

# Kinect

## General

Support for touch-free computing applications in Quest3D. The channel makes the Kinect sensor RGB and Depth video streams and most prominent body skeleton data available inside Quest3D. More than one person can be tracked. Several state changes and inquiries are allowed at runtime. Additional feature support includes: Active user tracking, near mode, seated mode, set/get elevation angle, send joint position and rotation, and set IR transmitter on/off.

Application scenarios include: skeleton tracking, gesture and posture recognition, and Augmented Reality.

## Children

- |     |        |                      |           |                                 |
|-----|--------|----------------------|-----------|---------------------------------|
| 1.  | OUT    | Skeleton Index       | (Value)   |                                 |
| 2.  | OUT    | Skeleton ID          | (Value)   |                                 |
| 3.  | OUT    | Player Position      | (Vector)  |                                 |
| 4.  | OUT    | Joint Position       | (Vector)  | - growing link list (20 joints) |
| 5.  | OUT    | Tracking State       | (Value)   |                                 |
| 6.  | OUT    | Quality Flags        | (Value)   |                                 |
| 7.  | OUT    | Processing Call      | (any)     |                                 |
| 8.  | OUT    | RGB Camera           | (Texture) |                                 |
| 9.  | OUT    | Depth Camera         | (Texture) |                                 |
| 10. | OUT    | Active TrackingIDs 1 | (Vextor)  |                                 |
| 11. | OUT    | Active TrackingIDs 2 | (Vextor)  |                                 |
| 12. | IN     | Fixed TrackingID 1   | (Value)   |                                 |
| 13. | IN     | Fixed TrackingID 2   | (Value)   |                                 |
| 14. | IN/OUT | Elevation Angle      | (Value)   |                                 |
| 15. | IN     | Smooth Parameter     | (Value)   |                                 |
| 16. | OUT    | Nearest Point        | (Vector)  |                                 |
| 17. | OUT    | Journal              | (Text)    |                                 |
| 18. | OUT    | Joint Rotation       | (Matrix)  | - growing link list (20 joints) |

**Skeleton Index** refers to the internal array position. **SkeletonID** is a growing number. **Player Position** can be used to display an alternate ghost avatar, if not enough joint data is present. **Joint Positions** is a growing link, it is ready to encompass all 20 joints. **Tracking State** can be 0 (not tracked) 1 (position only, passive tracking of 3<sup>rd</sup> to 6<sup>th</sup> person), or  $\geq 2$  (tracked). **Quality Flags** gives feedback with regard to clipping (see Quality Enumeration table in Appendix B). The **Processing Call** stub is the entry point to start a processing sub tree if multiple users are individually tracked.

The child values are undefined if no active tracks are encountered. Tracking one skeleton will result in filling all child positions including the final processing call. If two or more tracks are encountered, all child positions will be filled repeatedly. The typical way is to harvest the results by adding an evaluation sub tree to the last child link.

**RGB Camera** and **Depth Camera** contain the video streams. They can be used for Augmented Reality applications, or just as a normal high end video camera. The following resolutions are supported: 1280x1024 or 640x480 for RGB, and 640x480 or 320x240 for Depth.

The following more advanced children typically require a command and are available with the Kinect for Windows sensor. **Active User Tracking** in child 10 to 13: Active Tracking IDs provide up to six candidate IDs, out of which up to two can be chosen as active. The Sensor **Elevation Angle** can be controlled (child 14). Up to five different **Smooth Parameters** can be changed (child 15). **Nearest Point** delivers the 3D coordinates of the nearest object in camera space (child 16). This can be used for simple gesture recognition without using Skeletons. **Joint Rotation** -Experimental- Euler or Quaternion rotation angles of all 20 joints delivered in a growing link list of matrixes. Choices are selectable via commands, including the selection between absolute or hierarchical rotation angles (see appendix C).

## Description

The installation of a current Microsoft Kinect SDK or Runtime Version 1.7 or up is a prerequisite.

Calling this channel will result in multiple child calls, if more than one tracking is active. The processing sub tree must then be enabled to be work continuously. The return value reflects basically the tracking state:

>0	Number of active skeletons tracked (current limit is 2 active and 4 passive, imposed by the SDK).
0	Kinect ready, no skeleton
-1	Kinect not found or stopped
-2	Trial finished (Trial version only)
-3	Kinect Init pending
<= -4	Permanent failure

Kinect state commands are set by calling the channel with a **SetValue**. The following state settings are available:

Value	Command
-1	Disconnect Sensor
0	Connect Sensor -default-
3	Tracking active - default-
4	Tracking pause
5	RGB Camera active - default-
6	RGB Camera pause
7	Depth Stream active - default-
8	Depth Stream pause
10	Depth + player Index Visualization -default-
11	Player Index only Visualization (useful for alpha mapping)
12	Depth only Visualization

13	InfraredEmitter on –default-
14	InfraredEmitter off
16	Enable Active Player Selection
17	Disable Active Player Selection –default-
18	Enable Near Mode
19	Disable Near Mode –default-
20	Enable Depth Pointer
21	Disable Depth Pointer –default-
22	Set Elevation Angle of sensor (value in child 14)
23	Set Smooth Parameter (value in child 15)
24	Set Correction Parameter (value in child 15)
25	Set Prediction Parameter (value in child 15)
26	Set JitterRadius Parameter (value in child 15)
27	Set MaxDeviationRadius Parameter (value in child 15)
28	Enable Seated Tracking (upper 10 joints)
29	Enable Full Tracking (all 20 joints) – default-
30	Joint Output set to Position –default-
31	Joint Output set to Hierarchical Rotation
32	Joint Output set to Absolute Rotation
33	Rotation in Euler
34	Rotation in Quaternions (4 values)
96	Prepare Fast Exit - <i>see caveat</i> -
97	Prepare Normal Exit (default)
99	Collect Kinect State and Statistics (available at child 17)
100	Diagnostics off
101	Diagnostics standard (provided in editor debug window) -default-
102	Diagnostics moderate

Runtime duration and expiration restrictions are enforced with trial licenses. The applicable restrictions are listed in the editor debug window.

Expect Kinect updates at a rate of 30 per second at the best.

The Texture channels need to be **preconfigured** before the first use. The easiest way is to use a template with already preconfigured texture settings! Otherwise, follow the manual setting instruction:

1. Load the image `black1280.png` and `black320.png` into each video output texture
2. In each Texture Property: Set "MipMap Levels" to `none`

Failing to do so will result in a malfunction.

## Caveat

Great care was taken to ensure proper operation of the editor under all circumstances. However, due to the complicity of all processes involved, this cannot always be guaranteed. Please save your projects frequently.

Exiting of the Quest3D editor or runtime system will perform all Kinect related shutdowns too. This may take some seconds. The command "Prepare Fast Exit" shortens this process by forcing an intermediate exit. However, other pending save operations issued by other channels may be affected too. Check consequences in published projects. Invoking Fast Exit in the Editor is not recommended, as it may stall the next invocation of the editor.

## Tips

- Some features are not available due to Microsoft hardware of licence restrictions.
- Evaluate the Kinect channel frequently to react to mode changes.
- Don't expect foot and hand positions to be accurate.
- An application can handle multiple Kinect sensors
- Use preconfigured texture children
- RGB and Depth streams need to be aligned (see appendix A 1)
- Avoid resizing of the Output Window. You may need to stop and restart the Kinect engine by a SetValue command. Or restart Kinect Best practice is to use with an appropriate "Win32 SetWindow" channel setting.
- Both the classic Kinect 360 and the new Kinect for Windows hardware are supported. However, some features are available only with the Windows sensor.
- Deployments using Kinect for Windows hardware may use the runtime instead of the SDK.
- Do not disconnect Sensor (USB or power) while running.
- Avoid "Convert to Public Call Channel". Use Shortcuts instead!
- Near Pointer is best with Near Mode. It requires a texture at the depth child. Near Pointer coordinate space is similar to texture space: 0,0 is in the upper left corner, range is 0 .. 1.
- Avoid window resize and changes to full screen during runtime. This is particular true with video output. At worst, it may result in program failure. Best practice is to select in advance.
- Use the two commands stop engine and restart engine, if video transmission has stopped.
- Avoid having more than one sensor available.

**Legal note:** Permission granted for evaluation and educational purposes only (trial version), commercial use requires an explicit license agreement <http://godbersen.eu/shop/texte/license.txt>.

Volume, runtime duration, and expiration restrictions are enforced with trial licenses. Details can be found by reading the output of the Debug Window, and on my web page. Published trial projects are subject to occasional random termination. Be prepared to cope with this situation in your application.

Multiple sensors are not supported in this version, but is available upon request. Same applies to 3D avatar support.

## Supported Quest3D Versions

- 4.3.2
- 5.0 (x64). 32 on request only

## Known Problems/ Limitations

- The tight integration into the Quest3D system results in some annoyances. See Caveat above, and consult the tips section.

## Revision History:

2012 01 10 initial release Beta2

2012 03 20 updated to V1, tested with Kinect XBOX 360 and Kinect for Windows sensors

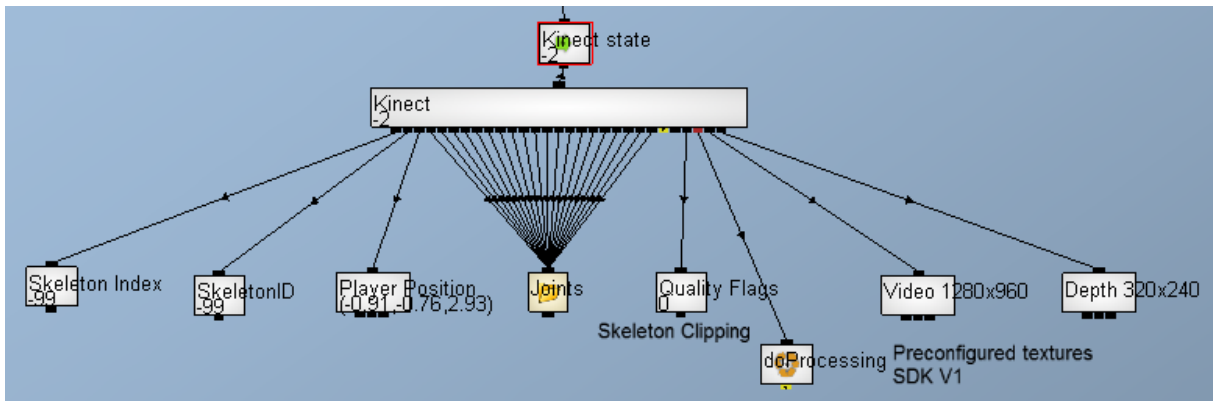
2013 05 24 updated to V1.7

**Contact:** [quest3d.godbensen.eu](http://quest3d.godbensen.eu)

# Tutorial

Examine the demo sources! You need previous knowledge of the Kinect sensor varieties and the API. Starting point: <http://www.microsoft.com/en-us/kinectforwindows/>

## Sample Scenario:



Do Processing will read all previous child values and displays each player’s skeleton or position. Do Processing is marked “continuous” in order to exemplify that more than one iteration may be needed, if you expect more than one player. If you are interested in only one player, you can do your processing in the main loop, and doProcessing is not needed.

For an annotated full example, see `KinectMultiTrackSkeletonDemo.cgr`.

## A list of a available Demos (distribution restrictions may apply)

Demo Name	Topic	Features
<b>KinectAugmentedReality</b>	Augmented Reality Scenario	Joint positions as gesture source
<b>KinectPan</b>	Panning Gesture	NearPointer as a gesture source
<b>KinectMultiTrackSkeleton</b>	Track up to 6 Players	
<b>KinectAdvancedFeatures</b>	Access to advanced features and commands	Active Player Selection, Near, Seated, ..

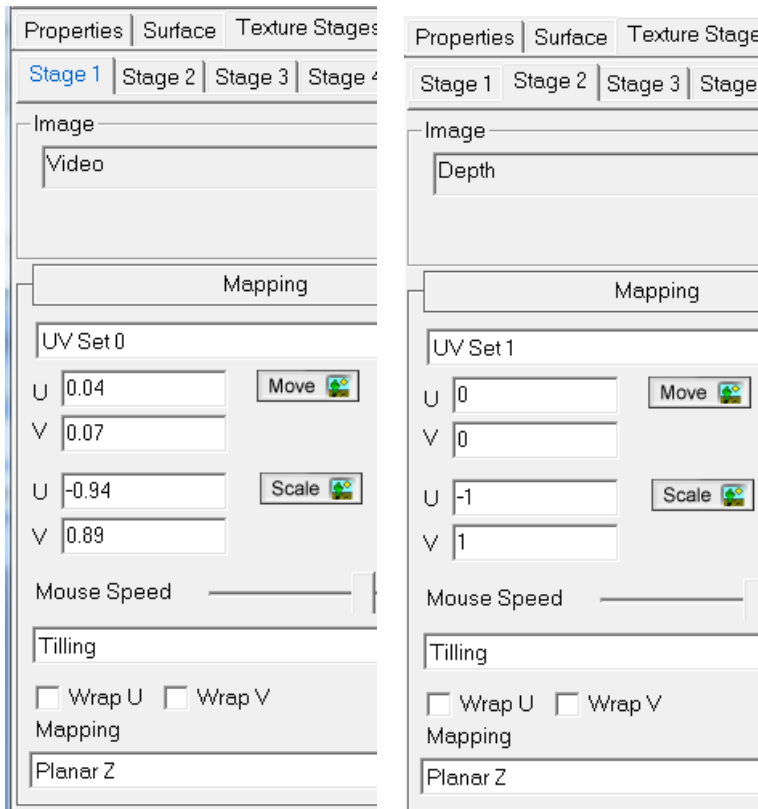
## Appendix A: Camera Output

### A1: Manual Alignment of RGB and Depth Camera Stream

Background: The two sensors are placed on Kinect in different positions, and they come with different field of view (FOV) settings Depth 70 degree vs. RGB 73.9 (diagonal).

Tip: Do coordinate transform on RGB only, leave Depth untouched.

My settings:



The values may differ for other Kinect hardware.

The FOV values of the cameras can be used for the Quest3D camera setup. The basic idea is to let the 3D scene camera have the same FOV as the sensors. (see KinectDemo.cgr)

### A2: Supported Resolutions

	320x240	640x480	1280x960
RGB	-	x	default
Depth	x	x	-
Depth + Player Index	default	-	-

x available, - not possible

## Appendix B: Skeleton State and Joints

Same as listed in the SDK:

### A) Skeleton

#### TrackingState

Member name	Description	KinectSkeleton
NotTracked	Not tracked.	0
PositionOnly	The overall position is tracked but not the individual joint positions. (occurs, if more than 2 users are tracked)	1
Tracked	All joint positions are tracked.	2

**Note:** The sum of all individual Joint Tracking State values is made available in the Tracking State child. In other words: If active tracking occurs, the return value is not 2 but 2 + the sum calculated. This allows for further assessment of the tracking quality.

#### Quality Enumeration

Member name	Description	KinectSkeleton (flags may add up)
-	none	0
ClippedRight	Part of the user's body is out of frame to the camera's right.	1
ClippedLeft	... to the camera's left.	2
ClippedTop	... above the camera's field of view.	4
ClippedBottom	... below the camera's field of view.	8

**Note:** More than one flag may be set.

### b) Joints

#### JointID (alphabetical order)

Member name	Description	KinectSkeleton growing link position
HipCenter	Center, between hips.	0
Spine	Spine.	1
ShoulderCenter	Center, between shoulders.	2
Head	Head.	3
ShoulderLeft	Left shoulder.	4
ElbowLeft	Left elbow.	5
WristLeft	Left wrist.	6
HandLeft	Left hand.	7
ShoulderRight	Right shoulder.	8
ElbowRight	Right elbow.	9
WristRight	Right wrist.	10
HandRight	Right hand.	11
HipLeft	Left hip.	12
KneeLeft	Left knee.	13
AnkleLeft	Left ankle.	14
FootLeft	Left foot.	15
HipRight	Right hip.	16
KneeRight	Right knee.	17
AnkleRight	Right ankle.	18
FootRight	Right foot.	19



## Appendix C: Advanced Topics

### C1: Active Player Selection

The ability to perform active player selection has to be requested by a command. Children 19 and 20 provide a list of up to 6 candidate player IDs. The application can choose up to two IDs for active tracking in child 21 and 22.

19	OUT	Active TrackingIDs 1	(vector)
20	OUT	Active TrackingIDs 2	(vector)
21	IN	Fixed TrackingID 1	(value)
22	IN	Fixed TrackingID 2	(value)

### C2: Commands with Parameter

Child 14 delivers the actual evaluation angle of the sensor. Provide a new value and call command 22 to set a new value. Child 15 is the parameter for the various smooth settings.

14	IN/OUT	Elevation Angle	(value)
15	IN	Smooth Parameter	(value)

22	Set Elevation Angle of sensor	(value in child 14)
23	Set Smooth Parameter	(value in child 15)
24	Set Correction Parameter	(value in child 15)
25	Set Prediction Parameter	(value in child 15)
26	Set JitterRadius Parameter	(value in child 15)
27	Set MaxDeviationRadius Parameter	(value in child 15)

### C3: Rotation Output

All joint rotations are delivered in a matrix channel as provided by the API. It is up to the application to do any further processing.

- The Euler rotation matrix can be processed immediately in Quest3D.
- The Quaternion 4-Element Vector is delivered in 4 elements of a Matrix. First column contains x, y, z values, the second column contains the w value. This is just a piggyback solution, and is not directly aligned with any Quest3D Quaternion channels.