

[Skip to content](#)

ID(bpy_struct)

base class — [bpy_struct](#)

subclasses — [Action](#), [Armature](#), [Brush](#), [CacheFile](#), [Camera](#), [Collection](#), [Curve](#), [Curves](#), [FreestyleLineStyle](#), [GreasePencil](#), [GreasePencilv3](#), [Image](#), [Key](#), [Lattice](#), [Library](#), [Light](#), [LightProbe](#), [Mask](#), [Material](#), [Mesh](#), [MetaBall](#), [MovieClip](#), [NodeTree](#), [Object](#), [PaintCurve](#), [Palette](#), [ParticleSettings](#), [PointCloud](#), [Scene](#), [Screen](#), [Sound](#), [Speaker](#), [Text](#), [Texture](#), [VectorFont](#), [Volume](#), [WindowManager](#), [WorkSpace](#), [World](#)

class `bpy.types.ID(bpy_struct)`

Base type for data-blocks, defining a unique name, linking from other libraries and garbage collection

asset_data

Additional data for an asset data-block

TYPE:

[AssetMetaData](#)

id_type

Type identifier of this data-block

TYPE:

enum in [Id Type Items](#), default 'ACTION', (readonly)

is_editable

This data-block is editable in the user interface. Linked datablocks are not editable, except if they were loaded as editable assets.

TYPE:

boolean, default False, (readonly)

is_embedded_data

This data-block is not an independent one, but is actually a sub-data of another ID (typical example: root node trees or master collections)

TYPE:

boolean, default False, (readonly)

is_evaluated

Whether this ID is runtime-only, evaluated data-block, or actual data from .blend file

TYPE:

boolean, default False, (readonly)

is_library_indirect

Is this ID block linked indirectly

TYPE:

boolean, default False, (readonly)

is_missing

This data-block is a place-holder for missing linked data (i.e. it is [an override of] a linked data that could not be found anymore)

TYPE:

boolean, default False, (readonly)

is_runtime_data

This data-block is runtime data, i.e. it won't be saved in .blend file. Note that some evaluated IDs are always runtime, so this value is only edited

This data-block is runtime data, i.e. it won't be saved in .blend file. Note that e.g. evaluated IDs are always runtime, so this value is only equal for data-blocks in Main data-base.

TYPE:

boolean, default False

library

Library file the data-block is linked from

TYPE:

`Library`, (readonly)

library_weak_reference

Weak reference to a data-block in another library .blend file (used to re-use already appended data instead of appending new copies)

TYPE:

`LibraryWeakReference`, (readonly)

name

Unique data-block ID name (within a same type and library)

TYPE:

string, default "", (never None)

name_full

Unique data-block ID name, including library one is any

TYPE:

string, default "", (readonly, never None)

original

Actual data-block from .blend file (Main database) that generated that evaluated one

TYPE:

`ID`, (readonly)

override_library

Library override data

TYPE:

`IDOverrideLibrary`, (readonly)

preview

Preview image and icon of this data-block (always None if not supported for this type of data)

TYPE:

`ImagePreview`, (readonly)

session_uid

A session-wide unique identifier for the data block that remains the same across renames and internal reallocations, unchanged when reloading the file

TYPE:

int in $[-\text{inf}, \text{inf}]$, default 0, (readonly)

tag

Tools can use this to tag data for their own purposes (initial state is undefined)

TYPE:

boolean, default False

use_extra_user

Indicates whether an extra user is set or not (mainly for internal/debug usages)

TYPE:

boolean, default False

use_fake_user

Save this data-block even if it has no users

TYPE:

boolean, default False

users

Number of times this data-block is referenced

TYPE:

int in [0, inf], default 0, (readonly)

rename(name, *, mode='NEVER')

More refined handling in case the new name collides with another ID's name

PARAMETERS:

- **name** (*string, (never None)*) – New name to rename the ID to, if empty will re-use the current ID name
- **mode** (*enum in ['NEVER', 'ALWAYS', 'SAME_ROOT'], (optional)*) – How to handle name collision, in case the requested new name is already used by another ID of the same type
 - **NEVER** Never Rename – Never rename an existing ID whose name would conflict, the currently renamed ID will get a numeric suffix appended to its new name.
 - **ALWAYS** Always Rename – Always rename an existing ID whose name would conflict, ensuring that the currently renamed ID will get the requested name.
 - **SAME_ROOT** Rename If Same Root – Only rename an existing ID whose name would conflict if its name root (everything besides the numerical suffix) is the same as the existing name of the currently renamed ID.

RETURNS:

How did the renaming of the data-block went on

- **UNCHANGED** Unchanged – The ID was not renamed, e.g. because it is already named as requested.
- **UNCHANGED_COLLISION** Unchanged Due to Collision – The ID was not renamed, because requested name would have collided with another existing ID's name, and the automatically adjusted name was the same as the current ID's name.
- **RENAMED_NO_COLLISION** Renamed Without Collision – The ID was renamed as requested, without creating any name collision.
- **RENAMED_COLLISION_ADJUSTED** Renamed With Collision – The ID was renamed with adjustment of the requested name, to avoid a name collision.
- **RENAMED_COLLISION_FORCED** Renamed Enforced With Collision – The ID was renamed as requested, also renaming another ID to avoid a name collision.

RETURN TYPE:

enum in ['UNCHANGED', 'UNCHANGED_COLLISION', 'RENAMED_NO_COLLISION', 'RENAMED_COLLISION_ADJUSTED', 'RENAMED_COLLISION_FORCED']

evaluated_get(depsgraph)

Get corresponding evaluated ID from the given dependency graph. Note that this does not ensure the dependency graph is fully evaluated, it just returns the result of the last evaluation.

PARAMETERS:

depsgraph (*Depsgraph*, (never None)) – Dependency graph to perform lookup in

RETURNS:

New copy of the ID

RETURN TYPE:

ID

copy()

Create a copy of this data-block (not supported for all data-blocks). The result is added to the Blend-File Data (Main database), with all references to other data-blocks ensured to be from within the same Blend-File Data.

RETURNS:

New copy of the ID

RETURN TYPE:

ID

asset_mark()

Enable easier reuse of the data-block through the Asset Browser, with the help of customizable metadata (like previews, descriptions and tags)

asset_clear()

Delete all asset metadata and turn the asset data-block back into a normal data-block

asset_generate_preview()

Generate preview image (might be scheduled in a background thread)

override_create(*, remap_local_usages=False)

Create an overridden local copy of this linked data-block (not supported for all data-blocks)

PARAMETERS:

remap_local_usages (*boolean, (optional)*) – Whether local usages of the linked ID should be remapped to the new library override or not

RETURNS:

New overridden local copy of the ID

RETURN TYPE:

ID

override_hierarchy_create(scene, view_layer, *, reference=None, do_fully_editable=False)

Create an overridden local copy of this linked data-block, and most of its dependencies when it is a Collection or and Object

PARAMETERS:

- **scene** ([Scene](#) , (never None)) – In which scene the new overrides should be instantiated
- **view_layer** ([ViewLayer](#) , (never None)) – In which view layer the new overrides should be instantiated
- **reference** ([ID](#) , (optional)) – Another ID (usually an Object or Collection) used as a hint to decide where to instantiate the new override
- **do_fully_editable** (*boolean, (optional)*) – Make all library overrides generated by this call fully editable by the user (none will be ‘system overrides’)

RETURNS:

New overridden local copy of the root ID

RETURN TYPE:

ID

user_clear()

Clear the user count of a data-block 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)
```

user_remap(new_id)

Replace all usage in the .blend file of this ID by new given one

PARAMETERS:

new_id ([ID](#), (never None)) – New ID to use

make_local(*, clear_proxy=True, clear_liboverride=False, clear_asset_data=True)

Make this datablock local, return local one (may be a copy of the original, in case it is also indirectly used)

PARAMETERS:

- **clear_proxy** (*boolean, (optional)*) – Deprecated, has no effect
- **clear_liboverride** (*boolean, (optional)*) – Remove potential library override data from the newly made local data
- **clear_asset_data** (*boolean, (optional)*) – Remove potential asset metadata so the newly local data-block is not treated as asset data-block and won't show up in asset libraries

RETURNS:

This ID, or the new ID if it was copied

RETURN TYPE:

[ID](#)

user_of_id(id)

Count the number of times that ID uses/references given one

PARAMETERS:

id ([ID](#), (never None)) – ID to count usages

RETURNS:

Number of usages/references of given id by current data-block

RETURN TYPE:

int in [0, inf]

animation_data_create()

Create animation data to this ID, note that not all ID types support this

RETURNS:

New animation data or nullptr

RETURN TYPE:

[AnimData](#)

animation_data_clear()

Clear animation on this ID

--- Data API (Blender 2.80) ---

update_tag(*^*, refresh={})

Tag the ID to update its display data, e.g. when calling `bpy.types.Scene.update`

PARAMETERS:

refresh (*enum set in {'OBJECT', 'DATA', 'TIME'}, (optional)*) – Type of updates to perform

preview_ensure()

Ensure that this ID has preview data (if ID type supports it)

RETURNS:

The existing or created preview

RETURN TYPE:

`ImagePreview`

classmethod bl_rna_get_subclass(*id*, default=None)

PARAMETERS:

id (*str*) – The RNA type identifier.

RETURNS:

The RNA type or default when not found.

RETURN TYPE:

`bpy.types.Struct` subclass

classmethod bl_rna_get_subclass_py(*id*, default=None)

PARAMETERS:

id (*str*) – The RNA type identifier.

RETURNS:

The class or default when not found.

RETURN TYPE:

type

Inherited Properties

- `bpy_struct.id_data`

Inherited Functions

- `bpy_struct.as_pointer`
- `bpy_struct.driver_add`
- `bpy_struct.driver_remove`
- `bpy_struct.get`
- `bpy_struct.id_properties_clear`
- `bpy_struct.id_properties_ensure`
- `bpy_struct.id_properties_ui`
- `bpy_struct.is_property_hidden`
- `bpy_struct.is_property_overridable_library`
- `bpy_struct.is_property_readonly`
- `bpy_struct.is_property_set`
- `bpy_struct.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`
- `bpy_struct.keys`
- `bpy_struct.path_from_id`
- `bpy_struct.path_resolve`
- `bpy_struct.pop`
- `bpy_struct.property_overridable_library_set`
- `bpy_struct.property_unset`
- `bpy_struct.type_recast`
- `bpy_struct.values`

References

- [bpy.context.annotation_data_owner](#)
- [bpy.context.id](#)
- [bpy.context.selected_ids](#)
- [bpy.context.texture_user](#)
- [Action.fcurve_ensure_for_datablock](#)
- [ActionSlot.users](#)
- [AssetRepresentation.local_id](#)
- [BlendDataObjects.new](#)
- [BlendImportContextItem.id](#)
- [BlendImportContextItem.library_override_id](#)
- [BlendImportContextItem.reusable_local_id](#)
- [Depsgraph.id_eval_get](#)
- [Depsgraph.id_eval_get](#)
- [Depsgraph.ids](#)
- [DepsgraphUpdate.id](#)
- [DopeSheet.source](#)
- [DriverTarget.id](#)
- [ID.copy](#)
- [ID.evaluated_get](#)
- [ID.make_local](#)
- [ID.original](#)
- [ID.override_create](#)
- [ID.override_hierarchy_create](#)
- [ID.override_hierarchy_create](#)
- [ID.user_of_id](#)
- [ID.user_remap](#)
- [IDOverrideLibrary.hierarchy_root](#)
- [IDOverrideLibrary.reference](#)
- [IDOverrideLibraryPropertyOperation.subitem_l](#)
- [IDOverrideLibraryPropertyOperation.subitem_r](#)
- [IDOverrideLibraryPropertyOperations.add](#)
- [IDOverrideLibraryPropertyOperations.add](#)
- [IDViewerPathElem.id](#)
- [Key.user](#)
- [KeyingSetPath.id](#)
- [KeyingSetPaths.add](#)
- [MaskParent.id](#)
- [NodeTree.get_from_context](#)
- [NodeTree.get_from_context](#)
- [NodesModifierDataBlock.id](#)
- [Object.data](#)
- [PropertyGroupItem.id](#)
- [SpaceFileBrowser.activate_asset_by_id](#)
- [SpaceNodeEditor.id](#)
- [SpaceNodeEditor.id_from](#)
- [SpaceProperties.pin_id](#)
- [UILayout.template_action](#)
- [UILayout.template_path_builder](#)
- [UILayout.template_preview](#)
- [UILayout.template_preview](#)