

Labels Tutorial

1. Introduction
2. Step-By-Step Model Construction

Introduction

This tutorial will introduce you to using Labels in your model. Labels can be used to store information in your objects and items that can be accessed at anytime. As a couple of examples, labels can be useful to specify the downstream flow of your items, or they can be used to store financial data about an object.

What You Will Learn

- How to create and access labels
- How you can use labels to change the way your model functions

New Features

- You will be introduced to Pull Requirements
- You will be introduced to the FlowItem Bin

Approximate Time to Complete this Lesson

This lesson should take about 20- 30 minutes to complete.

Model Overview

In this model we will create a simple model that creates items with three different itemtypes. We will then track how many items we create of each itemtype and then upon processing the item, we will modify the size of our items.

[Click here for the Step-By-Step Tutorial.](#)

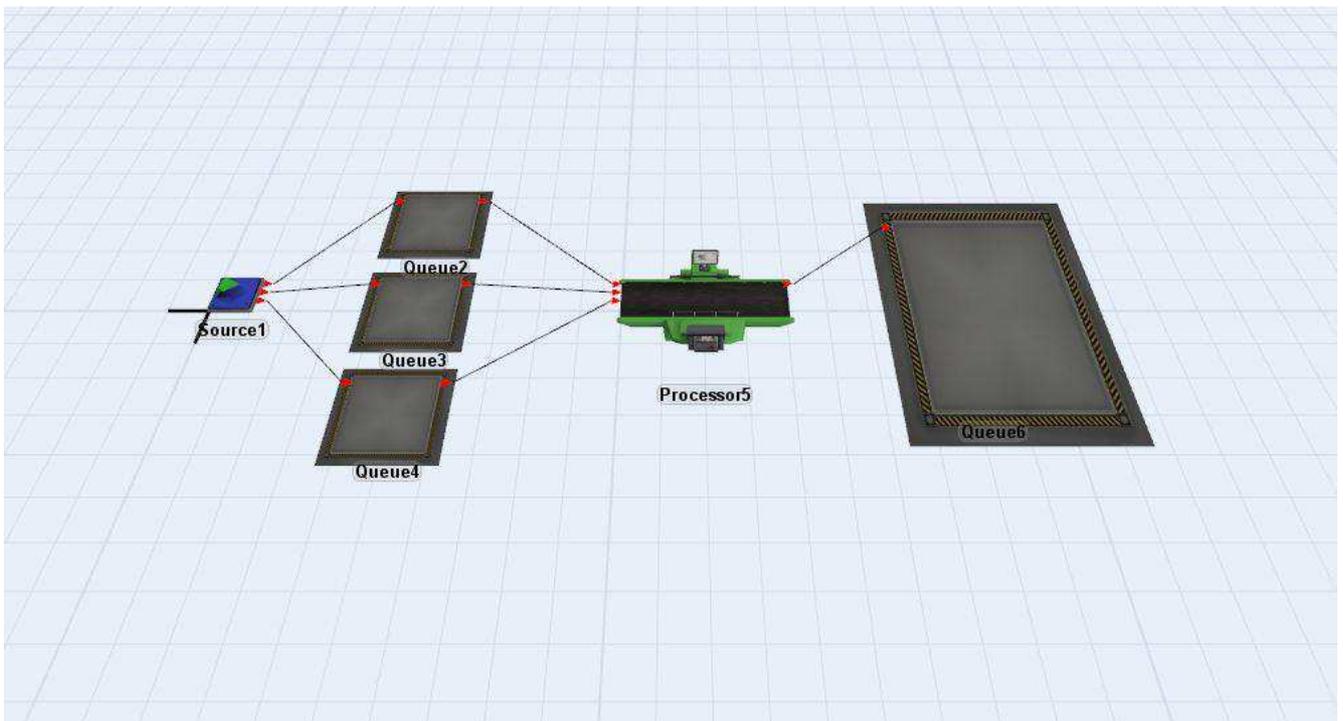
Step-By-Step Model Construction

Building Labels Model

Begin a new model by clicking the  button on the toolbar. Click **OK** on the Model Units window – we will use the default units for our model.

Step 1: Create the Objects

- Drag objects from your Library Icon Grid onto the 3D View to create the model shown below.



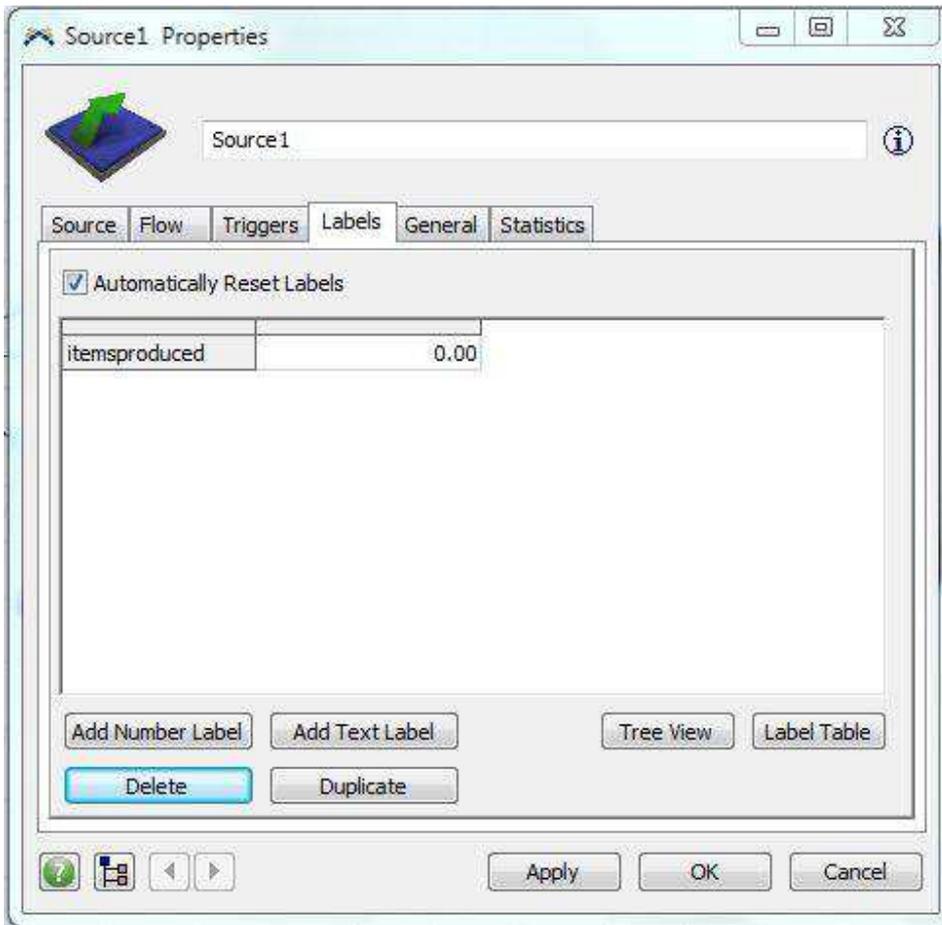
Connect all of the objects as shown:

- Connect *Source1* to *Queue2* *Queue3* and *Queue4*
- Connect *Queue2*, *Queue3* and *Queue4* to *Processor5*
- Connect *Processor5* to *Queue6*

Step 2: Setup the Source Properties

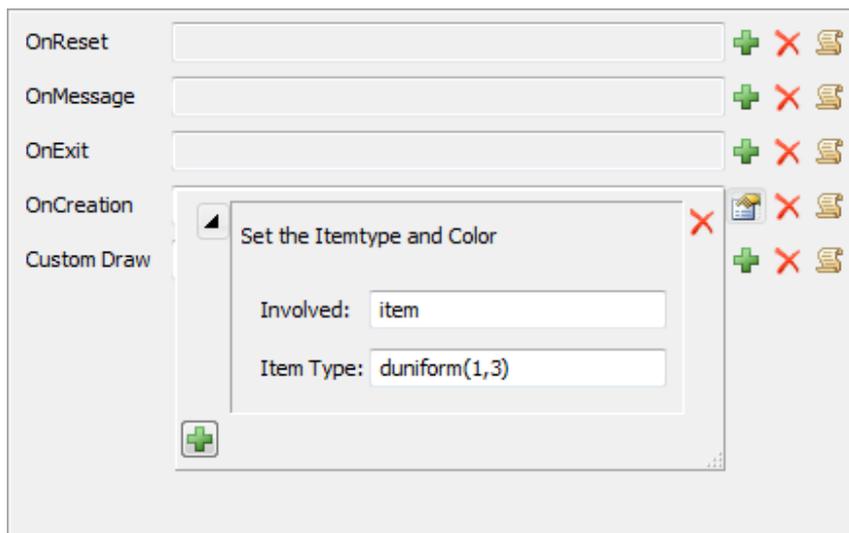
All objects can store their own Labels. The Labels tab will allow you to create, view and manipulate the labels on that object.

- Double-click on the *Source* to open its properties window.
- Go to the **Labels** tab.
- Check **Automatically Reset Labels**. This will cause the labels to reset to the value you give it when you reset the model.
- Click **Add Number Label** and double click "*newlabelname*" to rename the label "*itemsproduced*".



The *Source* will create items with 3 itemtypes. Each itemtype will be assigned a specific color.

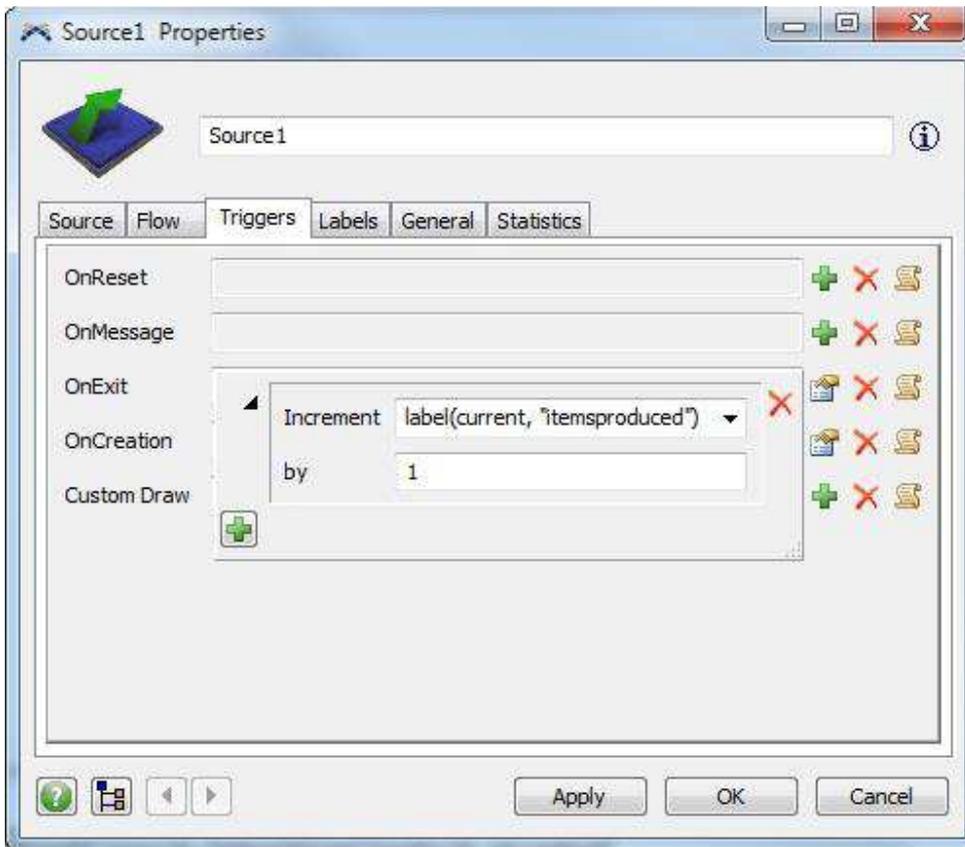
- Go to the **Triggers** tab.
- Click the add **+** button for the **OnCreation** trigger, and select the *Set Item Type and Color* pick option.
- We will be using 3 itemtypes, so keep the **Involved** and **Item Type** at their default values.



We also want to increment the *"itemsproduced"* label to keep track of how many items have been produced by the source.

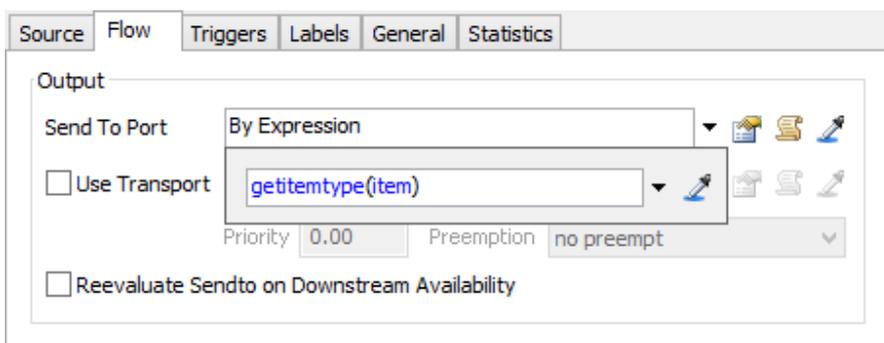
- Go to the **Triggers** tab.

- Click the add **+** button for the **OnExit** trigger, and select the **Increment Value** pick option.
- In the edit box for **Increment** replace *label(item, "labelname")* with *label(current, "itemsproduced")*.
- Leave the **By** at 1. This will increment the value of the "itemsproduced" label by 1 on each time an item leaves the Source.



We will also assign items of each itemtype to go to a specific queue based on the itemtype.

- Go to the **Flow** tab.
- Choose **By Expression** in the **Send to Port** option. Leave it at the default *getitemtype(item)*.



- Click **Apply** to apply the changes you made to the *Source*.

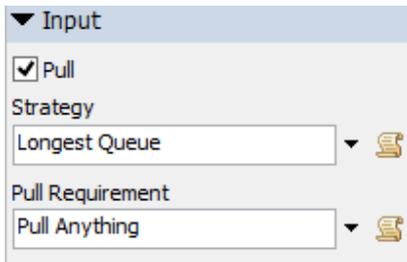
Reset and run your model. If you go back to the **Labels** tab of the source and watch your "itemsproduced" label, it will update the value as each new flow item is created.

Close the properties window of the Source. It may ask you if you want to "Update Label Reset Values?". Click No. (You want the labels to reset to zero when you reset the model.)

Step 3: Setup the Processor Properties

We want the Processor to grab items from the Queue that has the most items waiting in, or the **Longest Queue**. Since each of the Queues do not know how many items are in the other Queues, it works best to have the Processor determine which Queue it will pull an item from.

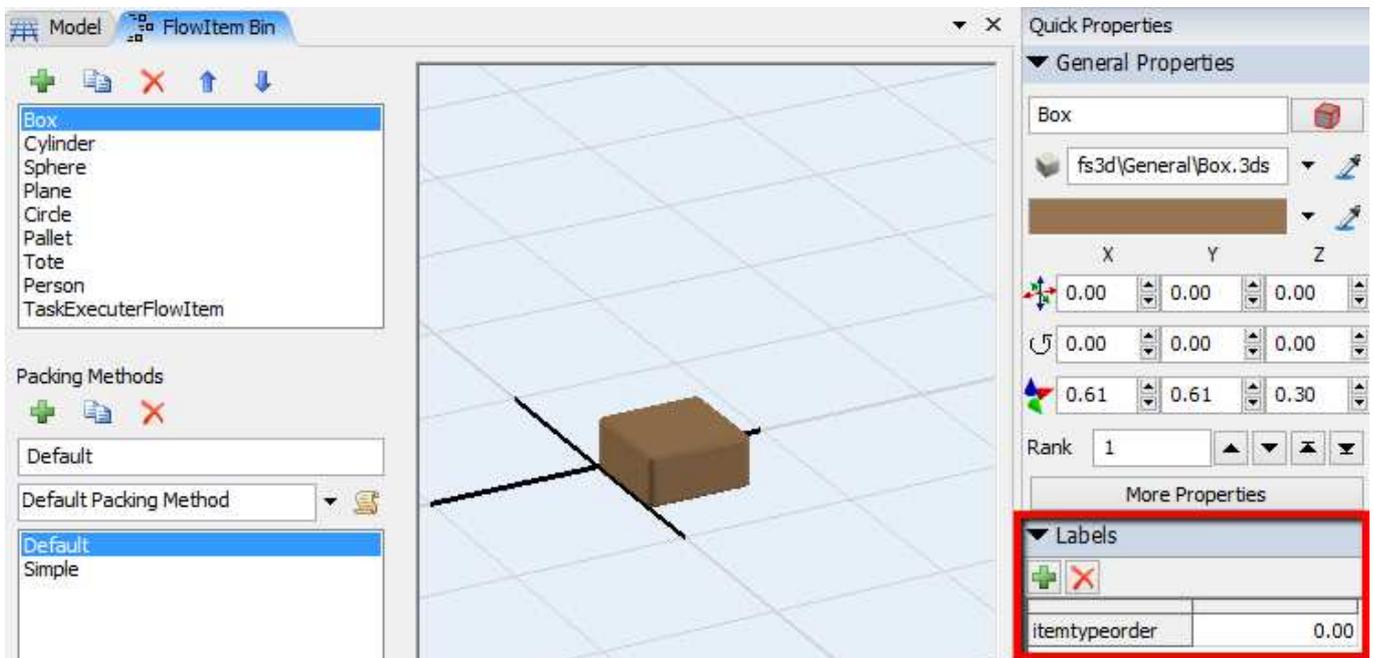
- Click on the *Processor* to open its properties in the Quick Properties window.
- Click the arrow next to the **Input** section to expand it.
- Check the **Pull** option and set the **Pull Strategy** to *Longest Queue*.



Step 4: Customizing Flowitems

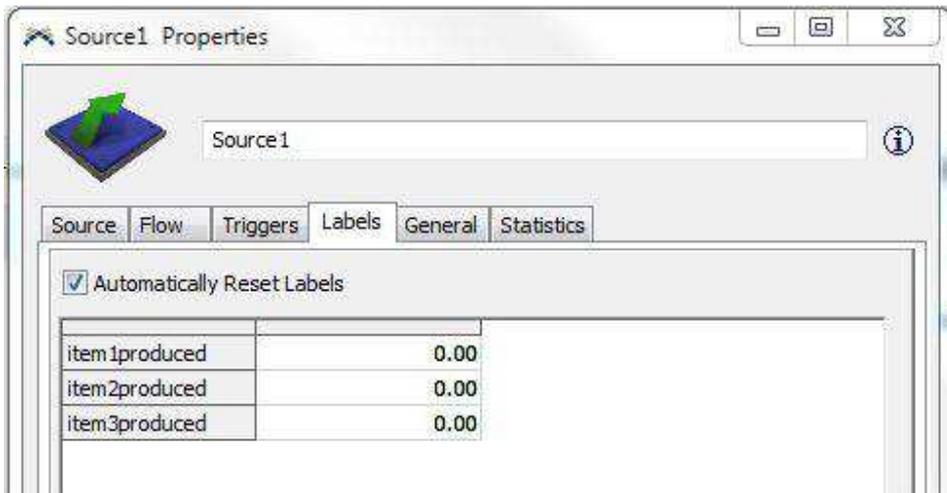
We are going to store information on all the flowitems so that they can be accessed throughout the model.

- Open the **Flowitem Bin** by going to **Tools > Flowitem Bin**
- With **Box** selected click the **+** under the Labels section of the Quick Properties window and add a number label.
- Double click "*LabelName*" to rename the label "*itemypeorder*".



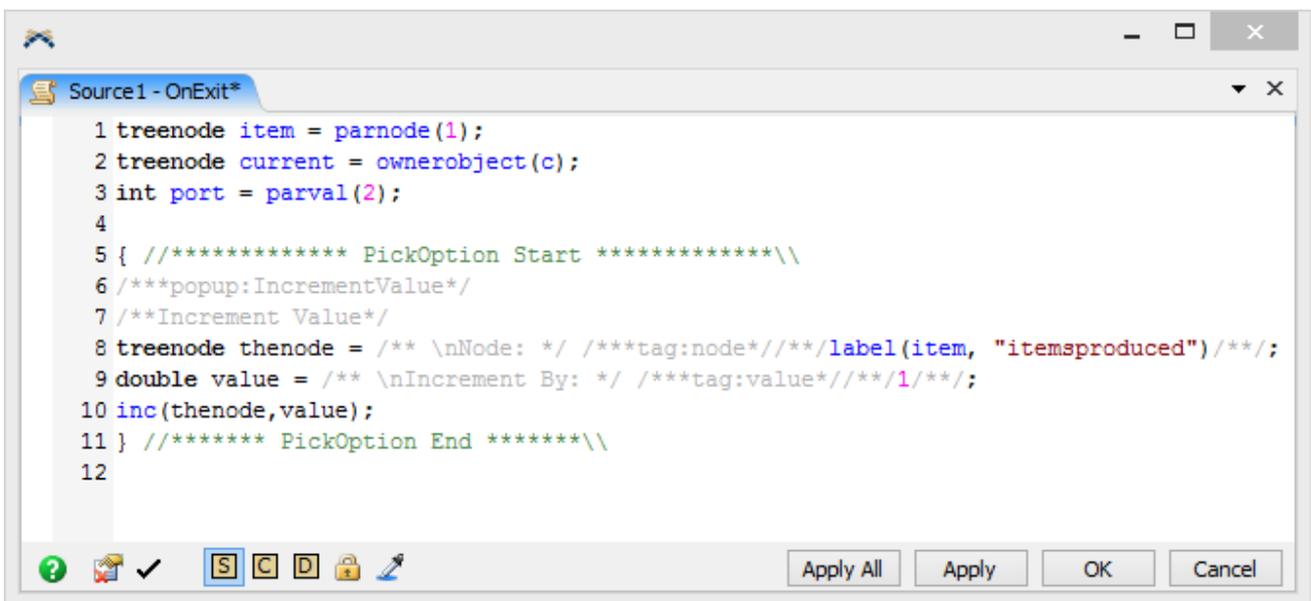
We will now have the source increment the "*itemypeorder*" label on each flowitem for each specific type of item. This will allow us to know in what order the flowitem was created.

- Open the properties window for the *Source*.
- Go to the **Labels** tab.
- Change the label "*itemsproduced*" to "*item1produced*".
- Select the "*item1produced*" label and hit the **Duplicate** button twice.
- Rename the two new labels to "*item2produced*" and "*item3produced*".



In order to increment each these labels correctly, we need to customize the OnExit trigger. This will require editing the code manually, but don't worry – writing code is easy!

- Go into the Triggers tab, and click on the  button, located to the left corner of the OnExit Trigger, to open the **Code Editor**.



- Click on the remove template  button to get rid of the comments.
- Feel free to delete the "PickOption Start/End" lines.

```

1 /**Custom Code*/
2 treenode item = parnode(1);
3 treenode current = ownerobject(c);
4 int port = parval(2);
5
6 { /******* PickOption Start *****\\
7
8
9 treenode thenode = label(item, "itemsproduced");
10 double value = 1;
11 inc(thenode,value);
12 } /******* PickOption End *****\\
13

```

- We want this "increment" function to increment each label individually. Write an "if" statement matching each itemtype to its own label. We will also set the new label's value to the flowitem.

```

if (getitemtype(item) == 1)
{
    int value = inc(label(current, "item1produced"), 1);
    setlabelnum(item, "itemtypeorder", value);
}

```

- Write the same statements for itemtype 2 and 3. The code should look like this when you are done:

```

1 /**Custom Code*/
2 treenode item = parnode(1);
3 treenode current = ownerobject(c);
4 int port = parval(2);
5
6 if (getitemtype(item) == 1)
7 {
8     int value = inc(label(current, "item1produced"), 1);
9     setlabelnum(item, "itemtypeorder", value);
10 }
11 if (getitemtype(item) == 2)
12 {
13     int value = inc(label(current, "item2produced"), 1);
14     setlabelnum(item, "itemtypeorder", value);
15 }
16 if (getitemtype(item) == 3)
17 {
18     int value = inc(label(current, "item3produced"), 1);
19     setlabelnum(item, "itemtypeorder", value);
20 }

```

- Click **OK** to apply and close the Code window.
- Click **OK** to apply and close the Properties window.

Step 5: Process the Flowitems

In this step, we will change the way the *Processor* processes the boxes.

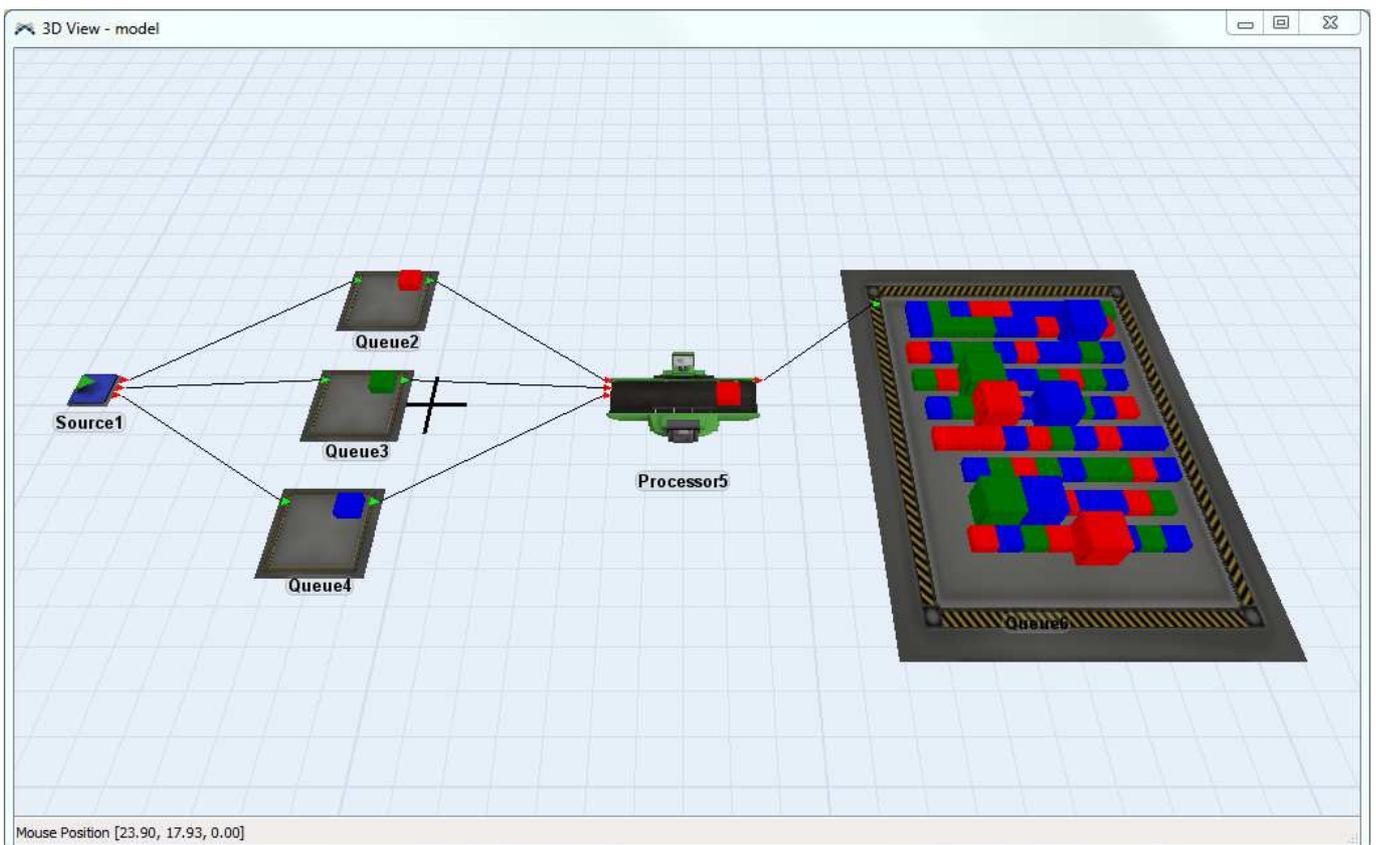
- Open the *Processor's* Properties window.
- Go to the *Triggers* tab.
- Click the code edit button  next to the *OnProcessFinish* trigger.
- Write an if statement that will only allow every tenth item of each type to be set to a specific size:

```
if (getlabelnum(item, "itemtypeorder") % 10 == 0)
{
    setsize(item,1,1,1);
}
```

Note: You don't need to do this two more times, because it applies to any flowitem with the "itemtypeorder" label.

- Click **OK** on the Edit Code window as well as the Properties window.

Reset and **Run** your model. Your model should begin to look something like this:



This completes the the Labels tutorial. Congratulations!