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`