

CompositorNodeChannelMatte(CompositorNode)

base classes — `bpy_struct`, `Node`, `NodeInternal`, `CompositorNode`

class `bpy.types.CompositorNodeChannelMatte(CompositorNode)`

Create matte based on differences in color channels

color_space

- `RGB` RGB – RGB (Red, Green, Blue) color space.
- `HSV` HSV – HSV (Hue, Saturation, Value) color space.
- `YUV` YUV – YUV (Y - luma, U V - chroma) color space.
- `YCC` YCbCr – YCbCr (Y - luma, Cb - blue-difference chroma, Cr - red-difference chroma) color space.

TYPE:

enum in ['RGB', 'HSV', 'YUV', 'YCC'], default 'RGB'

limit_channel

Limit by this channel's value

- `R` R – Red.
- `G` G – Green.
- `B` B – Blue.

TYPE:

enum in ['R', 'G', 'B'], default 'R'

limit_max

Values higher than this setting are 100% opaque

TYPE:

float in [-inf, inf], default 0.0

limit_method

Algorithm to use to limit channel

- `SINGLE` Single – Limit by single channel.
- `MAX` Max – Limit by maximum of other channels.

TYPE:

enum in ['SINGLE', 'MAX'], default 'SINGLE'

limit_min

Values lower than this setting are 100% keyed

TYPE:

float in [-inf, inf], default 0.0

matte_channel

Channel used to determine matte

- `R` R – Red.
- `G` G – Green.
- `B` B – Blue.

TYPE:

enum in ['R', 'G', 'B'], default 'R'

enum in `[K , G , B]`, default `K`

classmethod `is_registered_node_type()`

True if a registered node type

RETURNS:

Result

RETURN TYPE:

boolean

classmethod `input_template(index)`

Input socket template

PARAMETERS:

index (*int in $[0, inf]$*) – Index

RETURNS:

result

RETURN TYPE:

`NodeInternalSocketTemplate`

classmethod `output_template(index)`

Output socket template

PARAMETERS:

index (*int in $[0, inf]$*) – Index

RETURNS:

result

RETURN TYPE:

`NodeInternalSocketTemplate`

update()

classmethod `bl_ma_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_ma_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`
- `Node.select`

- `Node.type`
- `Node.location`
- `Node.location_absolute`
- `Node.width`
- `Node.height`
- `Node.dimensions`
- `Node.name`
- `Node.label`
- `Node.inputs`
- `Node.outputs`
- `Node.internal_links`
- `Node.parent`
- `Node.warning_propagation`
- `Node.use_custom_color`
- `Node.color`
- `Node.color_tag`
- `Node.show_options`
- `Node.show_preview`
- `Node.hide`
- `Node.mute`
- `Node.show_texture`
- `Node.bl_idname`
- `Node.bl_label`
- `Node.bl_description`
- `Node.bl_icon`
- `Node.bl_static_type`
- `Node.bl_width_default`
- `Node.bl_width_min`
- `Node.bl_width_max`
- `Node.bl_height_default`
- `Node.bl_height_min`
- `Node.bl_height_max`

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`
- `Node.socket_value_update`
- `Node.is_registered_node_type`
- `Node.poll`
- `Node.poll_instance`
- `Node.update`
- `Node.insert_link`
- `Node.init`
- `Node.copy`
- `Node.free`
- `Node.draw_buttons`
- `Node.draw_buttons_ext`
- `Node.draw_label`
- `Node.debug_zone_body_lazy_function_graph`
- `Node.debug_zone_lazy_function_graph`
- `Node.poll`
- `Node.bl_rna_get_subclass`
- `Node.bl_rna_get_subclass_py`
- `NodeInternal.poll`
- `NodeInternal.poll_instance`
- `NodeInternal.update`
- `NodeInternal.draw_buttons`
- `NodeInternal.draw_buttons_ext`
- `NodeInternal.bl_rna_get_subclass`
- `NodeInternal.bl_rna_get_subclass_py`
- `CompositorNode.tag_need_exec`
- `CompositorNode.poll`
- `CompositorNode.update`
- `CompositorNode.bl_rna_get_subclass`
- `CompositorNode.bl_rna_get_subclass_py`

