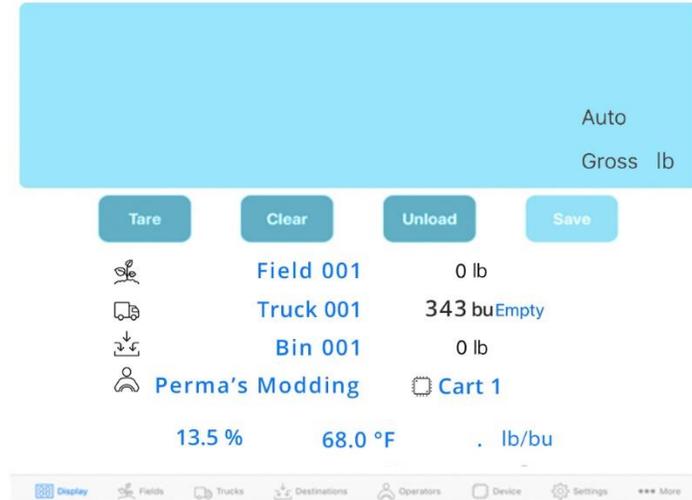


# Fill Level Dashboard



---

## *What is it?*

---

The Fill Level Dashboard is a script based specialization that can be added to a vehicle to display the fill level, weight of the filltype based upon the current fill level and the weight per volume. If you use the unit conversion mod by ThundR, when setting up the vehicle .xml you can specify which units the script should use.

---

## *What's Included?*

---

Included in the zip are some pre-made screens, as well as templates which can be modified using Photoshop or GIMP (GNU Image Manipulation Program). These templates are based off the iFarm Grain Cart Scale interfaces. There is also a blank template which you can use to create your own screen display. Also included is a monitor in a vertical and horizontal preset. (More to be added in the future)

---

## How to Add It

---

In order to add the script into your vehicle you will need a program capable of editing xml files as well as Giants Editor 8.

### Step 1. Changes to moddesc.xml

In order for the script to function, we need to first add it as a specialization in our modDesc, this requires also creating a custom vehicle type. For demonstration purposes, we'll be adding the script into Custom Modding's 8R US Series tractor.

In the modDesc.xml we'll need to either add or edit two sections, the first being specializations, this is where we can add new specializations for our vehicle type. For the game to recognize our script, we need to add the following line into the specialization section:

```
<specializations>
  <specialization name="fillLevelDashboard" className="FillLevelDashboard" filename="FillLevelDashboard.lua"/>
</specializations>
```

Next we need to define a new vehicle type so that we can allow it to use our specialization. To add a new vehicle type we can use the following code:

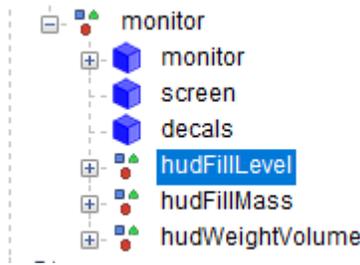
```
<vehicleTypes>
  <type name="customType" parent="baseDrivable" className="Vehicle" filename="$dataS/scripts/vehicles/Vehicle.lua" >
    <specialization name="fillLevelDashboard"/>
  </type>
</vehicleTypes>
```

**(If a custom vehicle type is already being used, then we can just add the specialization code itself)**

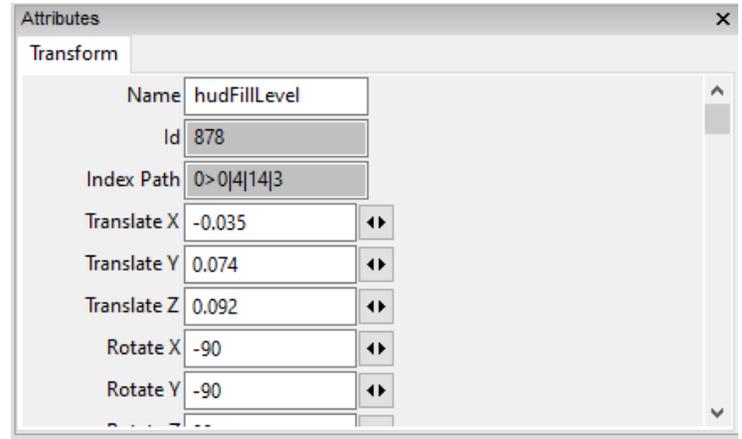
### Step 2. Giants Editor

The next step in the process is adding the monitor to the vehicle in Giants Editor. To do this first decide which monitor you wish to use, for this guide, we'll be using the horizontal TabPad monitor, copy the monitor i3d file, its corresponding .shapes file and the monitorTextures folder into the root folder of the vehicle. Now open the vehicle i3d in Giants Editor. Once the i3d is loaded, click on file -> import and choose the i3d of the monitor you chose to use. After importing, position the monitor using the transform tools in Giants Editor. You will also need to move it into the hierarchy of the scenegraph. I find it best to move it into position first and then use the middle mouse button and drag it to where you want it in the scenegraph. (This automatically changes the location and rotation of the object so that it keeps the location you moved it to.)

After positioning it, you will need to note the location of the index path of the 3 hud related transform groups and the screen.



You can find the index path in the attributes panel;



In the case of this guide, the index paths are;

- 0>0|4|14|1 for screen
- 0>0|4|14|3 for hudFillLevel
- 0>0|4|14|4 for hudFillMass
- 0>0|4|14|5 for hudWeightVolume

### Step 3. Changes to vehicle.xml

The next step is to make changes to the vehicle.xml itself. The first step is to change the vehicleType. The vehicle type is usually defined on the second line of the xml file. Remember the vehicleType we created in Step 1, this is where we specify the mod to use the new vehicleType.

After specifying the vehicle type we need to add some nodes to the i3dMapping of the mod, if your mod does not have an i3dMapping we can use the following code for it, otherwise, just use the <i3dMapping/> lines:

```
<i3dMappings>
  <i3dMapping id="screen" node="0>0|4|14|1" />
  <i3dMapping id="hudFillLevel" node="0>0|4|14|3" />
  <i3dMapping id="hudFillMass" node="0>0|4|14|4" />
  <i3dMapping id="hudWeightVolume" node="0>0|4|14|5" />
</i3dMappings>
```

We should add the nodes above to the dashboard section of the vehicle so that they only show up while the engine is active, that code will look like this:

```
<dashboard>
  <groups>
    <group name="MOTOR_STARTING" isMotorStarting="true"/>
    <group name="MOTOR_ACTIVE" isMotorStarting="true" isMotorRunning="true"/>
  </groups>
  <default>
    <dashboard displayType="VISIBILITY" node="screen" groups="MOTOR_ACTIVE"/>
    <dashboard displayType="VISIBILITY" node="hudFillLevel" groups="MOTOR_ACTIVE"/>
    <dashboard displayType="VISIBILITY" node="hudFillMass" groups="MOTOR_ACTIVE"/>
    <dashboard displayType="VISIBILITY" node="hudWeightVolume" groups="MOTOR_ACTIVE"/>
  </default>
</dashboard>
```

We have one more section to go, this is where we tell the script the information that it needs to be able to function. You can add this section any place you wish to in the xml but I generally tend to put it just above the i3dMappings section. Below is the code as well as a bit of explanation on how to use it correctly:

```
<fillLevelDashboard volumeUnit="BUSHEL_US" weightUnit="POUND">
  <dashboards>
    <dashboard valueType="fillLevel" displayType="NUMBER" precision="0" numbers="hudFillLevel" groups="MOTOR_ACTIVE" />
    <dashboard valueType="fillWeight" displayType="NUMBER" precision="0" numbers="hudFillMass" groups="MOTOR_ACTIVE" />
    <dashboard valueType="weightVolume" displayType="NUMBER" precision="1" numbers="hudWeightVolume" groups="MOTOR_ACTIVE" />
  </dashboards>
</fillLevelDashboard>
```

In the first line we have two tags, volumeUnit and weightUnit, these tell the script that if UnitConvert is active, use the values from these units. The valueType is telling the script which function should be used. The precision dictates how many decimal points to use in our number, by default, the fill level and weight are whole numbers, while the weight per volume is set to one decimal place. The numbers tag dictate which transform group's numbers should be affected by this function. The groups tells the script that it should only be running while the engine is on.

While you can specify your own units in UnitConvert, the following are the default units:

- BUSHEL\_US
- BUSHEL\_UK
- GALLON\_US
- GALLON\_UK
- TON\_SHORT
- TON\_LONG
- POUND

---

## Credits

---

ThundR: Scripting

Perma's Modding: Small scripting, Creative Thought, Modeling, Texturing

Klock32: Initial push of idea

Trailer Park Farms: Pushed me in the right direction in getting 4D Modding's baler script to function for a in cab fill level