NodeTree(ID)

Poll Function

The `NodeTree.poll` function determines if a node tree is visible in the given context (similar to how `Panel.poll` and `Menu.poll` define visibility). If it returns False, the node tree type will not be selectable in the node editor.

A typical condition for shader nodes would be to check the active render engine of the scene and only show nodes of the renderer they are designed for.

```
import bpy


class CyclesNodeTree(bpy.types.NodeTree):
    """ This operator is only visible when Cycles is the selected render engine"""
    bl_label = "Cycles Node Tree"
    bl_icon = 'NONE'

    @classmethod
    def poll(cls, context):
        return context.scene.render.engine == 'CYCLES'


bpy.utils.register_class(CyclesNodeTree)
```

base classes — `bpy_struct`, `ID`

subclasses — `CompositorNodeTree`, `GeometryNodeTree`, `ShaderNodeTree`, `TextureNodeTree`

class bpy.types.**NodeTree(ID)**

Node tree consisting of linked nodes used for shading, textures and compositing

animation_data

Animation data for this data-block

TYPE:

`AnimData`, (readonly)

bl_description

TYPE:

string, default “”, (never None)

bl_icon

The node tree icon

TYPE:

enum in Icon Items, default ‘NODETREE’

bl_idname

TYPE:

string, default “”, (never None)

bl_label

The node tree label

TYPE:

string, default “”, (never None)

bl_use_group_interface

Determines the visibility of some UI elements related to node groups

TYPE:

boolean, default True

color_tag

Color tag of the node group which influences the header color

- `NONE` None – Default color tag for new nodes and node groups.
- `ATTRIBUTE` Attribute.
- `COLOR` Color.
- `CONVERTER` Converter.
- `DISTORT` Distort.
- `FILTER` Filter.
- `GEOMETRY` Geometry.
- `INPUT` Input.
- `MATTE` Matte.
- `OUTPUT` Output.
- `SCRIPT` Script.
- `SHADER` Shader.
- `TEXTURE` Texture.
- `VECTOR` Vector.
- `PATTERN` Pattern.
- `INTERFACE` Interface.
- `GROUP` Group.

TYPE:

enum in [‘NONE’, ‘ATTRIBUTE’, ‘COLOR’, ‘CONVERTER’, ‘DISTORT’, ‘FILTER’, ‘GEOMETRY’, ‘INPUT’, ‘MATTE’, ‘OUTPUT’, ‘SCRIPT’, ‘SHADER’, ‘TEXTURE’, ‘VECTOR’, ‘PATTERN’, ‘INTERFACE’, ‘GROUP’], default ‘NONE’

default_group_node_width

The width for newly created group nodes

TYPE:

int in [40, 700], default 140

description

Description of the node tree

TYPE:

string, default “”, (never None)

grease_pencil

Grease Pencil data-block

TYPE:

`GreasePencil`

interface

Interface declaration for this node tree

TYPE:

NodeTreeInterface, (readonly)

links

TYPE:

NodeLinks bpy_prop_collection of NodeLink, (readonly)

nodes

TYPE:

Nodes bpy_prop_collection of Node, (readonly)

type

Node Tree type (deprecated, bl_idname is the actual node tree type identifier)

- UNDEFINED Undefined – Undefined type of nodes (can happen e.g. when a linked node tree goes missing).
- CUSTOM Custom – Custom nodes.
- SHADER Shader – Shader nodes.
- TEXTURE Texture – Texture nodes.
- COMPOSITING Compositing – Compositing nodes.
- GEOMETRY Geometry – Geometry nodes.

TYPE:

enum in ['UNDEFINED', 'CUSTOM', 'SHADER', 'TEXTURE', 'COMPOSITING', 'GEOMETRY'], default 'SHADER', (readon

view_center

The current location (offset) of the view for this Node Tree

TYPE:

mathutils.Vector of 2 items in [-inf, inf], default (0.0, 0.0), (readonly)

interface_update(context)

Updated node group interface

contains_tree(sub_tree)

Check if the node tree contains another. Used to avoid creating recursive node groups.

PARAMETERS:

sub_tree (NodeTree, (never None)) – Node Tree, Node tree for recursive check

RETURNS:

contained

RETURN TYPE:

boolean

classmethod poll(context)

Check visibility in the editor

RETURN TYPE:

boolean

update()

Update on editor changes

classmethod get_from_context(context)

Get a node tree from the context

RETURNS:

result_1, Active node tree from context, NodeTree

result_2, ID data-block that owns the node tree, ID

result_3, Original ID data-block selected from the context, ID

RETURN TYPE:

(NodeTree, ID, ID)

classmethod valid_socket_type(idname)

Check if the socket type is valid for the node tree

PARAMETERS:

idname (*string, (never None)*) – Socket Type, Identifier of the socket type

RETURN TYPE:

boolean

debug_lazy_function_graph()

Get the internal lazy-function graph for this node tree

RETURNS:

Dot Graph, Graph in dot format

RETURN TYPE:

string

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
- ID.name
- ID.name_full
- ID.id_type
- ID.session_uid
- ID.is_evaluated
- ID.original
- ID.users
- ID.use_fake_user
- ID.is_missing
- ID.is_runtime_data
- ID.is_editable
- ID.tag
- ID.is_library_indirect
- ID.library
- ID.library_weak_reference
- ID.asset_data

- ID.use_fake_user
- ID.use_extra_user
- ID.is_embedded_data
- ID.override_library
- ID.preview

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
- ID.rename
- ID.evaluated_get
- ID.copy
- ID.asset_mark
- ID.asset_clear
- ID.asset_generate_preview
- ID.override_create
- ID.override_hierarchy_create
- ID.user_clear
- ID.user_remap
- ID.make_local
- ID.user_of_id
- ID.animation_data_create
- ID.animation_data_clear
- ID.update_tag
- ID.preview_ensure
- ID.bl_rna_get_subclass
- ID.bl_rna_get_subclass_py

References

- BlendData.node_groups
- BlendDataNodeTrees.new
- BlendDataNodeTrees.remove
- CompositorNodeCustomGroup.node_tree
- CompositorNodeGroup.node_tree
- FreestyleLineStyle.node_tree
- GeometryNodeCustomGroup.node_tree
- GeometryNodeGroup.node_tree
- Light.node_tree
- Material.node_tree
- Node.poll
- Node.poll_instance
- NodeCustomGroup.node_tree
- NodeGroup.node_tree
- NodeInternal.poll
- NodeInternal.poll_instance
- NodeTree.contains_tree
- NodeTree.get_from_context
- NodeTreePath.node_tree
- NodesModifier.node_group
- Scene.node_tree
- ShaderNodeCustomGroup.node_tree
- ShaderNodeGroup.node_tree
- SpaceNodeEditor.edit_tree
- SpaceNodeEditor.geometry_nodes_tool_tree
- SpaceNodeEditor.node_tree
- SpaceNodeEditorPath.append
- SpaceNodeEditorPath.start
- Texture.node_tree
- TextureNodeGroup.node_tree
- UILayout.template_node_link
- UILayout.template_node_view
- World.node_tree