Motion Tutorial

Motion Version 2.6

www.motionnode.com
www.motionshadow.com

1 Getting Started

The first steps to using your MotionNode sensor are listed in our **Getting Started** guide. This guide, as well all of the Motion documentation is available on our web site at http://www.motionnode.com/documentation.html and on your PC once you install our software package.

The **Getting Started** guide is intended for first time users of the Motion system, or if you need a quick review. This tutorial outlines the most common usage of the Motion system in more detail.

2 Software Components

The Motion software package is comprised of multiple components. Each component includes its own user manual or inline help. This overview explains the purpose of each component and references the associated manual or help files.

2.1 Service

The Motion software runs in the background as a Windows Service. By default, the service runs at Windows start up. Use the Windows `Control Panel > System and Security > Administrative Tools > Services` to check the status of, start, stop, or change start up behavior of the `Motion Service`.

The Motion Service is responsible for configuring devices, reading data, and recording data streams to the PC disk. Users control the service with scripting commands, a specialized programming language. There are multiple ways to send commands to the Motion Service. The **User Interface** provides a simple point and click view, accessible through a web browser.

For more information about the scripting system and commands, refer to the **Scripting Reference** manual.

Linux Users: The Motion Service runs as a daemon. Refer to the documentation for your Linux distribution to load the `MotionService` at system start up.

Mac Users: The Motion Service runs in the background as a daemon. It is not required to load the daemon at system start up. Simply open the Motion application bundle to launch the daemon and start the **User Interface**.

2.2 User Interface

To start the user interface, run the `Programs > Motion > User Interface` shortcut in your Windows start menu. This opens a browser window with the

web address of the **Motion Service**, http://127.0.0.1:32080/. Alternatively, manually open this URL in any web browser with Javascript support.

The **User Interface** displays the list of configured MotionNode devices, the current status of the system, and a set of commands in a toolbar at the top of the window. Use the drop down menus to access the available commands.

The user interface includes inline tool tips with all commands and links. Hover your mouse over a command for a brief description of what the command will do. Hover over a disabled command to learn why it is currently unavailable.

The **Tutorial** section provides examples of common usage of the Motion Service. Refer to these examples for step by step usage of the user interface.

2.3 Viewer

The **Motion Viewer** is a graphical application that displays all real time outputs of the Motion Service. The viewer displays the orientation of the sensor as a three dimensional box and the sensor data as a set of rolling plots.

Start the viewer application from the `Programs > Motion > Viewer` shortcut in your Windows start menu.

Press the **F1** key in the viewer or refer to the **Motion Viewer** manual for a full list of options and usage.

2.4 Monitor

The **Motion Monitor** is a utility application that detects device arrival and removal. The monitor can automatically configure and start reading from MotionNode USB and wireless devices as they are plugged in. The monitor application also provides user feedback on the current state of the Motion Service.

The monitor application runs in the Windows system tray. Start the monitor application from the `Programs > Motion > Tools > Monitor` shortcut in your Windows start menu.

Refer to the **Motion Monitor** manual for a full list of options and usage.

3 Tutorial

This tutorial covers everything you need to configure a sensor, preview the output, record data, and export it for usage or analysis in your software.

3.1 Configure a Sensor

The Motion Service operates on a set of devices called the configuration. To record data from a MotionNode IMU device it must first be a member of the current configuration set. The current configuration is displayed as the main table in the user interface.

The MotionNode may already be in the current configuration. If not, add it manually. Run the **Node** > **Scan** command to insert all available devices into the configuration set. You are now ready to record data from your MotionNode.

Unless you manually change the sensor parameters, each MotionNode is set to sample at $100\,Hz$ with an accelerometer range of $2\,g$. Use the command **G Select** in either the **Node** menu in the main toolbar or in the device specific menu accessed by clicking on the **Id** field to choose another g range. The **G Select** command prompts you for a value. Enter your desired accelerometer range and press **OK**.

Note that by using the main **Node** menu, the command applies to all MotionNode devices in the configuration. Each configured device also has its own command menu. Click on the **Id** field of the device to access the device specific menu.

3.2 Save Configuration

Any changes that you make to the sensor parameters are not automatically saved when you reboot your computer. However, the Motion Service can save a configuration to a file and load it later. Run the **File** > **Save** command to write the current configuration state to a file. Run the **File** > **Open** command to read a configuration file from disk.

If you have a single set of devices and associated parameters that you always want to use, you can save a configuration file that is automatically loaded every time the Motion system starts. Run the **File** > **Save** command and enter `default/configuration.mNode` as the file name.

All file paths are specified relative to your data folder. The data folder is specified the first time you run the Motion Service. By default, the data folder is `[...]/Documents/Motion/`. All user configuration and take data is stored in this folder.

3.3 Preview Data in Real-Time

The **Viewer** application provides a real-time graphical preview of the output of a MotionNode sensor. The viewer application is separate from the **User**

Interface and does not provide any device configuration.

To view data from your sensor, first start the device in the user interface. Run the **Node** > **Start Reading** command. This will initialize all configured devices and start reading data. Then, start the viewer application from the `Programs > Motion > Viewer` shortcut in your Windows start menu.

The viewer application starts with a split view of a 3D box that represents the current orientation and a rolling plot of the raw (uncalibrated) acceleration data. To view the calibrated accelerometer data instead, press the **4** key on your keyboard.

For a full list of all commands available in the viewer application, press **F1** or refer to the **Motion Viewer** manual.

3.4 Record Data

The Motion Service provides logging of all output data to your PC disk. A single session of recorded data is called a *take*. Each *take* is stored in your data folder, ordered by date and sequence.

Even though a device is connected, sending data, and is available for preview does not mean that the data streams are saved to disk. To record a take, run the **Node** > **Start Take** command in the user interface. When the take is active, the menubar will display *Take in progress...* on the right side.

When are finished recording data, run the **Node** > **Stop Take** command. This creates a link in the menubar to the *Current take*. This means that a take is currently loaded and ready for export to standard file formats. Click on the link to preview the take information, such as start and stop time.

3.5 Export Data

Once you have a take, you can export the data for easy access from third party applications. The Motion system provides export of orientation data to the following formats. Note that the orientation data is converted to Euler angles for maximum compatiblity with external applications.

- Autodesk FBX (*.fbx)
- COLLADA Digital Asset Exchange Schema (*.dae)
- Biovision Hierarchy (*.bvh)
- Comma Separated Values (*.csv)

To export a take, run the **File** > **Export** command. Enter a filename in the prompt. The type of file exported is determined by the file extension. For example, to export an FBX file enter the file name *take.fbx*. To export a COLLADA file, enter the file name *take.dae*. As with all user data, the exported file is stored relative to your Motion data folder.

The Motion Service also provides export of all sensor and orientation data associated with a take to a Comma Separated Value (CSV) format. Run the **File** > **Export Stream** command to export the current take data streams rather than just the Euler angle orientation output. Stream data is only available in the CSV format.

3.6 Matlab

The Motion software package includes Matlab scripts to directly access take data files. The scripts are also compatible with the freely available Octave application, which is similar to Matlab. The easiest way to use the scripts is to add the folder to the Matlab/Octave search path. The Motion data loading scripts are installed in `C:/Program Files/Motion/tools/matlab`, by default.

Once the Matlab scripts are in the path they are usable like regular Matlab commands. To access take data, change the working directory to a take folder.

```
cd 'C:/Users/username/Documents/Motion/take/2012-08-24/0001';
```

Now the current working directory is the folder for the first take recorded on August 24, 2012. To load and plot the calibrated sensor data for this take, use the `plot_sensor` function. Note that the following functions all simply take a file name as the argument. Since we are in the take folder, we can use relative paths to access the data files.

```
data = plot_sensor('sensor/MotionNode');
% Copy the x accelerometer channel into a variable.
ax = data(:, 1);

% A similar function for the raw, uncalibrated data.
raw_data = plot_raw('raw/MotionNode');
```

To access the quaternion data for the take and plot the Euler angles, use the `plot_output_euler` function. Note that this command returns two sets of data. The first set is Euler angles derived from the global quaternion data. The second set is the quaternion data itself.

```
[euler,quaternion] = plot_output_euler('output/MotionNode');
```

3.7 Excel

To import orientation and sensor data into Microsoft Excel, or any other similar application, use the stream export functionality. First, export the current take using the **File > Export Stream** command. Note that if you do not have a current take, you can either record one or load an existing take from disk. Second, import the Comma Separated Value (CSV) stream file into your spreadsheet application. By default, the exported file name is `take_stream.csv` in the Motion data folder.

The CSV stream format contains the following fields. Note that some fields may not be present depending on sensor configuration. See the **SDK Reference** manual for more information on the format of the data streams.

- time, in seconds
- Gq[w, x, y, z], global quaternion
- Lq[w, x, y, z], local quaternion
- r[x, y, z], local Euler angle rotation in radians
- l[x, y, z], global linear acceleration, specified in g
- a[x, y, z], accelerometer measurement, specified in g
- m[x, y, z], magnetometer measurement, specified in μT (microtesla)
- g[x, y, z], gyroscope measurement, specified in $^{\text{degree}}/_{\text{second}}$
- temp, temperature measurement, specified in degrees Celsius

3.8 Application Development

The Motion software includes a Software Development Kit (SDK) to simplify integration with your C, C++, C#, Java, or Python application. The Motion Service publishes all output data streams for real-time access from your application. See the **SDK Reference** for more information and real usage examples.

The SDK classes and C API access data streams from the Motion software service. All sensor configuration and management is handled through the **User Interface** or through scripting commands.