Skip to content

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='CLEAR_ANIMDATA')**

Undocumented, consider contributing.

PARAMETERS:

type (*enum in ['CLEAR_ANIMDATA', 'SET_ACT', 'CLEAR_ACT', 'REFRESH_DRIVERS', 'CLEAR_DRIVERS'], (optional)*) – Animation Operation

- `CLEAR_ANIMDATA` Clear Animation Data – Remove this animation data container.
- `SET_ACT` Set Action.
- `CLEAR_ACT` Unlink Action.
- `REFRESH_DRIVERS` Refresh Drivers.
- `CLEAR_DRIVERS` Clear Drivers.

bpy.ops.outliner.**clear_filter()**

Clear the search filter

bpy.ops.outliner.**collection_color_tag_set(*, color='NONE')**

Set a color tag for the selected collections

PARAMETERS:

color (enum in Collection Color Items, (optional)) – Color Tag

bpy.ops.outliner.**collection_disable()**

Disable viewport display in the view layers

bpy.ops.outliner.**collection_disable_render()**

Do not render this collection

bpy.ops.outliner.**collection_drop()**

Drag to move to collection in Outliner

bpy.ops.outliner.**collection_duplicate()**

Recursively duplicate the collection, all its children, objects and object data

bpy.ops.outliner.**collection_duplicate_linked()**

Recursively duplicate the collection, all its children and objects, with linked object data

bpy.ops.outliner.**collection_enable()**

Enable viewport display in the view layers

bpy.ops.outliner.**collection_enable_render()**

Render the collection

bpy.ops.outliner.**collection_exclude_clear()**

Include collection in the active view layer

bpy.ops.outliner.**collection_exclude_set()**

Exclude collection from the active view layer

bpy.ops.outliner.**collection_hide()**

Hide the collection in this view layer

bpy.ops.outliner.**collection_hide_inside()**

Hide all the objects and collections inside the collection

bpy.ops.outliner.**collection_hierarchy_delete()**

Delete selected collection hierarchies

bpy.ops.outliner.**collection_holdout_clear()**

Clear masking of collection in the active view layer

bpy.ops.outliner.**collection_holdout_set()**

Mask collection in the active view layer

bpy.ops.outliner.**collection_indirect_only_clear()**

Clear collection contributing only indirectly in the view layer

bpy.ops.outliner.**collection_indirect_only_set()**

Set collection to only contribute indirectly (through shadows and reflections) in the view layer

bpy.ops.outliner.**collection_instance()**

Instance selected collections to active scene

bpy.ops.outliner.**collection_isolate(*, extend=False)**

Hide all but this collection and its parents

PARAMETERS:

extend (*boolean, (optional)*) – Extend, Extend current visible collections

bpy.ops.outliner.**collection_link()**

Link selected collections to active scene

bpy.ops.outliner.**collection_new(*, nested=True)**

Add a new collection inside selected collection

PARAMETERS:

nested (*boolean, (optional)*) – Nested, Add as child of selected collection

bpy.ops.outliner.**collection_objects_deselect()**

Deselect objects in collection

bpy.ops.outliner.**collection_objects_select()**

Select objects in collection

bpy.ops.outliner.**collection_show()**

Show the collection in this view layer

bpy.ops.outliner.**collection_show_inside()**

Show all the objects and collections inside the collection

bpy.ops.outliner.**constraint_operation(*, type='ENABLE')**

Undocumented, consider contributing.

PARAMETERS:

type (*enum in ['ENABLE', 'DISABLE', 'DELETE'], (optional)*) – Constraint Operation

bpy.ops.outliner.**data_operation(*, type='DEFAULT')**

Undocumented, consider contributing.

PARAMETERS:

type (*enum in ['DEFAULT'], (optional)*) – Data Operation

bpy.ops.outliner.**datastack_drop()**

Copy or reorder modifiers, constraints, and effects

bpy.ops.outliner.**delete(*, hierarchy=False)**

Delete selected objects and collections

PARAMETERS:

hierarchy (*boolean, (optional)*) – Hierarchy, Delete child objects and collections

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.**hide()**

Hide selected objects and collections

bpy.ops.outliner.**highlight_update()**

Update the item highlight based on the current mouse position

bpy.ops.outliner.**id_copy()**

Copy the selected data-blocks to the internal clipboard

bpy.ops.outliner.**id_delete()**

Delete the ID under cursor

bpy.ops.outliner.**id_operation(*, type='UNLINK')**

General data-block management operations

PARAMETERS:

type (*enum in ['UNLINK', 'LOCAL', 'SINGLE', 'DELETE', 'REMAP', 'COPY', 'PASTE', 'ADD_FAKE', 'CLEAR_FAKE', 'RENAME', 'SELECT_LINKED'], (optional)*) –

ID Data Operation

- `UNLINK` Unlink.
- `LOCAL` Make Local.
- `SINGLE` Make Single User.
- `DELETE` Delete.
- `REMAP` Remap Users – Make all users of selected data-blocks to use instead current (clicked) one.
- `COPY` Copy.
- `PASTE` Paste.

- `ADD_FAKE` Add Fake User – Ensure data-block gets saved even if it isn't in use (e.g. for motion and material libraries).
- `CLEAR_FAKE` Clear Fake User.
- `RENAME` Rename.
- `SELECT_LINKED` Select Linked.

bpy.ops.outliner.**id_paste()**

Paste data-blocks from the internal clipboard

bpy.ops.outliner.**id_remap(*, id_type='OBJECT', old_id='', new_id='')**

Undocumented, consider contributing.

PARAMETERS:

- **id_type** (enum in Id Type Items, (optional)) – ID Type
- **old_id** (*enum in [], (optional)*) – Old ID, Old ID to replace
- **new_id** (*enum in [], (optional)*) – New ID, New ID to remap all selected IDs' users to

bpy.ops.outliner.**item_activate(*, extend=False, extend_range=False, deselect_all=False, recurse=False)**

Handle mouse clicks to select and activate items

PARAMETERS:

- **extend** (*boolean, (optional)*) – Extend, Extend selection for activation
- **extend_range** (*boolean, (optional)*) – Extend Range, Select a range from active element
- **deselect_all** (*boolean, (optional)*) – Deselect On Nothing, Deselect all when nothing under the cursor
- **recurse** (*boolean, (optional)*) – Recurse, Select objects recursively from active element

bpy.ops.outliner.**item_drag_drop()**

Drag and drop element to another place

bpy.ops.outliner.**item_openclose(*, all=False)**

Toggle whether item under cursor is enabled or closed

PARAMETERS:

all (*boolean, (optional)*) – All, Close or open all items

bpy.ops.outliner.**item_rename(*, use_active=False)**

Rename the active element

PARAMETERS:

use_active (*boolean, (optional)*) – Use Active, Rename the active item, rather than the one the mouse is over

bpy.ops.outliner.**keyingset_add_selected()**

Add selected items (blue-gray rows) to active Keying Set

bpy.ops.outliner.**keyingset_remove_selected()**

Remove selected items (blue-gray rows) from active Keying Set

bpy.ops.outliner.**lib_operation(*, type='DELETE')**

Undocumented, consider contributing.

PARAMETERS:

type (*enum in ['DELETE', 'RELOCATE', 'RELOAD'], (optional)*) –

Library Operation

- `DELETE` Delete – Delete this library and all its items.
- `RELOCATE` Relocate – Select a new path for this library, and reload all its data.

- `RELOAD` Reload – Reload all data from this library.

bpy.ops.outliner.**lib_relocate()**

Relocate the library under cursor

bpy.ops.outliner.**liboverride_operation(*, type='OVERRIDE_LIBRARY_CREATE_HIERARCHY', selection_set='SELECTED')**

Create, reset or clear library override hierarchies

PARAMETERS:

- **type** (*enum in ['OVERRIDE_LIBRARY_CREATE_HIERARCHY', 'OVERRIDE_LIBRARY_RESET', 'OVERRIDE_LIBRARY_CLEAR_SINGLE'], (optional)*) –

  Library Override Operation

  - `OVERRIDE_LIBRARY_CREATE_HIERARCHY` Make – Create a local override of the selected linked data-blocks, and their hierarchy of dependencies.
  - `OVERRIDE_LIBRARY_RESET` Reset – Reset the selected local overrides to their linked references values.
  - `OVERRIDE_LIBRARY_CLEAR_SINGLE` Clear – Delete the selected local overrides and relink their usages to the linked data-blocks if possible, else reset them and mark them as non editable.

- **selection_set** (*enum in ['SELECTED', 'CONTENT', 'SELECTED_AND_CONTENT'], (optional)*) –

  Selection Set, Over which part of the tree items to apply the operation

  - `SELECTED` Selected – Apply the operation over selected data-blocks only.
  - `CONTENT` Content – Apply the operation over content of the selected items only (the data-blocks in their sub-tree).
  - `SELECTED_AND_CONTENT` Selected & Content – Apply the operation over selected data-blocks and all their dependencies.

bpy.ops.outliner.**liboverride_troubleshoot_operation(*, type='OVERRIDE_LIBRARY_RESYNC_HIERARCHY', selection_set='SELECTED')**

Advanced operations over library override to help fix broken hierarchies

PARAMETERS:

- **type** (*enum in ['OVERRIDE_LIBRARY_RESYNC_HIERARCHY', 'OVERRIDE_LIBRARY_RESYNC_HIERARCHY_ENFORCE', 'OVERRIDE_LIBRARY_DELETE_HIERARCHY'], (optional)*) –

  Library Override Troubleshoot Operation

  - `OVERRIDE_LIBRARY_RESYNC_HIERARCHY` Resync – Rebuild the selected local overrides from their linked references, as well as their hierarchies of dependencies.
  - `OVERRIDE_LIBRARY_RESYNC_HIERARCHY_ENFORCE` Resync Enforce – Rebuild the selected local overrides from their linked references, as well as their hierarchies of dependencies, enforcing these hierarchies to match the linked data (i.e. ignoring existing overrides on data-blocks pointer properties).
  - `OVERRIDE_LIBRARY_DELETE_HIERARCHY` Delete – Delete the selected local overrides (including their hierarchies of override dependencies) and relink their usages to the linked data-blocks.

- **selection_set** (*enum in ['SELECTED', 'CONTENT', 'SELECTED_AND_CONTENT'], (optional)*) –

  Selection Set, Over which part of the tree items to apply the operation

  - `SELECTED` Selected – Apply the operation over selected data-blocks only.
  - `CONTENT` Content – Apply the operation over content of the selected items only (the data-blocks in their sub-tree).
  - `SELECTED_AND_CONTENT` Selected & Content – Apply the operation over selected data-blocks and all their dependencies.

bpy.ops.outliner.**material_drop()**

Drag material to object in Outliner

bpy.ops.outliner.**modifier_operation(*, type='APPLY')**

Undocumented, consider contributing.

PARAMETERS:

type (*enum in ['APPLY', 'DELETE', 'TOGVIS', 'TOGREN'], (optional)*) – Modifier Operation

bpy.ops.outliner.**object_operation(*, type='SELECT')**

Undocumented, consider contributing.

PARAMETERS:

type (*enum in ['SELECT', 'DESELECT', 'SELECT_HIERARCHY', 'REMAP', 'RENAME'], (optional)*) –

Object Operation

- `SELECT` Select.
- `DESELECT` Deselect.
- `SELECT_HIERARCHY` Select Hierarchy.
- `REMAP` Remap Users – Make all users of selected data-blocks to use instead a new chosen one.
- `RENAME` Rename.

bpy.ops.outliner.**operation()**

Context menu for item operations

bpy.ops.outliner.**orphans_manage()**

Open a window to manage unused data

bpy.ops.outliner.**orphans_purge(*, do_local_ids=True, do_linked_ids=True, do_recursive=True)**

Clear all orphaned data-blocks without any users from the file

PARAMETERS:

- **do_local_ids** (*boolean, (optional)*) – Local Data-blocks, Include unused local data-blocks into deletion
- **do_linked_ids** (*boolean, (optional)*) – Linked Data-blocks, Include unused linked data-blocks into deletion
- **do_recursive** (*boolean, (optional)*) – Recursive Delete, Recursively check for indirectly unused data-blocks, ensuring that no orphaned data blocks remain after execution

bpy.ops.outliner.**parent_clear()**

Drag to clear parent in Outliner

bpy.ops.outliner.**parent_drop()**

Drag to parent in Outliner

bpy.ops.outliner.**scene_drop()**

Drag object to scene in Outliner

bpy.ops.outliner.**scene_operation(*, type='DELETE')**

Context menu for scene operations

PARAMETERS:

type (*enum in ['DELETE'], (optional)*) – Scene Operation

bpy.ops.outliner.**scroll_page(*, up=False)**

Scroll page up or down

PARAMETERS:

up (*boolean, (optional)*) – Up, Scroll up one page

bpy.ops.outliner.**select_all(*, action='TOGGLE')**

Toggle the Outliner selection of items

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.outliner.**select_box(*, tweak=False, xmin=0, xmax=0, ymin=0, ymax=0, wait_for_input=True, mode='SET')**

Use box selection to select tree elements

PARAMETERS:

- **tweak** (*boolean, (optional)*) – Tweak, Tweak gesture from empty space for box selection
- **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
- **wait_for_input** (*boolean, (optional)*) – Wait for Input
- **mode** (*enum in ['SET', 'ADD', 'SUB'], (optional)*) –
  Mode
  - `SET` Set – Set a new selection.
  - `ADD` Extend – Extend existing selection.
  - `SUB` Subtract – Subtract existing selection.

bpy.ops.outliner.**select_walk(*, direction='UP', extend=False, toggle_all=False)**

Use walk navigation to select tree elements

PARAMETERS:

- **direction** (*enum in ['UP', 'DOWN', 'LEFT', 'RIGHT'], (optional)*) – Walk Direction, Select/Deselect element in this direction
- **extend** (*boolean, (optional)*) – Extend, Extend selection on walk
- **toggle_all** (*boolean, (optional)*) – Toggle All, Toggle open/close hierarchy

bpy.ops.outliner.**show_active()**

Open up the tree and 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.**start_filter()**

Start entering filter text

bpy.ops.outliner.**unhide_all()**

Unhide all objects and collections