Device / Attributes

A **device attribute** is a named value saved on the **device** the experience is running on — a PC, a phone or tablet, a standalone VR headset, and so on. You give each value a name (for example `language`) and store a piece of text under it. These values belong to the device itself: they stay on it after the scene or app is closed and reopened (even across different experiences), they are *not* tied to whoever is signed in, and the person using the device can also see and change them in the viewer's Settings.

That makes device attributes the right tool for remembering something on one particular device for next time — a chosen language, measurement units, the name of a kiosk or station, and the like. The examples on this page all use the same running example: a setting named `language` holding the text `English`.

What's on this page

- **Set Device Attribute (String)** — save a piece of text on the device under a name.
- **Get Device Attribute (String)** — read back the text saved under a name.
- **Device Has Attribute** — check whether the device has a value saved under a name (true or false).
- **Remove Device Attribute** — delete a saved value from the device.

Attributes

These four nodes are the complete set for working with a device's saved values — store one, read it back, check whether it exists, and remove it. None of them is tied to who is signed in, and every value stays on the device for next time.

 Set Device Attribute (String)

Saves a piece of text on the device under a name you choose, so you can read it back later — even after the experience is closed and reopened.

What it does

This node stores a bit of text on the device the experience is running on — a PC, a phone or tablet, a standalone VR headset, and so on. You give it a name (such as `language`) and the text to store (such as `English`), and the node remembers that value on the device. If a value with that name doesn't exist yet, it creates one; if it already exists, the new text replaces the old.

The value stays on the device, so it's still there next time the experience runs — it's a good way to remember a setting for next time. These values belong to the device itself, not to whoever is signed in, so they aren't tied to a particular person. They also aren't hidden: the person using the device can see and change them in the viewer's Settings.

Inputs

Port	Type	What to connect
Execute	Trigger	Wire this from the previous node's Execute output.
Attribute Name	Text	The name to save the text under, such as `language`. You can pick a name from the drop-down list of names already known on the device, or simply type a brand-new name.
Value	Text	The text you want to store, such as `English`. If a value with this name already exists, this replaces it.

Outputs

Port	Type	What you get
Execute	Trigger	Fires once the node has finished saving the value.
Attribute Name	Text	The same attribute name you gave, passed straight through (handy for wiring into the next node).

Port	Type	What you get
Value	Text	The same text you stored, passed straight through.

Example

Attribute Name input	`language`
Value input	`English`
Attribute Name output	`language`
Value output	`English`

Tips

- Use this to remember a setting on the device for next time, such as the chosen `language` or measurement `units`.
- Saving again under the same name simply replaces the old text — you don’t need to remove the old value first.
- Pair it with **Get Device Attribute (String)** to read the value back later.

Get Device Attribute (String)

Reads back a piece of text that was saved on this device under a name you choose.

What it does

A device attribute is a value stored on the device the experience is running on — a PC, phone, tablet, or VR headset. You give this node the name of the attribute you want, and it hands back the text currently saved under that name. These values live on the device itself, not with whoever is logged in, and they stay put after the scene or app is closed and reopened — so this node is the way to read something you remembered on this device for next time.

This node only reads; it never changes the saved value. If nothing has been saved under that name yet, you simply get back empty text — the node carries on without any error.

Inputs

Port	Type	What to connect

Execute	Trigger	Wire this from the previous node's Execute output.
Attribute Name	Text	The name of the value you want to read. Pick one from the drop-down, which lists names already saved on the device, or type a new name yourself. For example, `language`.

Outputs

Port	Type	What you get
Execute	Trigger	Fires once the node has finished.
Attribute Name	Text	The same attribute name you gave, passed straight through (handy for wiring into the next node).
Result	Text	The text saved under that name. If nothing is saved under that name, you get back empty text.

Example

Attribute Name input	`language`
Attribute Name output	`language`
Result output	`English`

Tips

- Getting empty text back usually means nothing has been saved under that name yet — on this device, set it first with **Set Device Attribute (String)**, or check with **Device Has Attribute** before reading.
- These values are saved per device and are not tied to who is logged in. The person using the device can also see and change them in the viewer's Settings.

Device Has Attribute

Checks whether this device already has a saved setting with a given name, and tells you yes or no.

What it does

This node looks on the device the experience is running on — a PC, phone, tablet, VR headset, and so on — for a saved setting with the name you give it. If a setting with that name exists, the **Result** comes back as `true`; if there is no setting with that name, it comes back as `false`. Either way the node simply reports what it found and does not change or remove anything.

These device settings live on the device itself. They are not tied to whoever is signed in, and they stay on the device after the scene or app is closed and reopened — even across different experiences. (The person using the device can also view and change them in the viewer's Settings.) That makes this node handy for asking "has this device been set up before?" so you can decide what to do next.

Inputs

Port	Type	What to connect
Execute	Trigger	Wire this from the previous node's Execute output.
Attribute Name	Text	The name of the setting to look for, such as `language`. The field offers a drop-down of names already saved on the device, and you can also type a brand-new name that isn't in the list.

Outputs

Port	Type	What you get
Execute	Trigger	Fires once the node has finished — whether the setting was found or not.
Attribute Name	Text	The same attribute name you gave, passed straight through (handy for wiring into the next node).
Result	True / false	`true` if the device has a setting with that name, or `false` if it does not.

Example

Attribute Name input	`language`
Attribute Name output	`language`
Result output	`true` (this device has a `language` setting saved)

Tips

- Use this as a quick “first time on this device?” check: if **Result** is `false`, set up a default; if it’s `true`, just read the saved value.
- This node only checks for the setting — it never changes or removes it. Use Set Device Attribute to store a value and Remove Device Attribute to clear one.

Remove Device Attribute

Deletes a saved setting from the device the experience is running on.

What it does

A device attribute is a small named value kept on the device itself — the PC, phone, tablet, or VR headset that is running the experience. These values stay on the device and are saved for next time, so they are still there after the app or scene is closed and reopened. They aren’t tied to who is logged in, and the person using the device can see and change them in the viewer’s Settings. This node removes one of those saved values by name.

Give it the name of the attribute you want to remove. If a value with that name is saved on the device, it is deleted; if no value with that name exists, the node simply does nothing and carries on — no error. The name you gave is passed straight through on the output, so you can keep wiring the flow forward.

Inputs

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

Port	Type	What to connect
Attribute Name	Text	The name of the device value to remove, such as `language`. You can pick a name from the drop-down list of values already saved on the device, or type a name yourself.

Outputs

Port	Type	What you get
Execute	Trigger	Fires once the node has finished.
Attribute Name	Text	The same attribute name you gave, passed straight through (handy for wiring into the next node).

Example

Attribute Name input	`language`
Attribute Name output	`language`

Tips

- It’s safe to remove a name that may not be there — if nothing is saved under that name, the node just moves on without an error.
- Removing a value only affects this device. Saved values on other devices are not touched.

Revision #2
Created 2026-06-09 09:03:22 UTC by Rafat
Updated 2026-10-25 09:24:14 UTC by Rafat