# Peacock Console

## Dependencies

- Peacock Univarsal Utility :doc:`../utils/readme`
- Peacock AE Utility :doc:`../ae_utils/readme`
- Peacock UI Utility :doc:`../ui_utils/readme`
- Peacock Preferences :doc:`../preferences/readme`

# Table of Contents



The console is like a command line. Three different types of input are possible.

**After Effects Keyframes / Mocha Tracking Data**

You can either paste Mocha tracking data directly from Mocha into the console or Keyframes from a selected layer property in After Effects. Note that only Position, Scale and Rotation keyframes are supported yet. If you press: `Cmd+Enter` or the `'R' button` the keyframes are getting parsed into an internal keyframe data structure.

> ***Note***
>
> There is no use for the parsed keyframes yet. I plan to manipulate tracking data keyframes synced to the beat.

**Peacock midi note data**

The external standalone program "Midiconverter" converts a midi file (.mid) into 'Peacock midi note data'. For this to work the midi notes in the midi file have to be in the range from C3 - C4 and you need to set the proper bpm value. After the midi file is converted the 'Peacock midi note data' is automatically copied to the clipboard and a .txt file with the same 'Peacock midi note data' is created as a sibling of the midi file. The 'Peacock midi note data' can be directly pasted into the BpmSlicer console. By pressing `Cmd+Enter` or the `'R' button` the slice data is getting parsed into the internal slices array which can then be used to slice layers in a composition.

**Executable javascript**

You can write any javascript code you like and execute it directly from the console. Some useful code snippets are accessible through tabcompletion and shortcuts

# Tabcompletion

A list of all tab completion code snippets.

`for`

```
aeHelper.selectAllLayers(comp);
for(var i=0; i<comp.selectedLayers.length; i++){
  var layer = comp.selectedLayers[i];
```

```
  log.appendLog(i + " " + layer.name);
}
```

fors

```
for(var i=0; i<slices.slices.length; i++) {
  var slice = slices.slices[i];
  log.appendLog(i + " " + slice.getInPoint());
}
slices.slices.length;
```

form

```
for(var i=0; i<markers.markers.length; i++) {
  var marker = markers.markers[i];
  log.appendLog(i + " " + marker.getTime());
}
markers.markers.length;
```

if

```
if(markers.markers.length > 10) {
  log.appendLog("More than 10 markers exist");
}
(markers.markers.length > 10);
```

if else

```
if(markers.markers.length > 10) {
  log.appendLog("More than 10 markers exist");
}else {
  log.appendLog("Less than 10 (or equal) markers exist");
}
(markers.markers.length > 10);
```

# Shortcuts

A list of all tab shortcut code snippets.

select

```
var counter = 0;
for(var i=0; i<comp.selectedLayers.length;i++){
  var layer = comp.selectedLayers[i];
  if(layer.name != ""){
    layer.selected = true;
  }
  counter++;
}
counter;
```

bpm

```
beatManager.setBpm(166);
beatManager.getBpm();
```

beatRate

```
beatManager.calculateBeatRate(120, "1/4");
```

status

```
markers.markers.length + " markers; " + slices.slices.length + " slices";
```

rename

```
var name = "newName";
re = /^newName/;
aeHelper.selectAllLayers(comp);
var counter = 0;
for(var i=0; i<comp.selectedLayers.length; i++){
  var layer = comp.selectedLayers[i];
  if(re.test(layer.name)){
    layer.name = name + "_" + i;
    counter++;
  }
}
counter;
```

createfile

```
var text = "";
var filePath = Folder.desktop.fullName + "/_default.txt";
var file = new File(filePath);
//var file = File.saveDialog("Choose a txt file","*.txt*", Folder.desktop);
if(file === null)
  file = File.saveDialog("Choose a txt file","*.txt*", filePath);
file.open("w");
file.writeln(text.toString());
file.close();
```

# Peacock Commands

A list of all tab peacock commands code snippets.

marker

```
markers.addCompMarker(comp, new Marker(10, { duration:0.0 }));
```

slice

```
slices.addCompSlice(comp, new Slice(5,10, { velocity:1.0 }));
```