Performance

These nodes help you check how smoothly an experience runs on a device. **Get FPS** reads the current frame rate, and **Record FPS To CSV** saves a reading to a spreadsheet file or sends it to a server, so you can compare devices or find heavy moments in a scene.

What's on this page

- **Get FPS** — Tells you how smoothly your scene is running right now, as a frame rate in frames per second (FPS).
- **Record FPS To CSV** — Saves one frame-rate reading, together with the scene name, platform and time, either to a spreadsheet file (CSV) or to a web server.



Get FPS

Tells you how smoothly your scene is running right now, as a frame rate in frames per second (FPS).

What it does

When this node runs, it reads the viewer's current frame rate — how many times per second the picture is being redrawn. The number is a short running average rather than a single frame, so one brief hiccup won't throw it off. Higher is smoother: around `60` or more feels fluid on a desktop, while a value that drops into the 20s or below usually means the scene is too heavy for the device.

It only reads the value; it doesn't change anything in your scene. Each time the node runs you get a fresh reading, so run it again whenever you want an up-to-date number.

Inputs

Port	Type	What to connect

Execute	Trigger	Wire this from the previous node's **Execute** output.

Outputs

Port	Type	What you get
Execute	Trigger	Fires once the node has finished.
FPS	Number	The current frame rate, in frames per second, at the moment the node ran (for example `72.4`). It can include decimals.

Example

FPS output	`72.4`

Tips

- Wire the **FPS** output into **Record FPS To CSV** to keep a log of how the scene performs over time.
- The frame rate changes constantly as the user moves around. To watch it during a session, run this node repeatedly — for example every few seconds using a **Delay** node — rather than once at the start.

Record FPS To CSV

Saves one frame-rate reading, together with the scene name, platform and time, either to a spreadsheet file (CSV) or to a web server.

What it does

Each time this node runs, it records **one reading**: the FPS value you connect, plus the viewer version, the scene's name, the date and time, and the platform the viewer is running on. It does not keep recording on its own and has no duration setting — one run writes one record. To log performance over a whole session, run it repeatedly (for example every few seconds), and each run adds another record.

Where the reading goes depends on what you put in **File Path or URL**:

- **A file path** (for example `C:\SimLab\Reports\fps-log.csv`) — the reading is added as a new line at the end of that CSV file, which opens in Excel or any spreadsheet program. If the file doesn't exist yet, it is created first (along with any missing folders) and given a header line. Earlier lines are never changed or removed, so readings from many sessions, dates and devices build up in the same file.
- **Left empty** — the node picks the file for you: a CSV named after your scene (for example `Factory_Training.csv`), saved in the same folder as the scene file you opened. Running the same scene again keeps adding to that same file.
- **A web address** starting with `http://` or `https://` — instead of writing a file, the node sends the reading to that server (as a JSON object with the fields `Version`, `Timestamp`, `ModelName`, `Platform` and `FPS`) and waits for the server to answer. Use this to collect readings from many headsets in one place.

The CSV file has these columns, in this order: **Version** (the viewer version), **ModelName** (the scene's name), **Timestamp** (date and time, such as `2026-10-22 14:05:31`), **Platform** (such as `Windows` or `Android`) and **FPS** (rounded to two decimals). The header line is written only once, when the file is first created.

If the file can't be written (for example the folder is read-only), or the server rejects the reading or can't be reached, **Success** comes back false. The node doesn't say why, so check the path or address.

Inputs

Port	Type	What to connect
Execute	Trigger	Wire this from the previous node's **Execute** output.
File Path or URL	Text	Where to save the reading: the full path of a CSV file, or a web address starting with `http://` or `https://`. Leave it empty to save to a CSV named after your scene, next to the scene file.
FPS	Number	The frame rate to record. Usually wired from the **FPS** output of **Get FPS**, run just before this node.

Outputs

Port	Type	What you get

Execute	Trigger	For a file, fires once the line has been written (or the write has failed). For a web address, fires only once the server has answered or the attempt has failed — the rest of your scene keeps running meanwhile. It fires either way, so check **Success**.
Success	True / false	True when the reading was saved to the file or accepted by the server; false otherwise.
Link	Text	Where the reading went: the full path of the CSV file (including the one chosen for you when **File Path or URL** was empty), or the web address it was sent to. Handy for showing the user where to find the log.
Finished	True / false	Always the same value as **Success**: true when the reading was saved, false when it wasn't.
Model Name	Text	The name recorded for your scene — the scene file's name without its extension (for example `Factory_Training`).
Platform	Text	The platform the viewer is running on, as recorded, such as `Windows`, `Mac`, `Android` or `IOS`.

Example

File Path or URL input	(empty)
FPS input	`72.4`
Success output	`true`
Link output	`D:\Projects\Factory_Training\Factory_Training.csv`
Finished output	`true`
Model Name output	`Factory_Training`
Platform output	`Windows`

The line added to the file:

```
17.0,"Factory_Training",2026-10-22 14:05:31,Windows,72.40
```

Tips

- To log a whole session, repeat **Get FPS** → **Record FPS To CSV** every few seconds, for example with a **Delay** node. Each run adds one line.
- If the scene was opened from a web link rather than from a folder on the device, the scene file sits in a temporary folder, so an empty **File Path or URL** saves the log somewhere hard to find. Give a full file path instead, or use the **Link** output to see where it went.
- To compare devices, point every headset at the same web address; each reading carries its own **Platform** and **Timestamp**.

Revision #1
Created 2026-10-25 08:22:13 UTC by Rafat
Updated 2026-10-25 08:22:15 UTC by Rafat