SceneNode \ Attributes

Every object in your scene can carry extra pieces of information called **attributes** — small named values you attach to an object and read back later. Attributes can also be grouped into named **categories**, which lets one object keep separate attributes that share a name. The nodes on this page let you add, read, check for, and remove attributes, both on their own and inside a category.

Attributes are saved with the object. Reading one never changes it; setting or removing one changes only that single attribute and leaves the rest of the object untouched. There are also event nodes that react the moment an attribute is added, changed, or removed on an object — those are documented on the **Node Attribute Events** page.

Get a node attribute

Get Node Attribute (String)

Reads a named attribute from a scene object and gives it back as a piece of text.

What it does

Every object in your scene can carry extra pieces of information called **attributes**, each one saved under a name you choose. This node looks for the attribute called the name you provide on the object you connect and hands its value back as a piece of text. It only reads the object — it does not change the object or the attribute in any way.

The attribute should already exist and hold text. If it is missing, you get an empty result rather than an error, so it is still worth checking with **Node Has Attribute** first.

Inputs

Port	Type	What to connect
Execute	Trigger	Wire this from the previous node's **Execute** output.

Port	Type	What to connect
SceneNode	Scene node	The scene object you want to read the attribute from.
Attribute Name	Text	The name of the attribute you want to read.

Outputs

Port	Type	What you get
Execute	Trigger	Fires once the node has finished.
SceneNode	Scene node	The same object you connected, passed straight through so you can keep using it.
Attribute Name	Text	The attribute name you provided, passed through.
Result	Text	The text stored in the attribute.

Example

SceneNode input	the `<Forklift>` object
Attribute Name input	`label`
Result output	`Forklift A`

Get Node Attribute (Number)

Reads a named attribute from a scene object and gives it back as a number.

What it does

Every object in your scene can carry extra pieces of information called **attributes**, each one saved under a name you choose. This node looks for the attribute called the name you provide on the object you connect and hands its value back as a number. It only reads the object — it does not change the object or the attribute in any way.

The attribute should already exist and hold a number. If it is missing, or what is stored cannot be read as a number, the node stops with an error instead of giving you a result — so check with **Node Has Attribute** first if you are not sure.

Inputs

Port	Type	What to connect
Execute	Trigger	Wire this from the previous node's **Execute** output.
SceneNode	Scene node	The scene object you want to read the attribute from.
Attribute Name	Text	The name of the attribute you want to read.

Outputs

Port	Type	What you get
Execute	Trigger	Fires once the node has finished.
SceneNode	Scene node	The same object you connected, passed straight through so you can keep using it.
Attribute Name	Text	The attribute name you provided, passed through.
Value	Number	The number stored in the attribute.

Example

SceneNode input	the `<Forklift>` object
Attribute Name input	`maxLoad`
Value output	`1500`

Get Node Attribute (Boolean)

Reads a named attribute from a scene object and gives it back as a true / false value.

What it does

Every object in your scene can carry extra pieces of information called **attributes**, each one saved under a name you choose. This node looks for the attribute called the name you provide on the object you connect and hands its value back as a true / false value. It

only reads the object — it does not change the object or the attribute in any way.

The attribute should already exist and hold a true / false value. If it is missing, or what is stored cannot be read as a true / false value, the node stops with an error instead of giving you a result — so check with **Node Has Attribute** first if you are not sure.

Saved values of `true`, `yes` or `on` are read as true, and `false`, `no` or `off` are read as false.

Inputs

Port	Type	What to connect
Execute	Trigger	Wire this from the previous node's **Execute** output.
SceneNode	Scene node	The scene object you want to read the attribute from.
Attribute Name	Text	The name of the attribute you want to read.

Outputs

Port	Type	What you get
Execute	Trigger	Fires once the node has finished.
SceneNode	Scene node	The same object you connected, passed straight through so you can keep using it.
Attribute Name	Text	The attribute name you provided, passed through.
Result	True / false	The true / false value stored in the attribute.

Example

SceneNode input	the `<Forklift>` object
Attribute Name input	`inspected`
Result output	`true`

Get Node Attribute (SceneNode)

Reads a named attribute from a scene object and gives it back as a scene node.

What it does

Every object in your scene can carry extra pieces of information called **attributes**, each one saved under a name you choose. This node looks for the attribute called the name you provide on the object you connect and hands its value back as a scene node. It only reads the object — it does not change the object or the attribute in any way.

The attribute should already exist and hold a scene node. If it is missing, you get an empty result rather than an error, so it is still worth checking with **Node Has Attribute** first.

Inputs

Port	Type	What to connect
Execute	Trigger	Wire this from the previous node's **Execute** output.
SceneNode	Scene node	The scene object you want to read the attribute from.
Attribute Name	Text	The name of the attribute you want to read.

Outputs

Port	Type	What you get
Execute	Trigger	Fires once the node has finished.
SceneNode	Scene node	The same object you connected, passed straight through so you can keep using it.
Attribute Name	Text	The attribute name you provided, passed through.
Result	Scene node	The scene node stored in the attribute.

Example

SceneNode input	the `<Forklift>` object
Attribute Name input	`assignedTo`

Result output	`<Operator>`

Get Node Attribute (SceneState)

Reads a named attribute from a scene object and gives it back as a scene state.

What it does

Every object in your scene can carry extra pieces of information called **attributes**, each one saved under a name you choose. This node looks for the attribute called the name you provide on the object you connect and hands its value back as a scene state. It only reads the object — it does not change the object or the attribute in any way.

The attribute should already exist and hold a scene state. If it is missing, you get an empty result rather than an error, so it is still worth checking with **Node Has Attribute** first.

Inputs

Port	Type	What to connect
Execute	Trigger	Wire this from the previous node's **Execute** output.
SceneNode	Scene node	The scene object you want to read the attribute from.
Attribute Name	Text	The name of the attribute you want to read.

Outputs

Port	Type	What you get
Execute	Trigger	Fires once the node has finished.
SceneNode	Scene node	The same object you connected, passed straight through so you can keep using it.
Attribute Name	Text	The attribute name you provided, passed through.
Result	Scene state	The scene state stored in the attribute.

Example

SceneNode input	the `<Forklift>` object
Attribute Name input	`savedView`
Result output	`<FrontView>`

Get Node Attribute (Sequence)

Reads a named attribute from a scene object and gives it back as a sequence.

What it does

Every object in your scene can carry extra pieces of information called **attributes**, each one saved under a name you choose. This node looks for the attribute called the name you provide on the object you connect and hands its value back as a sequence. It only reads the object — it does not change the object or the attribute in any way.

The attribute should already exist and hold a sequence. If it is missing, you get an empty result rather than an error, so it is still worth checking with **Node Has Attribute** first.

Inputs

Port	Type	What to connect
Execute	Trigger	Wire this from the previous node's **Execute** output.
SceneNode	Scene node	The scene object you want to read the attribute from.
Attribute Name	Text	The name of the attribute you want to read.

Outputs

Port	Type	What you get
Execute	Trigger	Fires once the node has finished.
SceneNode	Scene node	The same object you connected, passed straight through so you can keep using it.

Port	Type	What you get
Attribute Name	Text	The attribute name you provided, passed through.
Result	Sequence	The sequence stored in the attribute.

Example

SceneNode input	the `<Forklift>` object
Attribute Name input	`startupRoutine`
Result output	`<PowerOn>`

Get Node Attribute (Variable)

Reads a named attribute from a scene object and gives it back as a variable.

What it does

Every object in your scene can carry extra pieces of information called **attributes**, each one saved under a name you choose. This node looks for the attribute called the name you provide on the object you connect and hands its value back as a variable, ready to pass on to any node that works with variables. It only reads the object — it does not change the object or the attribute in any way.

The attribute should already exist. If it is missing, you get an empty result rather than an error, so it is still worth checking with **Node Has Attribute** first.

Inputs

Port	Type	What to connect
Execute	Trigger	Wire this from the previous node's **Execute** output.
SceneNode	Scene node	The scene object you want to read the attribute from.
Attribute Name	Text	The name of the attribute you want to read.

Outputs

Port	Type	What you get
Execute	Trigger	Fires once the node has finished.
SceneNode	Scene node	The same object you connected, passed straight through so you can keep using it.
Attribute Name	Text	The attribute name you provided, passed through.
Result	Variable	The value stored in the attribute, handed back as a variable.

Example

SceneNode input	the `<Forklift>` object
Attribute Name input	`serviceCount`
Result output	a variable holding `12`

Set a node attribute

Set Node Attribute (String)

Saves a piece of text onto a scene object as a named attribute.

What it does

Every object in your scene can carry extra pieces of information called **attributes**, each one saved under a name you choose. This node attaches the attribute called the name you provide to the object you connect and stores a piece of text in it. If the object already has an attribute with that name, its value is replaced; if not, the attribute is created.

The value you provide is also sent straight back out, so you can keep using it further along.

Inputs

Port	Type	What to connect

Execute	Trigger	Wire this from the previous node's **Execute** output.
SceneNode	Scene node	The scene object you want to add or change the attribute on.
Attribute Name	Text	The name to save the attribute under.
Value	Text	The text to store.

Outputs

Port	Type	What you get
Execute	Trigger	Fires once the node has finished.
SceneNode	Scene node	The same object you connected, passed straight through so you can keep using it.
Attribute Name	Text	The attribute name you provided, passed through.
Value	Text	The same value you stored, passed through.

Example

SceneNode input	the `<Forklift>` object
Attribute Name input	`label`
Value input	`Forklift A`

Set Node Attribute (Number)

Saves a number onto a scene object as a named attribute.

What it does

Every object in your scene can carry extra pieces of information called **attributes**, each one saved under a name you choose. This node attaches the attribute called the name you provide to the object you connect and stores a number in it. If the object already has an attribute with that name, its value is replaced; if not, the attribute is created.

The value you provide is also sent straight back out, so you can keep using it further along.

Inputs

Port	Type	What to connect
Execute	Trigger	Wire this from the previous node's **Execute** output.
SceneNode	Scene node	The scene object you want to add or change the attribute on.
Attribute Name	Text	The name to save the attribute under.
Value	Number	The number to store.

Outputs

Port	Type	What you get
Execute	Trigger	Fires once the node has finished.
SceneNode	Scene node	The same object you connected, passed straight through so you can keep using it.
Attribute Name	Text	The attribute name you provided, passed through.
Value	Number	The same value you stored, passed through.

Example

SceneNode input	the `<Forklift>` object
Attribute Name input	`maxLoad`
Value input	`1500`

Set Node Attribute (Boolean)

Saves a true / false value onto a scene object as a named attribute.

What it does

Every object in your scene can carry extra pieces of information called **attributes**, each one saved under a name you choose. This node attaches the attribute called the name you provide to the object you connect and stores a true / false value in it. If the object already has an attribute with that name, its value is replaced; if not, the attribute is created.

The value you provide is also sent straight back out, so you can keep using it further along.

Inputs

Port	Type	What to connect
Execute	Trigger	Wire this from the previous node's **Execute** output.
SceneNode	Scene node	The scene object you want to add or change the attribute on.
Attribute Name	Text	The name to save the attribute under.
Value	True / false	The true / false value to store.

Outputs

Port	Type	What you get
Execute	Trigger	Fires once the node has finished.
SceneNode	Scene node	The same object you connected, passed straight through so you can keep using it.
Attribute Name	Text	The attribute name you provided, passed through.
Value	True / false	The same value you stored, passed through.

Example

SceneNode input	the `<Forklift>` object
Attribute Name input	`inspected`
Value input	`true`

Set Node Attribute (SceneNode)

Saves a scene node onto a scene object as a named attribute.

What it does

Every object in your scene can carry extra pieces of information called **attributes**, each one saved under a name you choose. This node attaches the attribute called the name you provide to the object you connect and stores a scene node in it. If the object already has an attribute with that name, its value is replaced; if not, the attribute is created.

The value you provide is also sent straight back out, so you can keep using it further along.

Inputs

Port	Type	What to connect
Execute	Trigger	Wire this from the previous node's **Execute** output.
SceneNode	Scene node	The scene object you want to add or change the attribute on.
Attribute Name	Text	The name to save the attribute under.
Value	Scene node	The scene node to save a link to.

Outputs

Port	Type	What you get
Execute	Trigger	Fires once the node has finished.
SceneNode	Scene node	The same object you connected, passed straight through so you can keep using it.
Attribute Name	Text	The attribute name you provided, passed through.
Value	Scene node	The same value you stored, passed through.

Example

SceneNode input	the `<Forklift>` object
Attribute Name input	`assignedTo`
Value input	`<Operator>`

Set Node Attribute (SceneState)

Saves a scene state onto a scene object as a named attribute.

What it does

Every object in your scene can carry extra pieces of information called **attributes**, each one saved under a name you choose. This node attaches the attribute called the name you provide to the object you connect and stores a scene state in it. If the object already has an attribute with that name, its value is replaced; if not, the attribute is created.

The value you provide is also sent straight back out, so you can keep using it further along.

Inputs

Port	Type	What to connect
Execute	Trigger	Wire this from the previous node's **Execute** output.
SceneNode	Scene node	The scene object you want to add or change the attribute on.
Attribute Name	Text	The name to save the attribute under.
Value	Scene state	The scene state to save a link to.

Outputs

Port	Type	What you get
Execute	Trigger	Fires once the node has finished.
SceneNode	Scene node	The same object you connected, passed straight through so you can keep using it.

Port	Type	What you get
Attribute Name	Text	The attribute name you provided, passed through.
Value	Scene state	The same value you stored, passed through.

Example

SceneNode input	the `<Forklift>` object
Attribute Name input	`savedView`
Value input	`<FrontView>`

Set Node Attribute (Sequence)

Saves a sequence onto a scene object as a named attribute.

What it does

Every object in your scene can carry extra pieces of information called **attributes**, each one saved under a name you choose. This node attaches the attribute called the name you provide to the object you connect and stores a sequence in it. If the object already has an attribute with that name, its value is replaced; if not, the attribute is created.

The value you provide is also sent straight back out, so you can keep using it further along.

Inputs

Port	Type	What to connect
Execute	Trigger	Wire this from the previous node's **Execute** output.
SceneNode	Scene node	The scene object you want to add or change the attribute on.
Attribute Name	Text	The name to save the attribute under.
Value	Sequence	The sequence to save a link to.

Outputs

Port	Type	What you get
Execute	Trigger	Fires once the node has finished.
SceneNode	Scene node	The same object you connected, passed straight through so you can keep using it.
Attribute Name	Text	The attribute name you provided, passed through.
Value	Sequence	The same value you stored, passed through.

Example

SceneNode input	the `<Forklift>` object
Attribute Name input	`startupRoutine`
Value input	`<PowerOn>`

Set Node Attribute (Variable)

Saves the value held in a variable onto a scene object as a named attribute.

What it does

Every object in your scene can carry extra pieces of information called **attributes**, each one saved under a name you choose. This node attaches the attribute called the name you provide to the object you connect and stores the value held in the variable you give it. If the object already has an attribute with that name, its value is replaced; if not, the attribute is created.

The value you provide is also sent straight back out, so you can keep using it further along.

Inputs

Port	Type	What to connect
Execute	Trigger	Wire this from the previous node's **Execute** output.

Port	Type	What to connect
SceneNode	Scene node	The scene object you want to add or change the attribute on.
Attribute Name	Text	The name to save the attribute under.
Value	Variable	The variable whose value you want to store.

Outputs

Port	Type	What you get
Execute	Trigger	Fires once the node has finished.
SceneNode	Scene node	The same object you connected, passed straight through so you can keep using it.
Attribute Name	Text	The attribute name you provided, passed through.
Value	Variable	The same value you stored, passed through.

Example

SceneNode input	the `<Forklift>` object
Attribute Name input	`serviceCount`
Value input	a variable holding `12`

Check or remove a node attribute

Node Has Attribute

Checks whether a scene object has an attribute with a given name.

What it does

This node looks at the object you connect and tells you whether it already has an attribute saved under the name you provide. You get back **true** if it does and **false** if it does not. It only checks — nothing on the object is changed.

It is handy to run before reading or changing an attribute, to be sure it is there.

Inputs

Port	Type	What to connect
Execute	Trigger	Wire this from the previous node's **Execute** output.
SceneNode	Scene node	The scene object you want to check.
Attribute Name	Text	The name of the attribute to look for.

Outputs

Port	Type	What you get
Execute	Trigger	Fires once the node has finished.
SceneNode	Scene node	The same object you connected, passed straight through so you can keep using it.
Attribute Name	Text	The attribute name you provided, passed through.
Result	True / false	**True** if the attribute exists, otherwise **false**.

Example

SceneNode input	the `<Forklift>` object
Attribute Name input	`inspected`
Result output	**true** — the attribute exists

Remove Node Attribute

Deletes a named attribute from a scene object.

What it does

This node removes the attribute saved under the name you provide from the object you connect. That one attribute and its value are deleted from the object; everything else about the object stays the same.

If the object has no attribute with that name, nothing happens.

Inputs

Port	Type	What to connect
Execute	Trigger	Wire this from the previous node's **Execute** output.
SceneNode	Scene node	The scene object you want to remove the attribute from.
Attribute Name	Text	The name of the attribute to remove.

Outputs

Port	Type	What you get
Execute	Trigger	Fires once the node has finished.
SceneNode	Scene node	The same object you connected, passed straight through so you can keep using it.
Attribute Name	Text	The attribute name you provided, passed through.

Example

SceneNode input	the `<Forklift>` object
Attribute Name input	`tempNote`

Get a node attribute in a category

Get Node Attribute in Category (String)

Reads a named attribute kept in a category on a scene object and gives it back as a piece of text.

What it does

Every object in your scene can carry extra pieces of information called **attributes**, each one saved under a name you choose. Attributes can also be grouped into named **categories**, so a single object can keep separate attributes that share a name in different categories. This node looks for the attribute called the name you provide inside the category you name, on the object you connect and hands its value back as a piece of text. It only reads the object — it does not change the object or the attribute in any way.

The attribute should already exist and hold text. If it is missing, you get an empty result rather than an error, so it is still worth checking with **Node Has Attribute In Category** first.

Inputs

Port	Type	What to connect
Execute	Trigger	Wire this from the previous node's **Execute** output.
SceneNode	Scene node	The scene object you want to read the attribute from.
Attribute Name	Text	The name of the attribute you want to read.
Category	Text	The name of the category the attribute is kept in.

Outputs

Port	Type	What you get
Execute	Trigger	Fires once the node has finished.
SceneNode	Scene node	The same object you connected, passed straight through so you can keep using it.
Attribute Name	Text	The attribute name you provided, passed through.
Category	Text	The category name you provided, passed through.
Result	Text	The text stored in the attribute.

Example

SceneNode input	the `<Forklift>` object
Attribute Name input	`label`
Category input	`Maintenance`
Result output	`Forklift A`

Get Node Attribute in Category (Number)

Reads a named attribute kept in a category on a scene object and gives it back as a number.

What it does

Every object in your scene can carry extra pieces of information called **attributes**, each one saved under a name you choose. Attributes can also be grouped into named **categories**, so a single object can keep separate attributes that share a name in different categories. This node looks for the attribute called the name you provide inside the category you name, on the object you connect and hands its value back as a number. It only reads the object — it does not change the object or the attribute in any way.

The attribute should already exist and hold a number. If it is missing, or what is stored cannot be read as a number, the node stops with an error instead of giving you a result — so check with **Node Has Attribute In Category** first if you are not sure.

Inputs

Port	Type	What to connect
Execute	Trigger	Wire this from the previous node's **Execute** output.
SceneNode	Scene node	The scene object you want to read the attribute from.
Attribute Name	Text	The name of the attribute you want to read.
Category	Text	The name of the category the attribute is kept in.

Outputs

Port	Type	What you get
Execute	Trigger	Fires once the node has finished.
SceneNode	Scene node	The same object you connected, passed straight through so you can keep using it.
Attribute Name	Text	The attribute name you provided, passed through.
Category	Text	The category name you provided, passed through.
Result	Number	The number stored in the attribute.

Example

SceneNode input	the `<Forklift>` object
Attribute Name input	`maxLoad`
Category input	`Maintenance`
Result output	`1500`

Get Node Attribute in Category (Boolean)

Reads a named attribute kept in a category on a scene object and gives it back as a true / false value.

What it does

Every object in your scene can carry extra pieces of information called **attributes**, each one saved under a name you choose. Attributes can also be grouped into named **categories**, so a single object can keep separate attributes that share a name in different categories. This node looks for the attribute called the name you provide inside the category you name, on the object you connect and hands its value back as a true / false value. It only reads the object — it does not change the object or the attribute in any way.

The attribute should already exist and hold a true / false value. If it is missing, or what is stored cannot be read as a true / false value, the node stops with an error instead of giving you a result — so check with **Node Has Attribute In Category** first if you are not sure.

Saved values of `true`, `yes` or `on` are read as true, and `false`, `no` or `off` are read as false.

Inputs

Port	Type	What to connect
Execute	Trigger	Wire this from the previous node's **Execute** output.
SceneNode	Scene node	The scene object you want to read the attribute from.
Attribute Name	Text	The name of the attribute you want to read.
Category	Text	The name of the category the attribute is kept in.

Outputs

Port	Type	What you get

Execute	Trigger	Fires once the node has finished.
SceneNode	Scene node	The same object you connected, passed straight through so you can keep using it.
Attribute Name	Text	The attribute name you provided, passed through.
Category	Text	The category name you provided, passed through.
Result	True / false	The true / false value stored in the attribute.

Example

SceneNode input	the `<Forklift>` object
Attribute Name input	`inspected`
Category input	`Maintenance`
Result output	`true`

Get Node Attribute in Category (SceneNode)

Reads a named attribute kept in a category on a scene object and gives it back as a scene node.

What it does

Every object in your scene can carry extra pieces of information called **attributes**, each one saved under a name you choose. Attributes can also be grouped into named **categories**, so a single object can keep separate attributes that share a name in different categories. This node looks for the attribute called the name you provide inside the category you name, on the object you connect and hands its value back as a scene node. It only reads the object — it does not change the object or the attribute in any way.

The attribute should already exist and hold a scene node. If it is missing, you get an empty result rather than an error, so it is still worth checking with **Node Has Attribute In Category** first.

Inputs

Port	Type	What to connect
Execute	Trigger	Wire this from the previous node's **Execute** output.
SceneNode	Scene node	The scene object you want to read the attribute from.
Attribute Name	Text	The name of the attribute you want to read.
Category	Text	The name of the category the attribute is kept in.

Outputs

Port	Type	What you get
Execute	Trigger	Fires once the node has finished.
SceneNode	Scene node	The same object you connected, passed straight through so you can keep using it.
Attribute Name	Text	The attribute name you provided, passed through.
Category	Text	The category name you provided, passed through.
Result	Scene node	The scene node stored in the attribute.

Example

SceneNode input	the `<Forklift>` object
Attribute Name input	`assignedTo`
Category input	`Maintenance`
Result output	`<Operator>`

Get Node Attribute in Category (SceneState)

Reads a named attribute kept in a category on a scene object and gives it back as a scene state.

What it does

Every object in your scene can carry extra pieces of information called **attributes**, each one saved under a name you choose. Attributes can also be grouped into named **categories**, so a single object can keep separate attributes that share a name in different categories. This node looks for the attribute called the name you provide inside the category you name, on the object you connect and hands its value back as a scene state. It only reads the object — it does not change the object or the attribute in any way.

The attribute should already exist and hold a scene state. If it is missing, you get an empty result rather than an error, so it is still worth checking with **Node Has Attribute In Category** first.

Inputs

Port	Type	What to connect
Execute	Trigger	Wire this from the previous node's **Execute** output.
SceneNode	Scene node	The scene object you want to read the attribute from.
Attribute Name	Text	The name of the attribute you want to read.
Category	Text	The name of the category the attribute is kept in.

Outputs

Port	Type	What you get
Execute	Trigger	Fires once the node has finished.
SceneNode	Scene node	The same object you connected, passed straight through so you can keep using it.
Attribute Name	Text	The attribute name you provided, passed through.
Category	Text	The category name you provided, passed through.

Port	Type	What you get
Result	Scene state	The scene state stored in the attribute.

Example

SceneNode input	the `<Forklift>` object
Attribute Name input	`savedView`
Category input	`Maintenance`
Result output	`<FrontView>`

Get Node Attribute in Category (Sequence)

Reads a named attribute kept in a category on a scene object and gives it back as a sequence.

What it does

Every object in your scene can carry extra pieces of information called **attributes**, each one saved under a name you choose. Attributes can also be grouped into named **categories**, so a single object can keep separate attributes that share a name in different categories. This node looks for the attribute called the name you provide inside the category you name, on the object you connect and hands its value back as a sequence. It only reads the object — it does not change the object or the attribute in any way.

The attribute should already exist and hold a sequence. If it is missing, you get an empty result rather than an error, so it is still worth checking with **Node Has Attribute In Category** first.

Inputs

Port	Type	What to connect
Execute	Trigger	Wire this from the previous node's **Execute** output.

Port	Type	What to connect
SceneNode	Scene node	The scene object you want to read the attribute from.
Attribute Name	Text	The name of the attribute you want to read.
Category	Text	The name of the category the attribute is kept in.

Outputs

Port	Type	What you get
Execute	Trigger	Fires once the node has finished.
SceneNode	Scene node	The same object you connected, passed straight through so you can keep using it.
Attribute Name	Text	The attribute name you provided, passed through.
Category	Text	The category name you provided, passed through.
Result	Sequence	The sequence stored in the attribute.

Example

SceneNode input	the `<Forklift>` object
Attribute Name input	`startupRoutine`
Category input	`Maintenance`
Result output	`<PowerOn>`

Get Node Attribute in Category (Variable)

Reads a named attribute kept in a category on a scene object and gives it back as a variable.

What it does

Every object in your scene can carry extra pieces of information called **attributes**, each one saved under a name you choose. Attributes can also be grouped into named **categories**, so a single object can keep separate attributes that share a name in different categories. This node looks for the attribute called the name you provide inside the category you name, on the object you connect, and hands its value back as a variable. It only reads the object — it does not change the object or the attribute in any way.

The attribute should already exist. If it is missing, you get an empty result rather than an error, so it is still worth checking with **Node Has Attribute In Category** first.

Inputs

Port	Type	What to connect
Execute	Trigger	Wire this from the previous node's **Execute** output.
SceneNode	Scene node	The scene object you want to read the attribute from.
Attribute Name	Text	The name of the attribute you want to read.
Category	Text	The name of the category the attribute is kept in.

Outputs

Port	Type	What you get
Execute	Trigger	Fires once the node has finished.
SceneNode	Scene node	The same object you connected, passed straight through so you can keep using it.
Attribute Name	Text	The attribute name you provided, passed through.
Category	Text	The category name you provided, passed through.
Result	Variable	The value stored in the attribute, handed back as a variable.

Example

SceneNode input	the `<Forklift>` object
Attribute Name input	`serviceCount`
Category input	`Maintenance`
Result output	a variable holding `12`

Set a node attribute in a category

Set Node Attribute in Category (String)

Saves a piece of text into a category on a scene object as a named attribute.

What it does

Every object in your scene can carry extra pieces of information called **attributes**, each one saved under a name you choose. Attributes can also be grouped into named **categories**, so a single object can keep separate attributes that share a name in different categories. This node attaches the attribute called the name you provide, inside the category you name, to the object you connect and stores a piece of text in it. If that category already has an attribute with that name, its value is replaced; if not, the attribute is created.

The value you provide is also sent straight back out, so you can keep using it further along.

Inputs

Port	Type	What to connect
Execute	Trigger	Wire this from the previous node's **Execute** output.
SceneNode	Scene node	The scene object you want to add or change the attribute on.
Attribute Name	Text	The name to save the attribute under.

Port	Type	What to connect
Category	Text	The name of the category to keep the attribute in.
Value	Text	The text to store.

Outputs

Port	Type	What you get
Execute	Trigger	Fires once the node has finished.
SceneNode	Scene node	The same object you connected, passed straight through so you can keep using it.
Attribute Name	Text	The attribute name you provided, passed through.
Category	Text	The category name you provided, passed through.
Value	Text	The same value you stored, passed through.

Example

SceneNode input	the `<Forklift>` object
Attribute Name input	`label`
Category input	`Maintenance`
Value input	`Forklift A`

Set Node Attribute in Category (Number)

Saves a number into a category on a scene object as a named attribute.

What it does

Every object in your scene can carry extra pieces of information called **attributes**, each one saved under a name you choose. Attributes can also be grouped into named **categories**, so a single object can keep separate attributes that share a name in different

categories. This node attaches the attribute called the name you provide, inside the category you name, to the object you connect and stores a number in it. If that category already has an attribute with that name, its value is replaced; if not, the attribute is created.

The value you provide is also sent straight back out, so you can keep using it further along.

Inputs

Port	Type	What to connect
Execute	Trigger	Wire this from the previous node's **Execute** output.
SceneNode	Scene node	The scene object you want to add or change the attribute on.
Attribute Name	Text	The name to save the attribute under.
Category	Text	The name of the category to keep the attribute in.
Value	Number	The number to store.

Outputs

Port	Type	What you get
Execute	Trigger	Fires once the node has finished.
SceneNode	Scene node	The same object you connected, passed straight through so you can keep using it.
Attribute Name	Text	The attribute name you provided, passed through.
Category	Text	The category name you provided, passed through.
Value	Number	The same value you stored, passed through.

Example

SceneNode input	the `<Forklift>` object

Attribute Name input	`maxLoad`
Category input	`Maintenance`
Value input	`1500`

Set Node Attribute in Category (Boolean)

Saves a true / false value into a category on a scene object as a named attribute.

What it does

Every object in your scene can carry extra pieces of information called **attributes**, each one saved under a name you choose. Attributes can also be grouped into named **categories**, so a single object can keep separate attributes that share a name in different categories. This node attaches the attribute called the name you provide, inside the category you name, to the object you connect and stores a true / false value in it. If that category already has an attribute with that name, its value is replaced; if not, the attribute is created.

The value you provide is also sent straight back out, so you can keep using it further along.

Inputs

Port	Type	What to connect
Execute	Trigger	Wire this from the previous node's **Execute** output.
SceneNode	Scene node	The scene object you want to add or change the attribute on.
Attribute Name	Text	The name to save the attribute under.
Category	Text	The name of the category to keep the attribute in.
Value	True / false	The true / false value to store.

Outputs

Port	Type	What you get
Execute	Trigger	Fires once the node has finished.
SceneNode	Scene node	The same object you connected, passed straight through so you can keep using it.
Attribute Name	Text	The attribute name you provided, passed through.
Category	Text	The category name you provided, passed through.
Value	True / false	The same value you stored, passed through.

Example

SceneNode input	the `<Forklift>` object
Attribute Name input	`inspected`
Category input	`Maintenance`
Value input	`true`

Set Node Attribute in Category (SceneNode)

Saves a scene node into a category on a scene object as a named attribute.

What it does

Every object in your scene can carry extra pieces of information called **attributes**, each one saved under a name you choose. Attributes can also be grouped into named **categories**, so a single object can keep separate attributes that share a name in different categories. This node attaches the attribute called the name you provide, inside the category you name, to the object you connect and stores a scene node in it. If that category already has an attribute with that name, its value is replaced; if not, the attribute is created.

The value you provide is also sent straight back out, so you can keep using it further along.

Inputs

Port	Type	What to connect
Execute	Trigger	Wire this from the previous node's **Execute** output.
SceneNode	Scene node	The scene object you want to add or change the attribute on.
Attribute Name	Text	The name to save the attribute under.
Category	Text	The name of the category to keep the attribute in.
Value	Scene node	The scene node to save a link to.

Outputs

Port	Type	What you get
Execute	Trigger	Fires once the node has finished.
SceneNode	Scene node	The same object you connected, passed straight through so you can keep using it.
Attribute Name	Text	The attribute name you provided, passed through.
Category	Text	The category name you provided, passed through.
Value	Scene node	The same value you stored, passed through.

Example

SceneNode input	the `<Forklift>` object
Attribute Name input	`assignedTo`
Category input	`Maintenance`
Value input	`<Operator>`

Set Node Attribute in Category (SceneState)

Saves a scene state into a category on a scene object as a named attribute.

What it does

Every object in your scene can carry extra pieces of information called **attributes**, each one saved under a name you choose. Attributes can also be grouped into named **categories**, so a single object can keep separate attributes that share a name in different categories. This node attaches the attribute called the name you provide, inside the category you name, to the object you connect and stores a scene state in it. If that category already has an attribute with that name, its value is replaced; if not, the attribute is created.

The value you provide is also sent straight back out, so you can keep using it further along.

Inputs

Port	Type	What to connect
Execute	Trigger	Wire this from the previous node's **Execute** output.
SceneNode	Scene node	The scene object you want to add or change the attribute on.
Attribute Name	Text	The name to save the attribute under.
Category	Text	The name of the category to keep the attribute in.
Value	Scene state	The scene state to save a link to.

Outputs

Port	Type	What you get
Execute	Trigger	Fires once the node has finished.

Port	Type	What you get
SceneNode	Scene node	The same object you connected, passed straight through so you can keep using it.
Attribute Name	Text	The attribute name you provided, passed through.
Category	Text	The category name you provided, passed through.
Value	Scene state	The same value you stored, passed through.

Example

SceneNode input	the `<Forklift>` object
Attribute Name input	`savedView`
Category input	`Maintenance`
Value input	`<FrontView>`

Set Node Attribute in Category (Sequence)

Saves a sequence into a category on a scene object as a named attribute.

What it does

Every object in your scene can carry extra pieces of information called **attributes**, each one saved under a name you choose. Attributes can also be grouped into named **categories**, so a single object can keep separate attributes that share a name in different categories. This node attaches the attribute called the name you provide, inside the category you name, to the object you connect and stores a sequence in it. If that category already has an attribute with that name, its value is replaced; if not, the attribute is created.

The value you provide is also sent straight back out, so you can keep using it further along.

Inputs

Port	Type	What to connect
Execute	Trigger	Wire this from the previous node's **Execute** output.
SceneNode	Scene node	The scene object you want to add or change the attribute on.
Attribute Name	Text	The name to save the attribute under.
Category	Text	The name of the category to keep the attribute in.
Value	Sequence	The sequence to save a link to.

Outputs

Port	Type	What you get
Execute	Trigger	Fires once the node has finished.
SceneNode	Scene node	The same object you connected, passed straight through so you can keep using it.
Attribute Name	Text	The attribute name you provided, passed through.
Category	Text	The category name you provided, passed through.
Value	Sequence	The same value you stored, passed through.

Example

SceneNode input	the `<Forklift>` object
Attribute Name input	`startupRoutine`
Category input	`Maintenance`
Value input	`<PowerOn>`

Set Node Attribute in Category (Variable)

Saves the value held in a variable into a category on a scene object as a named attribute.

What it does

Every object in your scene can carry extra pieces of information called **attributes**, each one saved under a name you choose. Attributes can also be grouped into named **categories**, so a single object can keep separate attributes that share a name in different categories. This node attaches the attribute called the name you provide, inside the category you name, to the object you connect and stores the value held in the variable you give it. If that category already has an attribute with that name, its value is replaced; if not, the attribute is created.

The value you provide is also sent straight back out, so you can keep using it further along.

Inputs

Port	Type	What to connect
Execute	Trigger	Wire this from the previous node's **Execute** output.
SceneNode	Scene node	The scene object you want to add or change the attribute on.
Attribute Name	Text	The name to save the attribute under.
Category	Text	The name of the category to keep the attribute in.
Value	Variable	The variable whose value you want to store.

Outputs

Port	Type	What you get
Execute	Trigger	Fires once the node has finished.
SceneNode	Scene node	The same object you connected, passed straight through so you can keep using it.
Attribute Name	Text	The attribute name you provided, passed through.

Port	Type	What you get
Category	Text	The category name you provided, passed through.
Value	Variable	The same value you stored, passed through.

Example

SceneNode input	the `<Forklift>` object
Attribute Name input	`serviceCount`
Category input	`Maintenance`
Value input	a variable holding `12`

Check, list, or remove attributes in a category

Node Has Attribute In Category

Checks whether a scene object has an attribute with a given name inside a category.

What it does

This node looks at the object you connect and tells you whether it already has an attribute saved under the name you provide inside the category you name. You get back **true** if it does and **false** if it does not. It only checks — nothing on the object is changed.

Inputs

Port	Type	What to connect
Execute	Trigger	Wire this from the previous node's **Execute** output.
SceneNode	Scene node	The scene object you want to check.

Port	Type	What to connect
Attribute Name	Text	The name of the attribute to look for.
Category	Text	The name of the category to look in.

Outputs

Port	Type	What you get
Execute	Trigger	Fires once the node has finished.
SceneNode	Scene node	The same object you connected, passed straight through so you can keep using it.
Attribute Name	Text	The attribute name you provided, passed through.
Category	Text	The category name you provided, passed through.
Result	True / false	**True** if the attribute exists in that category, otherwise **false**.

Example

SceneNode input	the `<Forklift>` object
Attribute Name input	`inspected`
Category input	`Maintenance`
Result output	**true** — the attribute exists

Get Node Attribute Category Names

Lists the names of the attributes kept inside a category on a scene object.

What it does

Attributes on an object can be grouped into named **categories**. This node looks at the object you connect and gives you back the names of the attributes that are stored inside the category you name. The names come back as a list of separate pieces of text, so you can connect a node that steps through a list to handle them one at a time. It only reads

the object — nothing is changed.

Inputs

Port	Type	What to connect
Execute	Trigger	Wire this from the previous node's **Execute** output.
SceneNode	Scene node	The scene object you want to look at.
Category	Text	The name of the category whose attribute names you want.

Outputs

Port	Type	What you get
Execute	Trigger	Fires once the node has finished.
SceneNode	Scene node	The same object you connected, passed straight through so you can keep using it.
Category	Text	The category name you provided, passed through.
Result	Text	The names of the attributes found in that category, returned as a list of separate pieces of text. Connect a node that steps through a list to read them one by one.

Example

SceneNode input	the `<Forklift>` object
Category input	`Maintenance`
Result output	a list of names — e.g. `inspected`, `maxLoad`, `assignedTo`

Remove Node Attribute From Category

Deletes a named attribute that is kept in a category on a scene object.

What it does

This node removes the attribute saved under the name you provide, inside the category you name, from the object you connect. That one attribute and its value are deleted from that category; everything else about the object stays the same.

If no attribute with that name exists in the category, nothing happens.

Inputs

Port	Type	What to connect
Execute	Trigger	Wire this from the previous node's **Execute** output.
SceneNode	Scene node	The scene object you want to remove the attribute from.
Attribute Name	Text	The name of the attribute to remove.
Category	Text	The name of the category to remove it from.

Outputs

Port	Type	What you get
Execute	Trigger	Fires once the node has finished.
SceneNode	Scene node	The same object you connected, passed straight through so you can keep using it.
Attribute Name	Text	The attribute name you provided, passed through.
Category	Text	The category name you provided, passed through.

Example

SceneNode input	the `<Forklift>` object
Attribute Name input	`tempNote`
Category input	`Maintenance`

Revision #3
Created 10 June 2026 12:44:38 by Rafat

Updated 11 June 2026 13:57:28 by Rafat