Skip to content

Node Operators

bpy.ops.node.**add_collection(*, name='', session_uid=0)**

Add a collection info node to the current node editor

PARAMETERS:

- **name** (*string, (optional, never None)*) – Name, Name of the data-block to use by the operator
- **session_uid** (*int in [-inf, inf], (optional)*) – Session UID, Session UID of the data-block to use by the operator

bpy.ops.node.**add_color(*, color=(0.0, 0.0, 0.0, 0.0), gamma=False, has_alpha=False)**

Add a color node to the current node editor

PARAMETERS:

- **color** (*float array of 4 items in [0, inf], (optional)*) – Color, Source color
- **gamma** (*boolean, (optional)*) – Gamma Corrected, The source color is gamma corrected
- **has_alpha** (*boolean, (optional)*) – Has Alpha, The source color contains an Alpha component

bpy.ops.node.**add_file(*, filepath='', directory='', files=None, hide_props_region=True, check_existing=False, filter_blender=False, filter_backup=False, filter_image=True, filter_movie=True, filter_python=False, filter_font=False, filter_sound=False, filter_text=False, filter_archive=False, filter_btx=False, filter_collada=False, filter_alembic=False, filter_usd=False, filter_obj=False, filter_volume=False, filter_folder=True, filter_blenlib=False, filemode=9, relative_path=True, show_multiview=False, use_multiview=False, display_type='DEFAULT', sort_method='', name='', session_uid=0)**

Add a file node to the current node editor

PARAMETERS:

- **filepath** (*string, (optional, never None)*) – File Path, Path to file
- **directory** (*string, (optional, never None)*) – Directory, Directory of the file
- **files** (`bpy_prop_collection` of `OperatorFileListElement`, (optional)) – Files
- **hide_props_region** (*boolean, (optional)*) – Hide Operator Properties, Collapse the region displaying the operator settings
- **check_existing** (*boolean, (optional)*) – Check Existing, Check and warn on overwriting existing files
- **filter_blender** (*boolean, (optional)*) – Filter .blend files
- **filter_backup** (*boolean, (optional)*) – Filter .blend files
- **filter_image** (*boolean, (optional)*) – Filter image files
- **filter_movie** (*boolean, (optional)*) – Filter movie files
- **filter_python** (*boolean, (optional)*) – Filter Python files
- **filter_font** (*boolean, (optional)*) – Filter font files
- **filter_sound** (*boolean, (optional)*) – Filter sound files
- **filter_text** (*boolean, (optional)*) – Filter text files
- **filter_archive** (*boolean, (optional)*) – Filter archive files
- **filter_btx** (*boolean, (optional)*) – Filter btx files
- **filter_collada** (*boolean, (optional)*) – Filter COLLADA files
- **filter_alembic** (*boolean, (optional)*) – Filter Alembic files
- **filter_usd** (*boolean, (optional)*) – Filter USD files
- **filter_obj** (*boolean, (optional)*) – Filter OBJ files
- **filter_volume** (*boolean, (optional)*) – Filter OpenVDB volume files
- **filter_folder** (*boolean, (optional)*) – Filter folders
- **filter_blenlib** (*boolean, (optional)*) – Filter Blender IDs
- **filemode** (*int in [1, 9], (optional)*) – File Browser Mode, The setting for the file browser mode to load a .blend file, a library or a special fil
- **relative_path** (*boolean, (optional)*) – Relative Path, Select the file relative to the blend file
- **show_multiview** (*boolean, (optional)*) – Enable Multi-View

- **use_multiview** (*boolean, (optional)*) – Use Multi-View
- **display_type** (*enum in ['DEFAULT', 'LIST_VERTICAL', 'LIST_HORIZONTAL', 'THUMBNAIL'], (optional)*) –
  Display Type
  - `DEFAULT` Default – Automatically determine display type for files.
  - `LIST_VERTICAL` Short List – Display files as short list.
  - `LIST_HORIZONTAL` Long List – Display files as a detailed list.
  - `THUMBNAIL` Thumbnails – Display files as thumbnails.
- **sort_method** (*enum in ['DEFAULT', 'FILE_SORT_ALPHA', 'FILE_SORT_EXTENSION', 'FILE_SORT_TIME', 'FILE_SORT_SIZE', 'ASSET_CATALOG'], (optional)*) –
  File sorting mode
  - `DEFAULT` Default – Automatically determine sort method for files.
  - `FILE_SORT_ALPHA` Name – Sort the file list alphabetically.
  - `FILE_SORT_EXTENSION` Extension – Sort the file list by extension/type.
  - `FILE_SORT_TIME` Modified Date – Sort files by modification time.
  - `FILE_SORT_SIZE` Size – Sort files by size.
  - `ASSET_CATALOG` Asset Catalog – Sort the asset list so that assets in the same catalog are kept together. Within a single catalog, asse are ordered by name. The catalogs are in order of the flattened catalog hierarchy..
- **name** (*string, (optional, never None)*) – Name, Name of the data-block to use by the operator
- **session_uid** (*int in [-inf, inf], (optional)*) – Session UID, Session UID of the data-block to use by the operator

bpy.ops.node.**add_foreach_geometry_element_zone(*, use_transform=False, settings=None, offset=(150.0, 0.0))**

Add a For Each Geometry Element zone that allows executing nodes e.g. for each vertex separately

PARAMETERS:

- **use_transform** (*boolean, (optional)*) – Use Transform, Start transform operator after inserting the node
- **settings** (`bpy_prop_collection` of `NodeSetting`, (optional)) – Settings, Settings to be applied on the newly created node
- **offset** (*float array of 2 items in [-inf, inf], (optional)*) – Offset, Offset of nodes from the cursor when added

FILE:

startup/bl_operators/node.py:179

bpy.ops.node.**add_group(*, name='', session_uid=0, show_datablock_in_node=True)**

Add an existing node group to the current node editor

PARAMETERS:

- **name** (*string, (optional, never None)*) – Name, Name of the data-block to use by the operator
- **session_uid** (*int in [-inf, inf], (optional)*) – Session UID, Session UID of the data-block to use by the operator
- **show_datablock_in_node** (*boolean, (optional)*) – Show the datablock selector in the node

bpy.ops.node.**add_group_asset(*, asset_library_type='LOCAL', asset_library_identifier='', relative_asset_identifier='')**

Add a node group asset to the active node tree

PARAMETERS:

- **asset_library_type** (enum in Asset Library Type Items, (optional)) – Asset Library Type
- **asset_library_identifier** (*string, (optional, never None)*) – Asset Library Identifier
- **relative_asset_identifier** (*string, (optional, never None)*) – Relative Asset Identifier

bpy.ops.node.**add_mask(*, name='', session_uid=0)**

Add a mask node to the current node editor

PARAMETERS:

- **name** (*string, (optional, never None)*) – Name, Name of the data-block to use by the operator

- **session_uid** (*int in [-inf, inf], (optional)*) – Session UID, Session UID of the data-block to use by the operator

bpy.ops.node.**add_material(*, name='', session_uid=0)**

Add a material node to the current node editor

PARAMETERS:

- **name** (*string, (optional, never None)*) – Name, Name of the data-block to use by the operator
- **session_uid** (*int in [-inf, inf], (optional)*) – Session UID, Session UID of the data-block to use by the operator

bpy.ops.node.**add_node(*, use_transform=False, settings=None, type='')**

Add a node to the active tree

PARAMETERS:

- **use_transform** (*boolean, (optional)*) – Use Transform, Start transform operator after inserting the node
- **settings** (`bpy_prop_collection` of `NodeSetting`, (optional)) – Settings, Settings to be applied on the newly created node
- **type** (*string, (optional, never None)*) – Node Type, Node type

FILE:

startup/bl_operators/node.py:143

bpy.ops.node.**add_object(*, name='', session_uid=0)**

Add an object info node to the current node editor

PARAMETERS:

- **name** (*string, (optional, never None)*) – Name, Name of the data-block to use by the operator
- **session_uid** (*int in [-inf, inf], (optional)*) – Session UID, Session UID of the data-block to use by the operator

bpy.ops.node.**add_repeat_zone(*, use_transform=False, settings=None, offset=(150.0, 0.0))**

Add a repeat zone that allows executing nodes a dynamic number of times

PARAMETERS:

- **use_transform** (*boolean, (optional)*) – Use Transform, Start transform operator after inserting the node
- **settings** (`bpy_prop_collection` of `NodeSetting`, (optional)) – Settings, Settings to be applied on the newly created node
- **offset** (*float array of 2 items in [-inf, inf], (optional)*) – Offset, Offset of nodes from the cursor when added

FILE:

startup/bl_operators/node.py:179

bpy.ops.node.**add_reroute(*, path=None, cursor=11)**

Add a reroute node

PARAMETERS:

- **path** (`bpy_prop_collection` of `OperatorMousePath`, (optional)) – Path
- **cursor** (*int in [0, inf], (optional)*) – Cursor

bpy.ops.node.**add_simulation_zone(*, use_transform=False, settings=None, offset=(150.0, 0.0))**

Add simulation zone input and output nodes to the active tree

PARAMETERS:

- **use_transform** (*boolean, (optional)*) – Use Transform, Start transform operator after inserting the node
- **settings** (`bpy_prop_collection` of `NodeSetting`, (optional)) – Settings, Settings to be applied on the newly created node
- **offset** (*float array of 2 items in [-inf, inf], (optional)*) – Offset, Offset of nodes from the cursor when added

FILE:

startup/bl_operators/node.py:179

bpy.ops.node.**attach()**

Attach active node to a frame

bpy.ops.node.**backimage_fit()**

Fit the background image to the view

bpy.ops.node.**backimage_move()**

Move node backdrop

bpy.ops.node.**backimage_sample()**

Use mouse to sample background image

bpy.ops.node.**backimage_zoom(*, factor=1.2)**

Zoom in/out the background image

PARAMETERS:

factor (*float in [0, 10], (optional)*) – Factor

bpy.ops.node.**bake_node_item_add()**

Add item below active item

bpy.ops.node.**bake_node_item_move(*, direction='UP')**

Move active item

PARAMETERS:

direction (*enum in ['UP', 'DOWN'], (optional)*) – Direction, Move direction

bpy.ops.node.**bake_node_item_remove()**

Remove active item

bpy.ops.node.**capture_attribute_item_add()**

Add item below active item

bpy.ops.node.**capture_attribute_item_move(*, direction='UP')**

Move active item

PARAMETERS:

direction (*enum in ['UP', 'DOWN'], (optional)*) – Direction, Move direction

bpy.ops.node.**capture_attribute_item_remove()**

Remove active item

bpy.ops.node.**clear_viewer_border()**

Clear the boundaries for viewer operations

bpy.ops.node.**clipboard_copy()**

Copy the selected nodes to the internal clipboard

bpy.ops.node.**clipboard_paste(*, offset=(0.0, 0.0))**

Paste nodes from the internal clipboard to the active node tree

PARAMETERS:

offset (*float array of 2 items in [-inf, inf], (optional)*) – Location, The 2D view location for the center of the new nodes, or unchanged if n
set

bpy.ops.node.**collapse_hide_unused_toggle()**

Toggle collapsed nodes and hide unused sockets

FILE:
startup/bl_operators/node.py:249

bpy.ops.node.**connect_to_output(*, run_in_geometry_nodes=True)**

Connect active node to the active output node of the node tree

PARAMETERS:
run_in_geometry_nodes (*boolean, (optional)*) – Run in Geometry Nodes Editor

FILE:
startup/bl_operators/connect_to_output.py:249

bpy.ops.node.**cryptomatte_layer_add()**

Add a new input layer to a Cryptomatte node

bpy.ops.node.**cryptomatte_layer_remove()**

Remove layer from a Cryptomatte node

bpy.ops.node.**deactivate_viewer()**

Deactivate selected viewer node in geometry nodes

bpy.ops.node.**default_group_width_set()**

Set the width based on the parent group node in the current context

bpy.ops.node.**delete()**

Remove selected nodes

bpy.ops.node.**delete_reconnect()**

Remove nodes and reconnect nodes as if deletion was muted

bpy.ops.node.**detach()**

Detach selected nodes from parents

bpy.ops.node.**detach_translate_attach(*, NODE_OT_detach=None, TRANSFORM_OT_translate=None, NODE_OT_attach=None)**

Detach nodes, move and attach to frame

PARAMETERS:

- **NODE_OT_detach** (`NODE_OT_detach`, (optional)) – Detach Nodes, Detach selected nodes from parents
- **TRANSFORM_OT_translate** (`TRANSFORM_OT_translate`, (optional)) – Move, Move selected items
- **NODE_OT_attach** (`NODE_OT_attach`, (optional)) – Attach Nodes, Attach active node to a frame

bpy.ops.node.**duplicate(*, keep_inputs=False, linked=True)**

Duplicate selected nodes

PARAMETERS:

- **keep_inputs** (*boolean, (optional)*) – Keep Inputs, Keep the input links to duplicated nodes
- **linked** (*boolean, (optional)*) – Linked, Duplicate node but not node trees, linking to the original data

bpy.ops.node.**duplicate_move(*, NODE_OT_duplicate=None, NODE_OT_translate_attach=None)**

Duplicate selected nodes and move them

PARAMETERS:

- **NODE_OT_duplicate** (`NODE_OT_duplicate`, (optional)) – Duplicate Nodes, Duplicate selected nodes
- **NODE_OT_translate_attach** (`NODE_OT_translate_attach`, (optional)) – Move and Attach, Move nodes and attach to frame

bpy.ops.node.**duplicate_move_keep_inputs(*, NODE_OT_duplicate=None, NODE_OT_translate_attach=None)** bpy.ops.node.**duplicate_move_keep_inputs(*, NODE_OT_duplicate=None, NODE_OT_translate_attach=None)**

Duplicate selected nodes keeping input links and move them

PARAMETERS:

- **NODE_OT_duplicate** (`NODE_OT_duplicate`, (optional)) – Duplicate Nodes, Duplicate selected nodes
- **NODE_OT_translate_attach** (`NODE_OT_translate_attach`, (optional)) – Move and Attach, Move nodes and attach to frame

bpy.ops.node.**duplicate_move_linked(*, NODE_OT_duplicate=None, NODE_OT_translate_attach=None)**

Duplicate selected nodes, but not their node trees, and move them

PARAMETERS:

- **NODE_OT_duplicate** (`NODE_OT_duplicate`, (optional)) – Duplicate Nodes, Duplicate selected nodes
- **NODE_OT_translate_attach** (`NODE_OT_translate_attach`, (optional)) – Move and Attach, Move nodes and attach to frame

bpy.ops.node.**enum_definition_item_add()**

Add item below active item

bpy.ops.node.**enum_definition_item_move(*, direction='UP')**

Move active item

PARAMETERS:

direction (*enum in ['UP', 'DOWN'], (optional)*) – Direction, Move direction

bpy.ops.node.**enum_definition_item_remove()**

Remove active item

bpy.ops.node.**find_node()**

Search for a node by name and focus and select it

bpy.ops.node.**foreach_geometry_element_zone_generation_item_add()**

Add item below active item

bpy.ops.node.**foreach_geometry_element_zone_generation_item_move(*, direction='UP')**

Move active item

PARAMETERS:

direction (*enum in ['UP', 'DOWN'], (optional)*) – Direction, Move direction

bpy.ops.node.**foreach_geometry_element_zone_generation_item_remove()**

Remove active item

bpy.ops.node.**foreach_geometry_element_zone_input_item_add()**

Add item below active item

bpy.ops.node.**foreach_geometry_element_zone_input_item_move(*, direction='UP')**

Move active item

PARAMETERS:

direction (*enum in ['UP', 'DOWN'], (optional)*) – Direction, Move direction

bpy.ops.node.**foreach_geometry_element_zone_input_item_remove()**

Remove active item

bpy.ops.node.**foreach_geometry_element_zone_main_item_add()**

Add item below active item

bpy.ops.node.**foreach_geometry_element_zone_main_item_move(*, direction='UP')**

[illegible]

Move active item

PARAMETERS:

direction (*enum in ['UP', 'DOWN'], (optional)*) – Direction, Move direction

bpy.ops.node.**foreach_geometry_element_zone_main_item_remove()**

Remove active item

bpy.ops.node.**gltf_settings_node_operator()**

Add a node to the active tree for glTF export

FILE:

addons_core/io_scene_gltf2/blender/com/gltf2_blender_ui.py:35

bpy.ops.node.**group_edit(*, exit=False)**

Edit node group

PARAMETERS:

exit (*boolean, (optional)*) – Exit

bpy.ops.node.**group_insert()**

Insert selected nodes into a node group

bpy.ops.node.**group_make()**

Make group from selected nodes

bpy.ops.node.**group_separate(*, type='COPY')**

Separate selected nodes from the node group

PARAMETERS:

type (*enum in ['COPY', 'MOVE'], (optional)*) –

Type

- `COPY` Copy – Copy to parent node tree, keep group intact.
- `MOVE` Move – Move to parent node tree, remove from group.

bpy.ops.node.**group_ungroup()**

Ungroup selected nodes

bpy.ops.node.**hide_socket_toggle()**

Toggle unused node socket display

bpy.ops.node.**hide_toggle()**

Toggle hiding of selected nodes

bpy.ops.node.**index_switch_item_add()**

Add bake item

bpy.ops.node.**index_switch_item_remove(*, index=0)**

Remove an item from the index switch

PARAMETERS:

index (*int in [0, inf], (optional)*) – Index, Index to remove

bpy.ops.node.**insert_offset()**

Automatically offset nodes on insertion

bpy.ops.node.**interface_item_duplicate()**

Add a copy of the active item to the interface

FILE:

startup/bl_operators/node.py:380

bpy.ops.node.**interface_item_new(*, item_type='INPUT')**

Add a new item to the interface

PARAMETERS:

item_type (*enum in ['INPUT', 'OUTPUT', 'PANEL'], (optional)*) – Item Type, Type of the item to create

FILE:

startup/bl_operators/node.py:335

bpy.ops.node.**interface_item_remove()**

Remove active item from the interface

FILE:

startup/bl_operators/node.py:399

bpy.ops.node.**join()**

Attach selected nodes to a new common frame

bpy.ops.node.**link(*, detach=False, drag_start=(0.0, 0.0), inside_padding=2.0, outside_padding=0.0, speed_ramp=1.0, max_speed=26.0, delay=0.5, zoom_influence=0.5)**

Use the mouse to create a link between two nodes

PARAMETERS:

- **detach** (*boolean, (optional)*) – Detach, Detach and redirect existing links
- **drag_start** (*float array of 2 items in [-6, 6], (optional)*) – Drag Start, The position of the mouse cursor at the start of the operation
- **inside_padding** (*float in [0, 100], (optional)*) – Inside Padding, Inside distance in UI units from the edge of the region within which to start panning
- **outside_padding** (*float in [0, 100], (optional)*) – Outside Padding, Outside distance in UI units from the edge of the region at which to stop panning
- **speed_ramp** (*float in [0, 100], (optional)*) – Speed Ramp, Width of the zone in UI units where speed increases with distance from the edg
- **max_speed** (*float in [0, 10000], (optional)*) – Max Speed, Maximum speed in UI units per second
- **delay** (*float in [0, 10], (optional)*) – Delay, Delay in seconds before maximum speed is reached
- **zoom_influence** (*float in [0, 1], (optional)*) – Zoom Influence, Influence of the zoom factor on scroll speed

bpy.ops.node.**link_make(*, replace=False)**

Make a link between selected output and input sockets

PARAMETERS:

replace (*boolean, (optional)*) – Replace, Replace socket connections with the new links

bpy.ops.node.**link_viewer()**

Link to viewer node

bpy.ops.node.**links_cut(*, path=None, cursor=15)**

Use the mouse to cut (remove) some links

PARAMETERS:

- **path** (`bpy_prop_collection` of `OperatorMousePath`, (optional)) – Path
- **cursor** (*int in [0, inf], (optional)*) – Cursor

bpy.ops.node.**links_detach()**

Remove all links to selected nodes, and try to connect neighbor nodes together

bpy.ops.node.**links_mute(*, path=None, cursor=38)**

Use the mouse to mute links

PARAMETERS:

- **path** (`bpy_prop_collection` of `OperatorMousePath`, (optional)) – Path
- **cursor** (*int in [0, inf], (optional)*) – Cursor

bpy.ops.node.**move_detach_links(*, NODE_OT_links_detach=None, TRANSFORM_OT_translate=None)**

Move a node to detach links

PARAMETERS:

- **NODE_OT_links_detach** (`NODE_OT_links_detach`, (optional)) – Detach Links, Remove all links to selected nodes, and try to connect neighbor nodes together
- **TRANSFORM_OT_translate** (`TRANSFORM_OT_translate`, (optional)) – Move, Move selected items

bpy.ops.node.**move_detach_links_release(*, NODE_OT_links_detach=None, NODE_OT_translate_attach=None)**

Move a node to detach links

PARAMETERS:

- **NODE_OT_links_detach** (`NODE_OT_links_detach`, (optional)) – Detach Links, Remove all links to selected nodes, and try to connect neighbor nodes together
- **NODE_OT_translate_attach** (`NODE_OT_translate_attach`, (optional)) – Move and Attach, Move nodes and attach to frame

bpy.ops.node.**mute_toggle()**

Toggle muting of selected nodes

bpy.ops.node.**new_geometry_node_group_assign()**

Create a new geometry node group and assign it to the active modifier

FILE:

startup/bl_operators/geometry_nodes.py:320

bpy.ops.node.**new_geometry_node_group_tool()**

Create a new geometry node group for a tool

FILE:

startup/bl_operators/geometry_nodes.py:341

bpy.ops.node.**new_geometry_nodes_modifier()**

Create a new modifier with a new geometry node group

FILE:

startup/bl_operators/geometry_nodes.py:297

bpy.ops.node.**new_node_tree(*, type='', name='NodeTree')**

Create a new node tree

PARAMETERS:

- **type** (*enum in [], (optional)*) – Tree Type
- **name** (*string, (optional, never None)*) – Name

bpy.ops.node.**node_color_preset_add(*, name='', remove_name=False, remove_active=False)**

Add or remove a Node Color Preset

Add or remove a Node Color Preset

PARAMETERS:

- **name** (*string, (optional, never None)*) – Name, Name of the preset, used to make the path name
- **remove_name** (*boolean, (optional)*) – remove_name
- **remove_active** (*boolean, (optional)*) – remove_active

FILE:

startup/bl_operators/presets.py:119

bpy.ops.node.**node_copy_color()**

Copy color to all selected nodes

bpy.ops.node.**options_toggle()**

Toggle option buttons display for selected nodes

bpy.ops.node.**output_file_add_socket(*, file_path='Image')**

Add a new input to a file output node

PARAMETERS:

file_path (*string, (optional, never None)*) – File Path, Subpath of the output file

bpy.ops.node.**output_file_move_active_socket(*, direction='DOWN')**

Move the active input of a file output node up or down the list

PARAMETERS:

direction (*enum in ['UP', 'DOWN'], (optional)*) – Direction

bpy.ops.node.**output_file_remove_active_socket()**

Remove the active input from a file output node

bpy.ops.node.**parent_set()**

Attach selected nodes

bpy.ops.node.**preview_toggle()**

Toggle preview display for selected nodes

bpy.ops.node.**read_viewlayers()**

Read all render layers of all used scenes

bpy.ops.node.**render_changed()**

Render current scene, when input node’s layer has been changed

bpy.ops.node.**repeat_zone_item_add()**

Add item below active item

bpy.ops.node.**repeat_zone_item_move(*, direction='UP')**

Move active item

PARAMETERS:

direction (*enum in ['UP', 'DOWN'], (optional)*) – Direction, Move direction

bpy.ops.node.**repeat_zone_item_remove()**

Remove active item

bpy.ops.node.**resize()**

Resize a node

bpy.ops.node.**select(*, extend=False, deselect=False, toggle=False, deselect_all=False, select_passthrough=False, location=(0, 0), socket_select=False, clear_viewer=False)**

Select the node under the cursor

PARAMETERS:

- **extend** (*boolean, (optional)*) – Extend, Extend selection instead of deselecting everything first
- **deselect** (*boolean, (optional)*) – Deselect, Remove from selection
- **toggle** (*boolean, (optional)*) – Toggle Selection, Toggle the selection
- **deselect_all** (*boolean, (optional)*) – Deselect On Nothing, Deselect all when nothing under the cursor
- **select_passthrough** (*boolean, (optional)*) – Only Select Unselected, Ignore the select action when the element is already selected
- **location** (*int array of 2 items in [-inf, inf], (optional)*) – Location, Mouse location
- **socket_select** (*boolean, (optional)*) – Socket Select
- **clear_viewer** (*boolean, (optional)*) – Clear Viewer, Deactivate geometry nodes viewer when clicking in empty space

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

(De)select all nodes

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

Use box selection to select nodes

PARAMETERS:

- **tweak** (*boolean, (optional)*) – Tweak, Only activate when mouse is not over a node (useful for tweak gesture)
- **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.node.**select_circle(*, x=0, y=0, radius=25, wait_for_input=True, mode='SET')**

Use circle selection to select nodes

PARAMETERS:

- **x** (*int in [-inf, inf], (optional)*) – X
- **y** (*int in [-inf, inf], (optional)*) – Y
- **radius** (*int in [1, inf], (optional)*) – Radius
- **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.node.**select_grouped(*, extend=False, type='TYPE')**

Select nodes with similar properties

PARAMETERS:

- **extend** (*boolean, (optional)*) – Extend, Extend selection instead of deselecting everything first
- **type** (*enum in ['TYPE', 'COLOR', 'PREFIX', 'SUFFIX'], (optional)*) – Type

bpy.ops.node.**select_lasso(*, tweak=False, path=None, use_smooth_stroke=False, smooth_stroke_factor=0.75, smooth_stroke_radius=3**
mode='SET')

Select nodes using lasso selection

PARAMETERS:

- **tweak** (*boolean, (optional)*) – Tweak, Only activate when mouse is not over a node (useful for tweak gesture)
- **path** (`bpy_prop_collection` of `OperatorMousePath`, (optional)) – Path
- **use_smooth_stroke** (*boolean, (optional)*) – Stabilize Stroke, Selection lags behind mouse and follows a smoother path
- **smooth_stroke_factor** (*float in [0.5, 0.99], (optional)*) – Smooth Stroke Factor, Higher values gives a smoother stroke
- **smooth_stroke_radius** (*int in [10, 200], (optional)*) – Smooth Stroke Radius, Minimum distance from last point before selection continue
- **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.node.**select_link_viewer(*, NODE_OT_select=None, NODE_OT_link_viewer=None)**

Select node and link it to a viewer node

PARAMETERS:

- **NODE_OT_select** (`NODE_OT_select`, (optional)) – Select, Select the node under the cursor
- **NODE_OT_link_viewer** (`NODE_OT_link_viewer`, (optional)) – Link to Viewer Node, Link to viewer node

bpy.ops.node.**select_linked_from()**

Select nodes linked from the selected ones

bpy.ops.node.**select_linked_to()**

Select nodes linked to the selected ones

bpy.ops.node.**select_same_type_step(*, prev=False)**

Activate and view same node type, step by step

PARAMETERS:

prev (*boolean, (optional)*) – Previous

bpy.ops.node.**shader_script_update()**

Update shader script node with new sockets and options from the script

bpy.ops.node.**simulation_zone_item_add()**

Add item below active item

bpy.ops.node.**simulation_zone_item_move(*, direction='UP')**

Move active item

PARAMETERS:

direction (*enum in ['UP', 'DOWN'], (optional)*) – Direction, Move direction

bpy.ops.node.**simulation_zone_item_remove()**

Remove active item

bpy.ops.node.**translate_attach(*, TRANSFORM_OT_translate=None, NODE_OT_attach=None)**

Move nodes and attach to frame

PARAMETERS:

- **TRANSFORM_OT_translate** (`TRANSFORM_OT_translate`, (optional)) – Move, Move selected items
- **NODE_OT_attach** (`NODE_OT_attach`, (optional)) – Attach Nodes, Attach active node to a frame

bpy.ops.node.**translate_attach_remove_on_cancel(*, TRANSFORM_OT_translate=None, NODE_OT_attach=None)**

Move nodes and attach to frame

PARAMETERS:

- **TRANSFORM_OT_translate** (`TRANSFORM_OT_translate`, (optional)) – Move, Move selected items
- **NODE_OT_attach** (`NODE_OT_attach`, (optional)) – Attach Nodes, Attach active node to a frame

bpy.ops.node.**tree_path_parent()**

Go to parent node tree

FILE:

startup/bl_operators/node.py:279

bpy.ops.node.**view_all()**

Resize view so you can see all nodes

bpy.ops.node.**view_selected()**

Resize view so you can see selected nodes

bpy.ops.node.**viewer_border(*, xmin=0, xmax=0, ymin=0, ymax=0, wait_for_input=True)**

Set the boundaries for viewer operations

PARAMETERS:

- **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

bpy.ops.node.**viewer_shortcut_get(*, viewer_index=0)**

Activate a specific compositor viewer node using 1,2,..,9 keys

PARAMETERS:

viewer_index (*int in [-inf, inf], (optional)*) – Viewer Index, Index corresponding to the shortcut, e.g. number key 1 corresponds to index 1 etc..

FILE:

startup/bl_operators/node.py:505

bpy.ops.node.**viewer_shortcut_set(*, viewer_index=0)**

Create a compositor viewer shortcut for the selected node by pressing ctrl+1,2,..9

PARAMETERS:

viewer_index (*int in [-inf, inf], (optional)*) – Viewer Index, Index corresponding to the shortcut, e.g. number key 1 corresponds to index 1 etc..

FILE:

startup/bl_operators/node.py:441