# Table of Contents

Skip to content

# ActionChannelbags(bpy_struct)

base class — `bpy_struct`

**class** bpy.types.**ActionChannelbags(bpy_struct)**

For each action slot, a list of animation channels that are meant for that slot

**new(slot)**

Add a new channelbag to the strip, to contain animation channels for a specific slot

**PARAMETERS:**

**slot** ( `ActionSlot` ) – Action Slot, The slot that should be animated by this channelbag

**RETURNS:**

Newly created channelbag

**RETURN TYPE:**

`ActionChannelbag`

**remove(channelbag)**

Remove the channelbag from the strip

**PARAMETERS:**

**channelbag** ( `ActionChannelbag` ) – The channelbag to remove

**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.items`
- `bpy_struct.keyframe_delete`
- `bpy_struct.keyframe_insert`

- 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.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

- ActionKeyframeStrip.channelbags

Previous
ActionChannelbagGroups(bpy_struct)

N
ActionConstraint(Constrai

Report issue on this page

Skip to content

# BlendDataLibraries(bpy_struct)

base class — `bpy_struct`

**class** bpy.types.**BlendDataLibraries(bpy_struct)**

Collection of libraries

**tag(value)**

tag

**PARAMETERS:**

**value** (*boolean*) – Value

**remove(library, *, do_unlink=True, do_id_user=True, do_ui_user=True)**

Remove a library from the current blendfile

**PARAMETERS:**

- **library** (`Library`, (never None)) – Library to remove
- **do_unlink** (*boolean, (optional)*) – Unlink all usages of this library before deleting it
- **do_id_user** (*boolean, (optional)*) – Decrement user counter of all datablocks used by this library
- **do_ui_user** (*boolean, (optional)*) – Make sure interface does not reference this library

**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

**load(filepath, link=False, relative=False, assets_only=False, create_liboverrides=False, reuse_liboverrides=False, create_liboverrides_runtime=False)**

Returns a context manager which exposes 2 library objects on entering. Each object has attributes matching bpy.data which are lists of strings be linked.

**PARAMETERS:**

- **filepath** (*str* | *bytes*) – 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.
- **assets_only** (*bool*) – If True, only list data-blocks marked as assets.
- **create_liboverrides** (*bool*) – If True and `link` is True, liboverrides will be created for linked data.
- **reuse_liboverrides** (*bool*) – If True and `create_liboverride` is True, search for existing liboverride first.
- **create_liboverrides_runtime** (*bool*) – If True and `create liboverride` is True, create (or search for existing) runtime

liboverride.

```
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)
```

**write(filepath, datablocks, path_remap=False, fake_user=False, compress=False)**

Write data-blocks into a blend file.

> Note
>
> Indirectly referenced data-blocks will be expanded and written too.

**PARAMETERS:**

- **filepath** (*str* | *bytes*) – The path to write the blend-file.
- **datablocks** (set[`bpy.types.ID`]) – set of data-blocks.
- **path_remap** (*str*) –

  Optionally remap paths when writing the file:

  - `NONE` No path manipulation (default).
  - `RELATIVE` Remap paths that are already relative to the new location.
  - `RELATIVE_ALL` Remap all paths to be relative to the new location.
  - `ABSOLUTE` Make all paths absolute on writing.

- **fake_user** (*bool*) – When True, data-blocks will be written with fake-user flag enabled.
- **compress** (*bool*) – When True, write a compressed blend file.

```python
import bpy

filepath = "//new_library.blend"

# write selected objects and their data to a blend file
data_blocks = set(bpy.context.selected_objects)
bpy.data.libraries.write(filepath, data_blocks)


# write all meshes starting with a capital letter and
# set them with fake-user enabled so they aren't lost on re-saving
data_blocks = {mesh for mesh in bpy.data.meshes if mesh.name[:1].isupper()}
bpy.data.libraries.write(filepath, data_blocks, fake_user=True)


# write all materials, textures and node groups to a library
data_blocks = {*bpy.data.materials, *bpy.data.textures, *bpy.data.node_groups}
bpy.data.libraries.write(filepath, data_blocks)
```

## 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

- BlendData.libraries

Skip to content

# BlendDataLights(bpy_struct)

base class — `bpy_struct`

**class** bpy.types.**BlendDataLights(bpy_struct)**

Collection of lights

**new(name, type)**

Add a new light to the main database

**PARAMETERS:**

- **name** (*string, (never None)*) – New name for the data-block
- **type** (enum in Light Type Items) – Type, The type of light to add

**RETURNS:**

New light data-block

**RETURN TYPE:**

`Light`

**remove(light, *, do_unlink=True, do_id_user=True, do_ui_user=True)**

Remove a light from the current blendfile

**PARAMETERS:**

- **light** ( `Light` , (never None)) – Light to remove
- **do_unlink** (*boolean, (optional)*) – Unlink all usages of this light before deleting it (WARNING: will also delete objects instancing that light data)
- **do_id_user** (*boolean, (optional)*) – Decrement user counter of all datablocks used by this light data
- **do_ui_user** (*boolean, (optional)*) – Make sure interface does not reference this light data

**tag(value)**

tag

**PARAMETERS:**

**value** (*boolean*) – Value

**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

- `BlendData.lights`

Previous
BlendDataLibraries(bpy_struct)

Report issue on this page

N
BlendDataLineStyles(bpy_stru

Skip to content

# BlendDataLineStyles(bpy_struct)

base class — `bpy_struct`

**class** bpy.types.**BlendDataLineStyles(bpy_struct)**

Collection of line styles

**tag(value)**

tag

**PARAMETERS:**

**value** (*boolean*) – Value

**new(name)**

Add a new line style instance to the main database

**PARAMETERS:**

**name** (*string, (never None)*) – New name for the data-block

**RETURNS:**

New line style data-block

**RETURN TYPE:**

`FreestyleLineStyle`

**remove(linestyle, *, do_unlink=True, do_id_user=True, do_ui_user=True)**

Remove a line style instance from the current blendfile

**PARAMETERS:**

- **linestyle** ( `FreestyleLineStyle` , (never None)) – Line style to remove
- **do_unlink** (*boolean, (optional)*) – Unlink all usages of this line style before deleting it
- **do_id_user** (*boolean, (optional)*) – Decrement user counter of all datablocks used by this line style
- **do_ui_user** (*boolean, (optional)*) – Make sure interface does not reference this line style

**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

- BlendData.linestyles

Previous
BlendDataLights(bpy_struct)

Report issue on this page

N
BlendDataMasks(bpy_stru

Skip to content

# BlendDataMasks(bpy_struct)

base class — `bpy_struct`

**class** bpy.types.**BlendDataMasks(bpy_struct)**

Collection of masks

**tag(value)**

tag

**PARAMETERS:**

**value** (*boolean*) – Value

**new(name)**

Add a new mask with a given name to the main database

**PARAMETERS:**

**name** (*string, (never None)*) – Mask, Name of new mask data-block

**RETURNS:**

New mask data-block

**RETURN TYPE:**

`Mask`

**remove(mask, *, do_unlink=True, do_id_user=True, do_ui_user=True)**

Remove a mask from the current blendfile

**PARAMETERS:**

- **mask** (`Mask`, (never None)) – Mask to remove
- **do_unlink** (*boolean, (optional)*) – Unlink all usages of this mask before deleting it
- **do_id_user** (*boolean, (optional)*) – Decrement user counter of all datablocks used by this mask
- **do_ui_user** (*boolean, (optional)*) – Make sure interface does not reference this mask

**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

- BlendData.masks

Previous
BlendDataLineStyles(bpy_struct)

N
BlendDataMaterials(bpy_stru

Report issue on this page

Skip to content

# BlendDataMaterials(bpy_struct)

base class — `bpy_struct`

**class** bpy.types.**BlendDataMaterials(bpy_struct)**

Collection of materials

**new(name)**

Add a new material to the main database

**PARAMETERS:**

**name** (*string, (never None)*) – New name for the data-block

**RETURNS:**

New material data-block

**RETURN TYPE:**

`Material`

**create_gpencil_data(material)**

Add Grease Pencil material settings

**PARAMETERS:**

**material** (`Material`, (never None)) – Material

**remove_gpencil_data(material)**

Remove Grease Pencil material settings

**PARAMETERS:**

**material** (`Material`, (never None)) – Material

**remove(material, *, do_unlink=True, do_id_user=True, do_ui_user=True)**

Remove a material from the current blendfile

**PARAMETERS:**

- **material** (`Material`, (never None)) – Material to remove
- **do_unlink** (*boolean, (optional)*) – Unlink all usages of this material before deleting it
- **do_id_user** (*boolean, (optional)*) – Decrement user counter of all datablocks used by this material
- **do_ui_user** (*boolean, (optional)*) – Make sure interface does not reference this material

**tag(value)**

tag

**PARAMETERS:**

**value** (*boolean*) – Value

**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)**

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

- BlendData.materials

Skip to content

# BlendDataMeshes(bpy_struct)

base class — `bpy_struct`

**class** bpy.types.**BlendDataMeshes(bpy_struct)**

Collection of meshes

**new(name)**

Add a new mesh to the main database

**PARAMETERS:**

**name** (*string, (never None)*) – New name for the data-block

**RETURNS:**

New mesh data-block

**RETURN TYPE:**

`Mesh`

**new_from_object(object, *, preserve_all_data_layers=False, depsgraph=None)**

Add a new mesh created from given object (undeformed geometry if object is original, and final evaluated geometry, with all modifiers etc., if object is evaluated)

**PARAMETERS:**

- **object** ( `Object` , (never None)) – Object to create mesh from
- **preserve_all_data_layers** (*boolean, (optional)*) – Preserve all data layers in the mesh, like UV maps and vertex groups. By default Blender only computes the subset of data layers needed for viewport display and rendering, for better performance.
- **depsgraph** ( `Depsgraph` , (optional)) – Dependency Graph, Evaluated dependency graph which is required when preserve_all_data_layers is true

**RETURNS:**

Mesh created from object, remove it if it is only used for export

**RETURN TYPE:**

`Mesh`

**remove(mesh, *, do_unlink=True, do_id_user=True, do_ui_user=True)**

Remove a mesh from the current blendfile

**PARAMETERS:**

- **mesh** ( `Mesh` , (never None)) – Mesh to remove
- **do_unlink** (*boolean, (optional)*) – Unlink all usages of this mesh before deleting it (WARNING: will also delete objects instancing that mesh data)
- **do_id_user** (*boolean, (optional)*) – Decrement user counter of all datablocks used by this mesh data
- **do_ui_user** (*boolean, (optional)*) – Make sure interface does not reference this mesh data

**tag(value)**

tag

**PARAMETERS:**

**value** (*boolean*) – Value

**classmethod bl_rna_get_subclass(id, default=None)**

**PARAMETERS:**

**id** (*str*) – The RNA type identifier.

**RETURNS:**

**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

- `BlendData.meshes`

Skip to content

# BlendDataMetaBalls(bpy_struct)

base class — `bpy_struct`

**class** bpy.types.**BlendDataMetaBalls(bpy_struct)**

Collection of metaballs

**new(name)**

Add a new metaball to the main database

**PARAMETERS:**

**name** (*string, (never None)*) – New name for the data-block

**RETURNS:**

New metaball data-block

**RETURN TYPE:**

`MetaBall`

**remove(metaball, *, do_unlink=True, do_id_user=True, do_ui_user=True)**

Remove a metaball from the current blendfile

**PARAMETERS:**

- **metaball** (`MetaBall`, (never None)) – Metaball to remove
- **do_unlink** (*boolean, (optional)*) – Unlink all usages of this metaball before deleting it (WARNING: will also delete objects instancing tl metaball data)
- **do_id_user** (*boolean, (optional)*) – Decrement user counter of all datablocks used by this metaball data
- **do_ui_user** (*boolean, (optional)*) – Make sure interface does not reference this metaball data

**tag(value)**

tag

**PARAMETERS:**

**value** (*boolean*) – Value

**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

- BlendData.metaballs

Previous
BlendDataMeshes(bpy_struct)

N
BlendDataMovieClips(bpy_stru

Report issue on this page

Skip to content

# BlendDataMovieClips(bpy_struct)

base class — `bpy_struct`

**class** bpy.types.**BlendDataMovieClips(bpy_struct)**

Collection of movie clips

**tag(value)**

tag

**PARAMETERS:**

**value** (*boolean*) – Value

**remove(clip, *, do_unlink=True, do_id_user=True, do_ui_user=True)**

Remove a movie clip from the current blendfile.

**PARAMETERS:**

- **clip** (`MovieClip`, (never None)) – Movie clip to remove
- **do_unlink** (*boolean, (optional)*) – Unlink all usages of this movie clip before deleting it
- **do_id_user** (*boolean, (optional)*) – Decrement user counter of all datablocks used by this movie clip
- **do_ui_user** (*boolean, (optional)*) – Make sure interface does not reference this movie clip

**load(filepath, *, check_existing=False)**

Add a new movie clip to the main database from a file (while `check_existing` is disabled for consistency with other load functions, behavior with multiple movie-clips using the same file may incorrectly generate proxies)

**PARAMETERS:**

- **filepath** (*string, (never None)*) – path for the data-block
- **check_existing** (*boolean, (optional)*) – Using existing data-block if this file is already loaded

**RETURNS:**

New movie clip data-block

**RETURN TYPE:**

`MovieClip`

**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

- BlendData.movieclips

Previous
BlendDataMetaBalls(bpy_struct)

N
BlendDataNodeTrees(bpy_stru

Made with Furo

Report issue on this page

Skip to content

# BlendDataNodeTrees(bpy_struct)

base class — `bpy_struct`

**class** bpy.types.**BlendDataNodeTrees(bpy_struct)**

Collection of node trees

**new(name, type)**

Add a new node tree to the main database

**PARAMETERS:**

- **name** (*string, (never None)*) – New name for the data-block
- **type** (*enum in ['DUMMY']*) – Type, The type of node_group to add

**RETURNS:**

New node tree data-block

**RETURN TYPE:**

`NodeTree`

**remove(tree, *, do_unlink=True, do_id_user=True, do_ui_user=True)**

Remove a node tree from the current blendfile

**PARAMETERS:**

- **tree** (`NodeTree`, (never None)) – Node tree to remove
- **do_unlink** (*boolean, (optional)*) – Unlink all usages of this node tree before deleting it
- **do_id_user** (*boolean, (optional)*) – Decrement user counter of all datablocks used by this node tree
- **do_ui_user** (*boolean, (optional)*) – Make sure interface does not reference this node tree

**tag(value)**

tag

**PARAMETERS:**

**value** (*boolean*) – Value

**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

- BlendData.node_groups

Previous
BlendDataMovieClips(bpy_struct)
Report issue on this page

Ne
BlendDataObjects(bpy_stru

Skip to content

# BlendDataObjects(bpy_struct)

base class — `bpy_struct`

**class** bpy.types.**BlendDataObjects(bpy_struct)**

Collection of objects

**new(name, object_data)**

Add a new object to the main database

**PARAMETERS:**

- **name** (*string, (never None)*) – New name for the data-block
- **object_data** ( `ID` ) – Object data or None for an empty object

**RETURNS:**

New object data-block

**RETURN TYPE:**

`Object`

**remove(object, *, do_unlink=True, do_id_user=True, do_ui_user=True)**

Remove an object from the current blendfile

**PARAMETERS:**

- **object** ( `Object` , (never None)) – Object to remove
- **do_unlink** (*boolean, (optional)*) – Unlink all usages of this object before deleting it
- **do_id_user** (*boolean, (optional)*) – Decrement user counter of all datablocks used by this object
- **do_ui_user** (*boolean, (optional)*) – Make sure interface does not reference this object

**tag(value)**

tag

**PARAMETERS:**

**value** (*boolean*) – Value

**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

- BlendData.objects

Previous
BlendDataNodeTrees(bpy_struct)

Ne
BlendDataPaintCurves(bpy_stru

Report issue on this page

Skip to content

# BlendDataPaintCurves(bpy_struct)

base class — `bpy_struct`

**class** bpy.types.**BlendDataPaintCurves(bpy_struct)**

Collection of paint curves

**tag(value)**

tag

**PARAMETERS:**

**value** (*boolean*) – Value

**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

- `BlendData.paint_curves`

Skip to content

# BlendDataPalettes(bpy_struct)

base class — `bpy_struct`

**class** bpy.types.**BlendDataPalettes(bpy_struct)**

Collection of palettes

**new(name)**

Add a new palette to the main database

**PARAMETERS:**

**name** (*string, (never None)*) – New name for the data-block

**RETURNS:**

New palette data-block

**RETURN TYPE:**

`Palette`

**remove(palette, *, do_unlink=True, do_id_user=True, do_ui_user=True)**

Remove a palette from the current blendfile

**PARAMETERS:**

- **palette** ( `Palette` , (never None)) – Palette to remove
- **do_unlink** (*boolean, (optional)*) – Unlink all usages of this palette before deleting it
- **do_id_user** (*boolean, (optional)*) – Decrement user counter of all datablocks used by this palette
- **do_ui_user** (*boolean, (optional)*) – Make sure interface does not reference this palette

**tag(value)**

tag

**PARAMETERS:**

**value** (*boolean*) – Value

**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

- BlendData.palettes

Previous
BlendDataPaintCurves(bpy_struct)

Report issue on this page

Ne
BlendDataParticles(bpy_stru

Skip to content

# BlendDataParticles(bpy_struct)

base class — `bpy_struct`

**class** bpy.types.**BlendDataParticles(bpy_struct)**

Collection of particle settings

**new(name)**

Add a new particle settings instance to the main database

**PARAMETERS:**

**name** (*string, (never None)*) – New name for the data-block

**RETURNS:**

New particle settings data-block

**RETURN TYPE:**

`ParticleSettings`

**remove(particle, *, do_unlink=True, do_id_user=True, do_ui_user=True)**

Remove a particle settings instance from the current blendfile

**PARAMETERS:**

- **particle** ( `ParticleSettings` , (never None)) – Particle Settings to remove
- **do_unlink** (*boolean, (optional)*) – Unlink all usages of those particle settings before deleting them
- **do_id_user** (*boolean, (optional)*) – Decrement user counter of all datablocks used by this particle settings
- **do_ui_user** (*boolean, (optional)*) – Make sure interface does not reference this particle settings

**tag(value)**

tag

**PARAMETERS:**

**value** (*boolean*) – Value

**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

- BlendData.particles

Report issue on this page

Skip to content

# BlendDataPointClouds(bpy_struct)

base class — `bpy_struct`

**class** bpy.types.**BlendDataPointClouds(bpy_struct)**

Collection of point clouds

**new(name)**

Add a new point cloud to the main database

**PARAMETERS:**

**name** (*string, (never None)*) – New name for the data-block

**RETURNS:**

New point cloud data-block

**RETURN TYPE:**

`PointCloud`

**remove(pointcloud, *, do_unlink=True, do_id_user=True, do_ui_user=True)**

Remove a point cloud from the current blendfile

**PARAMETERS:**

- **pointcloud** ( `PointCloud` , (never None)) – Point cloud to remove
- **do_unlink** (*boolean, (optional)*) – Unlink all usages of this point cloud before deleting it (WARNING: will also delete objects instancin that point cloud data)
- **do_id_user** (*boolean, (optional)*) – Decrement user counter of all datablocks used by this point cloud data
- **do_ui_user** (*boolean, (optional)*) – Make sure interface does not reference this point cloud data

**tag(value)**

tag

**PARAMETERS:**

**value** (*boolean*) – Value

**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

- BlendData.pointclouds

Skip to content

# BlendDataProbes(bpy_struct)

base class — `bpy_struct`

**class** bpy.types.**BlendDataProbes(bpy_struct)**

Collection of light probes

**new(name, type)**

Add a new light probe to the main database

**PARAMETERS:**

- **name** (*string, (never None)*) – New name for the data-block
- **type** (enum in Lightprobes Type Items) – Type, The type of light probe to add

**RETURNS:**

New light probe data-block

**RETURN TYPE:**

`LightProbe`

**remove(lightprobe, *, do_unlink=True, do_id_user=True, do_ui_user=True)**

Remove a light probe from the current blendfile

**PARAMETERS:**

- **lightprobe** (`LightProbe`, (never None)) – Light probe to remove
- **do_unlink** (*boolean, (optional)*) – Unlink all usages of this light probe before deleting it (WARNING: will also delete objects instancing that light probe data)
- **do_id_user** (*boolean, (optional)*) – Decrement user counter of all datablocks used by this light probe
- **do_ui_user** (*boolean, (optional)*) – Make sure interface does not reference this light probe

**tag(value)**

tag

**PARAMETERS:**

**value** (*boolean*) – Value

**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

- BlendData.lightprobes

Skip to content

# BlendDataScenes(bpy_struct)

base class — `bpy_struct`

**class** bpy.types.**BlendDataScenes(bpy_struct)**

Collection of scenes

**new(name)**

Add a new scene to the main database

**PARAMETERS:**

**name** (*string, (never None)*) – New name for the data-block

**RETURNS:**

New scene data-block

**RETURN TYPE:**

`Scene`

**remove(scene, *, do_unlink=True)**

Remove a scene from the current blendfile

**PARAMETERS:**

- **scene** ( `Scene` , (never None)) – Scene to remove
- **do_unlink** (*boolean, (optional)*) – Unlink all usages of this scene before deleting it

**tag(value)**

tag

**PARAMETERS:**

**value** (*boolean*) – Value

**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

- BlendData.scenes

Previous
BlendDataProbes(bpy_struct)

N
BlendDataScreens(bpy_stru

Report issue on this page

# BlendDataScreens(bpy_struct)

base class — `bpy_struct`

**class** bpy.types.**BlendDataScreens(bpy_struct)**

Collection of screens

**tag(value)**

tag

**PARAMETERS:**

**value** (*boolean*) – Value

**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

- `BlendData.screens`

Skip to content

# BlendDataSounds(bpy_struct)

base class — `bpy_struct`

**class** bpy.types.**BlendDataSounds(bpy_struct)**

Collection of sounds

**load(filepath, *, check_existing=False)**

Add a new sound to the main database from a file

**PARAMETERS:**

- **filepath** (*string, (never None)*) – path for the data-block
- **check_existing** (*boolean, (optional)*) – Using existing data-block if this file is already loaded

**RETURNS:**

New text data-block

**RETURN TYPE:**

`Sound`

**remove(sound, *, do_unlink=True, do_id_user=True, do_ui_user=True)**

Remove a sound from the current blendfile

**PARAMETERS:**

- **sound** ( `Sound` , (never None)) – Sound to remove
- **do_unlink** (*boolean, (optional)*) – Unlink all usages of this sound before deleting it
- **do_id_user** (*boolean, (optional)*) – Decrement user counter of all datablocks used by this sound
- **do_ui_user** (*boolean, (optional)*) – Make sure interface does not reference this sound

**tag(value)**

tag

**PARAMETERS:**

**value** (*boolean*) – Value

**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

- BlendData.sounds

Skip to content

# BlendDataSpeakers(bpy_struct)

base class — `bpy_struct`

**class** bpy.types.**BlendDataSpeakers(bpy_struct)**

Collection of speakers

**new(name)**

Add a new speaker to the main database

**PARAMETERS:**

**name** (*string, (never None)*) – New name for the data-block

**RETURNS:**

New speaker data-block

**RETURN TYPE:**

`Speaker`

**remove(speaker, *, do_unlink=True, do_id_user=True, do_ui_user=True)**

Remove a speaker from the current blendfile

**PARAMETERS:**

- **speaker** ( `Speaker` , (never None)) – Speaker to remove
- **do_unlink** (*boolean, (optional)*) – Unlink all usages of this speaker before deleting it (WARNING: will also delete objects instancing tl speaker data)
- **do_id_user** (*boolean, (optional)*) – Decrement user counter of all datablocks used by this speaker data
- **do_ui_user** (*boolean, (optional)*) – Make sure interface does not reference this speaker data

**tag(value)**

tag

**PARAMETERS:**

**value** (*boolean*) – Value

**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

- BlendData.speakers

Skip to content

# BlendDataTexts(bpy_struct)

base class — `bpy_struct`

**class** bpy.types.**BlendDataTexts(bpy_struct)**

Collection of texts

**new(name)**

Add a new text to the main database

**PARAMETERS:**

**name** (*string, (never None)*) – New name for the data-block

**RETURNS:**

New text data-block

**RETURN TYPE:**

`Text`

**remove(text, *, do_unlink=True, do_id_user=True, do_ui_user=True)**

Remove a text from the current blendfile

**PARAMETERS:**

- **text** ( `Text` , (never None)) – Text to remove
- **do_unlink** (*boolean, (optional)*) – Unlink all usages of this text before deleting it
- **do_id_user** (*boolean, (optional)*) – Decrement user counter of all datablocks used by this text
- **do_ui_user** (*boolean, (optional)*) – Make sure interface does not reference this text

**load(filepath, *, internal=False)**

Add a new text to the main database from a file

**PARAMETERS:**

- **filepath** (*string, (never None)*) – path for the data-block
- **internal** (*boolean, (optional)*) – Make internal, Make text file internal after loading

**RETURNS:**

New text data-block

**RETURN TYPE:**

`Text`

**tag(value)**

tag

**PARAMETERS:**

**value** (*boolean*) – Value

**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

- `BlendData.texts`

Previous
BlendDataSpeakers(bpy_struct)

N
BlendDataTextures(bpy_stru

Report issue on this page

Skip to content

# BlendDataTextures(bpy_struct)

base class — `bpy_struct`

**class** bpy.types.**BlendDataTextures(bpy_struct)**

Collection of textures

**new(name, type)**

Add a new texture to the main database

**PARAMETERS:**

- **name** (*string, (never None)*) – New name for the data-block
- **type** (enum in Texture Type Items) – Type, The type of texture to add

**RETURNS:**

New texture data-block

**RETURN TYPE:**

`Texture`

**remove(texture, *, do_unlink=True, do_id_user=True, do_ui_user=True)**

Remove a texture from the current blendfile

**PARAMETERS:**

- **texture** ( `Texture` , (never None)) – Texture to remove
- **do_unlink** (*boolean, (optional)*) – Unlink all usages of this texture before deleting it
- **do_id_user** (*boolean, (optional)*) – Decrement user counter of all datablocks used by this texture
- **do_ui_user** (*boolean, (optional)*) – Make sure interface does not reference this texture

**tag(value)**

tag

**PARAMETERS:**

**value** (*boolean*) – Value

**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

- BlendData.textures

Previous
BlendDataTexts(bpy_struct)

N
BlendDataVolumes(bpy_stru

Report issue on this page

Skip to content

# BlendDataVolumes(bpy_struct)

base class — `bpy_struct`

**class** bpy.types.**BlendDataVolumes(bpy_struct)**

Collection of volumes

**new(name)**

Add a new volume to the main database

**PARAMETERS:**

**name** (*string, (never None)*) – New name for the data-block

**RETURNS:**

New volume data-block

**RETURN TYPE:**

`Volume`

**remove(volume, *, do_unlink=True, do_id_user=True, do_ui_user=True)**

Remove a volume from the current blendfile

**PARAMETERS:**

- **volume** ( `Volume` , (never None)) – Volume to remove
- **do_unlink** (*boolean, (optional)*) – Unlink all usages of this volume before deleting it (WARNING: will also delete objects instancing tha volume data)
- **do_id_user** (*boolean, (optional)*) – Decrement user counter of all datablocks used by this volume data
- **do_ui_user** (*boolean, (optional)*) – Make sure interface does not reference this volume data

**tag(value)**

tag

**PARAMETERS:**

**value** (*boolean*) – Value

**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

- BlendData.volumes

Previous
BlendDataTextures(bpy_struct)

N
BlendDataWindowManagers(bpy_stru

Report issue on this page

Skip to content

# BlendDataWindowManagers(bpy_struct)

base class — `bpy_struct`

**class** bpy.types.**BlendDataWindowManagers(bpy_struct)**

Collection of window managers

**tag(value)**

tag

**PARAMETERS:**

**value** (*boolean*) – Value

**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

- `BlendData.window_managers`

Skip to content

# BlendDataWorkSpaces(bpy_struct)

base class — `bpy_struct`

**class** bpy.types.**BlendDataWorkSpaces(bpy_struct)**

Collection of workspaces

**tag(value)**

tag

**PARAMETERS:**

**value** (*boolean*) – Value

**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

- `BlendData.workspaces`

Made with Furo

# BlendDataWorlds(bpy_struct)

base class — `bpy_struct`

**class** bpy.types.**BlendDataWorlds(bpy_struct)**

Collection of worlds

**new(name)**

Add a new world to the main database

**PARAMETERS:**

**name** (*string, (never None)*) – New name for the data-block

**RETURNS:**

New world data-block

**RETURN TYPE:**

`World`

**remove(world, *, do_unlink=True, do_id_user=True, do_ui_user=True)**

Remove a world from the current blendfile

**PARAMETERS:**

- **world** (`World`, (never None)) – World to remove
- **do_unlink** (*boolean, (optional)*) – Unlink all usages of this world before deleting it
- **do_id_user** (*boolean, (optional)*) – Decrement user counter of all datablocks used by this world
- **do_ui_user** (*boolean, (optional)*) – Make sure interface does not reference this world

**tag(value)**

tag

**PARAMETERS:**

**value** (*boolean*) – Value

**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

- BlendData.worlds

Previous
BlendDataWorkSpaces(bpy_struct)

N
BlendImportContext(bpy_stru

Report issue on this page

Skip to content

# 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)

**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`

Previous

N
BoidRule(bpy_stru

Skip to content

# BlendImportContext(bpy_struct)

base class — `bpy_struct`

**class** bpy.types.**BlendImportContext(bpy_struct)**

Contextual data for a blendfile library/linked-data related operation. Currently only exposed as read-only data for the pre/post blendimport handler

**import_items**

**TYPE:**

`BlendImportContextItems` `bpy_prop_collection` of `BlendImportContextItem`, (readonly)

**options**

Options for this blendfile import operation

- `LINK` Only link data, instead of appending it.
- `MAKE_PATHS_RELATIVE` Make paths of used library blendfiles relative to current blendfile.
- `USE_PLACEHOLDERS` Generate a placeholder (empty ID) if not found in any library files.
- `FORCE_INDIRECT` Force loaded ID to be tagged as indirectly linked (used in reload context only).
- `APPEND_SET_FAKEUSER` Set fake user on appended IDs.
- `APPEND_RECURSIVE` Append (make local) also indirect dependencies of appended IDs coming from other libraries. NOTE: All IDs (including indirectly linked ones) coming from the same initial library are always made local.
- `APPEND_LOCAL_ID_REUSE` Try to re-use previously appended matching IDs when appending them again, instead of creating local duplicates.
- `APPEND_ASSET_DATA_CLEAR` Clear the asset data on append (it is always kept for linked data).
- `SELECT_OBJECTS` Automatically select imported objects.
- `USE_ACTIVE_COLLECTION` Use the active Collection of the current View Layer to instantiate imported collections and objects.
- `OBDATA_INSTANCE` Instantiate object data IDs (i.e. create objects for them if needed).
- `COLLECTION_INSTANCE` Instantiate collections as empties, instead of linking them into the current view layer.

**TYPE:**

enum set in {‘LINK’, ‘MAKE_PATHS_RELATIVE’, ‘USE_PLACEHOLDERS’, ‘FORCE_INDIRECT’, ‘APPEND_SET_FAKEUSER’, ‘APPEND_RECURSIVE’, ‘APPEND_LOCAL_ID_REUSE’, ‘APPEND_ASSET_DATA_CLEAI ‘SELECT_OBJECTS’, ‘USE_ACTIVE_COLLECTION’, ‘OBDATA_INSTANCE’, ‘COLLECTION_INSTANCE’}, default {‘LINK’}, (readonly)

**process_stage**

Current stage of the import process

- `INIT` Blendfile import context has been initialized and filled with a list of items to import, no data has been linked or appended yet.
- `DONE` All data has been imported and is available in the list of``import_items``.

**TYPE:**

enum in [‘INIT’, ‘DONE’], default ‘INIT’, (readonly)

**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`

Skip to content

# BlendImportContextItem(bpy_struct)

base class — `bpy_struct`

**class** bpy.types.**BlendImportContextItem(bpy_struct)**

An item (representing a data-block) in a BlendImportContext data. Currently only exposed as read-only data for the pre/post linking handlers

**append_action**

How this item has been handled by the append operation. Only set if the data has been appended

- `UNSET` Not yet defined.
- `KEEP_LINKED` ID has been kept linked.
- `REUSE_LOCAL` An existing matching local ID has been re-used.
- `MAKE_LOCAL` The newly linked ID has been made local.
- `COPY_LOCAL` The linked ID had other unrelated usages, so it has been duplicated into a local copy.

**TYPE:**

enum in [‘UNSET’, ‘KEEP_LINKED’, ‘REUSE_LOCAL’, ‘MAKE_LOCAL’, ‘COPY_LOCAL’], default ‘UNSET’, (readonly)

**id**

The imported ID. None until it has been linked or appended. May be the same as `reusable_local_id` when appended

**TYPE:**

`ID`, (readonly)

**id_type**

ID type of the item

**TYPE:**

enum in Id Type Items, default ‘ACTION’, (readonly)

**import_info**

Various status info about an item after it has been imported

- `INDIRECT_USAGE` That item was added for an indirectly imported ID, as a dependency of another data-block.
- `LIBOVERRIDE_DEPENDENCY` That item represents an ID also used as liboverride dependency (either directly, as a liboverride reference, or indirectly, as data used by a liboverride reference). It should never be directly made local. Mutually exclusive with `LIBOVERRIDE_DEPENDENCY_ONLY`.
- `LIBOVERRIDE_DEPENDENCY_ONLY` That item represents an ID only used as liboverride dependency (either directly or indirectly see `LIBOVERRIDE_DEPENDENCY` for precisions). It should not be considered during the ‘make local’ (append) process, and remai purely linked data. Mutually exclusive with `LIBOVERRIDE_DEPENDENCY`.

**TYPE:**

enum set in {‘INDIRECT_USAGE’, ‘LIBOVERRIDE_DEPENDENCY’, ‘LIBOVERRIDE_DEPENDENCY_ONLY’}, default {‘INDIRECT_USAGE’}, (readonly)

**library_override_id**

The library override of the linked ID. None until it has been created

**TYPE:**

`ID`, (readonly)

**name**

ID name of the item

**TYPE:**

**TYPE:**

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

**reusable_local_id**

The already existing local ID that may be reused in append & reuse case. None until it has been found

**TYPE:**

`ID`, (readonly)

**source_libraries**

List of libraries to search and import that ID from. The ID will be imported from the first file in that list that contains it

**TYPE:**

`BlendImportContextLibraries` `bpy_prop_collection` of `BlendImportContextLibrary`, (readonly

**source_library**

Library ID representing the blendfile from which the ID was imported. None until the ID has been linked or appended

**TYPE:**

`Library`, (readonly)

**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.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.is_property_overridable_library
- bpy_struct.is_property_readonly
- bpy_struct.is_property_set
- bpy_struct.property_unset
- bpy_struct.type_recast
- bpy_struct.values

## References

- BlendImportContext.import_items

Previous
BlendImportContext(bpy_struct)

Ne
BlendImportContextItems(bpy_stru

Report issue on this page

Skip to content

# BlendImportContextItems(bpy_struct)

base class — `bpy_struct`

**class** bpy.types.**BlendImportContextItems(bpy_struct)**

Collection of blendfile import context items

**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

- `BlendImportContext.import_items`

Previous
BlendImportContextItem(bpy_struct)

Ne
BlendImportContextLibraries(bpy_stru

Report issue on this page