, 0.0] white = [1.0, 1.0, 1.0].

**Type** list [r, g, b]

**specular**

The specular colour of the material. black = [0.0, 0.0, 0.0] white = [1.0, 1.0, 1.0].

**Type** list [r, g, b]

**shininess**

The shininess (specular exponent) of the material.  $0.0 \leq \text{shininess} \leq 128.0$ .

**Type** float

**specularity**

The amount of specular of the material.  $0.0 \leq \text{specularity} \leq 1.0$ .

**Type** float

**updateTexture** (*tface, rasty*)

Updates a realtime animation.

**Parameters**

- **tface** (*CObject*) – Texture face (eg mat.tface)
- **rasty** (*CObject*) – Rasterizer

**setTexture** (*tface*)

Sets texture render state.

**Parameters** **tface** (*CObject*) – Texture face

```
mat.setTexture(mat.tface)
```

**activate** (*rasty, cachingInfo*)

Sets material parameters for this object for rendering.

Material Parameters set:

- 1.Texture
- 2.Backface culling
- 3.Line drawing
- 4.Specular Colour
- 5.Shininess
- 6.Diffuse Colour
- 7.Polygon Offset.

**Parameters**

- **rasty** (*CObject*) – Rasterizer instance.
- **cachingInfo** (*CObject*) – Material cache instance.

**setCustomMaterial** (*material*)

Sets the material state setup object.

Using this method, you can extend or completely replace the gameengine material to do your own advanced multipass effects.

Use this method to register your material class. Instead of the normal material, your class's activate method will be called just before rendering the mesh. This should setup the texture, material, and any other state you would like. It should return True to render the mesh, or False if you are finished. You should clean up any state Blender does not set before returning False.

Activate Method Definition:

```
def activate(self, rasty, cachingInfo, material):
```

**Parameters** **material** (*instance*) – The material object.

```
class PyMaterial:  
    def __init__(self):  
        self.pass_no = -1
```

```

def activate(self, rasty, cachingInfo, material):
    # Activate the material here.
    #
    # The activate method will be called until it returns False.
    # Every time the activate method returns True the mesh will
    # be rendered.
    #
    # rasty is a CObject for passing to material.updateTexture()
    #     and material.activate()
    # cachingInfo is a CObject for passing to material.activate()
    # material is the KX_PolygonMaterial instance this material
    #     was added to

    # default material properties:
    self.pass_no += 1
    if self.pass_no == 0:
        material.activate(rasty, cachingInfo)
        # Return True to do this pass
        return True

    # clean up and return False to finish.
    self.pass_no = -1
    return False

# Create a new Python Material and pass it to the renderer.
mat.setCustomMaterial(PyMaterial())

```

**class** `bge.types.KX_RadarSensor` (*KX\_NearSensor*)

Radar sensor is a near sensor with a conical sensor object.

**coneOrigin**

The origin of the cone with which to test. The origin is in the middle of the cone. (read-only).

**Type** list of floats [x, y, z]

**coneTarget**

The center of the bottom face of the cone with which to test. (read-only).

**Type** list of floats [x, y, z]

**distance**

The height of the cone with which to test.

**Type** float

**angle**

The angle of the cone (in degrees) with which to test.

**Type** float from 0 to 360

**axis**

The axis on which the radar cone is cast.

**Type** integer from 0 to 5

`KX_RADAR_AXIS_POS_X`, `KX_RADAR_AXIS_POS_Y`, `KX_RADAR_AXIS_POS_Z`,  
`KX_RADAR_AXIS_NEG_X`, `KX_RADAR_AXIS_NEG_Y`, `KX_RADAR_AXIS_NEG_Z`

**getConeHeight** ()

**Returns** The height of the cone with which to test.

**Return type** float

**class** `bge.types.KX_RaySensor` (*SCA\_ISensor*)

A ray sensor detects the first object in a given direction.

**propName**

The property the ray is looking for.

**Type** string

**range**

The distance of the ray.

**Type** float

**useMaterial**

Whether or not to look for a material (false = property).

**Type** boolean

**useXRay**

Whether or not to use XRay.

**Type** boolean

**hitObject**

The game object that was hit by the ray. (read-only).

**Type** `KX_GameObject`

**hitPosition**

The position (in worldcoordinates) where the object was hit by the ray. (read-only).

**Type** list [x, y, z]

**hitNormal**

The normal (in worldcoordinates) of the object at the location where the object was hit by the ray. (read-only).

**Type** list [x, y, z]

**rayDirection**

The direction from the ray (in worldcoordinates). (read-only).

**Type** list [x, y, z]

**axis**

The axis the ray is pointing on.

**Type** integer from 0 to 5

- KX\_RAY\_AXIS\_POS\_X
- KX\_RAY\_AXIS\_POS\_Y
- KX\_RAY\_AXIS\_POS\_Z
- KX\_RAY\_AXIS\_NEG\_X
- KX\_RAY\_AXIS\_NEG\_Y
- KX\_RAY\_AXIS\_NEG\_Z

**class** `bge.types.KX_SCA_AddObjectActuator` (*SCA\_IActuator*)

Edit Object Actuator (in Add Object Mode)

**Warning:** An Add Object actuator will be ignored if at game start, the linked object doesn't exist (or is empty) or the linked object is in an active layer.

Error: GameObject 'Name' has a AddObjectActuator 'ActuatorName' without object (in 'nonactive'

**object**

the object this actuator adds.

**Type** `KX_GameObject` or `None`

**objectLastCreated**

the last added object from this actuator (read-only).

**Type** `KX_GameObject` or `None`

**time**

the lifetime of added objects, in frames. Set to 0 to disable automatic deletion.

**Type** `integer`

**linearVelocity**

the initial linear velocity of added objects.

**Type** `list [vx, vy, vz]`

**angularVelocity**

the initial angular velocity of added objects.

**Type** `list [vx, vy, vz]`

**instantAddObject ()**

adds the object without needing to calling `SCA_PythonController.activate()`

---

**Note:** Use `objectLastCreated` to get the newly created object.

---

**class** `bge.types.KX_SCA_DynamicActuator` (*SCA\_IActuator*)

Dynamic Actuator.

**mode**

**Type** `integer`

the type of operation of the actuator, 0-4

- `KX_DYN_RESTORE_DYNAMICS(0)`
- `KX_DYN_DISABLE_DYNAMICS(1)`
- `KX_DYN_ENABLE_RIGID_BODY(2)`
- `KX_DYN_DISABLE_RIGID_BODY(3)`
- `KX_DYN_SET_MASS(4)`

**mass**

the mass value for the `KX_DYN_SET_MASS` operation.

**Type** `float`

**class** `bge.types.KX_SCA_EndObjectActuator` (*SCA\_IActuator*)

Edit Object Actuator (in End Object mode)

This actuator has no python methods.

`class bge.types.KX_SCA_ReplaceMeshActuator (SCA_IActuator)`

Edit Object actuator, in Replace Mesh mode.

**Warning:** Replace mesh actuators will be ignored if at game start, the named mesh doesn't exist. This will generate a warning in the console

```
Error: GameObject 'Name' ReplaceMeshActuator 'ActuatorName' without object
```

```
# Level-of-detail
# Switch a game object's mesh based on its depth in the camera view.
# +-----+ +-----+ +-----+
# | Always +-----+ Python +-----+ Edit Object (Replace Mesh) LOD.Mesh |
# +-----+ +-----+ +-----+
import GameLogic

# List detail meshes here
# Mesh (name, near, far)
# Meshes overlap so that they don't 'pop' when on the edge of the distance.
meshes = (("Hi", 0.0, -20.0),
          ("Med", -15.0, -50.0),
          ("Lo", -40.0, -100.0)
         )

co = GameLogic.getCurrentController()
obj = co.owner
act = co.actuators["LOD." + obj.name]
cam = GameLogic.getCurrentScene().active_camera

def Depth(pos, plane):
    return pos[0]*plane[0] + pos[1]*plane[1] + pos[2]*plane[2] + plane[3]

# Depth is negative and decreasing further from the camera
depth = Depth(obj.position, cam.world_to_camera[2])

newmesh = None
curmesh = None
# Find the lowest detail mesh for depth
for mesh in meshes:
    if depth < mesh[1] and depth > mesh[2]:
        newmesh = mesh
    if "ME" + obj.name + mesh[0] == act.getMesh():
        curmesh = mesh

if newmesh != None and "ME" + obj.name + newmesh[0] != act.getMesh():
    # The mesh is a different mesh - switch it.
    # Check the current mesh is not a better fit.
    if curmesh == None or curmesh[1] < depth or curmesh[2] > depth:
        act.mesh = obj.getName() + newmesh[0]
        GameLogic.addActiveActuator(act, True)
```

#### mesh

MeshProxy or the name of the mesh that will replace the current one.

Set to None to disable actuator.

**Type** MeshProxy or None if no mesh is set

#### useDisplayMesh



when true the displayed mesh is replaced.

**Type** boolean

**usePhysicsMesh**

when true the physics mesh is replaced.

**Type** boolean

**instantReplaceMesh()**

Immediately replace mesh without delay.

**class** `bge.types.KX_Scene` (*PyObjectPlus*)

An active scene that gives access to objects, cameras, lights and scene attributes.

The activity culling stuff is supposed to disable logic bricks when their owner gets too far from the active camera. It was taken from some code lurking at the back of `KX_Scene` - who knows what it does!

```
import GameLogic

# get the scene
scene = GameLogic.getCurrentScene()

# print all the objects in the scene
for obj in scene.objects:
    print obj.name

# get an object named 'Cube'
obj = scene.objects["Cube"]

# get the first object in the scene.
obj = scene.objects[0]

# Get the depth of an object in the camera view.
import GameLogic

obj = GameLogic.getCurrentController().owner
cam = GameLogic.getCurrentScene().active_camera

# Depth is negative and decreasing further from the camera
depth = obj.position[0]*cam.world_to_camera[2][0] + obj.position[1]*cam.world_to_camera[2][1] +
```

@bug: All attributes are read only at the moment.

**name**

The scene's name, (read-only).

**Type** string

**objects**

A list of objects in the scene, (read-only).

**Type** `CListValue` of `KX_GameObject`

**objectsInactive**

A list of objects on background layers (used for the `addObject` actuator), (read-only).

**Type** `CListValue` of `KX_GameObject`

**lights**

A list of lights in the scene, (read-only).

**Type** `CListValue` of `KX_LightObject`

**cameras**

A list of cameras in the scene, (read-only).

**Type** `CListValue` of `KX_Camera`

**active\_camera**

The current active camera.

**Type** `KX_Camera`

---

**Note:** This can be set directly from python to avoid using the `KX_SceneActuator`.

---

**suspended**

True if the scene is suspended, (read-only).

**Type** `boolean`

**activity\_culling**

True if the scene is activity culling.

**Type** `boolean`

**activity\_culling\_radius**

The distance outside which to do activity culling. Measured in manhattan distance.

**Type** `float`

**dbvt\_culling**

True when Dynamic Bounding box Volume Tree is set (read-only).

**Type** `boolean`

**pre\_draw**

A list of callables to be run before the render step.

**Type** `list`

**post\_draw**

A list of callables to be run after the render step.

**Type** `list`

**addObject** (*object, other, time=0*)

Adds an object to the scene like the Add Object Actuator would.

**Parameters**

- **object** (`KX_GameObject` or string) – The object to add
- **other** (`KX_GameObject` or string) – The object's center to use when adding the object
- **time** (*integer*) – The lifetime of the added object, in frames. A time of 0 means the object will last forever.

**Returns** The newly added object.

**Return type** `KX_GameObject`

**end()**

Removes the scene from the game.

**restart()**

Restarts the scene.

**replace** (*scene*)

Replaces this scene with another one.

**Parameters** **scene** (*string*) – The name of the scene to replace this scene with.

**suspend** ()

Suspends this scene.

**resume** ()

Resume this scene.

**get** (*key*, *default=None*)

Return the value matching key, or the default value if its not found. :return: The key value or a default.

**class** `bge.types.KX_SceneActuator` (*SCA\_IActuator*)

Scene Actuator logic brick.

**Warning:** Scene actuators that use a scene name will be ignored if at game start, the named scene doesn't exist or is empty

This will generate a warning in the console:

```
Error: GameObject 'Name' has a SceneActuator 'ActuatorName' (SetScene) without scene
```

**scene**

the name of the scene to change to/overlay/underlay/remove/suspend/resume.

**Type** string

**camera**

the camera to change to.

**Type** `KX_Camera` on read, string or `KX_Camera` on write

---

**Note:** When setting the attribute, you can use either a `KX_Camera` or the name of the camera.

---

**useRestart**

Set flag to True to restart the sene.

**Type** boolean

**mode**

The mode of the actuator.

**Type** integer from 0 to 5.

**class** `bge.types.KX_SoundActuator` (*SCA\_IActuator*)

Sound Actuator.

The `startSound`, `pauseSound` and `stopSound` do not require the actuator to be activated - they act instantly provided that the actuator has been activated once at least.

**fileName**

The filename of the sound this actuator plays.

**Type** string

**volume**

The volume (gain) of the sound.

**Type** float

**pitch**

The pitch of the sound.

**Type** float

**rolloffFactor**

The roll off factor. Rolloff defines the rate of attenuation as the sound gets further away.

**Type** float

**looping**

The loop mode of the actuator.

**Type** integer

**position**

The position of the sound as a list: [x, y, z].

**Type** float array

**velocity**

The velocity of the emitter as a list: [x, y, z]. The relative velocity to the observer determines the pitch.  
List of 3 floats: [x, y, z].

**Type** float array

**orientation**

The orientation of the sound. When setting the orientation you can also use quaternion [float, float, float, float] or euler angles [float, float, float].

**Type** 3x3 matrix [[float]]

**mode**

The operation mode of the actuator. Can be one of *these constants*

**Type** integer

**class** `bge.types.KX_StateActuator` (*SCA\_IActuator*)

State actuator changes the state mask of parent object.

**operation**

Type of bit operation to be applied on object state mask.

You can use one of *these constants*

**Type** integer

**mask**

Value that defines the bits that will be modified by the operation.

The bits that are 1 in the mask will be updated in the object state.

The bits that are 0 are will be left unmodified expect for the Copy operation which copies the mask to the object state.

**Type** integer

**class** `bge.types.KX_TrackToActuator` (*SCA\_IActuator*)

Edit Object actuator in Track To mode.

**Warning:** Track To Actuators will be ignored if at game start, the object to track to is invalid.  
This will generate a warning in the console:

```
GameObject 'Name' no object in EditObjectActuator 'ActuatorName'
```

**object**

the object this actuator tracks.

**Type** `KX_GameObject` or `None`

**time**

the time in frames with which to delay the tracking motion.

**Type** `integer`

**use3D**

the tracking motion to use 3D.

**Type** `boolean`

**class** `bge.types.KX_VehicleWrapper` (*PyObjectPlus*)

`KX_VehicleWrapper`

TODO - description

**addWheel** (*wheel, attachPos, attachDir, axleDir, suspensionRestLength, wheelRadius, hasSteering*)

Add a wheel to the vehicle

**Parameters**

- **wheel** (`KX_GameObject` or a `KX_GameObject` name) – The object to use as a wheel.
- **attachPos** (*vector of 3 floats*) – The position that this wheel will attach to.
- **attachDir** (*vector of 3 floats*) – The direction this wheel points.
- **axleDir** (*vector of 3 floats*) – The direction of this wheels axle.
- **suspensionRestLength** (*float*) – TODO - Description
- **wheelRadius** (*float*) – The size of the wheel.

**applyBraking** (*force, wheelIndex*)

Apply a braking force to the specified wheel

**Parameters**

- **force** (*float*) – the brake force
- **wheelIndex** (*integer*) – index of the wheel where the force needs to be applied

**applyEngineForce** (*force, wheelIndex*)

Apply an engine force to the specified wheel

**Parameters**

- **force** (*float*) – the engine force
- **wheelIndex** (*integer*) – index of the wheel where the force needs to be applied

**getConstraintId** ()

Get the constraint ID

**Returns** the constraint id

**Return type** `integer`

**getConstraintType** ()

Returns the constraint type.

**Returns** constraint type

**Return type** integer

**getNumWheels** ()

Returns the number of wheels.

**Returns** the number of wheels for this vehicle

**Return type** integer

**getWheelOrientationQuaternion** (*wheelIndex*)

Returns the wheel orientation as a quaternion.

**Parameters** **wheelIndex** (*integer*) – the wheel index

**Returns** TODO Description

**Return type** TODO - type should be quat as per method name but from the code it looks like a matrix

**getWheelPosition** (*wheelIndex*)

Returns the position of the specified wheel

**Parameters** **wheelIndex** (*integer*) – the wheel index

**Returns** position vector

**Return type** list[x, y, z]

**getWheelRotation** (*wheelIndex*)

Returns the rotation of the specified wheel

**Parameters** **wheelIndex** (*integer*) – the wheel index

**Returns** the wheel rotation

**Return type** float

**setRollInfluence** (*rollInfluence*, *wheelIndex*)

Set the specified wheel's roll influence. The higher the roll influence the more the vehicle will tend to roll over in corners.

**Parameters**

- **rollInfluence** (*float*) – the wheel roll influence
- **wheelIndex** (*integer*) – the wheel index

**setSteeringValue** (*steering*, *wheelIndex*)

Set the specified wheel's steering

**Parameters**

- **steering** (*float*) – the wheel steering
- **wheelIndex** (*integer*) – the wheel index

**setSuspensionCompression** (*compression*, *wheelIndex*)

Set the specified wheel's compression

**Parameters**

- **compression** (*float*) – the wheel compression
- **wheelIndex** (*integer*) – the wheel index

**setSuspensionDamping** (*damping*, *wheelIndex*)

Set the specified wheel's damping

**Parameters**

- **damping** (*float*) – the wheel damping
- **wheelIndex** (*integer*) – the wheel index

**setSuspensionStiffness** (*stiffness, wheelIndex*)

Set the specified wheel's stiffness

**Parameters**

- **stiffness** (*float*) – the wheel stiffness
- **wheelIndex** (*integer*) – the wheel index

**setTyreFriction** (*friction, wheelIndex*)

Set the specified wheel's tyre friction

**Parameters**

- **friction** (*float*) – the tyre friction
- **wheelIndex** (*integer*) – the wheel index

**class** `bge.types.KX_VertexProxy` (*SCA\_IObject*)

A vertex holds position, UV, colour and normal information.

Note: The physics simulation is NOT currently updated - physics will not respond to changes in the vertex position.

**XYZ**

The position of the vertex.

**Type** list [x, y, z]

**UV**

The texture coordinates of the vertex.

**Type** list [u, v]

**normal**

The normal of the vertex.

**Type** list [nx, ny, nz]

**colour**

The colour of the vertex.

**Type** list [r, g, b, a]

Black = [0.0, 0.0, 0.0, 1.0], White = [1.0, 1.0, 1.0, 1.0]

**color**

Synonym for colour.

**x**

The x coordinate of the vertex.

**Type** float

**y**

The y coordinate of the vertex.

**Type** float

**z**

The z coordinate of the vertex.

**Type** float

**u**

The u texture coordinate of the vertex.

**Type** float

**v**

The v texture coordinate of the vertex.

**Type** float

**u2**

The second u texture coordinate of the vertex.

**Type** float

**v2**

The second v texture coordinate of the vertex.

**Type** float

**r**

The red component of the vertex colour.  $0.0 \leq r \leq 1.0$ .

**Type** float

**g**

The green component of the vertex colour.  $0.0 \leq g \leq 1.0$ .

**Type** float

**b**

The blue component of the vertex colour.  $0.0 \leq b \leq 1.0$ .

**Type** float

**a**

The alpha component of the vertex colour.  $0.0 \leq a \leq 1.0$ .

**Type** float

**getXYZ ()**

Gets the position of this vertex.

**Returns** this vertexes position in local coordinates.

**Return type** list [x, y, z]

**setXYZ (pos)**

Sets the position of this vertex.

**Type** list [x, y, z]

**Parameters** **pos** – the new position for this vertex in local coordinates.

**getUV ()**

Gets the UV (texture) coordinates of this vertex.

**Returns** this vertexes UV (texture) coordinates.

**Return type** list [u, v]

**setUV (uv)**

Sets the UV (texture) coordinates of this vertex.

**Type** list [u, v]



**getUV2 ()**

Gets the 2nd UV (texture) coordinates of this vertex.

**Returns** this vertexes UV (texture) coordinates.

**Return type** list [u, v]

**setUV2 (uv, unit)**

Sets the 2nd UV (texture) coordinates of this vertex.

**Type** list [u, v]

**Parameters**

- **unit** – optional argument, FLAT==1, SECOND\_UV==2, defaults to SECOND\_UV
- **unit** – integer

**getRGBA ()**

Gets the colour of this vertex.

The colour is represented as four bytes packed into an integer value. The colour is packed as RGBA.

Since Python offers no way to get each byte without shifting, you must use the struct module to access colour in a machine independent way.

Because of this, it is suggested you use the r, g, b and a attributes or the colour attribute instead.

```
import struct;
col = struct.unpack('4B', struct.pack('I', v.getRGBA()))
# col = (r, g, b, a)
# black = ( 0, 0, 0, 255)
# white = (255, 255, 255, 255)
```

**Returns** packed colour. 4 byte integer with one byte per colour channel in RGBA format.

**Return type** integer

**setRGBA (col)**

Sets the colour of this vertex.

See getRGBA() for the format of col, and its relevant problems. Use the r, g, b and a attributes or the colour attribute instead.

setRGBA() also accepts a four component list as argument col. The list represents the colour as [r, g, b, a] with black = [0.0, 0.0, 0.0, 1.0] and white = [1.0, 1.0, 1.0, 1.0]

```
v.setRGBA(0xff0000ff) # Red
v.setRGBA(0xff00ff00) # Green on little endian, transparent purple on big endian
v.setRGBA([1.0, 0.0, 0.0, 1.0]) # Red
v.setRGBA([0.0, 1.0, 0.0, 1.0]) # Green on all platforms.
```

**Parameters** **col** (*integer or list [r, g, b, a]*) – the new colour of this vertex in packed RGBA format.

**getNormal ()**

Gets the normal vector of this vertex.

**Returns** normalised normal vector.

**Return type** list [nx, ny, nz]

**setNormal** (*normal*)

Sets the normal vector of this vertex.

**Type** sequence of floats [r, g, b]

**Parameters** **normal** – the new normal of this vertex.

**class** `bge.types.KX_VisibilityActuator` (*SCA\_IActuator*)

Visibility Actuator.

**visibility**

whether the actuator makes its parent object visible or invisible.

**Type** boolean

**useOcclusion**

whether the actuator makes its parent object an occluder or not.

**Type** boolean

**useRecursion**

whether the visibility/occlusion should be propagated to all children of the object.

**Type** boolean

**class** `bge.types.SCA_2DFilterActuator` (*SCA\_IActuator*)

Create, enable and disable 2D filters

The following properties don't have an immediate effect. You must active the actuator to get the result. The actuator is not persistent: it automatically stops itself after setting up the filter but the filter remains active. To stop a filter you must activate the actuator with 'type' set to [RAS\\_2DFILTER\\_DISABLED](#) or [RAS\\_2DFILTER\\_NOFILTER](#).

**shaderText**

shader source code for custom shader.

**Type** string

**disableMotionBlur**

action on motion blur: 0=enable, 1=disable.

**Type** integer

**mode**

Type of 2D filter, use one of *these constants*

**Type** integer

**passNumber**

order number of filter in the stack of 2D filters. Filters are executed in increasing order of passNb.

Only be one filter can be defined per passNb.

**Type** integer (0-100)

**value**

argument for motion blur filter.

**Type** float (0.0-100.0)

**class** `bge.types.SCA_ANDController` (*SCA\_IController*)

An AND controller activates only when all linked sensors are activated.

There are no special python methods for this controller.

**class** `bge.types.SCA_ActuatorSensor` (*SCA\_ISensor*)

Actuator sensor detect change in actuator state of the parent object. It generates a positive pulse if the corresponding actuator is activated and a negative pulse if the actuator is deactivated.

**actuator**

the name of the actuator that the sensor is monitoring.

**Type** string

**class** `bge.types.SCA_AlwaysSensor` (*SCA\_ISensor*)

This sensor is always activated.

**class** `bge.types.SCA_DelaySensor` (*SCA\_ISensor*)

The Delay sensor generates positive and negative triggers at precise time, expressed in number of frames. The delay parameter defines the length of the initial OFF period. A positive trigger is generated at the end of this period.

The duration parameter defines the length of the ON period following the OFF period. There is a negative trigger at the end of the ON period. If duration is 0, the sensor stays ON and there is no negative trigger.

The sensor runs the OFF-ON cycle once unless the repeat option is set: the OFF-ON cycle repeats indefinitely (or the OFF cycle if duration is 0).

Use `SCA_ISensor.reset` at any time to restart sensor.

**delay**

length of the initial OFF period as number of frame, 0 for immediate trigger.

**Type** integer.

**duration**

length of the ON period in number of frame after the initial OFF period.

If duration is greater than 0, a negative trigger is sent at the end of the ON pulse.

**Type** integer

**repeat**

1 if the OFF-ON cycle should be repeated indefinitely, 0 if it should run once.

**Type** integer

**class** `bge.types.SCA_JoystickSensor` (*SCA\_ISensor*)

This sensor detects player joystick events.

**axisValues**

The state of the joysticks axis as a list of values `numAxis` long. (read-only).

**Type** list of ints.

Each specifying the value of an axis between -32767 and 32767 depending on how far the axis is pushed, 0 for nothing. The first 2 values are used by most joysticks and gamepads for directional control. 3rd and 4th values are only on some joysticks and can be used for arbitrary controls.

•left:[-32767, 0, ...]

•right:[32767, 0, ...]

•up:[0, -32767, ...]

•down:[0, 32767, ...]

**axisSingle**

like `axisValues` but returns a single axis value that is set by the sensor. (read-only).

**Type** integer

---

**Note:** Only use this for “Single Axis” type sensors otherwise it will raise an error.

---

**hatValues**

The state of the joysticks hats as a list of values `numHats` long. (read-only).

**Type** list of ints

Each specifying the direction of the hat from 1 to 12, 0 when inactive.

Hat directions are as follows...

- 0:None
- 1:Up
- 2:Right
- 4:Down
- 8:Left
- 3:Up - Right
- 6:Down - Right
- 12:Down - Left
- 9:Up - Left

**hatSingle**

Like `hatValues` but returns a single hat direction value that is set by the sensor. (read-only).

**Type** integer

**numAxis**

The number of axes for the joystick at this index. (read-only).

**Type** integer

**numButtons**

The number of buttons for the joystick at this index. (read-only).

**Type** integer

**numHats**

The number of hats for the joystick at this index. (read-only).

**Type** integer

**connected**

True if a joystick is connected at this joysticks index. (read-only).

**Type** boolean

**index**

The joystick index to use (from 0 to 7). The first joystick is always 0.

**Type** integer

**threshold**

Axis threshold. Joystick axis motion below this threshold wont trigger an event. Use values between (0 and 32767), lower values are more sensitive.

**Type** integer

**button**

The button index the sensor reacts to (first button = 0). When the “All Events” toggle is set, this option has no effect.

**Type** integer

**axis**

The axis this sensor reacts to, as a list of two values [axisIndex, axisDirection]

- axisIndex: the axis index to use when detecting axis movement, 1=primary directional control, 2=secondary directional control.
- axisDirection: 0=right, 1=up, 2=left, 3=down.

**Type** [integer, integer]

**hat**

The hat the sensor reacts to, as a list of two values: [hatIndex, hatDirection]

- hatIndex: the hat index to use when detecting hat movement, 1=primary hat, 2=secondary hat (4 max).
- hatDirection: 1-12.

**Type** [integer, integer]

**getButtonActiveList** ()

**Returns** A list containing the indicies of the currently pressed buttons.

**Return type** list

**getButtonStatus** (*buttonIndex*)

**Parameters** **buttonIndex** (*integer*) – the button index, 0=first button

**Returns** The current pressed state of the specified button.

**Return type** boolean

**class** `bge.types.SCA_KeyboardSensor` (*SCA\_ISensor*)

A keyboard sensor detects player key presses.

See module `bge.keys` for keycode values.

**key**

The key code this sensor is looking for.

**Type** keycode from `bge.keys` module

**hold1**

The key code for the first modifier this sensor is looking for.

**Type** keycode from `bge.keys` module

**hold2**

The key code for the second modifier this sensor is looking for.

**Type** keycode from `bge.keys` module

**toggleProperty**

The name of the property that indicates whether or not to log keystrokes as a string.

**Type** string

**targetProperty**

The name of the property that receives keystrokes in case in case a string is logged.

**Type** string

**useAllKeys**

Flag to determine whether or not to accept all keys.

**Type** boolean

**events**

a list of pressed keys that have either been pressed, or just released, or are active this frame. (read-only).

**Type** list *[[keycode, status], ...]*

**getKeyStatus** (*keycode*)

Get the status of a key.

**Parameters** **keycode** (*integer*) – The code that represents the key you want to get the state of, use one of *these constants*

**Returns** The state of the given key, can be one of *these constants*

**Return type** int

**class** `bge.types.SCA_NANDController` (*SCA\_IController*)

An NAND controller activates when all linked sensors are not active.

There are no special python methods for this controller.

**class** `bge.types.SCA_NORController` (*SCA\_IController*)

An NOR controller activates only when all linked sensors are de-activated.

There are no special python methods for this controller.

**class** `bge.types.SCA_ORController` (*SCA\_IController*)

An OR controller activates when any connected sensor activates.

There are no special python methods for this controller.

**class** `bge.types.SCA_PropertyActuator` (*SCA\_IActuator*)

Property Actuator

**propName**

the property on which to operate.

**Type** string

**value**

the value with which the actuator operates.

**Type** string

**mode**

TODO - add constants to game logic dict!.

**Type** integer

**class** `bge.types.SCA_PropertySensor` (*SCA\_ISensor*)

Activates when the game object property matches.

**mode**

Type of check on the property. Can be one of *these constants*

**Type** integer.

**propName**

the property the sensor operates.

**Type** string

**value**

the value with which the sensor compares to the value of the property.

**Type** string

**min**

the minimum value of the range used to evaluate the property when in interval mode.

**Type** string

**max**

the maximum value of the range used to evaluate the property when in interval mode.

**Type** string

**class** `bge.types.SCA_PythonController` (*SCA\_IController*)

A Python controller uses a Python script to activate it's actuators, based on it's sensors.

**script**

The value of this variable depends on the execution method.

- When 'Script' execution mode is set this value contains the entire python script as a single string (not the script name as you might expect) which can be modified to run different scripts.
- When 'Module' execution mode is set this value will contain a single line string - module name and function "module.func" or "package.module.func" where the module names are python textblocks or external scripts.

**Type** string

---

**Note:** Once this is set the script name given for warnings will remain unchanged.

---

**mode**

the execution mode for this controller (read-only).

- Script: 0, Execute the `script` as a python code.
- Module: 1, Execute the `script` as a module and function.

**Type** integer

**activate** (*actuator*)

Activates an actuator attached to this controller.

**Parameters** `actuator` (*actuator or the actuator name as a string*) – The actuator to operate on.

**deactivate** (*actuator*)

Deactivates an actuator attached to this controller.

**Parameters** `actuator` (*actuator or the actuator name as a string*) – The actuator to operate on.

**class** `bge.types.SCA_RandomActuator` (*SCA\_IActuator*)

Random Actuator

**seed**

Seed of the random number generator.

**Type** integer.

Equal seeds produce equal series. If the seed is 0, the generator will produce the same value on every call.

**para1**

the first parameter of the active distribution.

**Type** float, read-only.

Refer to the documentation of the generator types for the meaning of this value.

**para2**

the second parameter of the active distribution.

**Type** float, read-only

Refer to the documentation of the generator types for the meaning of this value.

**distribution**

Distribution type. (read-only). Can be one of *these constants*

**Type** integer

**propName**

the name of the property to set with the random value.

**Type** string

If the generator and property types do not match, the assignment is ignored.

**setBoolConst** (*value*)

Sets this generator to produce a constant boolean value.

**Parameters** **value** (*boolean*) – The value to return.

**setBoolUniform** ()

Sets this generator to produce a uniform boolean distribution.

The generator will generate True or False with 50% chance.

**setBoolBernouilli** (*value*)

Sets this generator to produce a Bernouilli distribution.

**Parameters** **value** (*float*) – Specifies the proportion of False values to produce.

- 0.0: Always generate True
- 1.0: Always generate False

**setIntConst** (*value*)

Sets this generator to always produce the given value.

**Parameters** **value** (*integer*) – the value this generator produces.

**setIntUniform** (*lower\_bound*, *upper\_bound*)

Sets this generator to produce a random value between the given lower and upper bounds (inclusive).

**setIntPoisson** (*value*)

Generate a Poisson-distributed number.

This performs a series of Bernouilli tests with parameter value. It returns the number of tries needed to achieve succes.



**setFloatConst** (*value*)

Always generate the given value.

**setFloatUniform** (*lower\_bound*, *upper\_bound*)

Generates a random float between *lower\_bound* and *upper\_bound* with a uniform distribution.

**setFloatNormal** (*mean*, *standard\_deviation*)

Generates a random float from the given normal distribution.

#### Parameters

- **mean** (*float*) – The mean (average) value of the generated numbers
- **standard\_deviation** (*float*) – The standard deviation of the generated numbers.

**setFloatNegativeExponential** (*half\_life*)

Generate negative-exponentially distributed numbers.

The half-life ‘time’ is characterized by *half\_life*.

**class** `bge.types.SCA_RandomSensor` (*SCA\_ISensor*)

This sensor activates randomly.

**lastDraw**

The seed of the random number generator.

**Type** integer

**seed**

The seed of the random number generator.

**Type** integer

**setSeed** (*seed*)

Sets the seed of the random number generator.

If the seed is 0, the generator will produce the same value on every call.

**getSeed** ()

**Returns** The initial seed of the generator. Equal seeds produce equal random series.

**Return type** integer

**getLastDraw** ()

**Returns** The last random number generated.

**Return type** integer

**class** `bge.types.SCA_XNORController` (*SCA\_IController*)

An XNOR controller activates when all linked sensors are the same (activated or inactive).

There are no special python methods for this controller.

**class** `bge.types.SCA_XORController` (*SCA\_IController*)

An XOR controller activates when there is the input is mixed, but not when all are on or off.

There are no special python methods for this controller.

**class** `bge.types.KX_Camera` (*KX\_GameObject*)

A Camera object.

**INSIDE**

See `sphereInsideFrustum` and `boxInsideFrustum`

**INTERSECT**

See `sphereInsideFrustum` and `boxInsideFrustum`

**OUTSIDE**

See `sphereInsideFrustum` and `boxInsideFrustum`

**lens**

The camera's lens value.

**Type** float

**ortho\_scale**

The camera's view scale when in orthographic mode.

**Type** float

**near**

The camera's near clip distance.

**Type** float

**far**

The camera's far clip distance.

**Type** float

**perspective**

True if this camera has a perspective transform, False for an orthographic projection.

**Type** boolean

**frustum\_culling**

True if this camera is frustum culling.

**Type** boolean

**projection\_matrix**

This camera's 4x4 projection matrix.

**Type** 4x4 Matrix [[float]]

**modelview\_matrix**

This camera's 4x4 model view matrix. (read-only).

**Type** 4x4 Matrix [[float]]

---

**Note:** This matrix is regenerated every frame from the camera's position and orientation.

---

**camera\_to\_world**

This camera's camera to world transform. (read-only).

**Type** 4x4 Matrix [[float]]

---

**Note:** This matrix is regenerated every frame from the camera's position and orientation.

---

**world\_to\_camera**

This camera's world to camera transform. (read-only).

**Type** 4x4 Matrix [[float]]

---

**Note:** Regenerated every frame from the camera's position and orientation.

---



---

**Note:** This is camera\_to\_world inverted.

---

**useViewport**

True when the camera is used as a viewport, set True to enable a viewport for this camera.

**Type** boolean

**sphereInsideFrustum** (*centre, radius*)

Tests the given sphere against the view frustum.

**Parameters**

- **centre** (*list [x, y, z]*) – The centre of the sphere (in world coordinates.)
- **radius** (*float*) – the radius of the sphere

**Returns** `INSIDE`, `OUTSIDE` or `INTERSECT`

**Return type** integer

---

**Note:** When the camera is first initialized the result will be invalid because the projection matrix has not been set.

---

```
import GameLogic
co = GameLogic.getCurrentController()
cam = co.owner

# A sphere of radius 4.0 located at [x, y, z] = [1.0, 1.0, 1.0]
if (cam.sphereInsideFrustum([1.0, 1.0, 1.0], 4) != cam.OUTSIDE):
    # Sphere is inside frustum !
    # Do something useful !
else:
    # Sphere is outside frustum
```

**boxInsideFrustum** (*box*)

Tests the given box against the view frustum.

**Parameters** **box** (*list of lists*) – Eight (8) corner points of the box (in world coordinates.)

**Returns** `INSIDE`, `OUTSIDE` or `INTERSECT`

---

**Note:** When the camera is first initialized the result will be invalid because the projection matrix has not been set.

---

```
import GameLogic
co = GameLogic.getCurrentController()
cam = co.owner

# Box to test...
box = []
```

```
box.append([-1.0, -1.0, -1.0])
box.append([-1.0, -1.0, 1.0])
box.append([-1.0, 1.0, -1.0])
box.append([-1.0, 1.0, 1.0])
box.append([ 1.0, -1.0, -1.0])
box.append([ 1.0, -1.0, 1.0])
box.append([ 1.0, 1.0, -1.0])
box.append([ 1.0, 1.0, 1.0])

if (cam.boxInsideFrustum(box) != cam.OUTSIDE):
    # Box is inside/intersects frustum !
    # Do something useful !
else:
    # Box is outside the frustum !
```

**pointInsideFrustum** (*point*)

Tests the given point against the view frustum.

**Parameters** **point** (*3D Vector*) – The point to test (in world coordinates.)

**Returns** True if the given point is inside this camera's viewing frustum.

**Return type** boolean

---

**Note:** When the camera is first initialized the result will be invalid because the projection matrix has not been set.

---

```
import GameLogic
co = GameLogic.getCurrentController()
cam = co.owner

# Test point [0.0, 0.0, 0.0]
if (cam.pointInsideFrustum([0.0, 0.0, 0.0])):
    # Point is inside frustum !
    # Do something useful !
else:
    # Box is outside the frustum !
```

**getCameraToWorld** ()

Returns the camera-to-world transform.

**Returns** the camera-to-world transform matrix.

**Return type** matrix (4x4 list)

**getWorldToCamera** ()

Returns the world-to-camera transform.

This returns the inverse matrix of `getCameraToWorld()`.

**Returns** the world-to-camera transform matrix.

**Return type** matrix (4x4 list)

**setOnTop** ()

Set this cameras viewport ontop of all other viewport.

**setViewport** (*left, bottom, right, top*)

Sets the region of this viewport on the screen in pixels.

Use `bge.render.getWindowHeight` and `bge.render.getWindowWidth` to calculate values relative to the entire display.

#### Parameters

- **left** (*integer*) – left pixel coordinate of this viewport
- **bottom** (*integer*) – bottom pixel coordinate of this viewport
- **right** (*integer*) – right pixel coordinate of this viewport
- **top** (*integer*) – top pixel coordinate of this viewport

#### `getScreenPosition` (*object*)

Gets the position of an object projected on screen space.

```
# For an object in the middle of the screen, coord = [0.5, 0.5]
coord = camera.getScreenPosition(object)
```

**Parameters** **object** (`KX_GameObject` or 3D Vector) – object name or list [x, y, z]

**Returns** the object's position in screen coordinates.

**Return type** list [x, y]

#### `getScreenVect` (*x, y*)

Gets the vector from the camera position in the screen coordinate direction.

#### Parameters

- **x** (*float*) – X Axis
- **y** (*float*) – Y Axis

**Return type** 3D Vector

**Returns** The vector from screen coordinate.

```
# Gets the vector of the camera front direction:
m_vect = camera.getScreenVect(0.5, 0.5)
```

#### `getScreenRay` (*x, y, dist=inf, property=None*)

Look towards a screen coordinate (x, y) and find first object hit within dist that matches prop. The ray is similar to `KX_GameObject->rayCastTo`.

#### Parameters

- **x** (*float*) – X Axis
- **y** (*float*) – Y Axis
- **dist** (*float*) – max distance to look (can be negative => look behind); 0 or omitted => detect up to other
- **property** (*string*) – property name that object must have; can be omitted => detect any object

**Return type** `KX_GameObject`

**Returns** the first object hit or None if no object or object does not match prop

```
# Gets an object with a property "wall" in front of the camera within a distance of 100:
target = camera.getScreenRay(0.5, 0.5, 100, "wall")
```

`class bge.types.BL_ArmatureObject` (`KX_GameObject`)

An armature object.

**constraints**

The list of armature constraint defined on this armature. Elements of the list can be accessed by index or string. The key format for string access is '<bone\_name>:<constraint\_name>'.

**Type** list of `BL_ArmatureConstraint`

**channels**

The list of armature channels. Elements of the list can be accessed by index or name the bone.

**Type** list of `BL_ArmatureChannel`

**update ()**

Ensures that the armature will be updated on next graphic frame.

This action is unnecessary if a `KX_ArmatureActuator` with mode run is active or if an action is playing. Use this function in other cases. It must be called on each frame to ensure that the armature is updated continuously.

**class** `bge.types.BL_ArmatureActuator` (`SCA_IActuator`)

Armature Actuators change constraint condition on armatures. Constants related to `type`

**KX\_ACT\_ARMATURE\_RUN**

Just make sure the armature will be updated on the next graphic frame. This is the only persistent mode of the actuator: it executes automatically once per frame until stopped by a controller

**Value** 0

**KX\_ACT\_ARMATURE\_ENABLE**

Enable the constraint.

**Value** 1

**KX\_ACT\_ARMATURE\_DISABLE**

Disable the constraint (runtime constraint values are not updated).

**Value** 2

**KX\_ACT\_ARMATURE\_SETTARGET**

Change target and subtarget of constraint.

**Value** 3

**KX\_ACT\_ARMATURE\_SETWEIGHT**

Change weight of (only for IK constraint).

**Value** 4

**type**

The type of action that the actuator executes when it is active.

Can be one of *these constants*

**Type** integer

**constraint**

The constraint object this actuator is controlling.

**Type** `BL_ArmatureConstraint`

**target**

The object that this actuator will set as primary target to the constraint it controls.

**Type** `KX_GameObject`

**subtarget**

The object that this actuator will set as secondary target to the constraint it controls.

**Type** `KX_GameObject`.

---

**Note:** Currently, the only secondary target is the pole target for IK constraint.

---

**weight**

The weight this actuator will set on the constraint it controls.

**Type** `float`.

---

**Note:** Currently only the IK constraint has a weight. It must be a value between 0 and 1.

---



---

**Note:** A weight of 0 disables a constraint while still updating constraint runtime values (see [BL\\_ArmatureConstraint](#))

---

**class** `bge.types.KX_ArmatureSensor` (*SCA\_ISensor*)

Armature sensor detect conditions on armatures. Constants related to `type`

**KX\_ARMSENSOR\_STATE\_CHANGED**

Detect that the constraint is changing state (active/inactive)

**Value** 0

**KX\_ARMSENSOR\_LIN\_ERROR\_BELOW**

Detect that the constraint linear error is above a threshold

**Value** 1

**KX\_ARMSENSOR\_LIN\_ERROR\_ABOVE**

Detect that the constraint linear error is below a threshold

**Value** 2

**KX\_ARMSENSOR\_ROT\_ERROR\_BELOW**

Detect that the constraint rotation error is above a threshold

**Value** 3

**KX\_ARMSENSOR\_ROT\_ERROR\_ABOVE**

Detect that the constraint rotation error is below a threshold

**Value** 4

**type**

The type of measurement that the sensor make when it is active.

Can be one of *these constants*

**Type** `integer`.

**constraint**

The constraint object this sensor is watching.

**Type** `BL_ArmatureConstraint`

**value**

The threshold used in the comparison with the constraint error The linear error is only updated on CopyPose/Distance IK constraint with iTaSC solver The rotation error is only updated on CopyPose+rotation IK constraint with iTaSC solver The linear error on CopyPose is always  $\geq 0$ : it is the norm of the distance between the target and the bone The rotation error on CopyPose is always  $\geq 0$ : it is the norm of the equivalent rotation vector between the bone and the target orientations The linear error on Distance

can be positive if the distance between the bone and the target is greater than the desired distance, and negative if the distance is smaller.

**Type** float

**class** `bge.types.BL_ArmatureConstraint` (*PyObjectPlus*)  
Proxy to Armature Constraint. Allows to change constraint on the fly. Obtained through `BL_ArmatureObject.constraints`.

---

**Note:** Not all armature constraints are supported in the GE.

---

Constants related to `type`

**CONSTRAINT\_TYPE\_TRACKTO**

**CONSTRAINT\_TYPE\_KINEMATIC**

**CONSTRAINT\_TYPE\_ROTLIKE**

**CONSTRAINT\_TYPE\_LOCLIKE**

**CONSTRAINT\_TYPE\_MINMAX**

**CONSTRAINT\_TYPE\_SIZELIKE**

**CONSTRAINT\_TYPE\_LOCKTRACK**

**CONSTRAINT\_TYPE\_STRETCHTO**

**CONSTRAINT\_TYPE\_CLAMPTO**

**CONSTRAINT\_TYPE\_TRANSFORM**

**CONSTRAINT\_TYPE\_DISTLIMIT**

Constants related to `ik_type`

**CONSTRAINT\_IK\_COPYPOSE**

constraint is trying to match the position and eventually the rotation of the target.

**Value** 0

**CONSTRAINT\_IK\_DISTANCE**

Constraint is maintaining a certain distance to target subject to `ik_mode`

**Value** 1

Constants related to `ik_flag`

**CONSTRAINT\_IK\_FLAG\_TIP**

Set when the constraint operates on the head of the bone and not the tail

**Value** 1

**CONSTRAINT\_IK\_FLAG\_ROT**

Set when the constraint tries to match the orientation of the target

**Value** 2

**CONSTRAINT\_IK\_FLAG\_STRETCH**

Set when the armature is allowed to stretch (only the bones with stretch factor > 0.0)

**Value** 16

**CONSTRAINT\_IK\_FLAG\_POS**

Set when the constraint tries to match the position of the target.



**Value** 32

Constants related to `ik_mode`

**CONSTRAINT\_IK\_MODE\_INSIDE**

The constraint tries to keep the bone within `ik_dist` of target

**Value** 0

**CONSTRAINT\_IK\_MODE\_OUTSIDE**

The constraint tries to keep the bone outside `ik_dist` of the target

**Value** 1

**CONSTRAINT\_IK\_MODE\_ONSURFACE**

The constraint tries to keep the bone exactly at `ik_dist` of the target.

**Value** 2

**type**

Type of constraint, (read-only).

Use one of *these constants*.

**Type** integer, one of `CONSTRAINT_TYPE_*` constants

**name**

Name of constraint constructed as `<bone_name>:<constraint_name>`. constraints list.

**Type** string

This name is also the key subscript on `BL_ArmatureObject`.

**enforce**

fraction of constraint effect that is enforced. Between 0 and 1.

**Type** float

**headtail**

Position of target between head and tail of the target bone: 0=head, 1=tail.

**Type** float.

---

**Note:** Only used if the target is a bone (i.e target object is an armature).

---

**lin\_error**

runtime linear error (in Blender units) on constraint at the current frame.

This is a runtime value updated on each frame by the IK solver. Only available on IK constraint and iTaSC solver.

**Type** float

**rot\_error**

Runtime rotation error (in radiant) on constraint at the current frame.

**Type** float.

This is a runtime value updated on each frame by the IK solver. Only available on IK constraint and iTaSC solver.

It is only set if the constraint has a rotation part, for example, a CopyPose+Rotation IK constraint.

**target**

Primary target object for the constraint. The position of this object in the GE will be used as target for the constraint.

**Type** `KX_GameObject`.

**subtarget**

Secondary target object for the constraint. The position of this object in the GE will be used as secondary target for the constraint.

**Type** `KX_GameObject`.

Currently this is only used for pole target on IK constraint.

**active**

True if the constraint is active.

**Type** `boolean`

---

**Note:** An inactive constraint does not update `lin_error` and `rot_error`.

---

**ik\_weight**

Weight of the IK constraint between 0 and 1.

Only defined for IK constraint.

**Type** `float`

**ik\_type**

Type of IK constraint, (read-only).

Use one of *these constants*.

**Type** `integer`.

**ik\_flag**

Combination of IK constraint option flags, read-only.

Use one of *these constants*.

**Type** `integer`

**ik\_dist**

Distance the constraint is trying to maintain with target, only used when `ik_type=CONSTRAINT_IK_DISTANCE`.

**Type** `float`

**ik\_mode**

Use one of *these constants*.

Additional mode for IK constraint. Currently only used for Distance constraint:

**Type** `integer`

**class** `bge.types.BL_ArmatureChannel` (*PyObjectPlus*)

Proxy to armature pose channel. Allows to read and set armature pose. The attributes are identical to RNA attributes, but mostly in read-only mode.

See `rotation_mode`

`PCHAN_ROT_QUAT`

`PCHAN_ROT_XYZ`

**PCHAN\_ROT\_XZY****PCHAN\_ROT\_YXZ****PCHAN\_ROT\_YZX****PCHAN\_ROT\_ZXY****PCHAN\_ROT\_ZYX****name**

channel name (=bone name), read-only.

**Type** string**bone**

return the bone object corresponding to this pose channel, read-only.

**Type** `BL_ArmatureBone`**parent**

return the parent channel object, None if root channel, read-only.

**Type** `BL_ArmatureChannel`**has\_ik**

true if the bone is part of an active IK chain, read-only. This flag is not set when an IK constraint is defined but not enabled (miss target information for example).

**Type** boolean**ik\_dof\_x**

true if the bone is free to rotation in the X axis, read-only.

**Type** boolean**ik\_dof\_y**

true if the bone is free to rotation in the Y axis, read-only.

**Type** boolean**ik\_dof\_z**

true if the bone is free to rotation in the Z axis, read-only.

**Type** boolean**ik\_limit\_x**

true if a limit is imposed on X rotation, read-only.

**Type** boolean**ik\_limit\_y**

true if a limit is imposed on Y rotation, read-only.

**Type** boolean**ik\_limit\_z**

true if a limit is imposed on Z rotation, read-only.

**Type** boolean**ik\_rot\_control**

true if channel rotation should applied as IK constraint, read-only.

**Type** boolean

**ik\_lin\_control**

true if channel size should applied as IK constraint, read-only.

**Type** boolean

**location**

displacement of the bone head in armature local space, read-write.

**Type** vector [X, Y, Z].

---

**Note:** You can only move a bone if it is unconnected to its parent. An action playing on the armature may change the value. An IK chain does not update this value, see `joint_rotation`.

---

**Note:** Changing this field has no immediate effect, the pose is updated when the armature is updated during the graphic render (see `BL_ArmatureObject.update`).

---

**scale**

scale of the bone relative to its parent, read-write.

**Type** vector [sizeX, sizeY, sizeZ].

---

**Note:** An action playing on the armature may change the value. An IK chain does not update this value, see `joint_rotation`.

---

**Note:** Changing this field has no immediate effect, the pose is updated when the armature is updated during the graphic render (see `BL_ArmatureObject.update`).

---

**rotation\_quaternion**

rotation of the bone relative to its parent expressed as a quaternion, read-write.

**Type** vector [qr, qi, qj, qk].

---

**Note:** This field is only used if `rotation_mode` is 0. An action playing on the armature may change the value. An IK chain does not update this value, see `joint_rotation`.

---

**Note:** Changing this field has no immediate effect, the pose is updated when the armature is updated during the graphic render (see `BL_ArmatureObject.update`).

---

**rotation\_euler**

rotation of the bone relative to its parent expressed as a set of euler angles, read-write.

**Type** vector [X, Y, Z].

---

**Note:** This field is only used if `rotation_mode` is > 0. You must always pass the angles in [X, Y, Z] order; the order of applying the angles to the bone depends on `rotation_mode`. An action playing on the armature may change this field. An IK chain does not update this value, see `joint_rotation`.

---

**Note:** Changing this field has no immediate effect, the pose is updated when the armature is updated during the graphic render (see `BL_ArmatureObject.update`).

---

**rotation\_mode**

Method of updating the bone rotation, read-write.

**Type** integer

Use the following constants (euler mode are named as in Blender UI but the actual axis order is reversed).

- PCHAN\_ROT\_QUAT(0) : use quaternion in rotation attribute to update bone rotation
- PCHAN\_ROT\_XYZ(1) : use euler\_rotation and apply angles on bone's Z, Y, X axis successively
- PCHAN\_ROT\_XZY(2) : use euler\_rotation and apply angles on bone's Y, Z, X axis successively
- PCHAN\_ROT\_YXZ(3) : use euler\_rotation and apply angles on bone's Z, X, Y axis successively
- PCHAN\_ROT\_YZX(4) : use euler\_rotation and apply angles on bone's X, Z, Y axis successively
- PCHAN\_ROT\_ZXY(5) : use euler\_rotation and apply angles on bone's Y, X, Z axis successively
- PCHAN\_ROT\_ZYX(6) : use euler\_rotation and apply angles on bone's X, Y, Z axis successively

**channel\_matrix**

pose matrix in bone space (deformation of the bone due to action, constraint, etc), Read-only. This field is updated after the graphic render, it represents the current pose.

**Type** matrix [4][4]

**pose\_matrix**

pose matrix in armature space, read-only, This field is updated after the graphic render, it represents the current pose.

**Type** matrix [4][4]

**pose\_head**

position of bone head in armature space, read-only.

**Type** vector [x, y, z]

**pose\_tail**

position of bone tail in armature space, read-only.

**Type** vector [x, y, z]

**ik\_min\_x**

minimum value of X rotation in degree ( $\leq 0$ ) when X rotation is limited (see ik\_limit\_x), read-only.

**Type** float

**ik\_max\_x**

maximum value of X rotation in degree ( $\geq 0$ ) when X rotation is limited (see ik\_limit\_x), read-only.

**Type** float

**ik\_min\_y**

minimum value of Y rotation in degree ( $\leq 0$ ) when Y rotation is limited (see ik\_limit\_y), read-only.

**Type** float

**ik\_max\_y**

maximum value of Y rotation in degree ( $\geq 0$ ) when Y rotation is limited (see ik\_limit\_y), read-only.

**Type** float

**ik\_min\_z**

minimum value of Z rotation in degree ( $\leq 0$ ) when Z rotation is limited (see ik\_limit\_z), read-only.

**Type** float

**ik\_max\_z**

maximum value of Z rotation in degree ( $\geq 0$ ) when Z rotation is limited (see `ik_limit_z`), read-only.

**Type** float

**ik\_stiffness\_x**

bone rotation stiffness in X axis, read-only.

**Type** float between 0 and 1

**ik\_stiffness\_y**

bone rotation stiffness in Y axis, read-only.

**Type** float between 0 and 1

**ik\_stiffness\_z**

bone rotation stiffness in Z axis, read-only.

**Type** float between 0 and 1

**ik\_stretch**

ratio of scale change that is allowed, 0=bone can't change size, read-only.

**Type** float

**ik\_rot\_weight**

weight of rotation constraint when `ik_rot_control` is set, read-write.

**Type** float between 0 and 1

**ik\_lin\_weight**

weight of size constraint when `ik_lin_control` is set, read-write.

**Type** float between 0 and 1

**joint\_rotation**

Control bone rotation in term of joint angle (for robotic applications), read-write.

When writing to this attribute, you pass a [x, y, z] vector and an appropriate set of euler angles or quaternion is calculated according to the `rotation_mode`.

When you read this attribute, the current pose matrix is converted into a [x, y, z] vector representing the joint angles.

The value and the meaning of the x, y, z depends on the `ik_dof_x/ik_dof_y/ik_dof_z` attributes:

- 1DoF joint X, Y or Z: the corresponding x, y, or z value is used as a joint angle in radian
- 2DoF joint X+Y or Z+Y: treated as 2 successive 1DoF joints: first X or Z, then Y. The x or z value is used as a joint angle in radian along the X or Z axis, followed by a rotation along the new Y axis of y radians.
- 2DoF joint X+Z: treated as a 2DoF joint with rotation axis on the X/Z plane. The x and z values are used as the coordinates of the rotation vector in the X/Z plane.
- 3DoF joint X+Y+Z: treated as a revolute joint. The [x, y, z] vector represents the equivalent rotation vector to bring the joint from the rest pose to the new pose.

**Type** vector [x, y, z]

---

**Note:** The bone must be part of an IK chain if you want to set the `ik_dof_x/ik_dof_y/ik_dof_z` attributes via the UI, but this will interfere with this attribute since the IK solver will overwrite the pose. You can stay in control of the armature if you create an IK constraint but do not finalize it (e.g. don't set a target) the IK solver will not run but the IK panel will show up on the UI for each bone in the chain.

---

**Note:** [0, 0, 0] always corresponds to the rest pose.

---

**Note:** You must request the armature pose to update and wait for the next graphic frame to see the effect of setting this attribute (see [BL\\_ArmatureObject.update](#)).

---

**Note:** You can read the result of the calculation in rotation or euler\_rotation attributes after setting this attribute.

---

**class** `bge.types.BL_ArmatureBone` (*PyObjectPlus*)

Proxy to Blender bone structure. All fields are read-only and comply to RNA names. All space attribute correspond to the rest pose.

**name**

bone name.

**Type** string

**connected**

true when the bone head is struck to the parent's tail.

**Type** boolean

**hinge**

true when bone doesn't inherit rotation or scale from parent bone.

**Type** boolean

**inherit\_scale**

true when bone inherits scaling from parent bone.

**Type** boolean

**bbone\_segments**

number of B-bone segments.

**Type** integer

**roll**

bone rotation around head-tail axis.

**Type** float

**head**

location of head end of the bone in parent bone space.

**Type** vector [x, y, z]

**tail**

location of head end of the bone in parent bone space.

**Type** vector [x, y, z]

**length**

bone length.

**Type** float

**arm\_head**

location of head end of the bone in armature space.

**Type** vector [x, y, z]

**arm\_tail**

location of tail end of the bone in armature space.

**Type** vector [x, y, z]

**arm\_mat**

matrix of the bone head in armature space.

**Type** matrix [4][4]

---

**Note:** This matrix has no scale part.

---

**bone\_mat**

rotation matrix of the bone in parent bone space.

**Type** matrix [3][3]

**parent**

parent bone, or None for root bone.

**Type** `BL_ArmatureBone`

**children**

list of bone's children.

**Type** list of `BL_ArmatureBone`

## 4.2 Game Logic (bge.logic)

### 4.2.1 Intro

Module to access logic functions, imported automatically into the python controllers namespace.

```
# To get the controller thats running this python script:
cont = bge.logic.getCurrentController() # bge.logic is automatically imported

# To get the game object this controller is on:
obj = cont.owner
```

`KX_GameObject` and `KX_Camera` or `KX_LightObject` methods are available depending on the type of object

```
# To get a sensor linked to this controller.
# "sensorname" is the name of the sensor as defined in the Blender interface.
# +-----+ +-----+
# | Sensor "sensorname" +---+ Python +
# +-----+ +-----+
sens = cont.sensors["sensorname"]

# To get a sequence of all sensors:
sensors = co.sensors
```

See the sensor's reference for available methods:

- `KX_MouseFocusSensor`
- `KX_NearSensor`



- `KX_NetworkMessageSensor`
- `KX_RadarSensor`
- `KX_RaySensor`
- `KX_TouchSensor`
- `SCA_DelaySensor`
- `SCA_JoystickSensor`
- `SCA_KeyboardSensor`
- `SCA_MouseSensor`
- `SCA_PropertySensor`
- `SCA_RandomSensor`

You can also access actuators linked to the controller

```
# To get an actuator attached to the controller:
#
#           +-----+ +-----+
#           + Python +--+ Actuator "actuatorname" |
#           +-----+ +-----+
actuator = co.actuators["actuatorname"]

# Activate an actuator
controller.activate(actuator)
```

See the actuator's reference for available methods

- `BL_ActionActuator`
- `BL_ShapeActionActuator`
- `KX_CameraActuator`
- `KX_ConstraintActuator`
- `KX_GameActuator`
- `KX_IpoActuator`
- `KX_NetworkMessageActuator`
- `KX_ObjectActuator`
- `KX_ParentActuator`
- `KX_SCA_AddObjectActuator`
- `KX_SCA_DynamicActuator`
- `KX_SCA_EndObjectActuator`
- `KX_SCA_ReplaceMeshActuator`
- `KX_SceneActuator`
- `KX_SoundActuator`
- `KX_StateActuator`
- `KX_TrackToActuator`
- `KX_VisibilityActuator`
- `SCA_2DFilterActuator`
- `SCA_PropertyActuator`
- `SCA_RandomActuator`

Most logic brick's methods are accessors for the properties available in the logic buttons. Consult the logic bricks documentation for more information on how each logic brick works.

There are also methods to access the current `bge.types.KX_Scene`

```
# Get the current scene
scene = bge.logic.getCurrentScene()

# Get the current camera
cam = scene.active_camera
```

Matrices as used by the game engine are **row major** `matrix[row][col] = float`

`bge.types.KX_Camera` has some examples using matrices.

## 4.2.2 Variables

`bge.logic.globalDict`

A dictionary that is saved between loading blend files so you can use it to store inventory and other variables you want to store between scenes and blend files. It can also be written to a file and loaded later on with the game load/save actuators.

---

**Note:** only python built in types such as int/string/bool/float/tuples/lists can be saved, GameObjects, Actuators etc will not work as expected.

---

`bge.logic.keyboard`

The current keyboard wrapped in an `SCA_PythonKeyboard` object.

`bge.logic.mouse`

The current mouse wrapped in an `SCA_PythonMouse` object.

## 4.2.3 General functions

`bge.logic.getCurrentController()`

Gets the Python controller associated with this Python script.

**Return type** `bge.types.SCA_PythonController`

`bge.logic.getCurrentScene()`

Gets the current Scene.

**Return type** `bge.types.KX_Scene`

`bge.logic.getSceneList()`

Gets a list of the current scenes loaded in the game engine.

**Return type** list of `bge.types.KX_Scene`

---

**Note:** Scenes in your blend file that have not been converted wont be in this list. This list will only contain scenes such as overlays scenes.

---

`bge.logic.loadGlobalDict()`

Loads `bge.logic.globalDict` from a file.

`bge.logic.saveGlobalDict()`

Saves `bge.logic.globalDict` to a file.

`bge.logic.startGame(blend)`

Loads the blend file.

**Parameters** `blend` (*string*) – The name of the blend file

`bge.logic.endGame()`

Ends the current game.

`bge.logic.restartGame()`

Restarts the current game by reloading the .blend file (the last saved version, not what is currently running).

`bge.logic.LibLoad` (*blend, type, data, load\_actions=False, verbose=False*)

Converts the all of the datablocks of the given type from the given blend.

#### Parameters

- **blend** (*string*) – The path to the blend file (or the name to use for the library if data is supplied)
- **type** (*string*) – The datablock type (currently only “Action”, “Mesh” and “Scene” are supported)
- **data** (*bytes*) – Binary data from a blend file (optional)
- **load\_actions** (*bool*) – Search for and load all actions in a given Scene and not just the “active” actions (Scene type only)
- **verbose** (*bool*) – Whether or not to print debugging information (e.g., “SceneName: Scene”)

`bge.logic.LibNew` (*name, type, data*)

Uses existing datablock data and loads in as a new library.

#### Parameters

- **name** (*string*) – A unique library name used for removal later
- **type** (*string*) – The datablock type (currently only “Mesh” is supported)
- **data** (*list of strings*) – A list of names of the datablocks to load

`bge.logic.LibFree` (*name*)

Frees a library, removing all objects and meshes from the currently active scenes.

**Parameters** **name** (*string*) – The name of the library to free (the name used in LibNew)

`bge.logic.LibList` ()

Returns a list of currently loaded libraries.

**Return type** list [str]

`bge.logic.addScene` (*name, overlay=1*)

Loads a scene into the game engine.

---

**Note:** This function is not effective immediately, the scene is queued and added on the next logic cycle where it will be available from `getSceneList`

---

#### Parameters

- **name** (*string*) – The name of the scene
- **overlay** (*integer*) – Overlay or underlay (optional)

`bge.logic.sendMessage` (*subject, body="" , to="" , message\_from=""*)

Sends a message to sensors in any active scene.

#### Parameters

- **subject** (*string*) – The subject of the message
- **body** (*string*) – The body of the message (optional)
- **to** (*string*) – The name of the object to send the message to (optional)

- **message\_from** (*string*) – The name of the object that the message is coming from (optional)

`bge.logic.setGravity` (*gravity*)

Sets the world gravity.

`bge.logic.getSpectrum` ()

Returns a 512 point list from the sound card. This only works if the fmod sound driver is being used.

**Return type** list [float], len(getSpectrum()) == 512

`bge.logic.stopDSP` ()

Stops the sound driver using DSP effects.

Only the fmod sound driver supports this. DSP can be computationally expensive.

`bge.logic.getMaxLogicFrame` ()

Gets the maximum number of logic frames per render frame.

**Returns** The maximum number of logic frames per render frame

**Return type** integer

`bge.logic.setMaxLogicFrame` (*maxlogic*)

Sets the maximum number of logic frames that are executed per render frame. This does not affect the physics system that still runs at full frame rate.

**Parameters** **maxlogic** (*integer*) – The new maximum number of logic frames per render frame.  
Valid values: 1..5

`bge.logic.getMaxPhysicsFrame` ()

Gets the maximum number of physics frames per render frame.

**Returns** The maximum number of physics frames per render frame

**Return type** integer

`bge.logic.setMaxPhysicsFrame` (*maxphysics*)

Sets the maximum number of physics timestep that are executed per render frame. Higher value allows physics to keep up with realtime even if graphics slows down the game. Physics timestep is fixed and equal to 1/tickrate (see setLogicTicRate) maxphysics/ticrate is the maximum delay of the renderer that physics can compensate.

**Parameters** **maxphysics** (*integer*) – The new maximum number of physics timestep per render frame. Valid values: 1..5.

`bge.logic.getLogicTicRate` ()

Gets the logic update frequency.

**Returns** The logic frequency in Hz

**Return type** float

`bge.logic.setLogicTicRate` (*ticrate*)

Sets the logic update frequency.

The logic update frequency is the number of times logic bricks are executed every second. The default is 60 Hz.

**Parameters** **ticrate** (*float*) – The new logic update frequency (in Hz).

`bge.logic.getPhysicsTicRate` ()

Gets the physics update frequency

**Returns** The physics update frequency in Hz

**Return type** float

`bge.logic.setPhysicsTicRate` (*ticrate*)

Sets the physics update frequency

The physics update frequency is the number of times the physics system is executed every second. The default is 60 Hz.

**Parameters** *ticrate* (*float*) – The new update frequency (in Hz).

## 4.2.4 Utility functions

`bge.logic.expandPath` (*path*)

Converts a blender internal path into a proper file system path.

Use / as directory separator in path You can use '/' at the start of the string to define a relative path; Blender replaces that string by the directory of the startup .blend or runtime file to make a full path name (doesn't change during the game, even if you load other .blend). The function also converts the directory separator to the local file system format.

**Parameters** *path* (*string*) – The path string to be converted/expanded.

**Returns** The converted string

**Return type** string

`bge.logic.getAverageFrameRate` ()

Gets the estimated/average framerate for all the active scenes, not only the current scene.

**Returns** The estimated average framerate in frames per second

**Return type** float

`bge.logic.getBlendFileList` (*path* = "/")

Returns a list of blend files in the same directory as the open blend file, or from using the option argument.

**Parameters** *path* (*string*) – Optional directory argument, will be expanded (like `expandPath`) into the full path.

**Returns** A list of filenames, with no directory prefix

**Return type** list

`bge.logic.getRandomFloat` ()

Returns a random floating point value in the range [0 - 1)

`bge.logic.PrintGLInfo` ()

Prints GL Extension Info into the console

## 4.2.5 Constants

`bge.logic.KX_TRUE`

True value used by some modules.

`bge.logic.KX_FALSE`

False value used by some modules.

## Sensors

### Sensor Status

`bge.logic.KX_SENSOR_INACTIVE`  
`bge.logic.KX_SENSOR_JUST_ACTIVATED`  
`bge.logic.KX_SENSOR_ACTIVE`  
`bge.logic.KX_SENSOR_JUST_DEACTIVATED`

### Property Sensor

`bge.logic.KX_PROPSENSOR_EQUAL`  
Activate when the property is equal to the sensor value.  
**Value 1**

`bge.logic.KX_PROPSENSOR_NOTEQUAL`  
Activate when the property is not equal to the sensor value.  
**Value 2**

`bge.logic.KX_PROPSENSOR_INTERVAL`  
Activate when the property is between the specified limits.  
**Value 3**

`bge.logic.KX_PROPSENSOR_CHANGED`  
Activate when the property changes  
**Value 4**

`bge.logic.KX_PROPSENSOR_EXPRESSION`  
Activate when the expression matches  
**Value 5**

### Radar Sensor

See `bge.types.KX_RadarSensor`

`bge.logic.KX_RADAR_AXIS_POS_X`  
`bge.logic.KX_RADAR_AXIS_POS_Y`  
`bge.logic.KX_RADAR_AXIS_POS_Z`  
`bge.logic.KX_RADAR_AXIS_NEG_X`  
`bge.logic.KX_RADAR_AXIS_NEG_Y`  
`bge.logic.KX_RADAR_AXIS_NEG_Z`

### Ray Sensor

See `bge.types.KX_RaySensor`

`bge.logic.KX_RAY_AXIS_POS_X`

`bge.logic.KX_RAY_AXIS_POS_Y`

`bge.logic.KX_RAY_AXIS_POS_Z`

`bge.logic.KX_RAY_AXIS_NEG_X`

`bge.logic.KX_RAY_AXIS_NEG_Y`

`bge.logic.KX_RAY_AXIS_NEG_Z`

## Actuators

### Action Actuator

See `bge.types.BL_ActionActuator`

`bge.logic.KX_ACTIONACT_PLAY`

`bge.logic.KX_ACTIONACT_FLIPPER`

`bge.logic.KX_ACTIONACT_LOOPSTOP`

`bge.logic.KX_ACTIONACT_LOOPEND`

`bge.logic.KX_ACTIONACT_PROPERTY`

### Constraint Actuator

See `bge.types.KX_ConstraintActuator.option`

- Applicable to Distance constraint:

`bge.logic.KX_ACT_CONSTRAINT_NORMAL`  
Activate alignment to surface

`bge.logic.KX_ACT_CONSTRAINT_DISTANCE`  
Activate distance control

`bge.logic.KX_ACT_CONSTRAINT_LOCAL`  
Direction of the ray is along the local axis

- Applicable to Force field constraint:

`bge.logic.KX_ACT_CONSTRAINT_DOROTFH`  
Force field act on rotation as well

- Applicable to both:

`bge.logic.KX_ACT_CONSTRAINT_MATERIAL`  
Detect material rather than property

`bge.logic.KX_ACT_CONSTRAINT_PERMANENT`  
No deactivation if ray does not hit target

See `bge.types.KX_ConstraintActuator.limit`

`bge.logic.KX_CONSTRAINTACT_LOCX`  
Limit X coord.

`bge.logic.KX_CONSTRAINTACT_LOCY`  
Limit Y coord

`bge.logic.KX_CONSTRAINTACT_LOCZ`  
Limit Z coord

`bge.logic.KX_CONSTRAINTACT_ROT_X`  
Limit X rotation

`bge.logic.KX_CONSTRAINTACT_ROT_Y`  
Limit Y rotation

`bge.logic.KX_CONSTRAINTACT_ROT_Z`  
Limit Z rotation

`bge.logic.KX_CONSTRAINTACT_DIRNX`  
Set distance along negative X axis

`bge.logic.KX_CONSTRAINTACT_DIRNY`  
Set distance along negative Y axis

`bge.logic.KX_CONSTRAINTACT_DIRNZ`  
Set distance along negative Z axis

`bge.logic.KX_CONSTRAINTACT_DIRPX`  
Set distance along positive X axis

`bge.logic.KX_CONSTRAINTACT_DIRPY`  
Set distance along positive Y axis

`bge.logic.KX_CONSTRAINTACT_DIRPZ`  
Set distance along positive Z axis

`bge.logic.KX_CONSTRAINTACT_ORIX`  
Set orientation of X axis

`bge.logic.KX_CONSTRAINTACT_ORIY`  
Set orientation of Y axis

`bge.logic.KX_CONSTRAINTACT_ORIZ`  
Set orientation of Z axis

`bge.logic.KX_ACT_CONSTRAINT_FHNX`  
Set force field along negative X axis

`bge.logic.KX_ACT_CONSTRAINT_FHNY`  
Set force field along negative Y axis

`bge.logic.KX_ACT_CONSTRAINT_FHNZ`  
Set force field along negative Z axis

`bge.logic.KX_ACT_CONSTRAINT_FHPX`  
Set force field along positive X axis

`bge.logic.KX_ACT_CONSTRAINT_FHPY`  
Set force field along positive Y axis

`bge.logic.KX_ACT_CONSTRAINT_FHPZ`  
Set force field along positive Z axis

### Dynamic Actuator

See `bge.types.KX_SCA_DynamicActuator`

`bge.logic.KX_DYN_RESTORE_DYNAMICS`



---

`bge.logic.KX_DYN_DISABLE_DYNAMICS`  
`bge.logic.KX_DYN_ENABLE_RIGID_BODY`  
`bge.logic.KX_DYN_DISABLE_RIGID_BODY`  
`bge.logic.KX_DYN_SET_MASS`

### Game Actuator

See `bge.types.KX_GameActuator`

`bge.logic.KX_GAME_LOAD`  
`bge.logic.KX_GAME_START`  
`bge.logic.KX_GAME_RESTART`  
`bge.logic.KX_GAME_QUIT`  
`bge.logic.KX_GAME_SAVECFG`  
`bge.logic.KX_GAME_LOADCFG`

### IPO Actuator

See `bge.types.KX_IpoActuator`

`bge.logic.KX_IPOACT_PLAY`  
`bge.logic.KX_IPOACT_PINGPONG`  
`bge.logic.KX_IPOACT_FLIPPER`  
`bge.logic.KX_IPOACT_LOOPSTOP`  
`bge.logic.KX_IPOACT_LOOPEND`  
`bge.logic.KX_IPOACT_FROM_PROP`

### Parent Actuator

`bge.logic.KX_PARENT_REMOVE`  
`bge.logic.KX_PARENT_SET`

### Random Distributions

See `bge.types.SCA_RandomActuator`

`bge.logic.KX_RANDOMACT_BOOL_CONST`  
`bge.logic.KX_RANDOMACT_BOOL_UNIFORM`  
`bge.logic.KX_RANDOMACT_BOOL_BERNOUILLI`  
`bge.logic.KX_RANDOMACT_INT_CONST`  
`bge.logic.KX_RANDOMACT_INT_UNIFORM`  
`bge.logic.KX_RANDOMACT_INT_POISSON`

bge.logic.KX\_RANDOMACT\_FLOAT\_CONST  
bge.logic.KX\_RANDOMACT\_FLOAT\_UNIFORM  
bge.logic.KX\_RANDOMACT\_FLOAT\_NORMAL  
bge.logic.KX\_RANDOMACT\_FLOAT\_NEGATIVE\_EXPONENTIAL

### Scene Actuator

See [bge.types.KX\\_SceneActuator](#)

bge.logic.KX\_SCENE\_RESTART  
bge.logic.KX\_SCENE\_SET\_SCENE  
bge.logic.KX\_SCENE\_SET\_CAMERA  
bge.logic.KX\_SCENE\_ADD\_FRONT\_SCENE  
bge.logic.KX\_SCENE\_ADD\_BACK\_SCENE  
bge.logic.KX\_SCENE\_REMOVE\_SCENE  
bge.logic.KX\_SCENE\_SUSPEND  
bge.logic.KX\_SCENE\_RESUME

### Shape Action Actuator

See [bge.types.BL\\_ActionActuator](#)

bge.logic.KX\_ACTIONACT\_PLAY  
bge.logic.KX\_ACTIONACT\_FLIPPER  
bge.logic.KX\_ACTIONACT\_LOOPSTOP  
bge.logic.KX\_ACTIONACT\_LOOPEND  
bge.logic.KX\_ACTIONACT\_PROPERTY

### Sound Actuator

See [bge.types.KX\\_SoundActuator](#)

bge.logic.KX\_SOUNDACT\_PLAYSTOP  
    Value 1  
bge.logic.KX\_SOUNDACT\_PLAYEND  
    Value 2  
bge.logic.KX\_SOUNDACT\_LOOPSTOP  
    Value 3  
bge.logic.KX\_SOUNDACT\_LOOPEND  
    Value 4  
bge.logic.KX\_SOUNDACT\_LOOPBIDIRECTIONAL

**Value 5**

`bge.logic.KX_SOUNDACT_LOOPBIDIRECTIONAL_STOP`

**Value 6****Various****Input Status**

See `bge.types.SCA_PythonKeyboard`, `bge.types.SCA_PythonMouse`,  
`bge.types.SCA_MouseSensor`, `bge.types.SCA_KeyboardSensor`

`bge.logic.KX_INPUT_NONE`

`bge.logic.KX_INPUT_JUST_ACTIVATED`

`bge.logic.KX_INPUT_ACTIVE`

`bge.logic.KX_INPUT_JUST_RELEASED`

**Mouse Buttons**

See `bge.types.SCA_MouseSensor`

`bge.logic.KX_MOUSE_BUT_LEFT`

`bge.logic.KX_MOUSE_BUT_MIDDLE`

`bge.logic.KX_MOUSE_BUT_RIGHT`

**States**

See `bge.types.KX_StateActuator`

`bge.logic.KX_STATE1`

`bge.logic.KX_STATE2`

`bge.logic.KX_STATE3`

`bge.logic.KX_STATE4`

`bge.logic.KX_STATE5`

`bge.logic.KX_STATE6`

`bge.logic.KX_STATE7`

`bge.logic.KX_STATE8`

`bge.logic.KX_STATE9`

`bge.logic.KX_STATE10`

`bge.logic.KX_STATE11`

`bge.logic.KX_STATE12`

`bge.logic.KX_STATE13`

`bge.logic.KX_STATE14`

`bge.logic.KX_STATE15`

`bge.logic.KX_STATE16`

`bge.logic.KX_STATE17`

`bge.logic.KX_STATE18`

`bge.logic.KX_STATE19`

`bge.logic.KX_STATE20`

`bge.logic.KX_STATE21`

`bge.logic.KX_STATE22`

`bge.logic.KX_STATE23`

`bge.logic.KX_STATE24`

`bge.logic.KX_STATE25`

`bge.logic.KX_STATE26`

`bge.logic.KX_STATE27`

`bge.logic.KX_STATE28`

`bge.logic.KX_STATE29`

`bge.logic.KX_STATE30`

See `bge.types.KX_StateActuator.operation`

`bge.logic.KX_STATE_OP_CLR`

Subtract bits to state mask

**Value 0**

`bge.logic.KX_STATE_OP_CPY`

Copy state mask

**Value 1**

`bge.logic.KX_STATE_OP_NEG`

Invert bits to state mask

**Value 2**

`bge.logic.KX_STATE_OP_SET`

Add bits to state mask

**Value 3**

## 2D Filter

`bge.logic.RAS_2DFILTER_BLUR`

**Value 2**

`bge.logic.RAS_2DFILTER_CUSTOMFILTER`

Customer filter, the code code is set via `shaderText` property.

**Value 12**

`bge.logic.RAS_2DFILTER_DILATION`

**Value 4**

`bge.logic.RAS_2DFILTER_DISABLED`  
Disable the filter that is currently active

**Value -1**

`bge.logic.RAS_2DFILTER_ENABLED`  
Enable the filter that was previously disabled

**Value -2**

`bge.logic.RAS_2DFILTER_EROSION`

**Value 5**

`bge.logic.RAS_2DFILTER_GRAYSCALE`

**Value 9**

`bge.logic.RAS_2DFILTER_INVERT`

**Value 11**

`bge.logic.RAS_2DFILTER_LAPLACIAN`

**Value 6**

`bge.logic.RAS_2DFILTER_MOTIONBLUR`  
Create and enable preset filters

**Value 1**

`bge.logic.RAS_2DFILTER_NOFILTER`  
Disable and destroy the filter that is currently active

**Value 0**

`bge.logic.RAS_2DFILTER_PREWITT`

**Value 8**

`bge.logic.RAS_2DFILTER_SEPIA`

**Value 10**

`bge.logic.RAS_2DFILTER_SHARPEN`

**Value 3**

`bge.logic.RAS_2DFILTER_SOBEL`

**Value 7**

## Shader

`bge.logic.VIEWMATRIX`

`bge.logic.VIEWMATRIX_INVERSE`

`bge.logic.VIEWMATRIX_INVERSETRANSPOSE`

`bge.logic.VIEWMATRIX_TRANSPOSE`

`bge.logic.MODELMATRIX`

`bge.logic.MODELMATRIX_INVERSE`

bge.logic.**MODELMATRIX\_INVERSETRANSPOSE**  
bge.logic.**MODELMATRIX\_TRANSPOSE**  
bge.logic.**MODELVIEWMATRIX**  
bge.logic.**MODELVIEWMATRIX\_INVERSE**  
bge.logic.**MODELVIEWMATRIX\_INVERSETRANSPOSE**  
bge.logic.**MODELVIEWMATRIX\_TRANSPOSE**  
bge.logic.**CAM\_POS**  
    Current camera position  
bge.logic.**CONSTANT\_TIMER**  
    User a timer for the uniform value.  
bge.logic.**SHD\_TANGENT**

### Blender Material

bge.logic.**BL\_DST\_ALPHA**  
bge.logic.**BL\_DST\_COLOR**  
bge.logic.**BL\_ONE**  
bge.logic.**BL\_ONE\_MINUS\_DST\_ALPHA**  
bge.logic.**BL\_ONE\_MINUS\_DST\_COLOR**  
bge.logic.**BL\_ONE\_MINUS\_SRC\_ALPHA**  
bge.logic.**BL\_ONE\_MINUS\_SRC\_COLOR**  
bge.logic.**BL\_SRC\_ALPHA**  
bge.logic.**BL\_SRC\_ALPHA\_SATURATE**  
bge.logic.**BL\_SRC\_COLOR**  
bge.logic.**BL\_ZERO**

## 4.3 Rasterizer (bge.render)

### 4.3.1 Intro

```
# Example Uses an L{SCA_MouseSensor}, and two L{KX_ObjectActuator}s to implement MouseLook::  
# To use a mouse movement sensor "Mouse" and a  
# motion actuator to mouse look:  
import bge.render  
import bge.logic  
  
# scale sets the speed of motion  
scale = 1.0, 0.5  
  
co = bge.logic.getCurrentController()  
obj = co.getOwner()  
mouse = co.getSensor("Mouse")
```

```

lmotion = co.getActuator("LMove")
wmotion = co.getActuator("WMove")

# Transform the mouse coordinates to see how far the mouse has moved.
def mousePos():
    x = (bge.render.getWindowWidth() / 2 - mouse.getXPosition()) * scale[0]
    y = (bge.render.getWindowHeight() / 2 - mouse.getYPosition()) * scale[1]
    return (x, y)

pos = mousePos()

# Set the amount of motion: X is applied in world coordinates...
lmotion.setTorque(0.0, 0.0, pos[0], False)
# ...Y is applied in local coordinates
wmotion.setTorque(-pos[1], 0.0, 0.0, True)

# Activate both actuators
bge.logic.addActiveActuator(lmotion, True)
bge.logic.addActiveActuator(wmotion, True)

# Centre the mouse
bge.render.setMousePosition(bge.render.getWindowWidth() / 2, bge.render.getWindowHeight() / 2)

```

### 4.3.2 Constants

**bge.render.KX\_TEXFACE\_MATERIAL**  
Materials as defined by the texture face settings.

**bge.render.KX\_BLENDER\_MULTITEX\_MATERIAL**  
Materials approximating blender materials with multitexturing.

**bge.render.KX\_BLENDER\_GLSL\_MATERIAL**  
Materials approximating blender materials with GLSL.

### 4.3.3 Functions

**bge.render.getWindowWidth()**  
Gets the width of the window (in pixels)

**Return type** integer

**bge.render.getWindowHeight()**  
Gets the height of the window (in pixels)

**Return type** integer

**bge.render.makeScreenshot(filename)**  
Writes a screenshot to the given filename.

If filename starts with // the image will be saved relative to the current directory. If the filename contains # it will be replaced with the frame number.

The standalone player saves .png files. It does not support colour space conversion or gamma correction.

When run from Blender, makeScreenshot supports Iris, IrisZ, TGA, Raw TGA, PNG, HamX, and Jpeg. Gamma, Colourspace conversion and Jpeg compression are taken from the Render settings panels.

`bge.render.enableVisibility` (*visible*)  
Doesn't really do anything...

`bge.render.showMouse` (*visible*)  
Enables or disables the operating system mouse cursor.

`bge.render.setMousePosition` (*x, y*)  
Sets the mouse cursor position.

`bge.render.setBackgroundColor` (*rgba*)  
Sets the window background colour.

`bge.render.setMistColor` (*rgb*)  
Sets the mist colour.

`bge.render.setAmbientColor` (*rgb*)  
Sets the color of ambient light.

`bge.render.setMistStart` (*start*)  
Sets the mist start value. Objects further away than start will have mist applied to them.

`bge.render.setMistEnd` (*end*)  
Sets the mist end value. Objects further away from this will be coloured solid with the colour set by `setMistColor()`.

`bge.render.disableMist` ()  
Disables mist.

---

**Note:** Set any of the mist properties to enable mist.

---

`bge.render.setEyeSeparation` (*eyesep*)  
Sets the eye separation for stereo mode. Usually Focal Length/30 provides a comfortable value.

**Parameters** `eyesep` (*float*) – The distance between the left and right eye.

`bge.render.getEyeSeparation` ()  
Gets the current eye separation for stereo mode.

**Return type** float

`bge.render.setFocalLength` (*focallength*)  
Sets the focal length for stereo mode. It uses the current camera focal length as initial value.

**Parameters** `focallength` (*float*) – The focal length.

`bge.render.getFocalLength` ()  
Gets the current focal length for stereo mode.

**Return type** float



`bge.render.setMaterialMode` (*mode*)  
Set the material mode to use for OpenGL rendering.

---

**Note:** Changes will only affect newly created scenes.

---

`bge.render.getMaterialMode` (*mode*)  
Get the material mode to use for OpenGL rendering.

**Return type** KX\_TEXFACE\_MATERIAL, KX\_BLENDER\_MULTITEX\_MATERIAL,  
KX\_BLENDER\_GLSL\_MATERIAL

`bge.render.setGLSLMaterialSetting` (*setting, enable*)  
Enables or disables a GLSL material setting.

`bge.render.getGLSLMaterialSetting` (*setting, enable*)  
Get the state of a GLSL material setting.

**Return type** boolean

`bge.render.setAnisotropicFiltering` (*level*)  
Set the anisotropic filtering level for textures.

**Parameters** *level* (*integer (must be one of 1, 2, 4, 8, 16)*) – The new anisotropic filtering level to use

`bge.render.getAnisotropicFiltering` ()  
Get the anisotropic filtering level used for textures.

**Return type** integer (one of 1, 2, 4, 8, 16)

`bge.render.drawLine` (*fromVec, toVec, color*)  
Draw a line in the 3D scene.

**Parameters**

- **fromVec** (*list [x, y, z]*) – the origin of the line
- **toVec** (*list [x, y, z]*) – the end of the line
- **color** (*list [r, g, b]*) – the color of the line

`bge.render.enableMotionBlur` (*factor*)  
Enable the motion blur effect.

**Parameters** *factor* (*float [0.0 - 1.0]*) – the ammount of motion blur to display.

`bge.render.disableMotionBlur` ()  
Disable the motion blur effect.

## 4.4 Video Texture (bge.texture)

### 4.4.1 Intro

The `bge.texture` module allows you to manipulate textures during the game.

Several sources for texture are possible: video files, image files, video capture, memory buffer, camera render or a mix of that.

The video and image files can be loaded from the internet using an URL instead of a file name.

In addition, you can apply filters on the images before sending them to the GPU, allowing video effect: blue screen, color band, gray, normal map.

bge.texture uses FFmpeg to load images and videos. All the formats and codecs that FFmpeg supports are supported by this module, including but not limited to:

- \* AVI
- \* Ogg
- \* Xvid
- \* Theora
- \* dvl394 camera
- \* video4linux capture card (this includes many webcams)
- \* videoForWindows capture card (this includes many webcams)
- \* JPG

The principle is simple: first you identify a texture on an existing object using the :materialID: function, then you create a new texture with dynamic content and swap the two textures in the GPU.

The GE is not aware of the substitution and continues to display the object as always, except that you are now in control of the texture.

When the texture object is deleted, the new texture is deleted and the old texture restored.

```
"""
Basic Video Playback
+++++
Example of how to replace a texture in game with a video. It needs to run everyframe
"""
import bge
from bge import texture
from bge import logic

cont = logic.getCurrentController()
obj = cont.owner

# the creation of the texture must be done once: save the
# texture object in an attribute of bge.logic module makes it persistent
if not hasattr(logic, 'video'):

    # identify a static texture by name
    matID = texture.materialID(obj, 'IMvideo.png')

    # create a dynamic texture that will replace the static texture
    logic.video = texture.Texture(obj, matID)

    # define a source of image for the texture, here a movie
    movie = logic.expandPath('//trailer_400p.ogg')
    logic.video.source = texture.VideoFFmpeg(movie)
    logic.video.source.scale = True

    # quick off the movie, but it wont play in the background
    logic.video.source.play()

# you need to call this function every frame to ensure update of the texture.
logic.video.refresh(True)
```

```

"""
Texture replacement
+++++
Example of how to replace a texture in game with an external image.
createTexture() and removeTexture() are to be called from a module Python
Controller.
"""
from bge import logic
from bge import texture

def createTexture(cont):
    """Create a new Dynamic Texture"""
    object = cont.owner

    # get the reference pointer (ID) of the internal texture
    ID = texture.materialID(object, 'IMoriginal.png')

    # create a texture object
    object_texture = texture.Texture(object, ID)

    # create a new source with an external image
    url = logic.expandPath("//newtexture.jpg")
    new_source = texture.ImageFFmpeg(url)

    # the texture has to be stored in a permanent Python object
    logic.texture = object_texture

    # update/replace the texture
    logic.texture.source = new_source
    logic.texture.refresh(False)

def removeTexture(cont):
    """Delete the Dynamic Texture, reversing back the final to its original state."""
    try:
        del logic.texture
    except:
        pass

class bge.texture.VideoFFmpeg(file[, capture=-1, rate=25.0, width=0, height=0])
    FFmpeg video source

    status
        video status

    range
        replay range

    repeat
        repeat count, -1 for infinite repeat

        Type int

    framerate
        frame rate

        Type float

```

**valid**  
Tells if an image is available  
**Type** bool

**image**  
image data

**size**  
image size

**scale**  
fast scale of image (near neighbour)

**flip**  
flip image vertically

**filter**  
pixel filter

**preseek**  
number of frames of preseek  
**Type** int

**deinterlace**  
deinterlace image  
**Type** bool

**play ()**  
Play (restart) video

**pause ()**  
pause video

**stop ()**  
stop video (play will replay it from start)

**refresh ()**  
Refresh video - get its status

**class** `bge.texture.ImageFFmpeg` (*file*)  
FFmpeg image source

**status**  
video status

**valid**  
Tells if an image is available  
**Type** bool

**image**  
image data

**size**  
image size

**scale**  
fast scale of image (near neighbour)

**flip**  
flip image vertically

**filter**

pixel filter

**refresh ()**

Refresh image, i.e. load it

**reload ([*newname* ])**

Reload image, i.e. reopen it

**class** `bge.texture.ImageBuff`

Image source from image buffer

**filter**

pixel filter

**flip**

flip image vertically

**image**

image data

**load (*imageBuffer*, *width*, *height*)**

Load image from buffer

**plot (*imageBuffer*, *width*, *height*, *positionX*, *positionY*)**

update image buffer

**scale**

fast scale of image (near neighbour)

**size**

image size

**valid**

bool to tell if an image is available

**class** `bge.texture.ImageMirror` (*scene*)

Image source from mirror

**alpha**

use alpha in texture

**background**

background color

**capsize**

size of render area

**clip**

clipping distance

**filter**

pixel filter

**flip**

flip image vertically

**image**

image data

**refresh (*imageMirror*)**

Refresh image - invalidate its current content

**scale**  
fast scale of image (near neighbour)

**size**  
image size

**valid**  
bool to tell if an image is available

**whole**  
use whole viewport to render

**class** `bge.texture.ImageMix`  
Image mixer

**filter**  
pixel filter

**flip**  
flip image vertically

**getSource** (*imageMix*)  
get image source

**getWeight** (*imageMix*)  
get image source weight

**image**  
image data

**refresh** (*imageMix*)  
Refresh image - invalidate its current content

**scale**  
fast scale of image (near neighbour)

**setSource** (*imageMix*)  
set image source

**setWeight** (*imageMix*)  
set image source weight

**valid**  
bool to tell if an image is available

**class** `bge.texture.ImageRender` (*scene, camera*)  
Image source from render

**alpha**  
use alpha in texture

**background**  
background color

**capsize**  
size of render area

**filter**  
pixel filter

**flip**  
flip image vertically

**image**  
image data

**refresh** (*imageRender*)  
Refresh image - invalidate its current content

**scale**  
fast scale of image (near neighbour)

**size**  
image size

**valid**  
bool to tell if an image is available

**whole**  
use whole viewport to render

**class** `bge.texture.ImageViewport`  
Image source from viewport

**alpha**  
use alpha in texture

**capsize**  
size of viewport area being captured

**filter**  
pixel filter

**flip**  
flip image vertically

**image**  
image data

**position**  
upper left corner of captured area

**refresh** (*imageViewport*)  
Refresh image - invalidate its current content

**scale**  
fast scale of image (near neighbour)

**size**  
image size

**valid**  
bool to tell if an image is available

**whole**  
use whole viewport to capture

**class** `bge.texture.Texture` (*gameObj*)  
Texture objects

**bindId**  
OpenGL Bind Name

**close** (*texture*)  
Close dynamic texture and restore original

**mipmap**  
mipmap texture

**refresh** (*texture*)  
Refresh texture from source

**source**  
source of texture

**class** `bge.texture.FilterBGR24`  
Source filter BGR24 objects

**class** `bge.texture.FilterBlueScreen`  
Filter for Blue Screen objects

**color**  
blue screen color

**limits**  
blue screen color limits

**previous**  
previous pixel filter

**class** `bge.texture.FilterColor`  
Filter for color calculations

**matrix**  
matrix [4][5] for color calculation

**previous**  
previous pixel filter

**class** `bge.texture.FilterGray`  
Filter for gray scale effect

**previous**  
previous pixel filter

**class** `bge.texture.FilterLevel`  
Filter for levels calculations

**levels**  
levels matrix [4] (min, max)

**previous**  
previous pixel filter

**class** `bge.texture.FilterNormal`  
Filter for Blue Screen objects

**colorIdx**  
index of color used to calculate normal (0 - red, 1 - green, 2 - blue)

**depth**  
depth of relief

**previous**  
previous pixel filter

**class** `bge.texture.FilterRGB24`  
Returns a new input filter object to be used with `ImageBuff` object when the image passed to the `ImageBuff.load()` function has the 3-bytes pixel format BGR.



**class** `bge.texture.FilterRGBA32`

Source filter RGBA32 objects

`bge.texture.getLastError()`

Last error that occurred in a `bge.texture` function.

**Returns** the description of the last error occurred in a `bge.texture` function.

**Return type** string

`bge.texture.imageToArray(image, mode)`

Returns a buffer corresponding to the current image stored in a texture source object.

**Parameters**

- **image** (object of type `VideoFFmpeg`, `ImageFFmpeg`, `ImageBuff`, `ImageMix`, `ImageRender`, `ImageMirror` or `ImageViewport`) – Image source object.
- **mode** (*string*) – optional argument representing the pixel format. You can use the characters R, G, B for the 3 color channels, A for the alpha channel, 0 to force a fixed 0 color channel and 1 to force a fixed 255 color channel. Example: “BGR” will return 3 bytes per pixel with the Blue, Green and Red channels in that order. “RGB1” will return 4 bytes per pixel with the Red, Green, Blue channels in that order and the alpha channel forced to 255. The default mode is “RGBA”.

**Return type** buffer

**Returns** A object representing the image as one dimensional array of bytes of size (`pixel_size*width*height`), line by line starting from the bottom of the image. The pixel size and format is determined by the mode parameter.

## 4.5 Game Keys (bge.events)

### 4.5.1 Intro

This module holds key constants for the `SCA_KeyboardSensor`.

```
# Set a connected keyboard sensor to accept F1
import bge

co = bge.logic.getCurrentController()
# 'Keyboard' is a keyboard sensor
sensor = co.sensors["Keyboard"]
sensor.key = bge.events.F1KEY

# Do the all keys thing
import bge

co = bge.logic.getCurrentController()
# 'Keyboard' is a keyboard sensor
sensor = co.sensors["Keyboard"]

for key, status in sensor.events:
    # key[0] == bge.events.keycode, key[1] = status
    if status == bge.logic.KX_INPUT_JUST_ACTIVATED:
        if key == bge.events.WKEY:
            # Activate Forward!
```

```
        if key == bge.events.SKEY:
            # Activate Backward!
        if key == bge.events.AKEY:
            # Activate Left!
        if key == bge.events.DKEY:
            # Activate Right!

# The all keys thing without a keyboard sensor (but you will
# need an always sensor with pulse mode on)
import bge

# Just shortening names here
keyboard = bge.logic.keyboard
JUST_ACTIVATED = bge.logic.KX_INPUT_JUST_ACTIVATED

if keyboard.events[bge.events.WKEY] == JUST_ACTIVATED:
    print("Activate Forward!")
if keyboard.events[bge.events.SKEY] == JUST_ACTIVATED:
    print("Activate Backward!")
if keyboard.events[bge.events.AKEY] == JUST_ACTIVATED:
    print("Activate Left!")
if keyboard.events[bge.events.DKEY] == JUST_ACTIVATED:
    print("Activate Right!")
```

## 4.5.2 Functions

`bge.events.EventToString(event)`

Return the string name of a key event. Will raise a `ValueError` error if its invalid.

**Parameters** `event (int)` – key event from `bge.keys` or the keyboard sensor.

**Return type** string

`bge.events.EventToCharacter(event, shift)`

Return the string name of a key event. Returns an empty string if the event cant be represented as a character.

**Parameters**

- `event (int)` – key event from `bge.keys` or the keyboard sensor.
- `shift (bool)` – set to true if shift is held.

**Return type** string

## 4.5.3 Keys (Constants)

### Mouse Keys

`bge.events.LEFTMOUSE`

`bge.events.MIDDLEMOUSE`

`bge.events.RIGHTMOUSE`

`bge.events.WHEELUPMOUSE`

`bge.events.WHEELDOWNMOUSE`

`bge.events.MOUSEX`

`bge.events.MOUSEY`

## Keyboard Keys

### Alphabet keys

`bge.events.AKEY`

`bge.events.BKEY`

`bge.events.CKEY`

`bge.events.DKEY`

`bge.events.EKEY`

`bge.events.FKEY`

`bge.events.GKEY`

`bge.events.HKEY`

`bge.events.IKEY`

`bge.events.JKEY`

`bge.events.KKEY`

`bge.events.LKEY`

`bge.events.MKEY`

`bge.events.NKEY`

`bge.events.OKEY`

`bge.events.PKEY`

`bge.events.QKEY`

`bge.events.RKEY`

`bge.events.SKEY`

`bge.events.TKEY`

`bge.events.UKEY`

`bge.events.VKEY`

`bge.events.WKEY`

`bge.events.XKEY`

`bge.events.YKEY`

`bge.events.ZKEY`

### Number keys

`bge.events.ZEROKEY`

`bge.events.ONEKEY`

bge.events.**TWOKEY**  
bge.events.**THREEKEY**  
bge.events.**FOURKEY**  
bge.events.**FIVEKEY**  
bge.events.**SIXKEY**  
bge.events.**SEVENKEY**  
bge.events.**EIGHTKEY**  
bge.events.**NINEKEY**

### Modifiers Keys

bge.events.**CAPSLOCKKEY**  
bge.events.**LEFTCTRLKEY**  
bge.events.**LEFTALTKEY**  
bge.events.**RIGHTALTKEY**  
bge.events.**RIGHTCTRLKEY**  
bge.events.**RIGHTSHIFTKEY**  
bge.events.**LEFTSHIFTKEY**

### Arrow Keys

bge.events.**LEFTARROWKEY**  
bge.events.**DOWNARROWKEY**  
bge.events.**RIGHTARROWKEY**  
bge.events.**UPARROWKEY**

### Numberpad Keys

bge.events.**PAD0**  
bge.events.**PAD1**  
bge.events.**PAD2**  
bge.events.**PAD3**  
bge.events.**PAD4**  
bge.events.**PAD5**  
bge.events.**PAD6**  
bge.events.**PAD7**  
bge.events.**PAD8**  
bge.events.**PAD9**

bge.events.**PADPERIOD**  
bge.events.**PADSLASHKEY**  
bge.events.**PADASTERKEY**  
bge.events.**PADMINUS**  
bge.events.**PADENTER**  
bge.events.**PADPLUSKEY**

### Function Keys

bge.events.**F1KEY**  
bge.events.**F2KEY**  
bge.events.**F3KEY**  
bge.events.**F4KEY**  
bge.events.**F5KEY**  
bge.events.**F6KEY**  
bge.events.**F7KEY**  
bge.events.**F8KEY**  
bge.events.**F9KEY**  
bge.events.**F10KEY**  
bge.events.**F11KEY**  
bge.events.**F12KEY**  
bge.events.**F13KEY**  
bge.events.**F14KEY**  
bge.events.**F15KEY**  
bge.events.**F16KEY**  
bge.events.**F17KEY**  
bge.events.**F18KEY**  
bge.events.**F19KEY**

### Other Keys

bge.events.**ACCENTGRAVEKEY**  
bge.events.**BACKSLASHKEY**  
bge.events.**BACKSPACEKEY**  
bge.events.**COMMAKEY**  
bge.events.**DELKEY**  
bge.events.**ENDKEY**  
bge.events.**EQUALKEY**

bge.events.**ESCKEY**  
bge.events.**HOMEKEY**  
bge.events.**INSERTKEY**  
bge.events.**LEFTBRACKETKEY**  
bge.events.**LINEFEEDKEY**  
bge.events.**MINUSKEY**  
bge.events.**PAGEDOWNKEY**  
bge.events.**PAGEUPKEY**  
bge.events.**PAUSEKEY**  
bge.events.**PERIODKEY**  
bge.events.**QUOTEKEY**  
bge.events.**RIGHTBRACKETKEY**  
bge.events.**RETKEY** (*Deprecated: use bge.events.ENTERKEY*)  
bge.events.**ENTERKEY**  
bge.events.**SEMICOLONKEY**  
bge.events.**SLASHKEY**  
bge.events.**SPACEKEY**  
bge.events.**TABKEY**

## 4.6 Physics Constraints (bge.constraints)

```
"""  
Basic Physics Constraint  
+++++  
Example of how to create a hinge Physics Constraint between two objects.  
"""  
from bge import logic  
from bge import constraints  
  
# get object list  
objects = logic.getCurrentScene().objects  
  
# get object named Object1 and Object 2  
object_1 = objects["Object1"]  
object_2 = objects["Object2"]  
  
# want to use Edge constraint type  
constraint_type = 2  
  
# get Object1 and Object2 physics IDs  
physics_id_1 = object_1.getPhysicsId()  
physics_id_2 = object_2.getPhysicsId()  
  
# Use bottom right edge of Object1 for hinge position  
edge_position_x = 1.0
```

```

edge_position_y = 0.0
edge_position_z = -1.0

# use Object1 y axis for angle to point hinge
edge_angle_x = 0.0
edge_angle_y = 1.0
edge_angle_z = 0.0

# create an edge constraint
constraints.createConstraint(physics_id_1, physics_id_2,
                             constraint_type,
                             edge_position_x, edge_position_y, edge_position_z,
                             edge_angle_x, edge_angle_y, edge_angle_z)

bge.constraints.createConstraint (physicsid, physicsid2, constrainttype, [pivotX, pivotY, pivotZ,
                                                                           [axisX, axisY, axisZ, [flag]]])

```

Creates a constraint.

#### Parameters

- **physicsid** (*int*) – the physics id of the first object in constraint
- **physicsid2** (*int*) – the physics id of the second object in constraint
- **constrainttype** – the type of the constraint. The constraint types are:

- POINTTOPOINT\_CONSTRAINT
- LINEHINGE\_CONSTRAINT
- ANGULAR\_CONSTRAINT
- CONETWIST\_CONSTRAINT
- VEHICLE\_CONSTRAINT

#### Parameters

- **pivotX** (*float*) – pivot X position
- **pivotY** (*float*) – pivot Y position
- **pivotZ** (*float*) – pivot Z position
- **axisX** (*float*) – X axis
- **axisY** (*float*) – Y axis
- **axisZ** (*float*) – Z axis
- **flag** (*int*) –

`bge.constraints.error`

Symbolic constant string that indicates error.

`bge.constraints.exportBulletFile` (*filename*)

export a .bullet file

**Parameters** `filename` (*string*) – File name

`bge.constraints.getAppliedImpulse` (*constraintId*)

**Parameters** `constraintId` (*int*) – The id of the constraint.

**Returns** the most recent applied impulse.

**Return type** float

`bge.constraints.getVehicleConstraint` (*constraintId*)

**Parameters** `constraintId` (*int*) – The id of the vehicle constraint.

**Returns** a vehicle constraint object.

**Return type** `bge.types.KX_VehicleWrapper`

`bge.constraints.removeConstraint` (*constraintId*)

Removes a constraint.

**Parameters** `constraintId` (*int*) – The id of the constraint to be removed.

`bge.constraints.setCcdMode` (*ccdMode*)

---

**Note:** Very experimental, not recommended

---

Sets the CCD (Continous Colision Detection) mode in the Physics Environment.

**Parameters** `ccdMode` (*int*) – The new CCD mode.

`bge.constraints.setContactBreakingTreshold` (*breakingTreshold*)

---

**Note:** Reasonable default is 0.02 (if units are meters)

---

Sets tresholds to do with contact point management.

**Parameters** `breakingTreshold` (*float*) – The new contact breaking treshold.

`bge.constraints.setDeactivationAngularTreshold` (*angularTreshold*)

Sets the angular velocity treshold.

**Parameters** `angularTreshold` (*float*) – New deactivation angular treshold.

`bge.constraints.setDeactivationLinearTreshold` (*linearTreshold*)

Sets the linear velocity treshold.

**Parameters** `linearTreshold` (*float*) – New deactivation linear treshold.

`bge.constraints.setDeactivationTime` (*time*)

Sets the time after which a resting rigidbody gets deactivated.

**Parameters** `time` (*float*) – The deactivation time.

`bge.constraints.setDebugMode` (*mode*)

Sets the debug mode.

**Debug modes:**

- `DBG_NODEDEBUG`
- `DBG_DRAWWIREFRAME`
- `DBG_DRAWAABB`
- `DBG_DRAWFREATURESTEXT`
- `DBG_DRAWCONTACTPOINTS`
- `DBG_NOHELPTTEXT`



- `DBG_DRAWTEXT`
- `DBG_PROFILETIMINGS`
- `DBG_ENABLESATCOMPARISION`
- `DBG_DISABLEBULLETLCP`
- `DBG_ENABLECCD`
- `DBG_DRAWCONSTRAINTS`
- `DBG_DRAWCONSTRAINTLIMITS`
- `DBG_FASTWIREFRAME`

**Parameters** `mode` (*int*) – The new debug mode.

`bge.constraints.setGravity` (*x, y, z*)  
Sets the gravity force.

**Parameters**

- `x` (*float*) – Gravity X force.
- `y` (*float*) – Gravity Y force.
- `z` (*float*) – Gravity Z force.

`bge.constraints.setLinearAirDamping` (*damping*)

---

**Note:** Not implemented.

---

Sets the linear air damping for rigidbodies.

`bge.constraints.setNumIterations` (*numiter*)  
Sets the number of iterations for an iterative constraint solver.

**Parameters** `numiter` (*int*) – New number of iterations.

`bge.constraints.setNumTimeSubSteps` (*numsubstep*)  
Sets the number of substeps for each physics proceed. Tradeoff quality for performance.

**Parameters** `numsubstep` (*int*) – New number of substeps.

`bge.constraints.setSolverDamping` (*damping*)

---

**Note:** Very experimental, not recommended

---

Sets the damper constant of a penalty based solver.

**Parameters** `damping` (*float*) – New damping for the solver.

`bge.constraints.setSolverTau` (*tau*)

---

**Note:** Very experimental, not recommended

---

Sets the spring constant of a penalty based solver.

**Parameters** `tau` (*float*) – New tau for the solver.

`bge.constraints.setSolverType` (*solverType*)

---

**Note:** Very experimental, not recommended

---

Sets the solver type.

**Parameters** `solverType` (*int*) – The new type of the solver.

`bge.constraints.setSorConstant` (*sor*)

---

**Note:** Very experimental, not recommended

---

Sets the successive overrelaxation constant.

**Parameters** `sor` (*float*) – New sor value.

`bge.constraints.setUseEpa` (*epa*)

Not implemented.

`bge.constraints.DBG_NODEBUG`

---

**Note:** Debug mode to be used with function `setDebugMode`

---

No debug.

`bge.constraints.DBG_DRAWWIREFRAME`

---

**Note:** Debug mode to be used with function `setDebugMode`

---

Draw wireframe in debug.

`bge.constraints.DBG_DRAWAABB`

---

**Note:** Debug mode to be used with function `setDebugMode`

---

Draw Axis Aligned Bounding Box in debug.

`bge.constraints.DBG_DRAWFEATURESTEXT`

---

**Note:** Debug mode to be used with function `setDebugMode`

---

Draw features text in debug.

`bge.constraints.DBG_DRAWCONTACTPOINTS`

---

**Note:** Debug mode to be used with function `setDebugMode`

---

---

Draw contact points in debug.

`bge.constraints.DBG_NOHELPTTEXT`

---

**Note:** Debug mode to be used with function `setDebugMode`

---

Debug without help text.

`bge.constraints.DBG_DRAWTEXT`

---

**Note:** Debug mode to be used with function `setDebugMode`

---

Draw text in debug.

`bge.constraints.DBG_PROFILETIMINGS`

---

**Note:** Debug mode to be used with function `setDebugMode`

---

Draw profile timings in debug.

`bge.constraints.DBG_ENABLESATCOMPARISION`

---

**Note:** Debug mode to be used with function `setDebugMode`

---

Enable sat comparision in debug.

`bge.constraints.DBG_DISABLEBULLETLCP`

---

**Note:** Debug mode to be used with function `setDebugMode`

---

Disable Bullet LCP.

`bge.constraints.DBG_ENABLECCD`

---

**Note:** Debug mode to be used with function `setDebugMode`

---

Enable Continous Colision Detection in debug.

`bge.constraints.DBG_DRAWCONSTRAINTS`

---

**Note:** Debug mode to be used with function `setDebugMode`

---

Draw constraints in debug.

`bge.constraints.DBG_DRAWCONSTRAINTLIMITS`

---

**Note:** Debug mode to be used with function `setDebugMode`

---

Draw constraint limits in debug.

`bge.constraints.DBG_FASTWIREFRAME`

---

**Note:** Debug mode to be used with function `setDebugMode`

---

Draw a fast wireframe in debug.

`bge.constraints.POINTTOPOINT_CONSTRAINT`

---

**Note:** Constraint type to be used with function `createConstraint`

---

`bge.constraints.LINEHINGE_CONSTRAINT`

---

**Note:** Constraint type to be used with function `createConstraint`

---

`bge.constraints.ANGULAR_CONSTRAINT`

---

**Note:** Constraint type to be used with function `createConstraint`

---

`bge.constraints.CONETWIST_CONSTRAINT`

---

**Note:** Constraint type to be used with function `createConstraint`

---

`bge.constraints.VEHICLE_CONSTRAINT`

---

**Note:** Constraint type to be used with function `createConstraint`

---

# API INFO

## 5.1 Blender API Change Log

### 5.1.1 2.56 to 2.57

#### **bpy.types.SplineBezierPoints**

##### Function Arguments

- `bpy.types.SplineBezierPoints.friction` (`count`), was (*number*)

#### **bpy.types.RenderSettings**

##### Added

- `bpy.types.RenderSettings.use_stamp_lens`

##### Removed

- `use_backbuf`

#### **bpy.types.ActionPoseMarkers**

##### Added

- `bpy.types.ActionPoseMarkers.active`
- `bpy.types.ActionPoseMarkers.active_index`

#### **bpy.types.SpaceImageEditor**

##### Renamed

- `curves` -> `bpy.types.SpaceImageEditor.curve`

## **bpy.types.Scene**

### **Removed**

- **network\_render**

## **bpy.types.GameObjectSettings**

### **Added**

- `bpy.types.GameObjectSettings.use_material_physics_fh`

### **Removed**

- **use\_material\_physics**

## **bpy.types.SplinePoints**

### **Function Arguments**

- `bpy.types.SplinePoints.use_material_physics` (count), *was (number)*

## **bpy.types.Area**

### **Added**

- `bpy.types.Area.height`
- `bpy.types.Area.width`

## **bpy.types.SolidifyModifier**

### **Added**

- `bpy.types.SolidifyModifier.material_offset`
- `bpy.types.SolidifyModifier.material_offset_rim`

### **Removed**

- **use\_rim\_material**

## **bpy.types.UserPreferencesEdit**

### **Removed**

- **use\_keyframe\_insert\_keyingset**

## **bpy.types.MaterialTextureSlot**

### **Added**

- `bpy.types.MaterialTextureSlot.bump_method`
- `bpy.types.MaterialTextureSlot.bump_objectspace`

### **Removed**

- `use_old_bump`

## **bpy.types.ExplodeModifier**

### **Added**

- `bpy.types.ExplodeModifier.particle_uv`
- `bpy.types.ExplodeModifier.use_edge_cut`

### **Removed**

- `use_edge_split`

## **bpy.types.Node**

### **Added**

- `bpy.types.Node.label`

## **bpy.types.RigidBodyJointConstraint**

### **Added**

- `bpy.types.RigidBodyJointConstraint.limit_angle_max_x`
- `bpy.types.RigidBodyJointConstraint.limit_angle_max_y`
- `bpy.types.RigidBodyJointConstraint.limit_angle_max_z`
- `bpy.types.RigidBodyJointConstraint.limit_angle_min_x`
- `bpy.types.RigidBodyJointConstraint.limit_angle_min_y`
- `bpy.types.RigidBodyJointConstraint.limit_angle_min_z`
- `bpy.types.RigidBodyJointConstraint.limit_max_x`
- `bpy.types.RigidBodyJointConstraint.limit_max_y`
- `bpy.types.RigidBodyJointConstraint.limit_max_z`
- `bpy.types.RigidBodyJointConstraint.limit_min_x`
- `bpy.types.RigidBodyJointConstraint.limit_min_y`

- `bpy.types.RigidBodyJointConstraint.limit_min_z`

#### Removed

- **limit\_cone\_max**
- **limit\_cone\_min**
- **limit\_generic\_max**
- **limit\_generic\_min**

#### `bpy.types.KeyMap`

##### Renamed

- **items** -> `bpy.types.KeyMap.keymap_items`

#### `bpy.types.SpaceNodeEditor`

##### Added

- `bpy.types.SpaceNodeEditor.backdrop_channels`
- `bpy.types.SpaceNodeEditor.backdrop_x`
- `bpy.types.SpaceNodeEditor.backdrop_y`
- `bpy.types.SpaceNodeEditor.backdrop_zoom`
- `bpy.types.SpaceNodeEditor.use_auto_render`

#### `bpy.types.SPHFluidSettings`

##### Added

- `bpy.types.SPHFluidSettings.factor_density`
- `bpy.types.SPHFluidSettings.factor_radius`
- `bpy.types.SPHFluidSettings.factor_repulsion`
- `bpy.types.SPHFluidSettings.factor_rest_length`
- `bpy.types.SPHFluidSettings.factor_stiff_viscosity`
- `bpy.types.SPHFluidSettings.plasticity`
- `bpy.types.SPHFluidSettings.repulsion`
- `bpy.types.SPHFluidSettings.spring_frames`
- `bpy.types.SPHFluidSettings.stiff_viscosity`
- `bpy.types.SPHFluidSettings.use_initial_rest_length`
- `bpy.types.SPHFluidSettings.use_viscoelastic_springs`
- `bpy.types.SPHFluidSettings.yield_ratio`



**Removed**

- **stiffness\_near**
- **viscosity\_beta**

**Renamed**

- **viscosity\_omega** -> `bpy.types.SPHFluidSettings.linear_viscosity`

**bpy.types.ConstraintActuator****Added**

- `bpy.types.ConstraintActuator.direction_axis_pos`
- `bpy.types.ConstraintActuator.fh_force`

**Removed**

- **spring**

**bpy.types.UILayout****Renamed**

- **operator\_enums** -> `bpy.types.UILayout.operator_enum`

**bpy.types.SpaceDopeSheetEditor****Added**

- `bpy.types.SpaceDopeSheetEditor.show_pose_markers`

**bpy.types.ToolSettings****Added**

- `bpy.types.ToolSettings.edge_path_live_unwrap`
- `bpy.types.ToolSettings.proportional_size`
- `bpy.types.ToolSettings.use_keyframe_insert_keyingset`

## **bpy.types.EditBone**

### **Added**

- `bpy.types.EditBone.bbone_x`
- `bpy.types.EditBone.bbone_z`

### **Function Arguments**

- `bpy.types.EditBone.bbone_z` (self, matrix, scale, roll), was (*self, matrix*)

## **bpy.types.ID**

### **Renamed**

- **update** -> `bpy.types.ID.update_tag`

## **bpy.types.SpaceGraphEditor**

### **Added**

- `bpy.types.SpaceGraphEditor.use_fancy_drawing`

## **bpy.types.ParticleSystem**

### **Added**

- `bpy.types.ParticleSystem.child_seed`

## **bpy.types.SpaceTimeline**

### **Removed**

- **use\_play\_3d\_editors**
- **use\_play\_animation\_editors**
- **use\_play\_image\_editors**
- **use\_play\_node\_editors**
- **use\_play\_properties\_editors**
- **use\_play\_sequence\_editors**
- **use\_play\_top\_left\_3d\_editor**

## bpy.types.Mesh

### Added

- `bpy.types.Mesh.validate`

### Renamed

- `show_extra_edge_angle` -> `bpy.types.Mesh.show_extra_face_angle`

### Function Arguments

- `bpy.types.Mesh.show_extra_face_angle` (`self`, `vertices`, `edges`, `faces`), was (`self`, `verts`, `edges`, `faces`)

## bpy.types.EnumProperty

### Added

- `bpy.types.EnumProperty.default_flag`

### Renamed

- `items` -> `bpy.types.EnumProperty.enum_items`

## bpy.types.Screen

### Added

- `bpy.types.Screen.use_play_3d_editors`
- `bpy.types.Screen.use_play_animation_editors`
- `bpy.types.Screen.use_play_image_editors`
- `bpy.types.Screen.use_play_node_editors`
- `bpy.types.Screen.use_play_properties_editors`
- `bpy.types.Screen.use_play_sequence_editors`
- `bpy.types.Screen.use_play_top_left_3d_editor`

## bpy.types.MirrorModifier

### Added

- `bpy.types.MirrorModifier.use_mirror_merge`

## **bpy.types.Operator**

### **Added**

- `bpy.types.Operator.cancel`

## **bpy.types.Brush**

### **Added**

- `bpy.types.Brush.height`
- `bpy.types.Brush.use_fixed_texture`

### **Renamed**

- `imagepaint_tool` -> `bpy.types.Brush.image_tool`
- `use_paint_texture` -> `bpy.types.Brush.use_paint_image`
- `vertexpaint_tool` -> `bpy.types.Brush.vertex_tool`

## **bpy.types.Key**

### **Renamed**

- `keys` -> `bpy.types.Key.key_blocks`

## **bpy.types.CompositorNodeBlur**

### **Added**

- `bpy.types.CompositorNodeBlur.aspect_correction`

## **bpy.types.SpaceTextEditor**

### **Added**

- `bpy.types.SpaceTextEditor.margin_column`
- `bpy.types.SpaceTextEditor.show_margin`

## **bpy.types.GPencilLayer**

### **Added**

- `bpy.types.GPencilLayer.show_x_ray`

**Removed**

- `active`

**bpy.types.MarbleTexture**

**Renamed**

- `noisebasis_2` -> `bpy.types.MarbleTexture.noise_basis_2`

**bpy.types.Particle**

**Removed**

- `is_hair`

**Renamed**

- `keys` -> `bpy.types.Particle.hair_keys`
- `keys` -> `bpy.types.Particle.particle_keys`

**bpy.types.Modifier**

**Added**

- `bpy.types.Modifier.use_apply_on_spline`

**bpy.types.Property**

**Added**

- `bpy.types.Property.is_enum_flag`

**bpy.types.SpaceProperties**

**Added**

- `bpy.types.SpaceProperties.texture_context`

**Removed**

- `show_brush_texture`

## **bpy.types.VertexGroups**

### **Added**

- `bpy.types.VertexGroups.remove`

### **Removed**

- **assign**

## **bpy.types.Material**

### **Added**

- `bpy.types.Material.shadow_only_type`

## **bpy.types.RenderLayer**

### **Function Arguments**

- `bpy.types.RenderLayer.shadow_only_type (filename, x, y), was (filename)`

## **bpy.types.Object**

### **Added**

- `bpy.types.Object.is_modified`

### **Renamed**

- **create\_dupli\_list** -> `bpy.types.Object.dupli_list_create`
- **create\_mesh** -> `bpy.types.Object.to_mesh`
- **free\_dupli\_list** -> `bpy.types.Object.dupli_list_clear`

## **bpy.types.NodeTree**

### **Added**

- `bpy.types.NodeTree.inputs`
- `bpy.types.NodeTree.outputs`

## bpy.types.DopeSheet

### Added

- `bpy.types.DopeSheet.filter_fcurve_name`
- `bpy.types.DopeSheet.show_lattices`
- `bpy.types.DopeSheet.show_only_matching_fcurves`

## bpy.types.ActionFCurves

### Function Arguments

- `bpy.types.ActionFCurves.show_only_matching_fcurves` (`data_path`, `index`, `action_group`), was (`data_path`, `array_index`, `action_group`)

## bpy.types.ShrinkwrapModifier

### Added

- `bpy.types.ShrinkwrapModifier.cull_face`

### Removed

- `use_cull_back_faces`
- `use_cull_front_faces`

## bpy.types.WindowManager

### Added

- `bpy.types.WindowManager.addon_filter`
- `bpy.types.WindowManager.addon_search`
- `bpy.types.WindowManager.addon_support`
- `bpy.types.WindowManager.event_timer_add`
- `bpy.types.WindowManager.event_timer_remove`

## bpy.types.WoodTexture

### Renamed

- `noisebasis_2` -> `bpy.types.WoodTexture.noise_basis_2`

## **bpy.types.VertexGroup**

### **Added**

- `bpy.types.VertexGroup.add`
- `bpy.types.VertexGroup.remove`
- `bpy.types.VertexGroup.weight`

## **bpy.types.FCurveKeyframePoints**

### **Added**

- `bpy.types.FCurveKeyframePoints.insert`

### **Function Arguments**

- `bpy.types.FCurveKeyframePoints.insert` (*count*), *was* (*frame, value, replace, needed, fast*)

## **bpy.types.ThemeView3D**

### **Added**

- `bpy.types.ThemeView3D.outline_width`

## **bpy.types.Image**

### **Added**

- `bpy.types.Image.pixels`

## **bpy.types.Bone**

### **Added**

- `bpy.types.Bone.bbone_x`
- `bpy.types.Bone.bbone_z`

## **bpy.types.InputKeyMapPanel**

### **Removed**

- `draw_entry`
- `draw_filtered`
- `draw_hierarchy`



- **draw\_keymaps**
- **draw\_km**
- **draw\_kmi**
- **draw\_kmi\_properties**
- **indented\_layout**

## **bpy.types.ParticleSettings**

### **Added**

- `bpy.types.ParticleSettings.active_texture`
- `bpy.types.ParticleSettings.active_texture_index`
- `bpy.types.ParticleSettings.child_parting_factor`
- `bpy.types.ParticleSettings.child_parting_max`
- `bpy.types.ParticleSettings.child_parting_min`
- `bpy.types.ParticleSettings.color_maximum`
- `bpy.types.ParticleSettings.create_long_hair_children`
- `bpy.types.ParticleSettings.draw_color`
- `bpy.types.ParticleSettings.effector_amount`
- `bpy.types.ParticleSettings.grid_random`
- `bpy.types.ParticleSettings.hair_length`
- `bpy.types.ParticleSettings.hexagonal_grid`
- `bpy.types.ParticleSettings.is_fluid`
- `bpy.types.ParticleSettings.kink_amplitude_clump`
- `bpy.types.ParticleSettings.kink_flat`
- `bpy.types.ParticleSettings.texture_slots`
- `bpy.types.ParticleSettings.timestep`
- `bpy.types.ParticleSettings.use_advanced_hair`

### **Removed**

- **reaction\_shape**
- **show\_material\_color**
- **use\_animate\_branching**
- **use\_branching**
- **use\_symmetric\_branching**

## **bpy.types.SceneGameData**

### **Added**

- `bpy.types.SceneGameData.show_mouse`

## **bpy.types.MaterialPhysics**

### **Renamed**

- **damping** -> `bpy.types.MaterialPhysics.fh_damping`
- **distance** -> `bpy.types.MaterialPhysics.fh_distance`
- **force** -> `bpy.types.MaterialPhysics.fh_force`
- **use\_normal\_align** -> `bpy.types.MaterialPhysics.use_fh_normal`

## **5.1.2 2.57 to 2.58**

### **bpy\_extras**

#### **Added**

- `bpy_extras`
- `bpy_extras.view3d_utils`

#### **Moved**

- `io_utils` -> `bpy_extras.io_utils`
- `image_utils` -> `bpy_extras.image_utils`
- `mesh_utils` -> `bpy_extras.mesh_utils`
- `object_utils` -> `bpy_extras.object_utils`

## **bpy.types.RenderSettings**

### **Added**

- `bpy.types.RenderSettings.use_bake_lores_mesh`
- `bpy.types.RenderSettings.use_bake_multires`

## **bpy.types.Camera**

### **Added**

- `bpy.types.Camera.show_guide`

**bpy.types.SpaceImageEditor****Added**

- `bpy.types.SpaceImageEditor.zoom`

**bpy.types.SpaceView3D****Added**

- `bpy.types.SpaceView3D.lock_camera`

**bpy.types.RegionView3D****Added**

- `bpy.types.RegionView3D.is_perspective`

**bpy.types.Scene****Added**

- `bpy.types.Scene.frame_subframe`

**bpy.types.Area****Removed**

- `active_space`

**bpy.types.DisplaceModifier****Renamed**

- `texture_coordinate_object` -> `bpy.types.DisplaceModifier.texture_coords_object`

**bpy.types.UserPreferencesView****Added**

- `bpy.types.UserPreferencesView.use_camera_lock_parent`

### **bpy.types.DomainFluidSettings**

#### **Added**

- `bpy.types.DomainFluidSettings.fluid_mesh_vertices`
- `bpy.types.DomainFluidSettings.surface_noobs`

### **bpy.types.Sculpt**

#### **Added**

- `bpy.types.Sculpt.use_deform_only`

### **bpy.types.ClothCollisionSettings**

#### **Added**

- `bpy.types.ClothCollisionSettings.distance_repel`
- `bpy.types.ClothCollisionSettings.repel_force`

### **bpy.types.UILayout**

#### **Added**

- `bpy.types.UILayout.template_edit_mode_selection`

### **bpy.types.ToolSettings**

#### **Added**

- `bpy.types.ToolSettings.use_snap_project_self`

### **bpy.types.Mesh**

#### **Removed**

- `edge_face_count`
- `edge_face_count_dict`
- `edge_loops_from_edges`
- `edge_loops_from_faces`

**bpy.types.PointDensity****Added**

- `bpy.types.PointDensity.falloff_curve`
- `bpy.types.PointDensity.falloff_speed_scale`
- `bpy.types.PointDensity.use_falloff_curve`

**bpy.types.SpaceTextEditor****Added**

- `bpy.types.SpaceTextEditor.use_match_case`

**bpy.types.CameraActuator****Added**

- `bpy.types.CameraActuator.damping`

**bpy.types.Property****Added**

- `bpy.types.Property.is_skip_save`

**bpy.types.UserPreferencesSystem****Added**

- `bpy.types.UserPreferencesSystem.anisotropic_filter`

**bpy.types.Object****Added**

- `bpy.types.Object.empty_image_offset`

**bpy.types.Image****Added**

- `bpy.types.Image.resolution`

## **bpy.types.SceneGameData**

### **Added**

- `bpy.types.SceneGameData.use_gls_l_color_management`

## **5.1.3 2.58 to 2.59**

## **bpy.types.Scene**

### **Function Arguments**

- `bpy.types.Scene.collada_export (filepath, selected)`, was (*filepath*)

## **bpy.types.MultiresModifier**

### **Added**

- `bpy.types.MultiresModifier.use_subsurf_uv`

## **bpy.types.KeyMap**

### **Removed**

- `copy_to_user`

### **Renamed**

- `is_user_defined` -> `bpy.types.KeyMap.is_user_modified`

## **bpy.types.SceneRenderLayer**

### **Added**

- `bpy.types.SceneRenderLayer.use_pass_material_index`

## **bpy.types.ToolSettings**

### **Renamed**

- `use_snap_project_self` -> `bpy.types.ToolSettings.use_snap_self`

## **bpy.types.UserPreferencesInput**

### **Added**

- `bpy.types.UserPreferencesInput.ndof_fly_helicopter`
- `bpy.types.UserPreferencesInput.ndof_lock_horizon`
- `bpy.types.UserPreferencesInput.ndof_orbit_invert_axes`
- `bpy.types.UserPreferencesInput.ndof_sensitivity`
- `bpy.types.UserPreferencesInput.ndof_show_guide`
- `bpy.types.UserPreferencesInput.ndof_zoom_invert`
- `bpy.types.UserPreferencesInput.ndof_zoom_updown`

### **Removed**

- `edited_keymaps`
- `ndof_pan_speed`
- `ndof_rotate_speed`

## **bpy.types.IDMaterials**

### **Function Arguments**

- `bpy.types.IDMaterials.pop` (`index`, `update_data`), *was* (`index`)

## **bpy.types.Material**

### **Added**

- `bpy.types.Material.pass_index`

## **bpy.types.RenderLayer**

### **Added**

- `bpy.types.RenderLayer.use_pass_material_index`

## **bpy.types.Object**

### **Added**

- `bpy.types.Object.closest_point_on_mesh`

### **bpy.types.ThemeNodeEditor**

Added

- `bpy.types.ThemeNodeEditor.noodle_curving`

### **bpy.types.ChildOfConstraint**

Added

- `bpy.types.ChildOfConstraint.inverse_matrix`

### **bpy.types.KeyConfigurations**

Added

- `bpy.types.KeyConfigurations.addon`
- `bpy.types.KeyConfigurations.user`

### **bpy.types.Image**

Added

- `bpy.types.Image.use_generated_float`

### **bpy.types.KeyMapItem**

Added

- `bpy.types.KeyMapItem.is_user_modified`

## **5.1.4 2.59 to 2.60**

### **bpy.types.MeshTextureFace**

Removed

- `use_image`
- `use_object_color`
- `use_blend_shared`



## Moved

- **hide** -> `bpy.types.Material.game_settings.invisible`
- **use\_collision** -> `bpy.types.Material.game_settings.physics`
- **use\_light** -> `bpy.types.Material.game_settings.use_shadeless`
- **use\_twoside** -> `bpy.types.Material.game_settings.backface_culling`
- **use\_bitmap\_text** -> `bpy.types.Material.game_settings.text`
- **blend\_type** -> `bpy.types.Material.game_settings.alpha_blend`
- **use\_alpha\_sort** -> `bpy.types.Material.game_settings.alpha_blend`
- **use\_billboard** -> `bpy.types.Material.game_settings.face_orientation`
- **use\_halo** -> `bpy.types.Material.game_settings.face_orientation`
- **use\_shadow\_cast** -> `bpy.types.Material.game_settings.face_orientation`

## `bpy.types.RenderSettings`

### Added

- `bpy.types.RenderSettings.ffmpeg_audio_channels`

## `bpy.types.DriverTarget`

### Added

- `bpy.types.DriverTarget.transform_space`

### Removed

- **use\_local\_space\_transform**

## `bpy.types.Sound`

### Added

- `bpy.types.Sound.factory`
- `bpy.types.Sound.use_mono`

## `bpy.types.Camera`

### Added

- `bpy.types.Camera.view_frame`

## **bpy.types.Scene**

### **Added**

- `bpy.types.Scene.audio_volume`

## **bpy.types.KeyingSet**

### **Added**

- `bpy.types.KeyingSet.refresh`

## **bpy.types.Armature**

### **Added**

- `bpy.types.Armature.deform_method`

## **bpy.types.GameObjectSettings**

### **Added**

- `bpy.types.GameObjectSettings.obstacle_radius`
- `bpy.types.GameObjectSettings.use_obstacle_create`

## **bpy.types.BlendData**

### **Added**

- `bpy.types.BlendData.speakers`

## **bpy.types.SolidifyModifier**

### **Added**

- `bpy.types.SolidifyModifier.thickness_vertex_group`

## **bpy.types.ThemeGraphEditor**

### **Added**

- `bpy.types.ThemeGraphEditor.handle_auto_clamped`
- `bpy.types.ThemeGraphEditor.handle_sel_auto_clamped`

**bpy.types.CompositorNodeIDMask****Added**

- `bpy.types.CompositorNodeIDMask.use_smooth_mask`

**bpy.types.Node****Added**

- `bpy.types.Node.parent`

**bpy.types.Texture****Added**

- `bpy.types.Texture.evaluate`

**bpy.types.UILayout****Added**

- `bpy.types.UILayout.template_keymap_item_properties`

**bpy.types.ToolSettings****Added**

- `bpy.types.ToolSettings.use_multipaint`

**bpy.types.UserPreferencesInput****Added**

- `bpy.types.UserPreferencesInput.ndof_panx_invert_axis`
- `bpy.types.UserPreferencesInput.ndof_pany_invert_axis`
- `bpy.types.UserPreferencesInput.ndof_panz_invert_axis`
- `bpy.types.UserPreferencesInput.ndof_roll_invert_axis`
- `bpy.types.UserPreferencesInput.ndof_rotate_invert_axis`
- `bpy.types.UserPreferencesInput.ndof_tilt_invert_axis`

### **bpy.types.LockedTrackConstraint**

#### **Added**

- `bpy.types.LockedTrackConstraint.head_tail`

### **bpy.types.SpaceGraphEditor**

#### **Moved**

- `use_fancy_drawing` -> `bpy.types.SpaceGraphEditor.use_beauty_drawing`

### **bpy.types.ParticleSystem**

#### **Added**

- `bpy.types.ParticleSystem.dt_frac`

### **bpy.types.Mesh**

#### **Added**

- `bpy.types.Mesh.use_paint_mask_vertex`

### **bpy.types.FCurve**

#### **Removed**

- `use_auto_handle_clamp`

### **bpy.types.DampedTrackConstraint**

#### **Added**

- `bpy.types.DampedTrackConstraint.head_tail`

### **bpy.types.ImageTexture**

#### **Added**

- `bpy.types.ImageTexture.use_derivative_map`

## **bpy.types.SoundSequence**

### **Added**

- `bpy.types.SoundSequence.pan`
- `bpy.types.SoundSequence.pitch`

### **Removed**

- **attenuation**

## **bpy.types.FModifier**

### **Added**

- `bpy.types.FModifier.blend_in`
- `bpy.types.FModifier.blend_out`
- `bpy.types.FModifier.frame_end`
- `bpy.types.FModifier.frame_start`
- `bpy.types.FModifier.influence`
- `bpy.types.FModifier.use_influence`
- `bpy.types.FModifier.use_restricted_range`

## **bpy.types.EnvironmentMap**

### **Added**

- `bpy.types.EnvironmentMap.clear`
- `bpy.types.EnvironmentMap.is_valid`
- `bpy.types.EnvironmentMap.save`

## **bpy.types.UserPreferencesSystem**

### **Added**

- `bpy.types.UserPreferencesSystem.use_translate_interface`

### **Removed**

- **use\_translate\_buttons**
- **use\_translate\_toolbox**

### **bpy.types.LimitDistanceConstraint**

#### **Added**

- `bpy.types.LimitDistanceConstraint.head_tail`
- `bpy.types.LimitDistanceConstraint.use_transform_limit`

### **bpy.types.MovieSequence**

#### **Added**

- `bpy.types.MovieSequence.stream_index`

### **bpy.types.Material**

#### **Added**

- `bpy.types.Material.game_settings`

### **bpy.types.Object**

#### **Added**

- `bpy.types.Object.matrix_parent_inverse`

### **bpy.types.SequenceProxy**

#### **Added**

- `bpy.types.SequenceProxy.build_100`
- `bpy.types.SequenceProxy.build_25`
- `bpy.types.SequenceProxy.build_50`
- `bpy.types.SequenceProxy.build_75`
- `bpy.types.SequenceProxy.build_free_run`
- `bpy.types.SequenceProxy.build_free_run_rec_date`
- `bpy.types.SequenceProxy.build_record_run`
- `bpy.types.SequenceProxy.quality`
- `bpy.types.SequenceProxy.timecode`

### **bpy.types.Sequence**

#### **Added**

- `bpy.types.Sequence.waveform`

## **bpy.types.DopeSheet**

### **Added**

- `bpy.types.DopeSheet.show_datablock_filters`
- `bpy.types.DopeSheet.show_speakers`

## **bpy.types.ActionActuator**

### **Added**

- `bpy.types.ActionActuator.apply_to_children`
- `bpy.types.ActionActuator.layer`
- `bpy.types.ActionActuator.layer_weight`
- `bpy.types.ActionActuator.use_additive`
- `bpy.types.ActionActuator.use_force`
- `bpy.types.ActionActuator.use_local`

## **bpy.types.VertexGroup**

### **Added**

- `bpy.types.VertexGroup.lock_weight`

## **bpy.types.ThemeView3D**

### **Added**

- `bpy.types.ThemeView3D.speaker`

## **bpy.types.Image**

### **Added**

- `bpy.types.Image.pack`
- `bpy.types.Image.unpack`

## **bpy.types.Curve**

### **Added**

- `bpy.types.Curve.fill_mode`

#### Removed

- `use_fill_back`
- `use_fill_front`

#### `bpy.types.ParticleSettings`

##### Added

- `bpy.types.ParticleSettings.adaptive_subframes`
- `bpy.types.ParticleSettings.courant_target`

#### `bpy.types.SceneGameData`

##### Added

- `bpy.types.SceneGameData.level_height`
- `bpy.types.SceneGameData.obstacle_simulation`
- `bpy.types.SceneGameData.recast_data`
- `bpy.types.SceneGameData.restrict_animation_updates`
- `bpy.types.SceneGameData.show_obstacle_simulation`

---

**Note:** The Blender Python API has areas which are still in development.

#### **The following areas are subject to change.**

- operator behavior, names and arguments
- mesh creation and editing functions

#### **These parts of the API are relatively stable and are unlikely to change significantly**

- data API, access to attributes of blender data such as mesh verts, material color, timeline frames and scene objects
  - user interface functions for defining buttons, creation of menus, headers, panels
  - render engine integration
  - modules: `bgl`, `mathutils` & game engine.
-



# PYTHON MODULE INDEX

## a

aud, ??

## b

bge.constraints, ??

bge.events, ??

bge.logic, ??

bge.render, ??

bge.texture, ??

bge.types, ??

bgl, ??

blf, ??

bpy, ??

bpy.app, ??

bpy.app.handlers, ??

bpy.context, ??

bpy.ops.action, ??

bpy.ops.anim, ??

bpy.ops.armature, ??

bpy.ops.boid, ??

bpy.ops.brush, ??

bpy.ops.buttons, ??

bpy.ops.cloth, ??

bpy.ops.console, ??

bpy.ops.constraint, ??

bpy.ops.curve, ??

bpy.ops.ed, ??

bpy.ops.export\_anim, ??

bpy.ops.export\_mesh, ??

bpy.ops.export\_scene, ??

bpy.ops.file, ??

bpy.ops.fluid, ??

bpy.ops.font, ??

bpy.ops.gpencil, ??

bpy.ops.graph, ??

bpy.ops.group, ??

bpy.ops.image, ??

bpy.ops.import\_anim, ??

bpy.ops.import\_curve, ??

bpy.ops.import\_mesh, ??

bpy.ops.import\_scene, ??

bpy.ops.info, ??

bpy.ops.lamp, ??

bpy.ops.lattice, ??

bpy.ops.logic, ??

bpy.ops.marker, ??

bpy.ops.material, ??

bpy.ops.mball, ??

bpy.ops.mesh, ??

bpy.ops.nla, ??

bpy.ops.node, ??

bpy.ops.object, ??

bpy.ops.outliner, ??

bpy.ops.paint, ??

bpy.ops.particle, ??

bpy.ops.pose, ??

bpy.ops.poselib, ??

bpy.ops.ptcache, ??

bpy.ops.render, ??

bpy.ops.scene, ??

bpy.ops.screen, ??

bpy.ops.script, ??

bpy.ops.sculpt, ??

bpy.ops.sequencer, ??

bpy.ops.sketch, ??

bpy.ops.sound, ??

bpy.ops.surface, ??

bpy.ops.text, ??

bpy.ops.texture, ??

bpy.ops.time, ??

bpy.ops.transform, ??

bpy.ops.ui, ??

bpy.ops.uv, ??

bpy.ops.view2d, ??

bpy.ops.view3d, ??

bpy.ops.wm, ??

bpy.ops.world, ??

bpy.path, ??

bpy.props, ??

bpy.types, ??

bpy.utils, ??

bpy\_extras, ??

bpy\_extras.anim\_utils, ??  
bpy\_extras.image\_utils, ??  
bpy\_extras.io\_utils, ??  
bpy\_extras.keyconfig\_utils, ??  
bpy\_extras.mesh\_utils, ??  
bpy\_extras.object\_utils, ??  
bpy\_extras.view3d\_utils, ??

## g

gpu, ??

## m

mathutils, ??  
mathutils.geometry, ??