

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 remain 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:

.....

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\(bpy_struct\)](#)
[Report issue on this page](#)

Copyright © Blender Authors
Made with [Furo](#)

[Next](#)
[BlendImportContextItems\(bpy_struct\)](#)