# Overlap

The nodes on this page are **states**. Each one keeps an eye on part of your scene and continuously reports a simple true/false answer about it — for example “are these two objects touching?” or “is the viewer standing inside this area?” A state is not an action you run once and finish; it is a live condition the scene keeps checking the whole time it is playing.

Every one of these nodes gives you the same three things to work with:

- **OnTrue** — a trigger that fires the moment the answer becomes true.
- **OnFalse** — a trigger that fires the moment the answer becomes false.
- **Output Boolean** — the current true/false value, ready to read at any time or to send into a **Branch** node.

The two triggers fire only when the answer *changes* — not over and over while it stays the same. These nodes only watch and report; they never move, assemble, or change anything in your scene.

This page covers the **overlap** states. For states that follow a part through a **VR assembly** — whether it can be put on or taken off, or is fully assembled — see the **Assembly** page.

---

# Overlap

These nodes report whether things are sharing the same space in your scene. **Overlap** means two volumes intersect — they are touching or passing through one another. One node compares two objects you choose; the other compares an object against the **viewer** (the person moving through the scene in VR).

## Nodes Overlap

Watches two objects in your scene and reports whether they are overlapping in space — that is, whether their 3D shapes share the same area.

# What it does

You give this node two objects from your scene. It keeps watching them while your scene runs and answers one simple yes/no question: are these two objects overlapping right now? The answer is true whenever the two objects' 3D volumes share the same space, and false whenever they are apart.

This node only watches and reports — it never moves, changes, or touches the objects themselves. It gives you a true/false answer plus two triggers so you can make something happen the moment the two objects start touching, or the moment they come apart.

# Inputs

| Port | Type | What to connect |
|---|---|---|
| **SceneNode A** | Scene node | The first object to check — for example a `Gear`. It must be a different object from SceneNode B. The viewer's start position cannot be used here; to test the viewer against an object, use the User Overlap Node instead. |
| **SceneNode B** | Scene node | The second object to check — for example a `Housing`. It must be a different object from SceneNode A. The viewer's start position cannot be used here either. |

# Outputs

| Port | Type | What you get |
|---|---|---|
| **OnTrue** | Trigger | Fires once the instant the two objects start overlapping. Wire it to whatever should happen the moment they come together. |
| **OnFalse** | Trigger | Fires once the instant the two objects stop overlapping and move apart. Wire it to whatever should happen the moment they separate. |

| Port | Type | What you get |
| --- | --- | --- |
| **Output Boolean** | True / false | The current answer: true while the two objects are overlapping, false while they are apart. You can read it at any time or feed it into a Branch node. |

## Example

| | |
| --- | --- |
| **SceneNode A** input | `Gear` |
| **SceneNode B** input | `Housing` |
| **OnTrue** output | Fires the moment the `Gear` slides into the `Housing` and they begin to overlap |
| **OnFalse** output | Fires the moment the `Gear` is pulled back out and the two no longer overlap |
| **Output Boolean** output | `true` while the `Gear` is inside the `Housing` |

## Tips

- OnTrue and OnFalse each fire just once, at the moment the answer changes — they do not keep firing while the two objects stay overlapping. If you need to act on the ongoing state rather than the moment it changes, read the Output Boolean instead.
- The two objects must be different. Pick two separate scene objects for SceneNode A and SceneNode B.
- To check whether the viewer is overlapping an object, use the User Overlap Node — the viewer’s start position is not accepted here.

# User Overlap Node

This node reports whether the person viewing the scene is currently sharing the same space as a chosen object — in other words, whether the viewer is standing in or passing through it.

## What it does

You give it one object from your scene. While the scene runs, the node keeps watching the viewer’s position and answers a single live question: is the viewer overlapping that object right now? The answer is `true` while the viewer’s space and the object’s space

share the same area, and `false` the rest of the time.

This node only watches and reports — it never moves, hides, or changes the object or the viewer in any way. You decide what happens by wiring its outputs to other nodes. The two triggers fire at the moment the answer changes: one the instant the viewer steps into the object, the other the instant they step back out. The true / false output always holds the current answer, so you can read it whenever you like.

# Inputs

| Port | Type | What to connect |
|---|---|---|
| **SceneNode** | Scene node | The object in your scene you want to test the viewer against — for example a doorway, a safety zone, or a part such as `Housing`. The node reports `true` while the viewer is sharing the same space as this object. (The viewer start position cannot be used here.) |

# Outputs

| Port | Type | What you get |
|---|---|---|
| **OnTrue** | Trigger | Fires once the moment the viewer starts overlapping the object. Wire it to whatever should happen when the viewer steps in — play a sound, show a message, open a door. |
| **OnFalse** | Trigger | Fires once the moment the viewer stops overlapping the object. Wire it to whatever should happen when the viewer leaves — the matching “step out” reaction. |
| **Output Boolean** | True / false | The current answer as a `true` / `false` value: `true` while the viewer is inside the object, `false` otherwise. Read it directly or send it into a Branch node to choose between two paths. |

# Example

| | |
|---|---|
| **SceneNode** input | `RestrictedZone` |
| **OnTrue** output | Fires when the viewer walks into `RestrictedZone` — wire it to a warning sound. |
| **OnFalse** output | Fires when the viewer leaves `RestrictedZone` — wire it to stop the warning. |
| **Output Boolean** output | `true` while the viewer is inside the zone, `false` when they are outside |

# Tips

- Use **OnTrue** and **OnFalse** as a matched pair when you want one reaction as the viewer enters and another as they leave — for example showing and then hiding a hint panel.
- When you only need to check the answer at a specific point (rather than react the instant it changes), feed **Output Boolean** into a Branch node instead.
- Pick an object whose space is big enough to comfortably contain the viewer, such as a room or a marked floor area, so the overlap is easy to trigger.

---

Revision #1
Created 9 June 2026 12:52:40 by Rafat
Updated 9 June 2026 12:53:55 by Rafat