

# Use Kinect Anywhere

## Stream Data

Although only a limited set of software, for example TouchDesigner, offers native support for Kinect over the Kinect SDK, you can stream Kinect data from one of these applications to almost anywhere, including in different environment like Python, Node.js or even Web Apps including p5.js.

Within a local network, you can effectively setup one Windows computer as the Kinect Host and stream data to the same machine or different machines, utilizing Kinect on Raspberry Pi, MacOS or even Arduino.

Depending on the use case, you can stream the raw camera data as a video feed, or stream the skeletal tracking data as a live data feed. TouchDesigner is good as the host to do both.

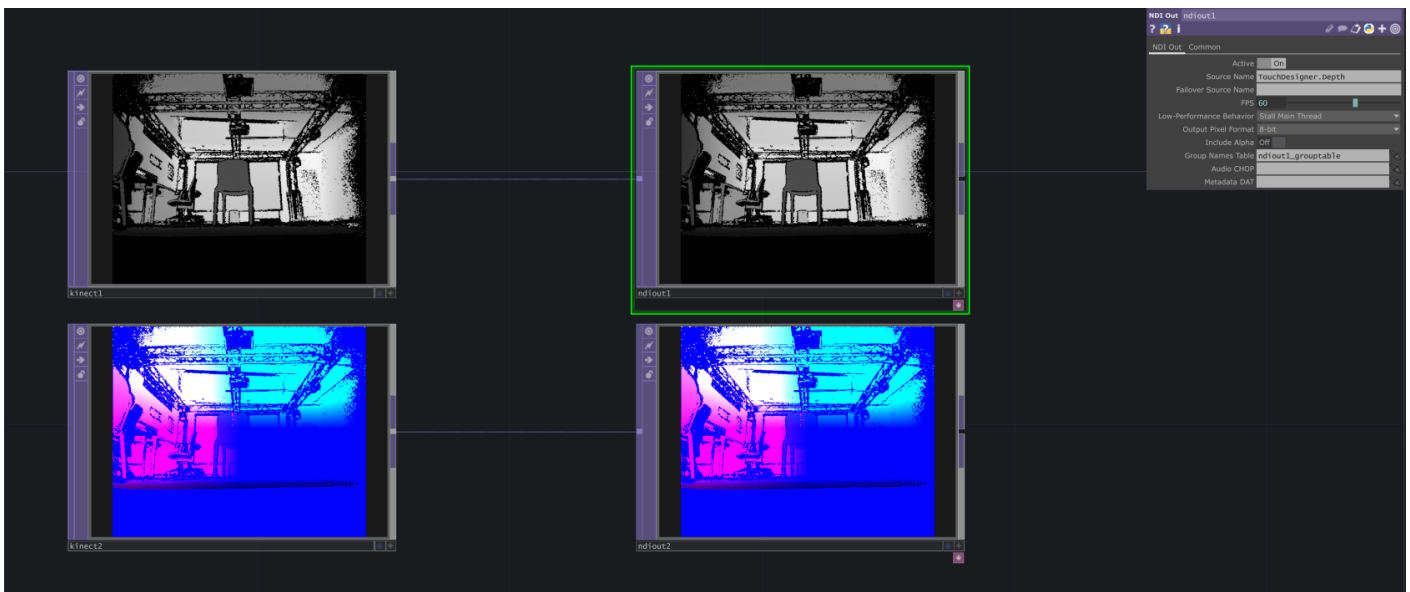
## Stream Image Feed via NDI

NDI (Network Device Interface) is a real-time video-over-IP protocol developed by NewTek. It's designed for sending high-quality, low-latency video and audio over a local network (LAN), with minimal setup and high performance. Read [NDI Documentation](#) for more information.

You can use NDI to stream Kinect video feeds (color, depth, IR) from TouchDesigner to:

- Another TouchDesigner instance (on same or different machine)
- OBS Studio for recording or streaming
- Unreal Engine, Unity, Max/MSP, etc.
- Custom apps using NDI SDK or NDI-compatible plugins

Setup NDI stream in TouchDesigner with `NDI OUT CHOP`, you can create different streams for different image source (TOP) with different names.



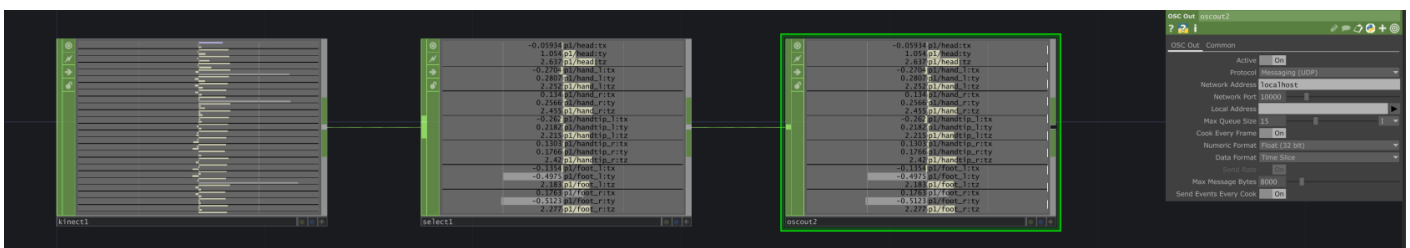
## Stream Body Tracking Data via OSC

OpenSoundControl (OSC) is a data transport specification (an encoding) for realtime message communication among applications and hardware over network, typically TCP or UDP. OSC is originally created for highly accurate, low latency, lightweight, and flexible method of communication for use in realtime musical performance and is widely used in music, art, motion tracking, lighting, robotics, and more.

You can use OSC to stream body tracking data from Kinect in TouchDesigner to other software (or vice versa), such as:

- Unity, Unreal Engine
- Processing, openFrameworks
- Max/MSP, Pure Data, Isadora
- Web apps (via websocket bridge)
- Python, Node.js apps
- Other TouchDesigner instance

Send OSC data from TouchDesigner with **OSC Out CHOP** or **OSC Out DAT**. Use **CHOP** when sending multiple channels straight from a chop; Use **DAT** with custom python script to further manipulate and format the data before sending.



## Stream with Kinectron for Web

OSC communication is typically implemented with UDP, which is fast and easy for native application to send and receive over the same local network. However web application in the browser runs in an isolated sandbox and does not have the access to local UDP port. To get data into your web application, you need a bridge for communicating with your web app through WebSocket or WebRTC.

Kinectron enables real-time stream of Azure Kinect Data to web browser. Visit Kinctron Release page to download the latest version of the server side application and client side library for using the data.

\*\* Notice that Kinectron V1.0.0 only support Azure Kinect. Support for Kinect Windows V2 could be found on the older version 0. Find more about Kinectron V0 and usage examples.

## Receiving data

OSC is widely supported in a range of applications and programming languages. Find the package or library to receive OSC data or you can bind a socket to the UDP port then listen and parse any OSC message.

## Unity

Script and examples for receiving OSC message

<https://t-o-f.info/UnityOSC/>

## Unreal

Unreal has built-in OSC Plugin, include the plugin from plug-in manager and start with a blueprint. Find the documentation

OSC Plug-in Overview

## Processing

Use oscP5 library for processsing

oscP5

## openFrameworks

openFrameworks has add-on to support OSC natively, find the documentation

ofxOsc Documentation

## MaxMSP and Pure Data

Use `udpsend` and `udpreceive` objects to send and receive osc message

## Ableton

Use Connection Kit in Max4Live to send and receive OSC data in Ableton. More info in [Connection Kit](#)

## Python

There are multiple packages available for osc in python.

One example is `python-osc`. Install with `pip install python-osc` and find the documentation [python-osc](#)

## Node.js

There are multiple osc packages in node as well.

One example is `osc`, install with `npm install osc` and find the documentation [osc.js](#)

With `osc.js`, you can also create a WebSocket bridge that forward osc messages to browser application.

## Browser /Web Application

To use Kinect data in the browser, there are two options

- Use Kinectron to stream data and receive with Kinectron client-side script, full support for video stream and body tracking data
- Use `osc.js` in Node.js to create WebSocket bridge and forward selective tracking data through websocket. Use JavaScript native WebSocket Client API to receive the data

## Use Kinect Directly in Unreal and Unity

In game engines, you should be able to use the Kinect SDK directly, which still involves some lower level programming and a lot of experience. There are some plugins developed for Kinect, but some of them are paid and some haven't been updated for years. Look for the right plugin for your need, depending on whether you want to get Skeletal Tracking or Depth Image.

## Unreal

[Neo Kinect](#) (paid)

[Azure Kinect for Unreal Engine](#)

## Unity

[Azure Kinect and Femto Bolt Examples for Unity](#) (paid)

[Unity\\_Kinect](#) (free)

---

Revision #3

Created 26 June 2025 10:01:58 by Ruiyang Wang

Updated 1 July 2025 11:14:57 by Ruiyang Wang