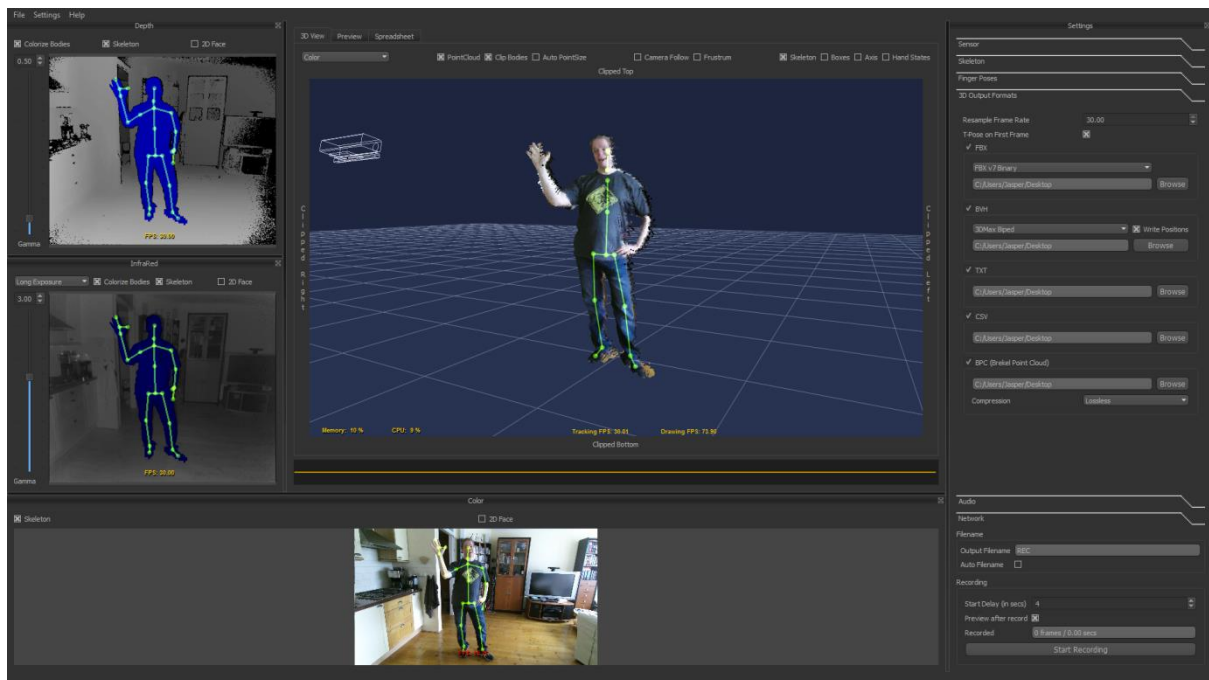


BREKEL

BODY

V2

WHAT IS “BREKEL BODY”



“Brekel Body” is a Windows application that does markerless body Motion Capture of up to 6 people from your living room or office using a Kinect sensor.

It is written by Jasper Brekelmans, so in case you’re wondering that’s what “Brekel” refers to.

“Brekel” is pronounced as “Break–uhl”.

REQUIREMENTS

- Kinect for Windows v2 sensor or Kinect for Xbox One with separately available adapter
- Windows 8 / 8.1 (USB stack of Windows 7 or below can't handle bandwidth requirements of v2 sensor and is NOT supported)
- USB 3.0 port, Intel and Renesas chipsets only! (others brands may or may not work)
- DirectX 11 capable GPU (Intel HD4000, AMD Radeon HD6470M / HD6570, NVIDIA Geforce 610M or above)
- 4 GB or more RAM
- Dual Core 3.1Ghz i7/i5 CPU or equivalent (slower should work but may drop frames)
- 1280×1024 screen (recommended: 1920×1080 or higher)

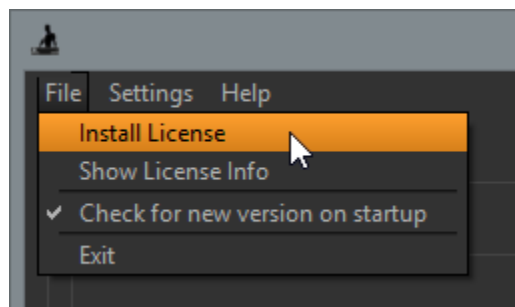
Note: when using a Kinect for Xbox One you will need to purchase a separately available power adapter to connect it to your PC.

LICENSING

With your purchase you should have received a download link for the retail version of the software and an email with your license code. (Please allow up to 24 hours for processing your request and double check your spam filter)

The retail version will work immediately and with no restrictions even without a valid license, but will need activation within 5 days for continued operation.

You can activate your copy by using the “File > Install License” option from the menu.

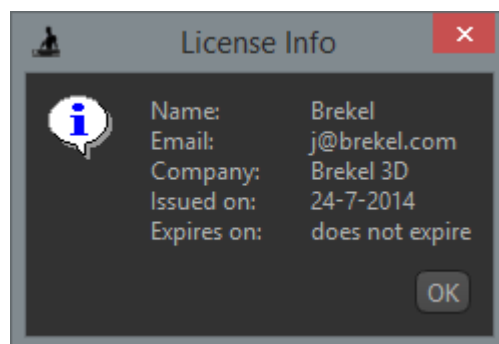


It will ask you to browse for your license file (which was attached to the mail).

The license will automatically be copied to your “C:\ProgramData” folder and be active upon restart of the application.

If license installation fails (for example due to user permissions) you also manually extract the .zip file and copy the .key file manually to the “C:\ProgramData” folder.

To display the current license info use the “File > Show License Info” option from the menu to display a window like the following:

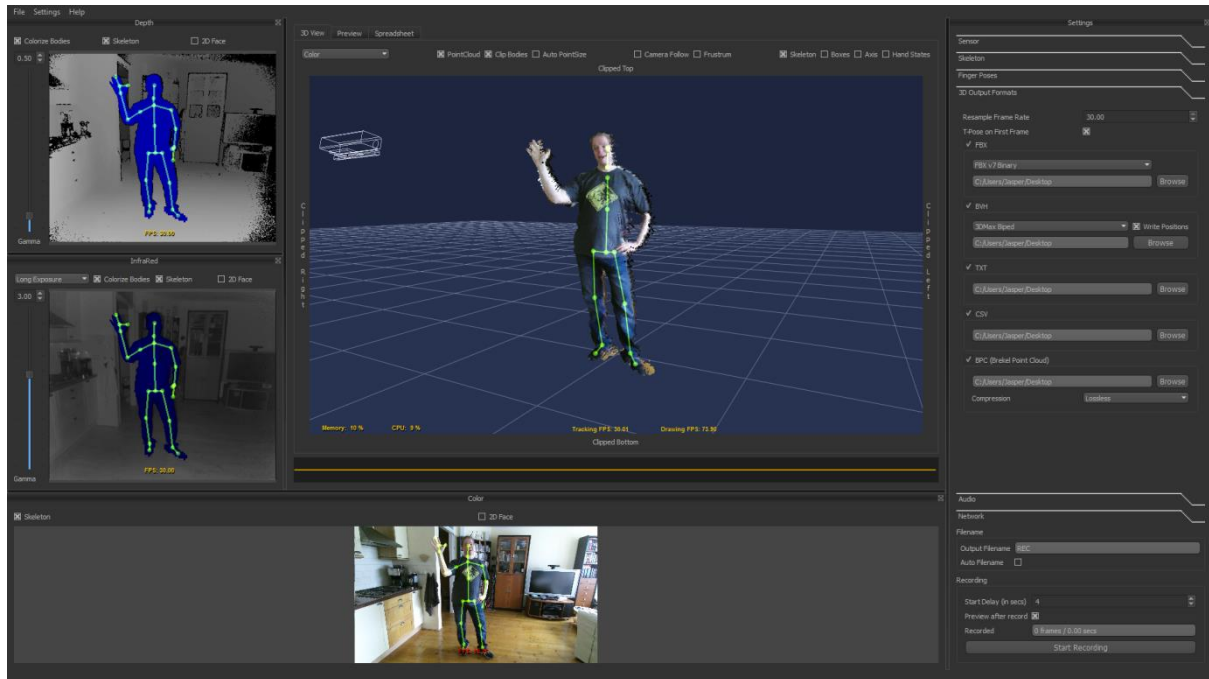


SENSOR PLACEMENT

Although the software can compensate for any sensor placement it's usually best to:

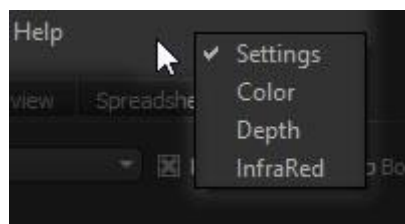
- place the sensor on a tripod (using the screw in the bottom)
- have it at about chest/head height
- point it down slightly so the floor can be seen (for automatic sensor alignment)
- make sure all bodies to be tracked are fully in view

THE MAIN INTERFACE

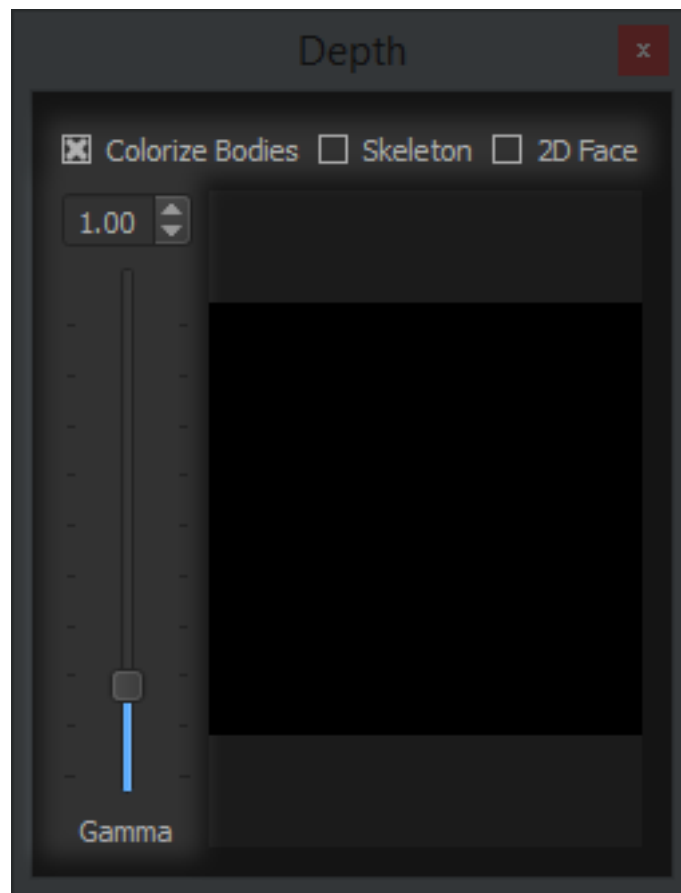


The interface is divided in 4 main docks:

- Top Left: Depth
 - Mid Left: InfraRed
 - Bottom: Color
 - Middle: 3D Viewport
 - Right: Settings
-
- These docks can be undocked and turned into a floating window by dragging from their title bar.
 - Double clicking on the title bar of a floating window will dock it back into the main window.
 - The overall GUI state will automatically be saved on exit and reloaded on start.
 - Right Clicking on the title bar of the main window (next to Help) will popup a list that allows showing hidden windows again.



DEPTH



This window visualizes the Depth stream which is used as the source for tracking.

Internally this is calculated from the InfraRed stream, it has a resolution of 512x424 at 30 fps.

In the visualization pixels closer to the camera have a lighter color, pixels further away have a darker color.

Black pixels indicate that the sensor wasn't able to measure depth, which can be due to being too close from the sensor (< 50cm), too dark, too reflective material, or too light. (for example strong sunlight)

Gamma

Adjusts the gamma of the Depth stream, you can use this to make it lighter/darker.

Colorize Bodies

Will color the pixels of detected bodies (up to 6).

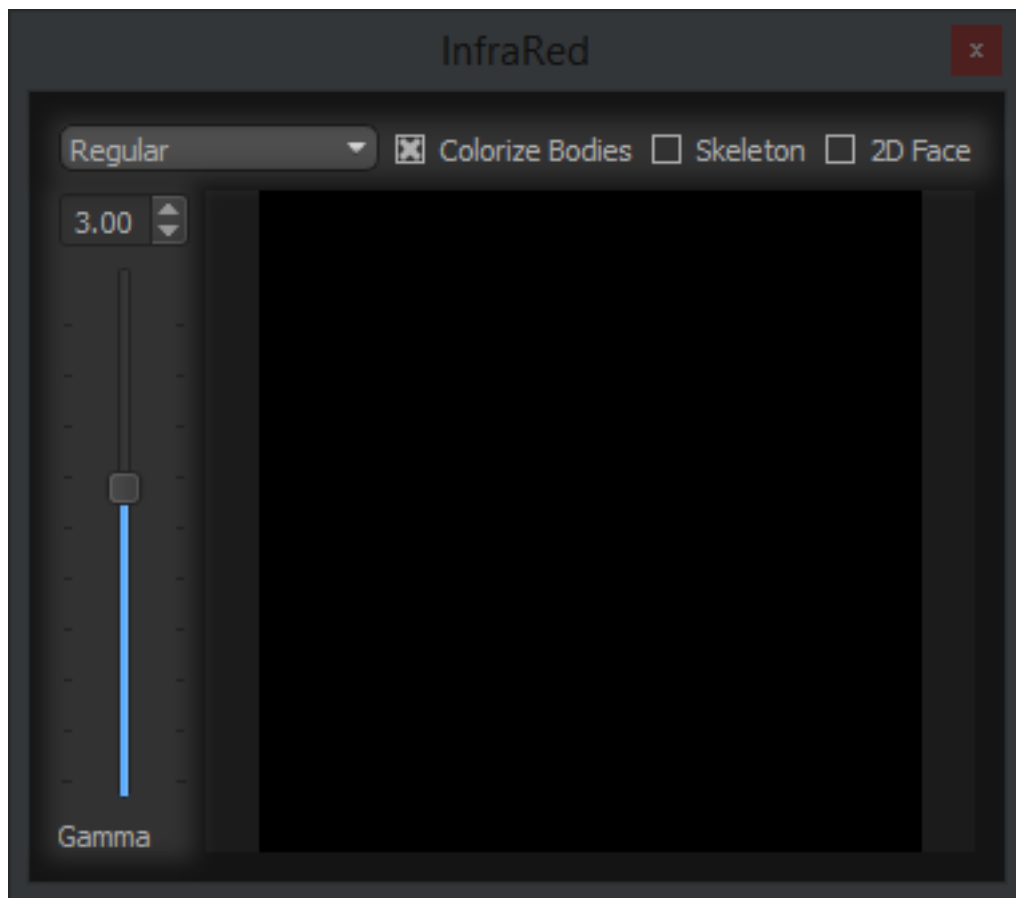
Skeleton

Draws the raw joints and bones of detected bodies in the 2D view.

2D Face

Draws the 2D face detection in the form of a rectangle and points in the 2D view.

INFRARED



This window visualizes the InfraRed stream which is the source for the Depth stream (and therefore tracking).

Internally this extracts all ambient lighting and should provide a very stable image as a base.

It has a resolution of 512x424 at 30 fps.

Gamma

Adjusts the gamma of the InfraRed video stream, you can use this to make it lighter/darker.

Regular / Long Exposure

Regular uses a single exposure frame with as little motion blur as possible but possibly some extra noise.

Long Exposure internally averages 3 frames (from the raw 300 fps stream) to reduce noise at the cost of slightly more motion blur.

Colorize Bodies

Will color the pixels of detected bodies (up to 6).

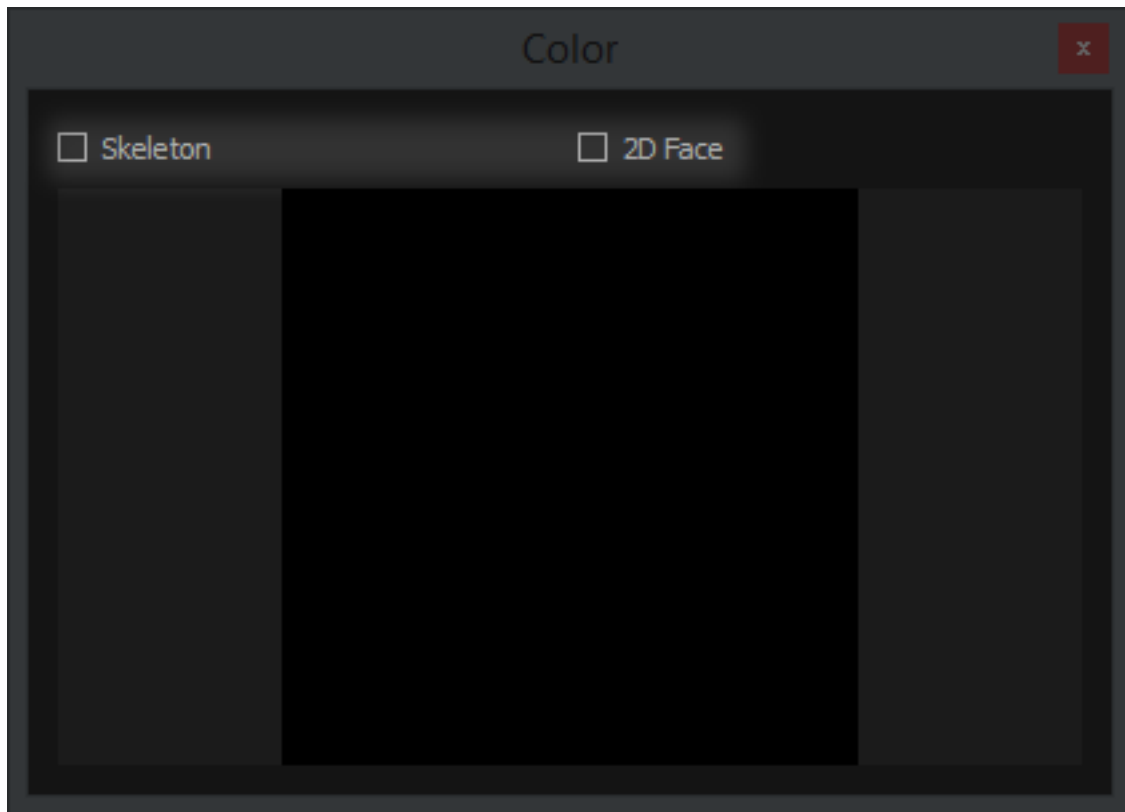
Skeleton

Draws the raw joints and bones of detected bodies in the 2D view.

2D Face

Draws the 2D face detection in the form of a rectangle and points in the 2D view.

COLOR



This window visualizes the Color stream, it is not used for tracking and purely exists for visualization purposes.

Note that this has a wider aspect ratio than the InfraRed and Depth windows and so it may show areas where tracking will not occur.

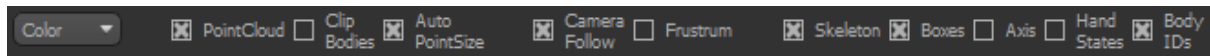
Skeleton

Draws the raw joints and bones of detected bodies in the 2D view.

2D Face

Draws the 2D face detection in the form of a rectangle and points in the 2D view.

3D VIEWPORT



The 3D viewport visualizes the 3D PointCloud data as well as any tracked skeletons.

Texture Selection, selects how to texture the pointcloud for visualization as well as recording in BPC format.

- None: do not use a texture but visualize white points
- Color: use the color camera to texture the pointcloud
- Infrared: use the infrared camera to texture the pointcloud
- Depth: use a depth gradient to texture the pointcloud, the further away a point is the darker

Note that Infrared and Depth are sampled from the same lens but Color is not. Color is automatically mapped to compensate for this but there may be small artifacts on object edges.

PointCloud

Toggles pointcloud rendering, note that on slow machines turning this OFF can gain extra performance.

Clip Bodies

Only pixels that belong to detected bodies are displayed, the background is clipped out in a 'greenscreen' like effect.

Auto PointSize

Tries to adjust the size of the points depending on the distance of the camera. (for visualization only)

Camera Follow

Turning this ON will automatically keep the camera centered on any tracked bodies.

Frustrum

Toggles drawing the camera frustrum in the viewport to give an indication of the field of view of the sensor.

Skeleton

Toggles drawing of joints & bones for tracked bodies.

Boxes

Toggles drawing of boxes for bones for tracked bodies, can give a better visual indication of rotation.

Axis

Toggles drawing of joint axis for tracked bodies, can give a better visual indication of rotation.

Hand States

Toggles drawing of the hand state detection as text in the 3D viewport.

Body IDs

Draws the ID (0-5) of the tracker body above the head in the 3D viewport.

3D VIEWPORT NAVIGATION

To orbit:

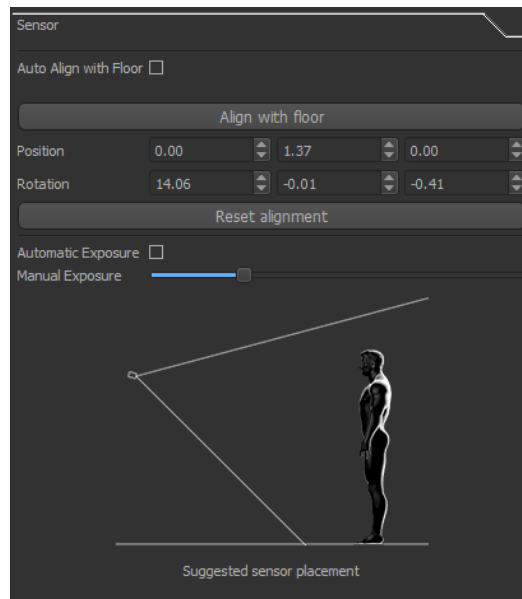
- Left mouse button, click & drag

To dolly in/out:

- Middle mouse button, click & drag
- Shift + Left mouse button, click & drag
- Mouse wheel

To pan:

- Right mouse button, click & drag
- CTRL + Left mouse button, click & drag



Auto Align with Floor

Automatically tries to align the sensor with the floor at every frame (a body needs to be seen in order to estimate where the floor is), not that this sometimes can result in some jitters so it's better to align once manually.

Align with floor

Tries to detect the floor and automatically adjust the Position/Rotation fields once.

(a body needs to be seen in order to estimate where the floor is)

Position

Adjusting these XYZ values will translate the pointcloud, the main value of interest is the Y value depicting the height of the sensor above the floor.

Rotation

Adjusting these XYZ values will rotate the pointcloud, which can be of use when the sensor is tilted.

Reset Alignment

Resets the Position and Rotation values back to 0

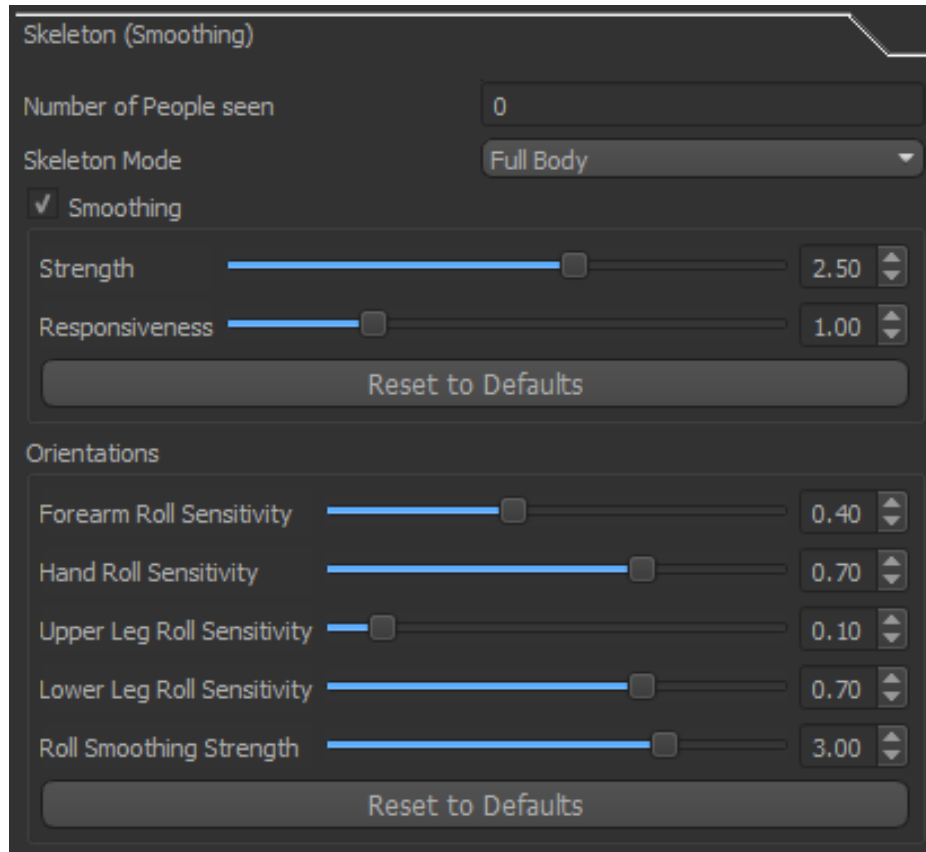
Automatic Exposure

When turned ON the color camera will automatically adjust exposure depending on the amount of light it sees.

Note that the color camera is not used for body tracking.

Manual Exposure

Allows for manual exposure adjustment when Automatic Exposure is turned OFF.



Number of People seen

Read only value showing how many bodies are detected and tracked from the Depth stream.

Skeleton Mode

- Full Body: tracks all joints as expected
- Upper Body Only (Standing): skips tracking of the legs and locks those in a standing pose
- Upper Body Only (Sitting): skips tracking of the legs and locks those in a sitting pose

Smoothing

Since the sensor data and markerless tracking inherently contains noise this enables specially designed filters to reduce the noise but retain the data as well as possible. Note that this will always introduce some lag.

Strength

The strength of the smoothing filtering that is applied.

Higher values smooth the data more but also increase lag and remove some detail.

Responsiveness

Works in conjunction with strength to try and reduce lag.

Higher values reduce lag but may reintroduce a bit of noise.

Reset to Default

Resets strength and responsiveness to their default values.

Forearm Roll Sensitivity

Controls sensitivity of the forearm roll, higher values apply more roll rotation but may also be a bit noisier.

Hand Roll Sensitivity

Controls sensitivity of the hand roll, higher values apply more roll rotation but may also be a bit noisier.

Upper Leg Roll Sensitivity

Controls sensitivity of the upper leg roll, higher values apply more roll rotation but may also be a bit noisier.

Lower Leg Roll Sensitivity

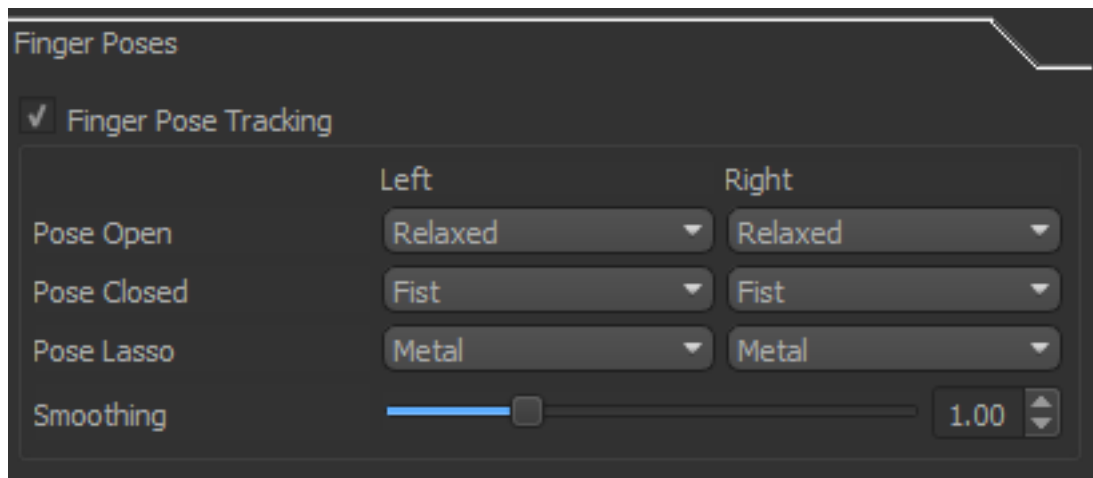
Controls sensitivity of the lower leg roll, higher values apply more roll rotation but may also be a bit noisier.

Roll Smoothing Strength

Amount of smoothing filtering to apply to the roll channels only, higher values apply more smoothing.

Note that arm roll relies on tracking of the thumb and may become less stable when a hand is in a fist pose.

Note that leg roll relies on tracking of the foot and may become less stable on very reflective floors or when the feet are obscured by clothing.



The tracker can detect 3 different hand states on the actor which can be connected to various finger skeleton poses using these settings, the 3 states are:

- Open
- Closed (fist)
- Lasso (pointing with 2 fingers at the same time)

Note that hand state tracking can only be done on 2 bodies simultaneously.

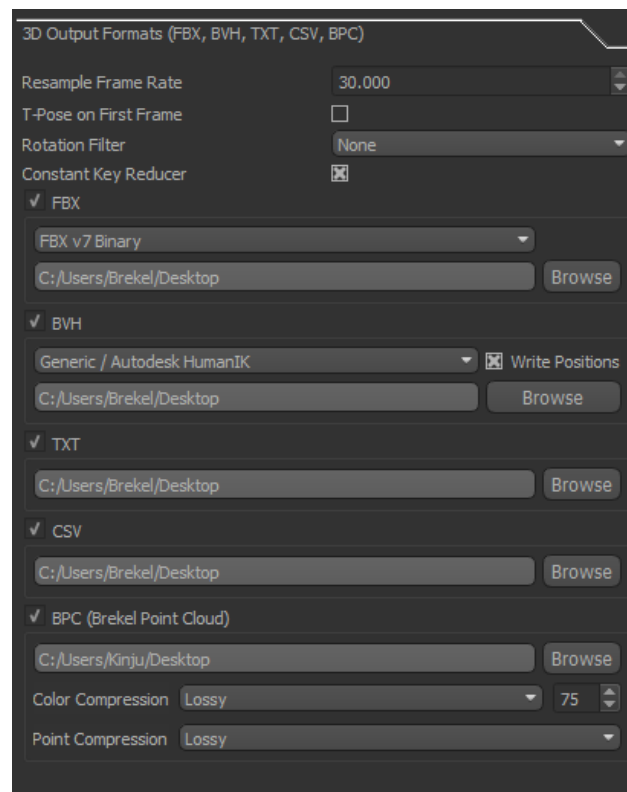
Each state can be connected to the following finger skeleton poses:

- Straight
- Relaxed
- Fist
- Pointing
- Metal

Smoothing tries to apply a smoothing filter to the somewhat noisy data the state detection can provide.

Higher values produce more smoothing.

Disabling Finger Pose Tracking locks the fingers in the Relaxed pose.



Resample Frame Rate

The sensor operates at 30 frames per second, by adjusting this value you can resample the animation data to a lower framerate. (Not supported for BPC files)

T-Pose on First Frame

By enabling this the first frame of the file will be the skeleton(s) in T-Pose, which can be convenient for certain workflows and retargeters.

Note that it is usually more accurate to actually capture a T-Pose instead of relying on a mathematically generated one.

Rotation Filter

Unroll & Gimble Killer analyze the rotation animation curves and try to prevent gimble flips, depending on your 3D application you may want to experiment with these.

Constant Key Reducer

Removes keys on static channels with no motion to keep file sizes small.

FBX

Exports to the FBX file format used by most 3D applications.

This is the most versatile and accurate file format, it includes additional data like hand states, clipping info, lean and 2D face tracking data as custom attributes on the “Hips” joint of each skeleton.

FBX version

You can choose between the current v7 and older v6 and between human readable Ascii files and the smaller Binary files. In case of a problem experiment with this to see what your favorite 3D application supports.

BVH

Exports to the BVH file format, this is a more limited format than FBX but still widely used.

Note that only skeletons can be saved and no additional data like hand states, clipping info, lean and 2D face tracking data.

Also note that due to the 20-year old format design specifications it has restrictions as to how accurately it can store rotational data since it uses only basic information to describe the skeleton and animation data.

BVH filetype

You can choose between the following file types which are designed to work with certain applications:

- Generic / Autodesk HumanIK
- 3DMax Bipod
- Poser / DAZ
- Blender

If your application isn't explicitly listed it may still work with one (or more) of the above presets.

The main difference is the naming scheme of the bones.

Write Positions

Turning this ON will write positions & rotations for each joint which preserves small fluctuations in limb lengths due to the tracking algorithm, turning it OFF will only write rotations fixing limb lengths to the average of that take.

Note that this is disabled for applications that are known to not be able to handle this.

TXT

Exports to a text file which can for example be used to read the data into custom code.

CSV

Exports to a Comma Separated Values file, which can for example be read into Excel or other spreadsheet applications.

BPC (Brekel Point Cloud)

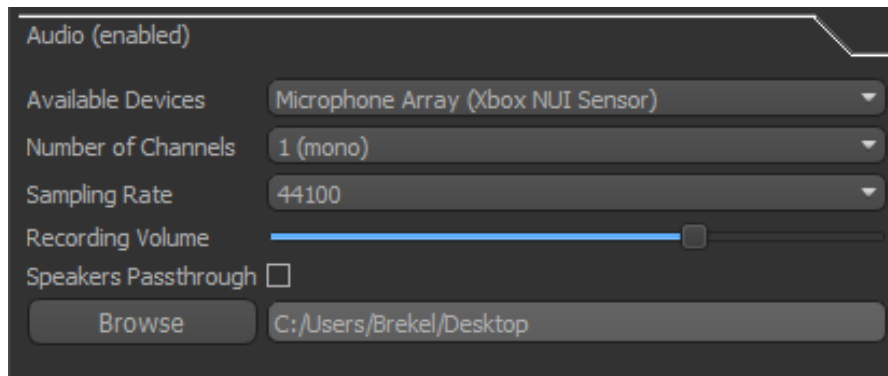
Records the pointcloud data to the custom BPC file format used by Brekel Pro PointCloud 2.

You can use this to record both skeletons and pointcloud data in sync and use Pro PointCloud 2 to export the pointcloud data to mesh and particle cache formats.

Compression

Lossless will result in the best quality but biggest filesize.

Higher amount of compression will produce smaller files at the loss of quality.



Available Devices

List the available audio devices that can be used for recording and allows you to select one of them.

Number of Channels

Switches between Mono and Stereo recording.

Sampling Rate

Sample rate to record (in Hz).

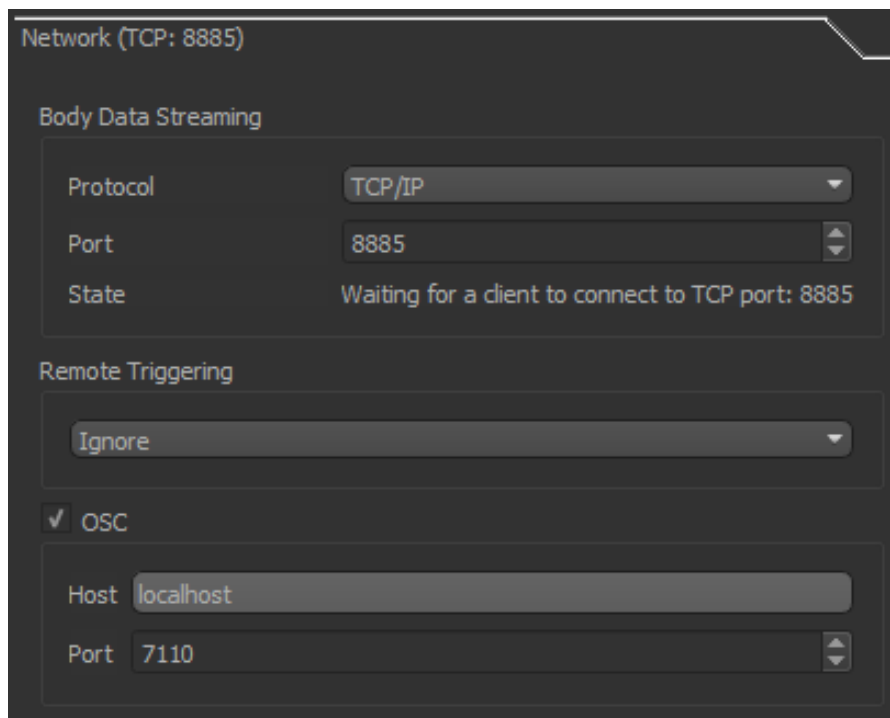
Recording Volume

Sets the volume used for audio recording

Speakers Passthrough

Playback the audio over the speakers while recording

Note that audio is always recorded in the WAV format.



The application has the ability to live stream the tracking data over a network port out to a different application (for example MotionBuilder). This can be done on the same machine, or across machines connected to the same network.

Protocol

Selects either TCP/IP or UDP as the streaming protocol, or turn streaming off.

Port

The network port over which data will be streamed. (default 8885)

State

The current state of the network streaming.

Please make sure security software (firewall) is not blocking the port/protocol.

Record Triggering

Allows synchronized recording across multiple Brekel applications.

One application can be in Primary mode, all others in Secondary or Ignore mode.

The Primary application will send a signal when recording is started and stopped so all applications start/stop at the same time and are using matching filenames.

Note that this works across multiple apps on the same machine and even across multiple machines on the same network.

Make sure your firewall isn't blocking port 8880-8890.

OSC

Enable/disable sending data out over UDP in OSC (Open Sound Control) format.

See chapter at the end of this manual for a description of the OSC messages that are sent.

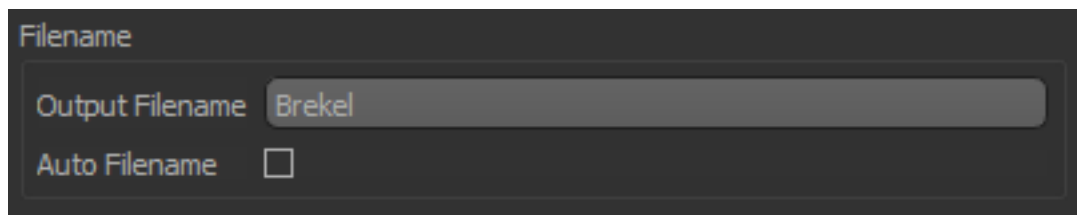
Host

Hostname / IP of machine to send the OSC data to (localhost if receiving application is running on the same machine as the Brekel application).

Port

Network port over which to send the OSC data.

SETTINGS > FILENAME



Filename

Output Filename Brekel

Auto Filename ☐

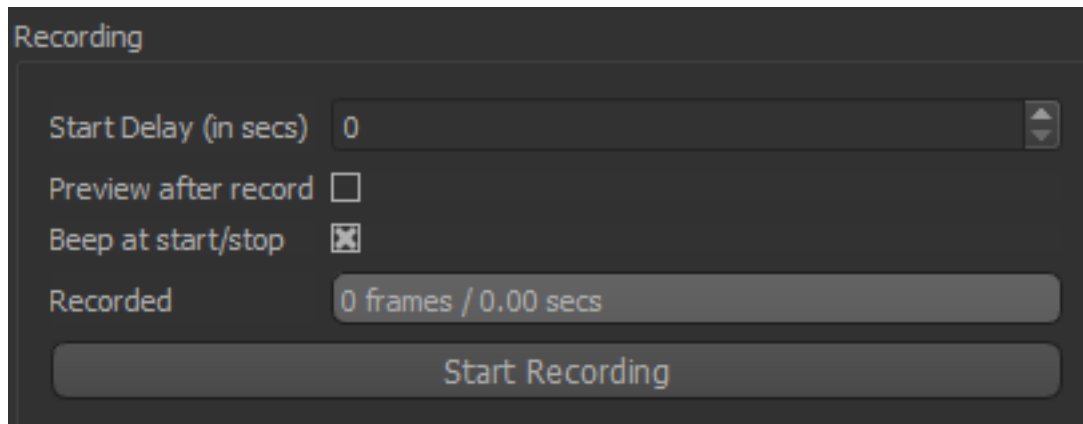
Output Filename

The name of the output file.

Note that the output folder(s) can be defined for each output format.

Auto Filename

When turned ON the filename will be automatically generated using the current date & time.



Start Delay (in secs)

When bigger than 0 a countdown will start after starting a recording, giving the actor a few seconds to get into position before capture will start.

During countdown beeps will be played over the speakers every second, and the amount of remaining seconds will also be shown visually in the 3D window.

Preview after record

Automatically switches to the Preview tab where you can replay the last take and decide to keep or delete it.

Beep at start/stop

Play a beeping sound when starting/stopping the recording.

Start/Stop Recording

Toggle button for starting & stopping the recording.

PREVIEW



The Preview tab shows the last recorded take.

With the standard video controls you can rewind, play and pause playback and scrub through the take.

Delete Take

Will delete the files of the last recorded take.

Keep Take

Will switch back to Live mode ready to record another take.

TOP MENU

File > Install License

Let's you browse to a license file and installs it to be used for the next time the application is run.

File > Manual License Installation

Open folder to which you can manually copy the license .key file.

File > Show License Info

Shows if a license is installed and who it belongs to.

File > Check for new version on startup

Checks for the availability of a new version upon startup, if a new version exists a summary of new/fixed features is shown and an option to take you to the download page.

Note that you will need a working internet connection for this.

File > Exit

Saves settings and exits the application (same as clicking the X in the top right corner)

Settings > Restore interface to default

Shows all hidden docking windows and places them back into their default positions.

Settings > Draw FPS

Toggle to draw the framerate in the bottom of the viewport.

Settings > Draw Memory/CPU usage

Toggle to draw the memory and CPU usage in the bottom of the viewport.

Settings > Don't enlarge video streams

When ON the 2D video windows will not enlarge beyond the raw data size.

When OFF the 2D video windows will proportionally stretch the content to match the window size.

Settings > Warn for Low FPS Streams

Will show a small system tray warning when detecting drawing is slower than the expected frame rate of the raw data stream. This can indicate your computer is too slow and may drop some frames to keep up.

Settings > Warn for File Overwriting

Toggle to warn for overwriting of existing files.

Settings > Warn for Delete Preview Take

Toggle warning when deleting a preview take.

Settings > Flash Sensor's LEDs during Recording

Flash the LED's on the Kinect sensor while recording.

Settings > DoubleClick 3D Viewport for Fullscreen

Allows making the 3D Viewport fullscreen when double clicking on it.

Settings > Viewport Settings

Brings up window to adjust some display settings like colors.

Help > Downloads Page

Opens your default browser and takes you to the Brekel download page.

Help > Forum

Opens your default browser and takes you to the Brekel forums. (hosted on Google Groups)

Help > Kinect v2 Configuration Verifier

Tries to start the Configuration Verifier (if you've installed the full SDK) which can help troubleshoot if your machine is capable of running a Kinect v2 sensor.

Help > Microsoft Troubleshooting Guide

Opens your default browser and takes you to the Microsoft Troubleshooting Guide page.

Help > Microsoft Forum

Opens your default browser and takes you to the Microsoft Kinect v2 forums.

Help > About

Shows the about window with version information.

HEADLESS MODE

You can run the application in a headless mode, without a GUI, in case you only want to stream data to other applications with minimal overhead.

To start in headless mode use “-headless” as one of the commandline options when starting the program, or use the start menu shortcut.

Upon startup settings will automatically be used from the last known startup of the GUI.

Furthermore the following commandline options are available to force particular settings:

`-synchronizeStreams 0`

Toggle stream synchronization (0 for OFF, 1 for ON)

`-autoAlignWithFloor 0`

Toggle automatic alignment with the floor (0 for OFF, 1 for ON)

`-sensorPosition 0.0 1.1 2.2`

Set sensor position where the 3 values are for X,Y,Z

`-sensorRotation 0.0 1.1 2.2`

Set sensor rotation where the 3 values are for X,Y,Z

`-skeletonMode 0`

Set skeleton tracking mode, where:

0 is Full Body tracking

1 is Upper Body Only (Standing)

2 is Upper Body Only (Sitting)

-smoothing 1

Toggles skeleton smoothing (0 for OFF, 1 for ON)

-smoothingStrength 2.5

Sets skeleton smoothing strength (0.0 for OFF, 4.0 for maximum)

-smoothingResponsiveness 1.0

Sets skeleton smoothing responsiveness (0.0 for OFF, 4.0 for maximum)

-rollSensitivityForearm 0.4

Sets roll sensitivity for the forearm

(0.0 for locked and stable, 1.0 for free but less stable)

-rollSensitivityHand 0.7

Sets roll sensitivity for the hand

(0.0 for locked and stable, 1.0 for free but less stable)

-rollSensitivityUpperLeg 0.1

Sets roll sensitivity for the upper leg

(0.0 for locked and stable, 1.0 for free but less stable)

-rollSensitivityLowerLeg 0.7

Sets roll sensitivity for the lower leg

(0.0 for locked and stable, 1.0 for free but less stable)

-rollSmoothingStrength 2.0

Sets strength for smoothing roll axis (0.0 for OFF, 4.0 for maximum)

-fingerposeOpenL 1

Sets the pose for an open left hand

0 = straight

1 = relaxed

2 = fist

3 = pointing

4 = metal

-fingerposeOpenR 1

Sets the pose for an open right hand (number description same as above)

-fingerposeClosedL 2

Sets the pose for a closed left hand (number description same as above)

-fingerposeClosedR 2

Sets the pose for a closed right hand (number description same as above)

-fingerposeLassoL 4

Sets the pose for a lasso (2 fingers pointing) left hand

(number description same as above)

-fingerposeLassoR 4

Sets the pose for a lasso (2 fingers pointing) right hand

(number description same as above)

- fingerposeSmoothing 1.0

Sets strength for smoothing finger poses (0.0 for OFF, 4.0 for maximum)

-networkProtocol 0

Sets the network protocol for streaming out body data to other applications

0 for TCP/IP

1 for UDP

-networkPort 8885

Sets the network port for streaming out body data to other applications

-ignoreRecordTriggering

When used the app will ignore Record Triggering signals from other Brekel apps in Master mode. (By default the headless mode will be in Slave mode)

-OSC_enable 0

0 disable OSC data streaming

1 enable OSC data streaming

-OSC_host localhost

Sets the hostname/IP of the machine to which to stream OSC data

-OSC_port 7775

Sets the port over which to stream OSC data

-OSC_mode 0

Sets the formatting of the OSC data

0 Brekel

1 Synapse

2 OSCeleton

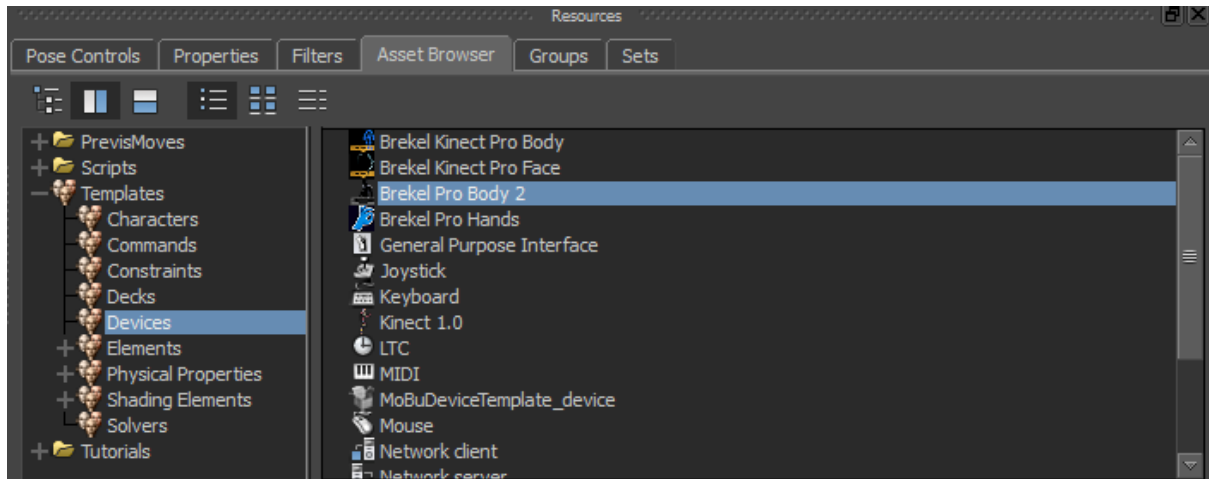
3 OSCeleton Kitchen

4 OSCeleton Quartz Composer

MOTIONBUILDER PLUGIN INSTALLATION

The installer should have automatically installed the plugins for the available versions of MotionBuilder.

And you should now have “Brekel Pro Body 2” listed under the “Devices” folder of the “Asset Browser”.



If it doesn't show up, you can find all the plugins for the various MotionBuilder versions (2009-2015 both 32- and 64 bit) in the “Brekel Pro Body 2” installation folder, usually in:

C:\Program Files\Brekel Pro Body 2 x64\MotionBuilder plugins

Simply copy the .dll file of your particular MotionBuilder version to the plugin folder.

Typically for a 32 bit version:

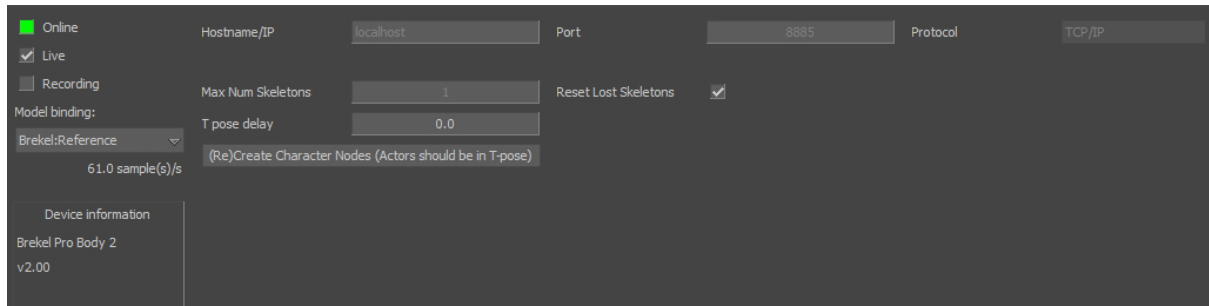
C:\Program Files (x86)\Autodesk\MotionBuilder 2013\bin\win32\plugins

For a 64 bit version:

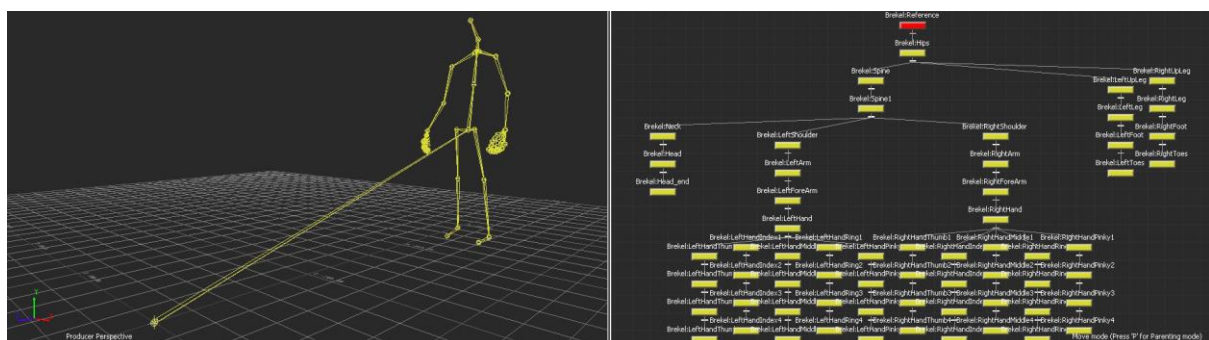
C:\Program Files\Autodesk\MotionBuilder 2013\bin\x64\plugins

USING THE MOTIONBUILDER PLUGIN

First make sure Brekel Pro Body 2 is running, and it's set to stream out data in the Network settings.



- Drag a Brekel Pro Body 2 device from the Devices folder in the Asset Browser into your scene
 - In the GUI for the device make sure the Hostname/IP points to the machine running the Brekel Pro Body 2 application
(Note: if both applications are running on the same machine you can simply use the default Hostname/IP of “localhost” or “127.0.0.1”)
 - Make sure Port and Protocol match in both Brekel Pro Body 2 and MotionBuilder
 - Make sure your security software (firewall) isn’t block this port/protocol
-
- If you want to track more than 1 person increase the Max Num Skeletons setting
 - Toggle the “Online” button to turn the device on
 - If it doesn’t turn green go back to the steps above
 - Toggle the “Live” button to start receiving data
 - Under “Model binding:” hit the “None” option and then “Create”, this will create a model hierarchy in your scene containing the skeletons



You should now see one or more moving skeletons in your 3D viewport.

Hostname/IP

Should point to the machine running Brekel Pro Body 2.

If both MotionBuilder and Pro Body 2 run on the same machine you can also leave this at the default of "localhost" or "127.0.0.1"

Port

Should be the same port as used in the Brekel Pro Body 2 application. (default 8885)

Protocol

Should be the same protocol (TCP/IP or UDP) as used in the Brekel Pro Body 2 application.

Max Num Skeletons

Defines how many skeletons to create at startup.

(You will need to create new model bindings when changing this)

Reset Lost Skeletons

When turned ON skeletons belonging to bodies that are not tracked are set to 0,0,0

When turned OFF skeletons belonging to bodies that are not tracked are left in their last known state.

T pose delay

When bigger than 0 a countdown will start after hitting the (Re)Create Character Nodes button, giving the actor a few seconds to get into position in a T-pose.

(Re)Create Character Nodes

Creates Character nodes (or delete/recreate if they already exists) for the Kinect skeletons, allowing for quick and easy retargeting setups to other characters.

Note that the actors should be in a T-pose for this to work correctly, since MotionBuilder uses that poses to map from one character to another.

To record a sequence:

- Make sure “Online” on the device is toggled ON
- Make sure “Live” on the device is toggled ON
- Make sure “Recording” on the device is toggled ON
- Make sure the “Record” button on the timeline transport controls is toggled ON
- Hit the Play button, and Stop button once the action is over

To play back a recording:

- Make sure “Live” on the device is toggled OFF
- Hit Play on the timeline transport controls or scrub through the timeline

To setup Character Nodes and Retargeting

- Set T pose delay to 4 (or more) seconds
- Hit the (Re)Create Character Nodes button
- Have the actor(s) in the capture volume assume a T-pose (increase T pose delay to give you enough time to get into position if needed)
- You can now go to the Character Node of the character you want to drive with the Kinect data and set
 - Input Type to “Character”
 - Input Source to one of the Kinect character nodes
 - Active to ON
- Your character should now be driven by the Kinect data

LIVE STREAMING INTO UNITY

You can find the Unity example script/scene for streaming live network data in the “Unity3D” folder inside the Brekel installation folder, usually something like here:

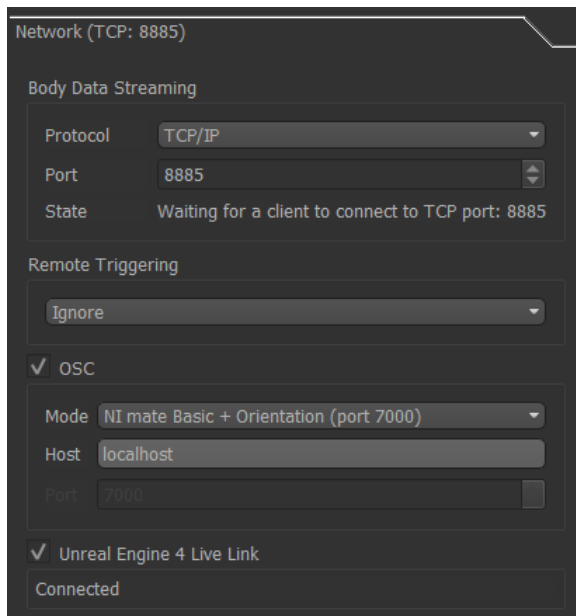
C:\Program Files\Brekel Pro Body 2 x64\Unity3D

You can also find the latest ReadMe.txt file here with instructions

LIVE STREAMING INTO UNREAL ENGINE 4

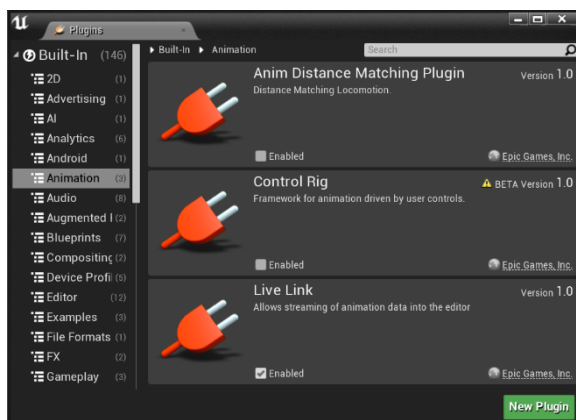
Live data can be streamed into the Unreal Engine 4 engine using its Live Link feature.

In the Brekel app make sure "Unreal Engine 4 Live Link" is enabled on the "Network" panel

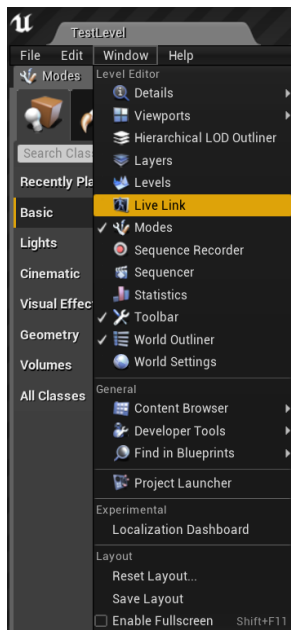


First make sure you enable this feature in the UE4 engine:

- Click Edit > Plugins
- Enable "Live Link" under "Animation"

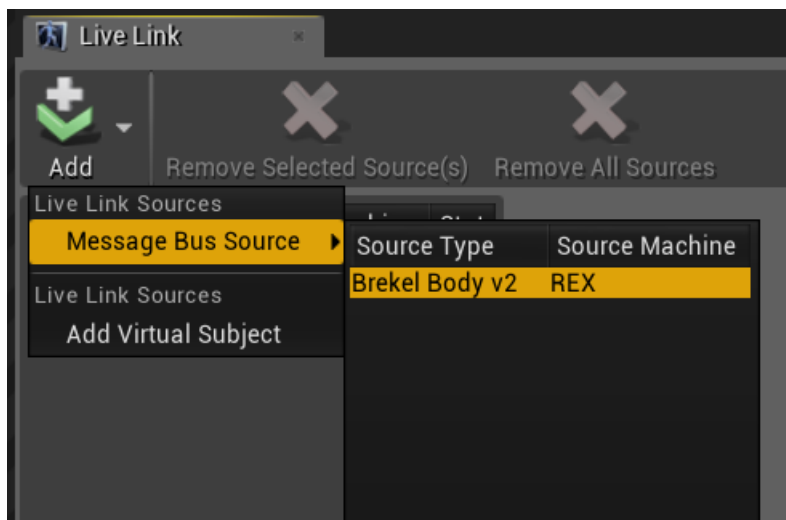


Click Window > Live Link



In Live Link Window

- click Add > Message Bus Source
- "Brekel Body v2" should be listed under Source Type/Machine, click it and hit Ok

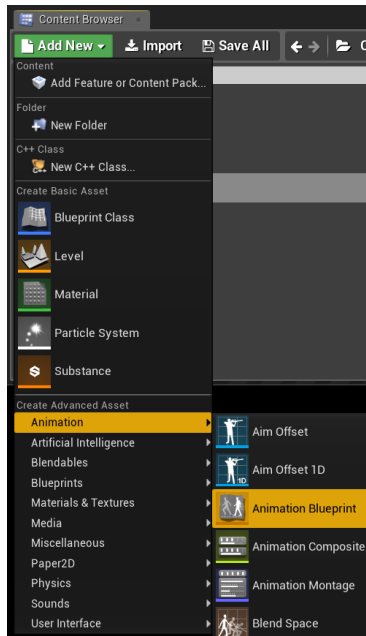


- Any visible and tracked bodies should be listed under "Subject Name"
- Take note of the body name you want to use
- "Closest_Body" for the body that is closest to the sensor
- "Body_0" - "Body_5" for a specific body, note that the body IDs can be displayed in the Brekel 3D viewport for reference

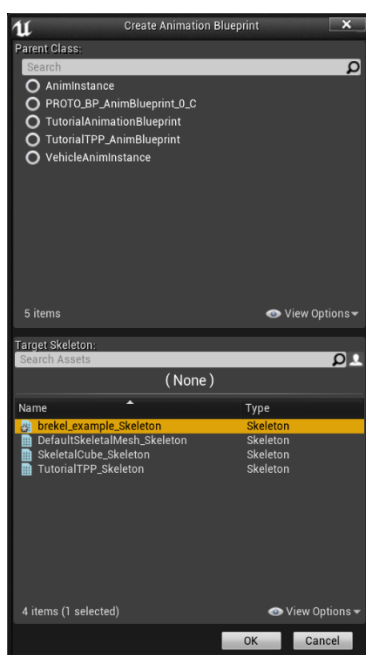


In the Content Browser:

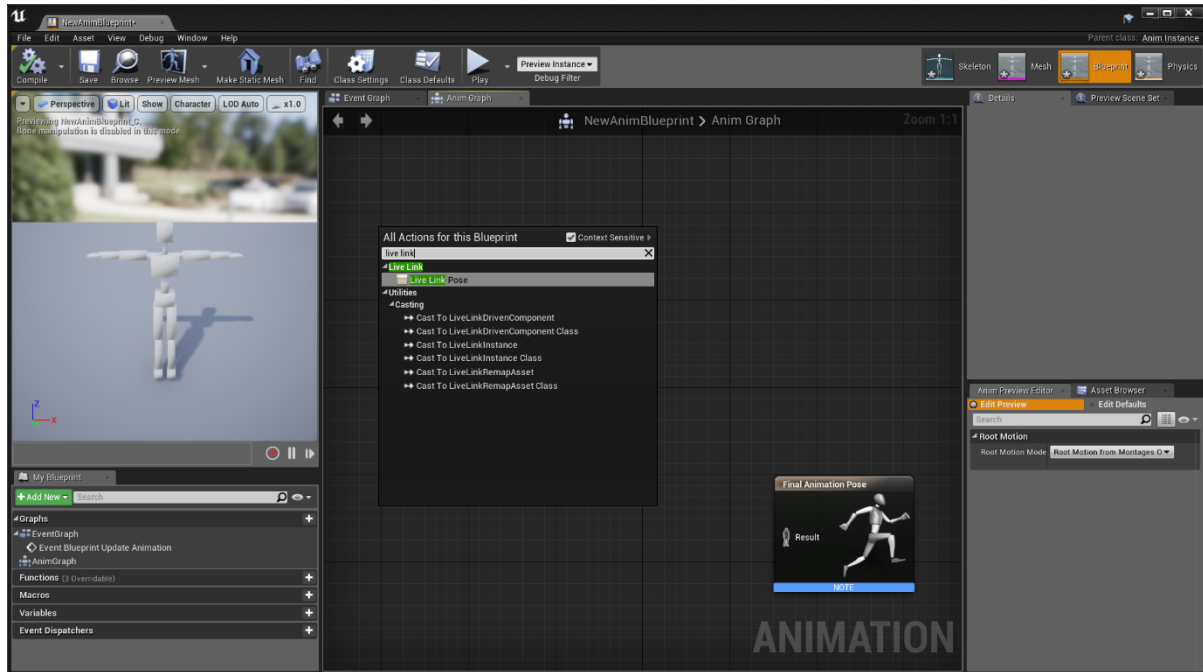
- Import "base_skeleton.fbx"
 - You can find this file in the "UnrealEngine4" folder inside the Brekel installation folder, usually something like here:
 - C:\Program Files\Brekel Pro Body 2 x64\UnrealEngine4
-
- Add New - Animation BluePrint



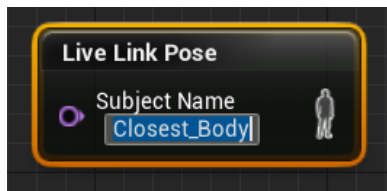
- Pick "brekel_example_Skeleton" from the list



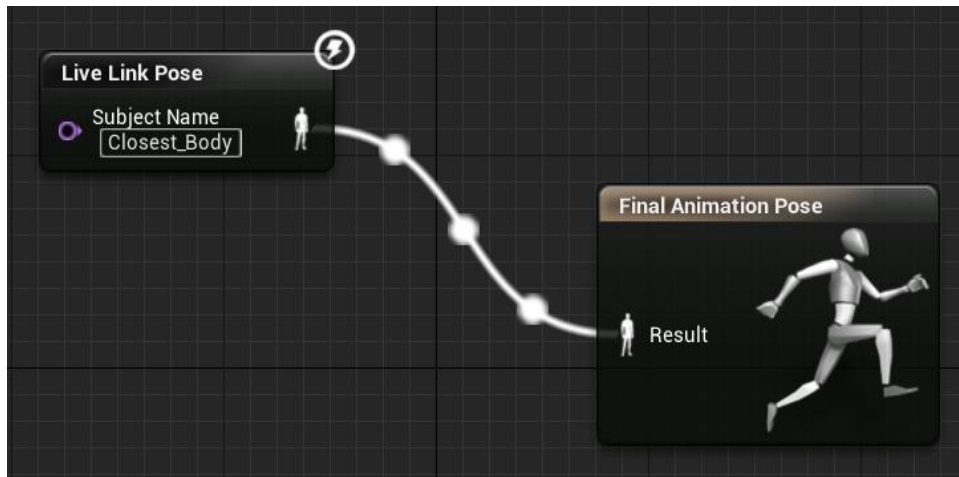
- Open the newly created blueprint by double clicking on it
- Right click in empty blueprint area
- Create a new "Live Link Pose" node



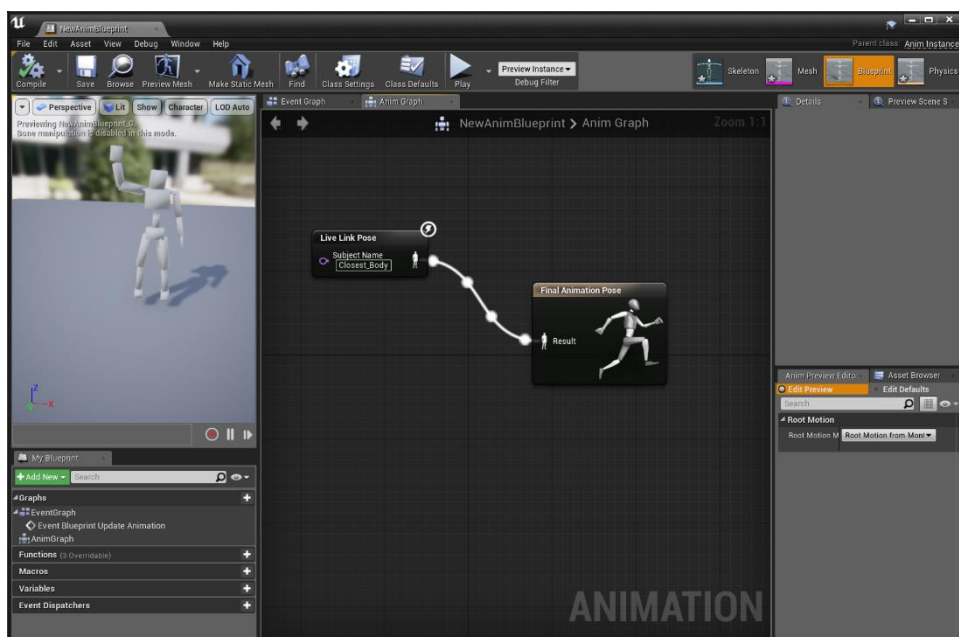
- Change Subject Name - None into the body you want to use ("Closest_Body" most probably)



- Connect Live Link Pose node to Final Animation Pose node



- Hit save and compile
 - You should now see a live data connection between the nodes
 - You should now see live data in the 3D window



NETWORK STREAMING PROTOCOL

Network packets can be sent through TCP or UDP, on a user definable port (default 8885)

Note that the free trial version doesn't include network streaming for security reasons, only the retail and full evaluation version do.

The beginning of each packet is as follows:

```
char      '2'  
u_short   bufferLength
```

The bufferLength is currently expected to be 12368, it is advised to check your incoming data for these values

Then it contains the following struct for the data:

```
struct networkDataBuffer_body
{
    const int    max_num_bodies    = BODY_COUNT;

    const int    num_joints        = bJoint_numJoints;


    int          isTracked[BODY_COUNT];

    int          trackingID[BODY_COUNT];

    float        timestamp[BODY_COUNT];

    float        isEngaged[BODY_COUNT];          // 0.0=no, 0.5=maybe, 1.0=yes

    int          isRestricted[BODY_COUNT];

    int          isClippedLeft[BODY_COUNT];

    int          isClippedRight[BODY_COUNT];

    int          isClippedTop[BODY_COUNT];

    int          isClippedBottom[BODY_COUNT];

    float        lean[BODY_COUNT][2];

    float        lean_confidence[BODY_COUNT];

    int          hand_L_state[BODY_COUNT];        // 0=unknown, 1=not tracked, 2=open, 3=closed, 4=lasso

    int          hand_R_state[BODY_COUNT];        // 0=unknown, 1=not tracked, 2=open, 3=closed, 4=lasso

    float        hand_L_confidence[BODY_COUNT];

    float        hand_R_confidence[BODY_COUNT];


    float        joint_confidence[BODY_COUNT][bJoint_numJoints];

    float        joint_lcl_pos[BODY_COUNT][bJoint_numJoints][3];

    float        joint_lcl_rot[BODY_COUNT][bJoint_numJoints][3];


    int          face_isTracked[BODY_COUNT];

    float        face_isHappy[BODY_COUNT];        // 0.0=no, 0.5=maybe, 1.0=yes

    float        face_isEngaged[BODY_COUNT];        // 0.0=no, 0.5=maybe, 1.0=yes

    float        face_isWearingGlasses[BODY_COUNT]; // 0.0=no, 0.5=maybe, 1.0=yes

    float        face_isEyeLeftClosed[BODY_COUNT]; // 0.0=no, 0.5=maybe, 1.0=yes

    float        face_isEyeRightClosed[BODY_COUNT]; // 0.0=no, 0.5=maybe, 1.0=yes

    float        face_isMouthOpen[BODY_COUNT];      // 0.0=no, 0.5=maybe, 1.0=yes

    float        face_isMouthMoved[BODY_COUNT];     // 0.0=no, 0.5=maybe, 1.0=yes

    float        face_isLookingAway[BODY_COUNT];    // 0.0=no, 0.5=maybe, 1.0=yes

};
```

Where:

BODY_COUNT = 6

bJoint_numJoints = 70

Note that all variables use 4 bytes (int/float) with no additional padding added to make parsing a packet as simple as possible.

Most of the data contents should be pretty self-explanatory and what you'd expect from the names.

The coordinate system uses: +X going screen right, +Y going up and +Z coming out of the screen.

Rotations are in Euler angles and degrees and are all in local space. (so in relation to their parent)

Some of these things may be subject to change in future versions.

The joints are defined in the following order:

```
enum bJoint_type
{
    bJoint_waist = 0,

    bJoint_spine,

    bJoint_chest,

    bJoint_neck,

    bJoint_head,

    bJoint_head_tip,


    bJoint_upperLeg_L,

    bJoint_lowerLeg_L,

    bJoint_foot_L,

    bJoint_toes_L,


    bJoint_upperLeg_R,

    bJoint_lowerLeg_R,

    bJoint_foot_R,

    bJoint_toes_R,


    bJoint_collar_L,

    bJoint_upperArm_L,

    bJoint_foreArm_L,

    bJoint_hand_L,

    bJoint_fingersBase_L,

    bJoint_fingersTip_L,

    bJoint_thumbBase_L,

    bJoint_thumbTip_L,


    bJoint_collar_R,

    bJoint_upperArm_R,

    bJoint_foreArm_R,

    bJoint_hand_R,

    bJoint_fingersBase_R,

    bJoint_fingersTip_R,

    bJoint_thumbBase_R,
```

bJoint_thumbTip_R,

bJoint_thumb_L_0,

bJoint_thumb_L_1,

bJoint_thumb_L_2,

bJoint_thumb_L_3,

bJoint_index_L_0,

bJoint_index_L_1,

bJoint_index_L_2,

bJoint_index_L_3,

bJoint_middle_L_0,

bJoint_middle_L_1,

bJoint_middle_L_2,

bJoint_middle_L_3,

bJoint_ring_L_0,

bJoint_ring_L_1,

bJoint_ring_L_2,

bJoint_ring_L_3,

bJoint_pinky_L_0,

bJoint_pinky_L_1,

bJoint_pinky_L_2,

bJoint_pinky_L_3,

```
bJoint_thumb_R_0,  
bJoint_thumb_R_1,  
bJoint_thumb_R_2,  
bJoint_thumb_R_3,  
bJoint_index_R_0,  
bJoint_index_R_1,  
bJoint_index_R_2,  
bJoint_index_R_3,  
bJoint_middle_R_0,  
bJoint_middle_R_1,  
bJoint_middle_R_2,  
bJoint_middle_R_3,  
bJoint_ring_R_0,  
bJoint_ring_R_1,  
bJoint_ring_R_2,  
bJoint_ring_R_3,  
bJoint_pinky_R_0,  
bJoint_pinky_R_1,  
bJoint_pinky_R_2,  
bJoint_pinky_R_3  
};
```

OSC (OPEN SOUND CONTROL) MESSAGES

The following messages are sent:

Address pattern: /lean

int: ID of the body this belongs to
float: timestamp
float: how far the person leans in the left/right direction
float: how far the person leans in the forward/backward direction

Address pattern: /handState

int: ID of the body this belongs to
float: timestamp
int: state of left hand: 0=unknown, 1=not tracked, 2=open, 3=closed, 4=lasso (pointing with 2 fingers)
int: state of right hand: 0=unknown, 1=not tracked, 2=open, 3=closed, 4=lasso (pointing with 2 fingers)
float: confidence of left hand: 0.0 not confident - 1.0 very confident
float: confidence of right hand: 0.0 not confident - 1.0 very confident

Address pattern: /face2D

int: ID of the body this belongs to
float: timestamp
bool: was the face seen and tracked by the sensor or not
float happy 0.0=no, 0.5=maybe, 1.0=yes
float engaged 0.0=no, 0.5=maybe, 1.0=yes
float wearing glasses 0.0=no, 0.5=maybe, 1.0=yes
float left eye closed 0.0=no, 0.5=maybe, 1.0=yes
float right eye closed 0.0=no, 0.5=maybe, 1.0=yes
float mouth open 0.0=no, 0.5=maybe, 1.0=yes
float looking away 0.0=no, 0.5=maybe, 1.0=yes

Address pattern: /<jointName>_joint_global (see list below for possible jointNames)

int: ID of the body this belongs to

float: timestamp

float: confidence 0.0 not confident - 1.0 very confident

float: X position in global space (relative to sensor)

float: Y position in global space (relative to sensor)

float: Z position in global space (relative to sensor)

float: X rotation in global space (relative to sensor) and in euler angles

float: Y rotation in global space (relative to sensor) and in euler angles

float: Z rotation in global space (relative to sensor) and in euler angles

Address pattern: /<jointName>_joint_local (see list below for possible jointNames)

int: ID of the body this belongs to

float: timestamp

float: confidence 0.0 not confident - 1.0 very confident

float: X position in local space (relative to parent joint)

float: Y position in local space (relative to parent joint)

float: Z position in local space (relative to parent joint)

float: X rotation in local space (relative to parent joint) and in euler angles

float: Y rotation in local space (relative to parent joint) and in euler angles

float: Z rotation in local space (relative to parent joint) and in euler angles

Address pattern: /<jointName>_joint_2D (see list below for possible jointNames)

int: ID of the body this belongs to

float: timestamp

float: confidence 0.0 not confident - 1.0 very confident

float: X position in screen space (of the IR/depth sensor)

float: Y position in screen space (of the IR/depth sensor)

Available joint names for <jointName>

waist

spine

chest

neck

head

head_tip

upperLeg_L

lowerLeg_L

foot_L

toes_L

upperLeg_R

lowerLeg_R

foot_R

toes_R

collar_L

upperArm_L

foreArm_L

hand_L

collar_R

upperArm_R

foreArm_R

hand_R

So for example you can search for matches on:

/head_joint_global

/lowerLeg_L_joint_local

/hand_L_joint_2D