

Assistive Technologies and Silver Science Group (leader: Assist. Professor Panagiotis Bamidis)  
of Medical Physics Laboratory, Medical School Aristotle University of Thessaloniki  
Thessaloniki, Greece

# Controler Application Communication (CAC) Framework Javascript API Reference

---

The CAC framework brings contemporary input devices (e.g. Kinect, wii, mindwave) to modern rich applications. The JavaScript API (decribed in this document) gives access to HTML5 applications to Kinect, Wii remote controller (Wiimote), Wii Balanceboard and Mindwave data for interaction and visualization. It allows the HTML5 applications running in a browser to connect to the sensor through a server. The CAC Javascript Reference (this document) provides all the required information to enable the developer utilizing the modern input device in modern Rich Internet Applications.

The CAC Framework server exploits the advantages of the web socket communication (found on most of the modern browsers). Besides this, restful api implementation immunizes backwards compatibility with old browsers.

The following sections demonstrate how to use the CAC Framework JavaScript API.

Assistive Technologies and Silver Science Group (leader: Assist. Professor Panagiotis Bamidis)  
of Medical Physics Laboratory, Medical School Aristotle University of Thessaloniki  
Thessaloniki, Greece

## **1. Include the Javascript source files**

Add the required JavaScript source files to your HTML page. The CAC Framework JavaScript API is defined in functionsInputDevice.js. Some of the UI elements such as cursors and buttons require the JQuery API.

```
<script src="http://code.jquery.com/jquery-1.10.1.min.js"></script>
```

```
<script  
src="http://kedip16.med.auth.gr/5on5/public/assets/scripts/functionsInputDevice.js"></scri  
pt>
```

Assistive Technologies and Silver Science Group (leader: Assist. Professor Panagiotis Bamidis)  
of Medical Physics Laboratory, Medical School Aristotle University of Thessaloniki  
Thessaloniki, Greece

## 2. Define SessionID

A group of input devices and clients define a session. Once a device streams information the rest members of the session receive the information. In order to define the session, the developer has to implement the following javascript function before the declaration of the functionsInputDevice.js. The following example register the web application to the Session DEMO for testing purposes. Data and information of all the supported devices is currently streamed (playback) at the session with ID="DEMO"

```
<script>
    function getInputDeviceGuid() {
        inputDeviceGuid = "DEMO";
        return inputDeviceGuid;
    }
</script>
<script
src="http://kedip16.med.auth.gr/5on5/public/assets/scripts/functionsInputDevice.js"></scri
pt>
```

### 3. Add a handler to process incoming stream frames

To process incoming data streams from the input devices, call the appropriate `addEventListener` function to set a callback function that is called whenever a data frame is received. The call back function may be fired on event **skeletonEvent** when a skeleton of the Kinect sensor is received, **wiiEvent** when a wiiremote or wii balanceboard frame is received, **MindwaveEvent** when a mindwave frame is received and **RGBVideoEvent** when an image is available. The following examples implement the `addEventListener`s.

```
<script type="text/javascript">
document.addEventListener("skeletonEvent", function (e) {
  //e.detail.SkeletonSourceData.Skeletons[0].Joints[0].Position.X
});

document.addEventListener("wiiEvent", function (e) {
  //e.detail.WiiSourceData.BalanceBoardState.WeightKg;
});

document.addEventListener("MindwaveEvent", function (e) {
  // e.detail.MindwaveSourceData.Attention;
});

document.addEventListener("RGBVideoEvent", function (e) {
  //e.detail.RGBVideoSourceData.RGBVideo.base64String;
});
</script>
```

## 4. Data Structures / Objects on events

The previous examples define the callback functions with the parameter e. This parameter contains information in respect to the type of the event. The streaming information is under e.detail.<SkeletonSourceData>|< WiiSourceData >|< MindwaveSourceData >|< RGBVideoSourceData >. Each object includes the implementation of a Device Object which contains information regarding the device that streamed the data.

| Device             |  |
|--------------------|--|
| DeviceID           | The device hardware unique id  |
| DeviceType         | 1: Skeleton, 2: Wiimote, 3:BalanceBoard, 4: Mindwave, 5: RgbColorImage |
| LastUpdateDateTime | The date and time of the last device capture                           |
| SessionID          | The session ID the device belongs to.                                  |

Table 1 Device Object

### 4.1 SkeletonSourceData



Figure 1 Kinect sensor

<http://www.microsoft.com/en-us/kinectforwindows/>

Skeleton contains an array of Skeletons. Each of the object of the array include the arrays of Joints, the Position and the TrackingState. The TrackingState represents the tracking status of an object according to Kinect (0: NotTracked, 1: PositionOnly, 2: Tracked). A skeleton with a tracking state of "position only" has information about the position of the user, but no details about the joints. The array of Joints contains 20 Joints each of them representing a Join of the body. Each Joint implements a Position and a TrackingState. The corresponding number of the Joints is presents in Table 2.

| JOINTS             |                  |                |                |
|--------------------|------------------|----------------|----------------|
| 0: HipCenter       | 5: ElbowLeft     | 10: WristRight | 15: FootLeft   |
| 1: Spine           | 6: WristLeft     | 11: HandRight  | 16: HipRight   |
| 2: ShoulderCentert | 7: HandLeft      | 12: HipLeft    | 17: KneeRight  |
| 3: Head            | 8: ShoulderRight | 13: KneeLeft   | 18: AnkleRight |
| 4: ShoulderLeft    | 9: ElbowRight    | 14: AnkleLeft  | 19: FootRight  |

Table 2 Skeleton joints corresponding number

Assistive Technologies and Silver Science Group (leader: Assist. Professor Panagiotis Bamidis)  
of Medical Physics Laboratory, Medical School Aristotle University of Thessaloniki  
Thessaloniki, Greece

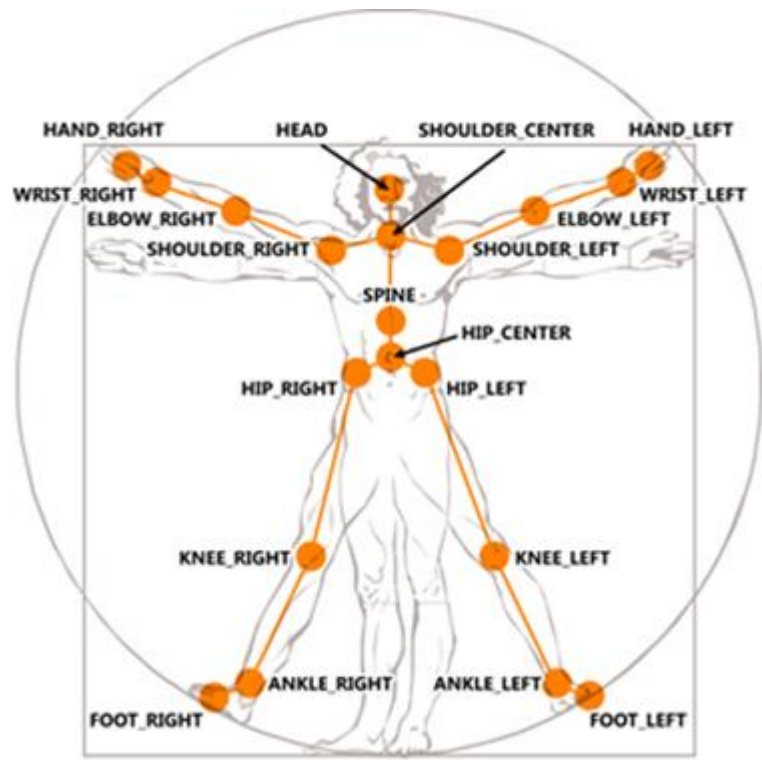


Fig.

<http://msdn.microsoft.com/en-us/magazine/jj159883.aspx>

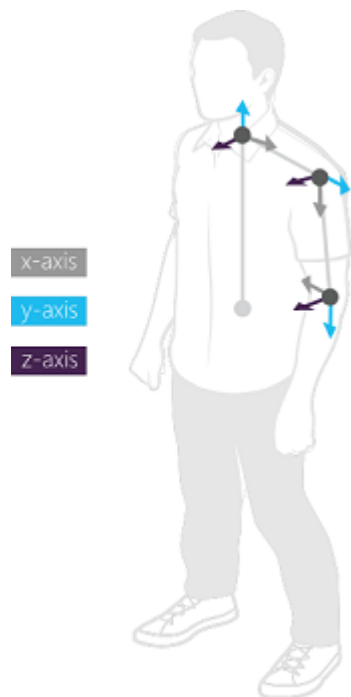


Fig. 2 <http://msdn.microsoft.com/en-us/library/hh973073.aspx>

Assistive Technologies and Silver Science Group (leader: Assist. Professor Panagiotis Bamidis)  
of Medical Physics Laboratory, Medical School Aristotle University of Thessaloniki  
Thessaloniki, Greece

## 4.2 WiiSourceData



Figure 2 Wii Balanceboard device (Nintendo)

<http://www.nintendo.com/consumer/downloads/wiiBalanceBoard.pdf>

The WiiSourceData contains AccelState (in case of wii remote controller) BalanceBoardState and ButtonState. The BalanceBoardState contains Weight in Kg, the CenterOfGravity (x and y axis) and the components of the weight at the 4 corners of the balance board (BottomLeft, BottomRight, TopLeft, TopRight). The ButtonState includes information of the buttons status being BalanceBoard or Wiimote.

## 4.3 MindwaveSourceData



Figure 3 Mindwave neurosky

<http://store.neurosky.com/products/mindwave-1>

The MindwaveSourceData contains the values of the processed EEG power spectrums (Alpha1, Alpha2, Beta1, Beta2, Delta, Gamma1, Gamma2, Theta), output of NeuroSky proprietary eSense meter for Attention, Meditation, and other future meters and signal quality analysis (can be used to detect poor contact and whether the device is off the head)

## 4.3 RGBVideoSourceData

<http://www.microsoft.com/en-us/kinectforwindows/>

The RGBVideoSourceData contains the RGBVideo object which incorporates the width and height of the image and the base64String of the image data. An image encoded into a base64 string and place it directly withing an HTML image tag or as a CSS background URL.

The base64String can be place whitin an HTML image tag as:

```
image.setAttribute("src", "data:image/jpeg;base64," + e.detail.RGBVideoSourceData.RGBVideo.  
base64String);
```



## 5. Example

```
<!DOCTYPE html>
<html>
  <head>
    <title>Usecase for Developers</title>
    <script src="http://code.jquery.com/jquery-1.10.1.min.js"></script>
    <script>
      function getInputDeviceGuid() {
        inputDeviceGuid = "DEMO";
        return inputDeviceGuid;
      }
    </script>
    <script src="http://kedip16.med.auth.gr/5on5/public/assets/scripts/functionsInputDevice.js"></script>
  </head>
  <body>
    <form>
      <label>Y position of user's HipCenter
        <input id="y-position" type="text" name="y-position" />
      </label><br>
      <label>User's alpha rhythm
        <input id="alpha-rhythm" type="text" name="alpha-rhythm" />
      </label><br>
      <label>User's center of mass (x-axis)
        <input id="center-of-gravity" type="text" name="center-of-gravity" />
      </label>
    </form>
    <p>RGB Video</p>
    <img src="" alt="RGB image" id="video">

    <script>
      document.addEventListener("skeletonEvent", function(e) {
        if (e.detail.SkeletonSourceData.Skeletons[0].Joints[0].TrackingState === 2) {
          document.getElementById("y-position").value =
e.detail.SkeletonSourceData.Skeletons[0].Joints[0].Position.Y;
        } else {
          document.getElementById("y-position").value = 'Not tracked';
        }
      });

      document.addEventListener("wiiEvent", function(e) {
        document.getElementById("center-of-gravity").value =
e.detail.WiiSourceData.BalanceBoardState.WeightKg;
      });

      document.addEventListener("MindwaveEvent", function(e) {
        document.getElementById("alpha-rhythm").value = e.detail.MindwaveSourceData.Attention;
      });

      document.addEventListener("RGBVideoEvent", function(e) {
        document.getElementById("video").setAttribute("src", "data:image/jpeg;base64," +
e.detail.RGBVideoSourceData.RGBVideo.base64String);
      });
    </script>
  </body>
</html>
```