

Here's a simple tutorial on how to make a Live Tab. Building a Live Tab is a relatively simple task assuming you know how to write an automation script. I'm going to assume that you know how to write an automation script and some of the basics of the DOM. If you need more information on that topic, check out the scripting handbook in your installation of LiveMotion. It is located at: LiveMotion 2.0\Scripting Resources\Documentation.

This tutorial will take one of the scripts that is installed in LiveMotion and convert that into a script. I'll go through the Star Maker because you guys already have the beginning and end pieces.

First, open up Star Maker.js from your Automation/Scripts folder. You can either use your own text editor or LiveMotion to view *.js files. If you look through this file quickly you'll see that this script has already been conveniently packaged into a function. This is the first step in taking a script into a Live Tab. For most cases, you'll want the action of the Live Tab to be inside of a function. It makes calling it easier later on.

This star function has four arguments: `star(inRad, outRad, thepoints, offsetPoints)`. That gives us a clue as to what kind of UI we will need to make. Now that we've seen the script, we'll put it aside for a minute. We'll come back to it when we're ready to put the engine behind our Live Tab. We have to build the chassis first.

Create a new composition. Open up the export palette (Window > Export). In the drop down select Live Tab. Then close this palette.

For the sake of this example, I won't go into building complex UI. We'll just be using simple text fields for our parameters. Create four dynamic text fields. Use whatever font you'd like and a border probably wouldn't be a bad idea. Make sure the input check mark is set. We'll need to give the text fields variable names. Use: `inRad`, `outRad`, `thePoints`, `offsetPoints`. You might also want to give it some initial values. So use the following:

```
inRad: 25
outRad: 45
thePoints: 5
offsetPoints: 0
```

Next, create a button that will apply the script. Just make sure it has a down state.

Ok, we've got all the major pieces of our Live Tab done. Let's just wire the whole thing together. Pop open the script editor for the Live Tab. Either use CTRL/CMD+J or Scripts > Editor from the menu.

Go to the onLoad event for the composition. Take the automation script that we had opened earlier and copy and paste the function. Here it is below:

```
function star(inRad, outRad, thepoints, offsetPoints) {
    comp = application.currentComposition;
```

```

        if (comp.selection.length < 1) {
            thestar = comp.createObject(LMGeometricType.path, comp.size.x/2,
comp.size.y/2, 0);
        } else {
            thestar = comp.selection[0].convertType(LMGeometricType.path);
        }

        comp.saveSelection()
        var degToRads = Math.PI/180;
        var outOffset = offsetPoints * degToRads;
        var anglePerPoint = 2*Math.PI/thePoints;
        var anglePerKnot = anglePerPoint/2 + outOffset;
        var angleOffset = Math.PI/2; // so a point is straight up
        var startAngle = 0 + angleOffset;
        var endAngle = 2*Math.PI + angleOffset;

        //the first point
        var path = thestar.addPath(outRad*Math.cos(startAngle), -outRad*Math.sin(startAngle),
0);

        while(thestar.paths.length > 1)
            thestar.deletePath(1);

        // the second point
        path.addKnot(inRad*Math.cos(startAngle+anglePerKnot), -
inRad*Math.sin(startAngle+anglePerKnot), 1);
        var currentAngle;

        for(currentAngle = startAngle+anglePerPoint; currentAngle < endAngle; currentAngle +=
anglePerPoint)
        {
            //outerPoint
            path.addKnot(outRad*Math.cos(currentAngle), -
outRad*Math.sin(currentAngle), path.knots.length);
            //innerPoint
            path.addKnot(inRad*Math.cos(currentAngle+anglePerKnot), -
inRad*Math.sin(currentAngle+anglePerKnot), path.knots.length);
        }
        path.closed = true;
        comp.restoreSelection();
    }
}

```

Ok, that will give us the engine to make our star. Now let's hook the engine to the starter, the button that we made earlier.

Go to the button's down state. Open up the script for the state. Type the following code:

```

this.inRad = parseInt(_root.inRad);
this.outRad = parseInt(_root.outRad);
this.thePoints = parseInt(_root.thePoints);
this.offsetPoints = parseInt(_root.offsetPoints);

_root.star(this.inRad, this.outRad, this.thePoints, this.offsetPoints);

```

The button will call the `star()` function with the values that are in the dynamic text fields. You may have noticed that I use the `parseInt()` method. The reason why I do this is because all text fields are considered strings. With `parseInt()` I convert the string into integer values. The variables that I'm using `this.inRad`, `this.outRad`, etc. are basically temporary variables to hold my numerized values.

Now save this file somewhere. If you save it in your Automation/Live Tabs folder it will appear in the drop down menu after either running the refresh menu command (Automation > Refresh Menu) or restarting LiveMotion.

That's it! It's a pretty easy path to go from script to Live Tab. Try embellishing this by adding buttons to increment the values, or make another script into a Live Tab.

