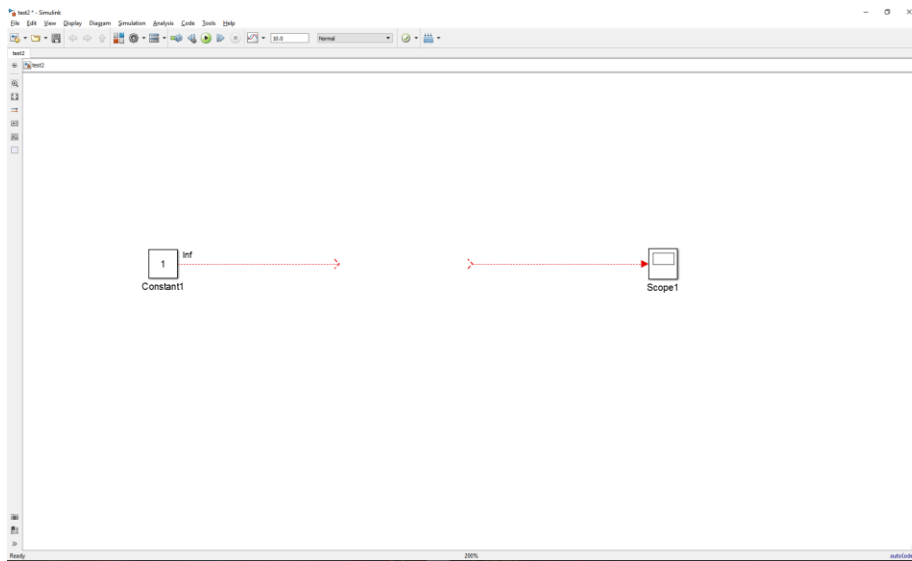


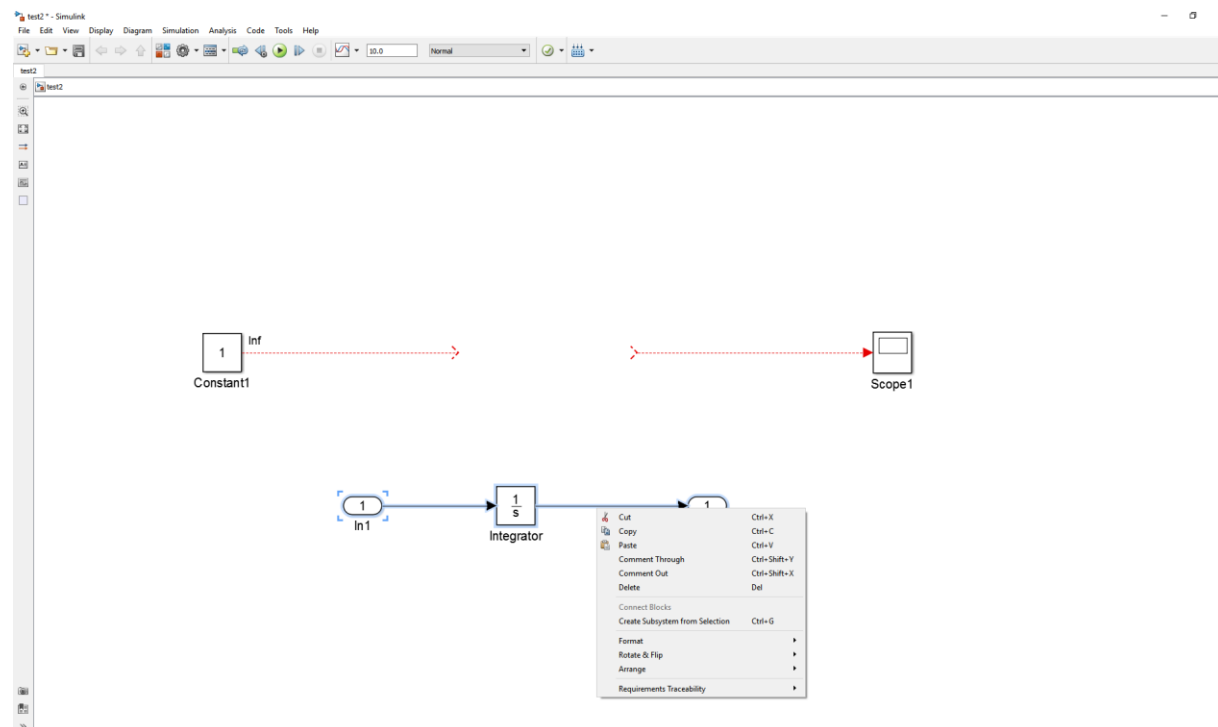
In this example, we will make use of our top-level model, “test2.slx”



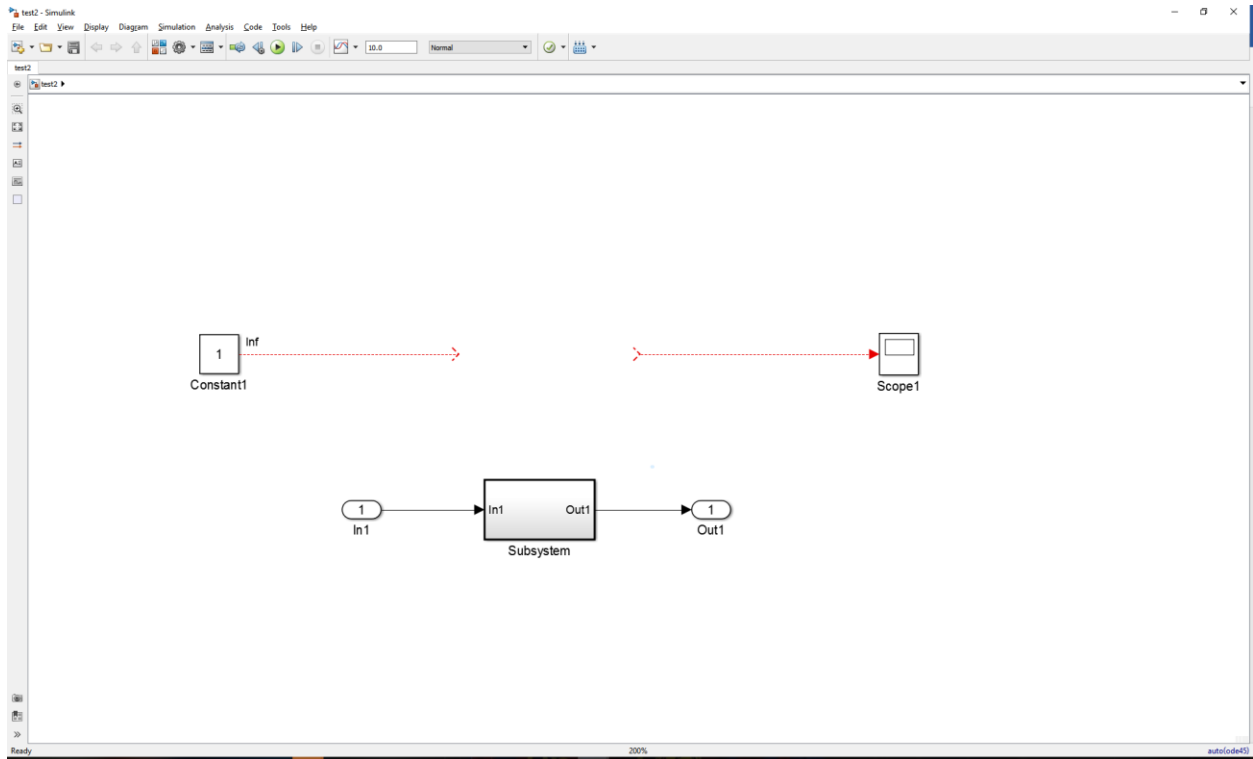
We will connect the source with the scope using a model reference that we create out of an atomic subsystem.

Step 1

Create a simple subsystem out of an “integrator” block as follows:



After making the subsystem atomic, you should see something like this:



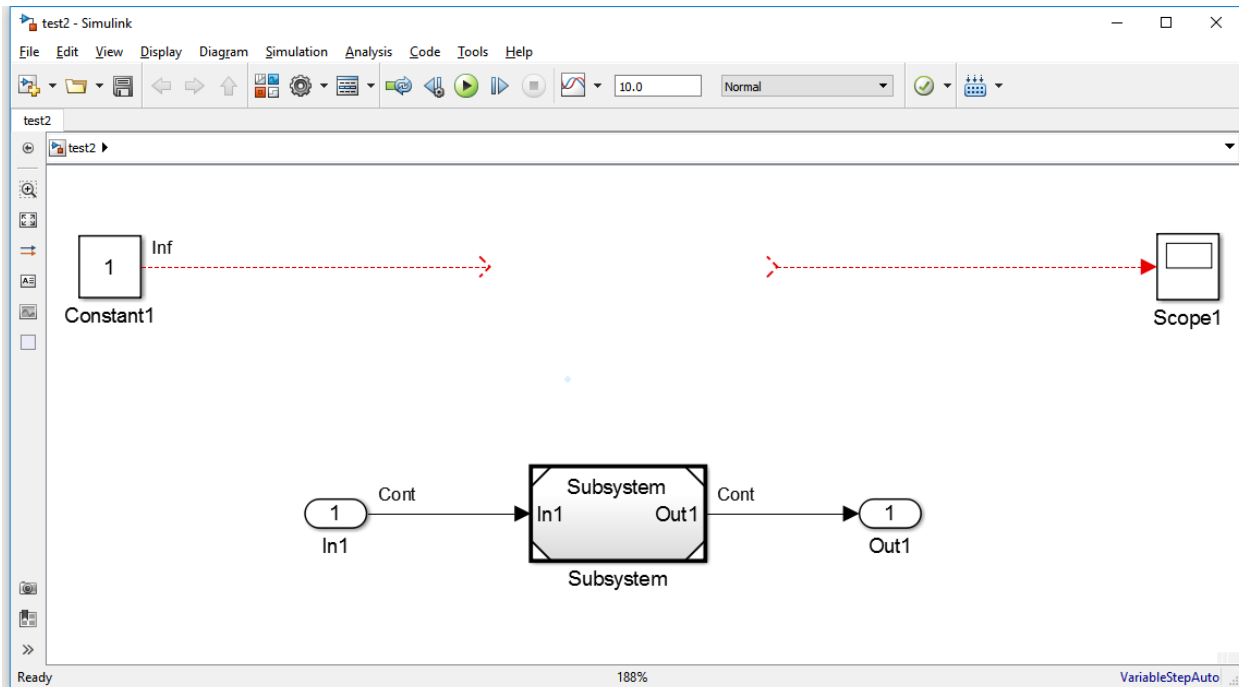
Step 2

Create a model reference for the Subsystem the same way as your earlier workflow.

Step 3

In your working directory, you should see two SLX files:

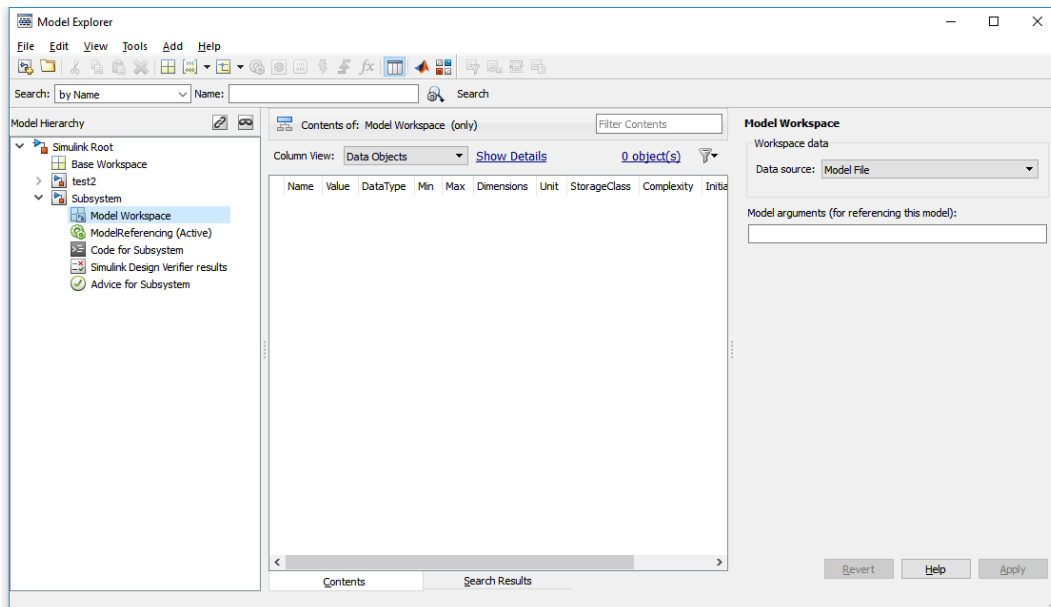
- 1) "test2.slx". This is your top-level model.
- 2) "Subsystem.slx". This is a new model created, in which the "Subsystem" model reference block in "test2.slx" is referencing.



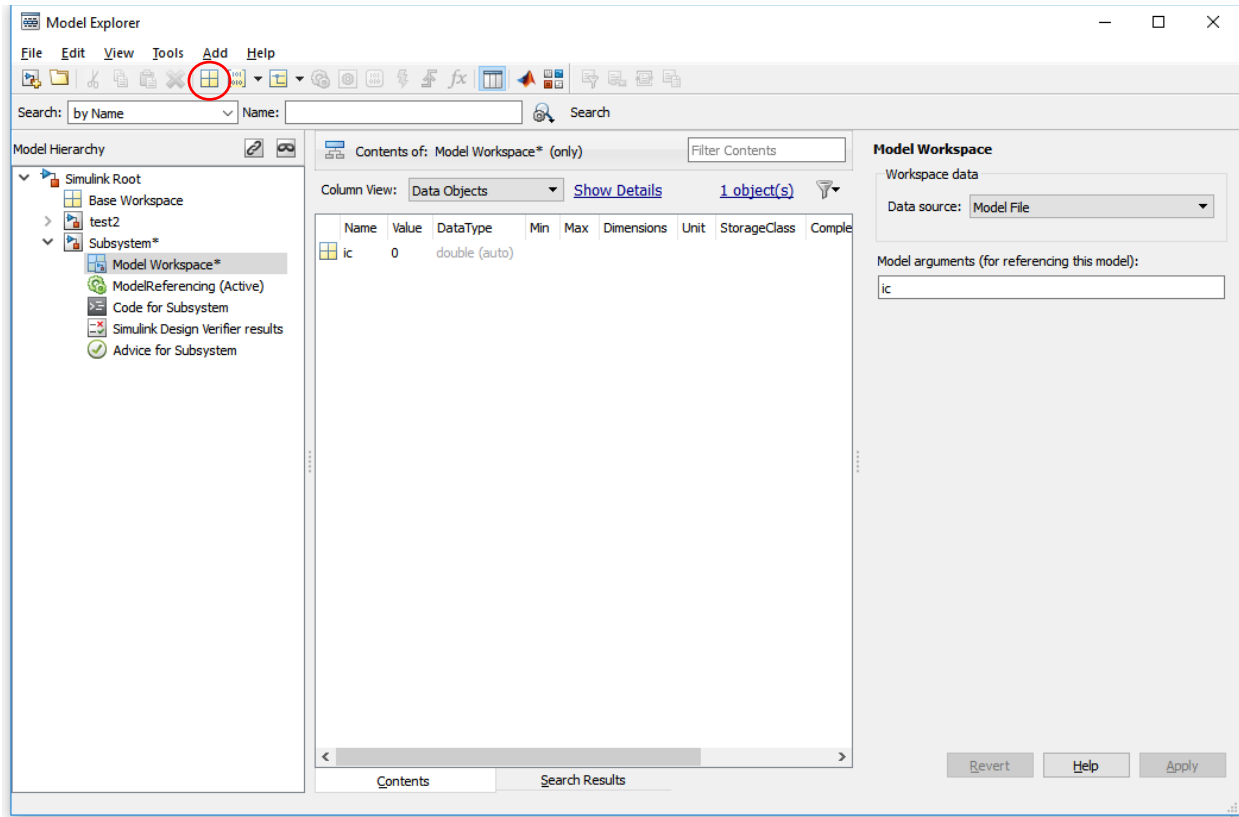
Step 4

This is a key step which is different from your original workflow. We are now trying to parametrize the initial condition of the Integrator block so that other users can change this value even if the model reference is protected.

Open "Subsystem.slx". Press CTRL + H to bring up the Model Explorer. Expand "Subsystem" and select "Model Workspace", as shown in the screenshot below:



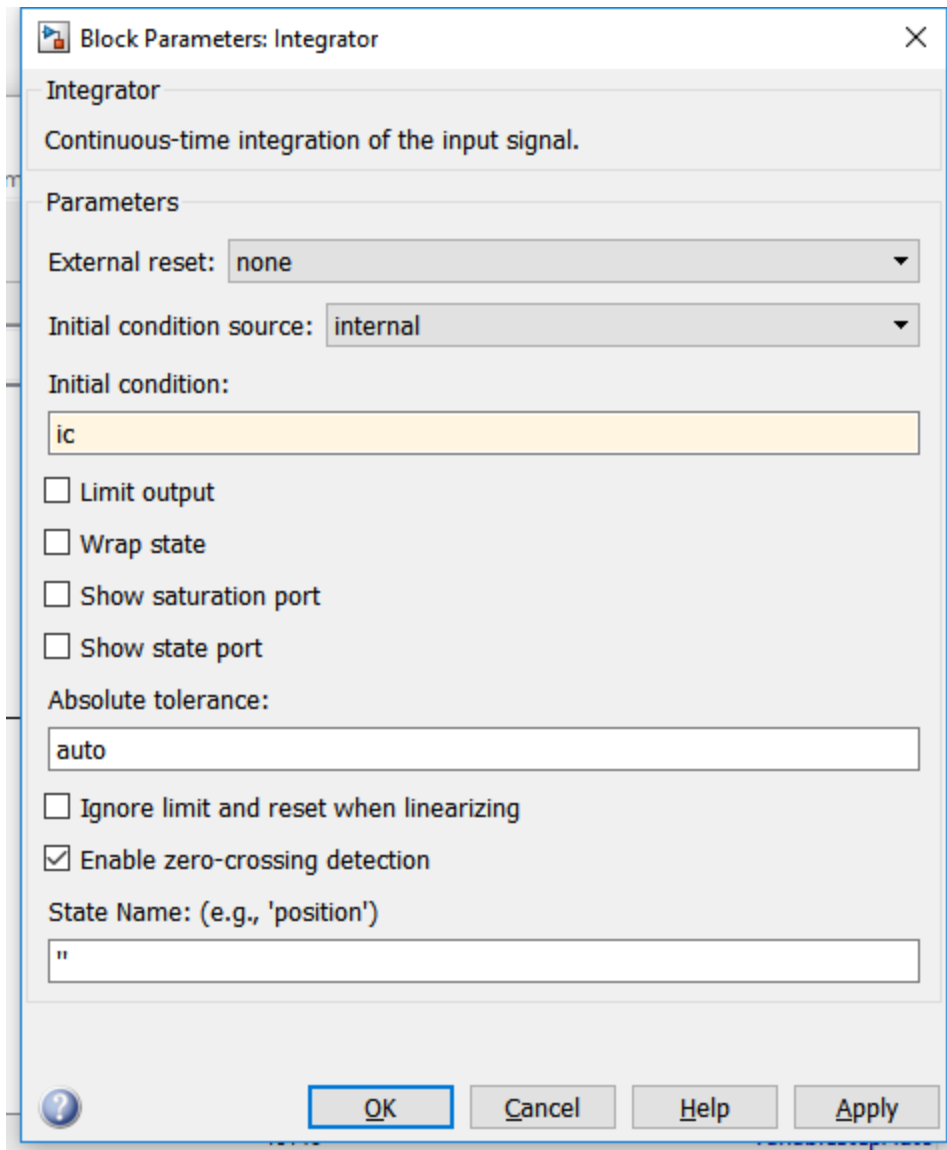
In the edit box under “Model arguments (for referencing this model):” which is on the right-hand side of the model explorer, type “ic”. Hit “Apply”. Then, in the toolbar, select the grid icon just beneath “Add” menu to add a MATLAB variable into the model workspace. Double-click on the name and change it to “ic”. Hit “Apply” again.



We have just added the variable “ic” as a model argument, in which one can assign any value to it.

Step 5

Now we need to link this variable to the parameter we want to control, which is the initial condition of the integrator. To do this, close the Model Explorer and double click on the Integrator block to bring up the block parameters dialog box. Under Initial Condition, instead of “0” (or some other number), change this to “ic”.



And then hit “Ok”. Save the “Subsystem.slx” model.

Step 6

Now we are ready to create a protected version of this model. Instead of using the dialog box in which you are missing the “read-only view” option, we can issue a command in the MATLAB Command Window.

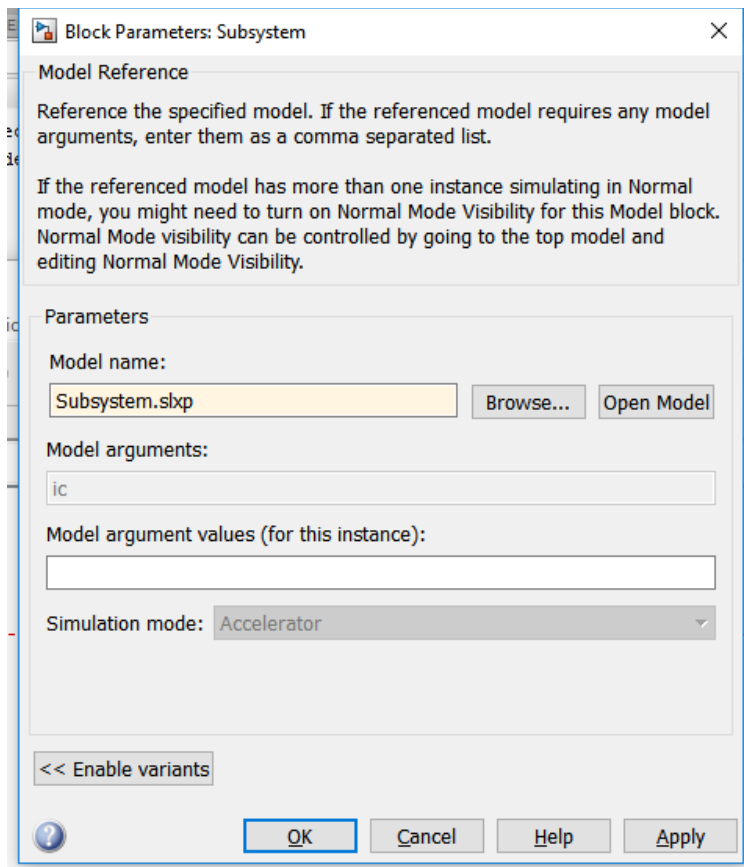
```
>> Simulink.ModelReference.protect('Subsystem', 'Webview', true)
```

This command creates a protected model that is being referenced that allows other users to both view and simulate the model reference in the top-level model. This protected model created is called “Subsystem.slxp”.

Note: if there is an error in this step, then please carry on with your original workflow of checking the “Simulate” option in the dialog box for model protection. This means that somehow “Webview” functionality is not enabled. We can dig into this later.

Step 7

Going back to the “test2.slx”, you will notice that the “Subsystem” block has black edges as it is now automatically referencing to “Subsystem.slxp” and not “Subsystem.slx”. However, just so that there is absolutely no confusion, right-click on the Subsystem model reference, and select “Block Parameters (ModelReference)”. In the ensuing dialog box under Model name, add the .slxp extension to “Subsystem”



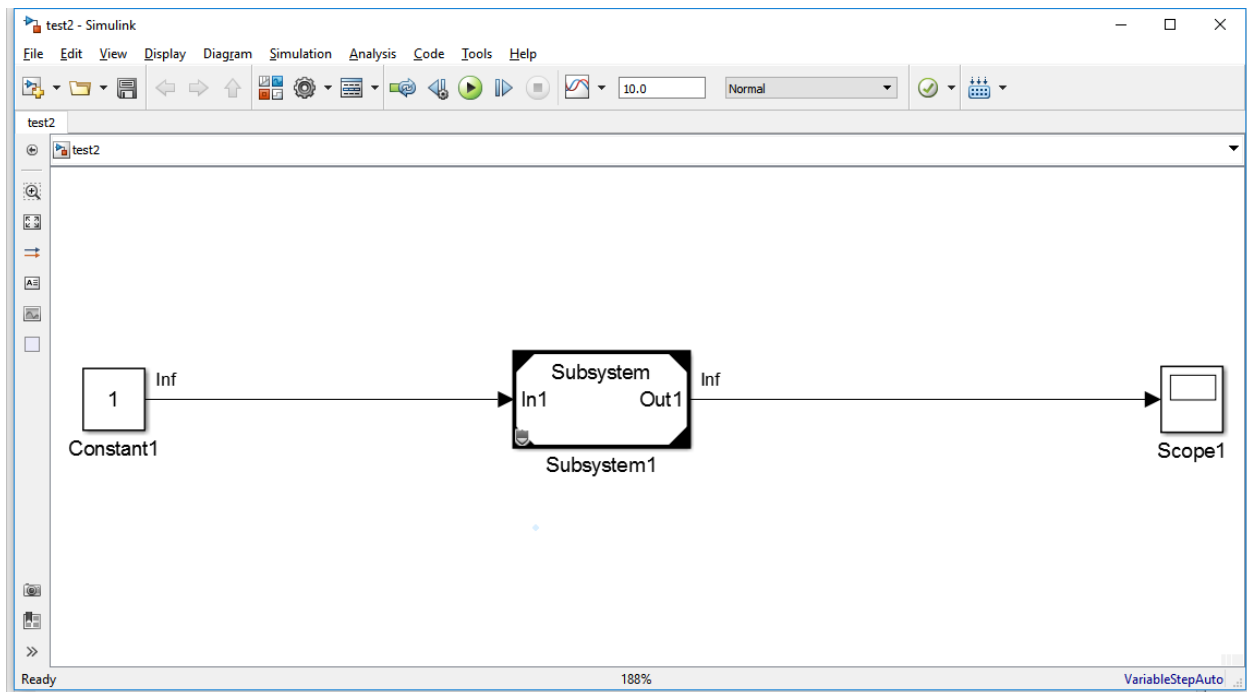
Tip: if you specify the “Model name” as “Subsystem.slx”, it will reference the unprotected model.

Note that under “Model arguments”, there is the “ic” variable.

Hit “Okay”.

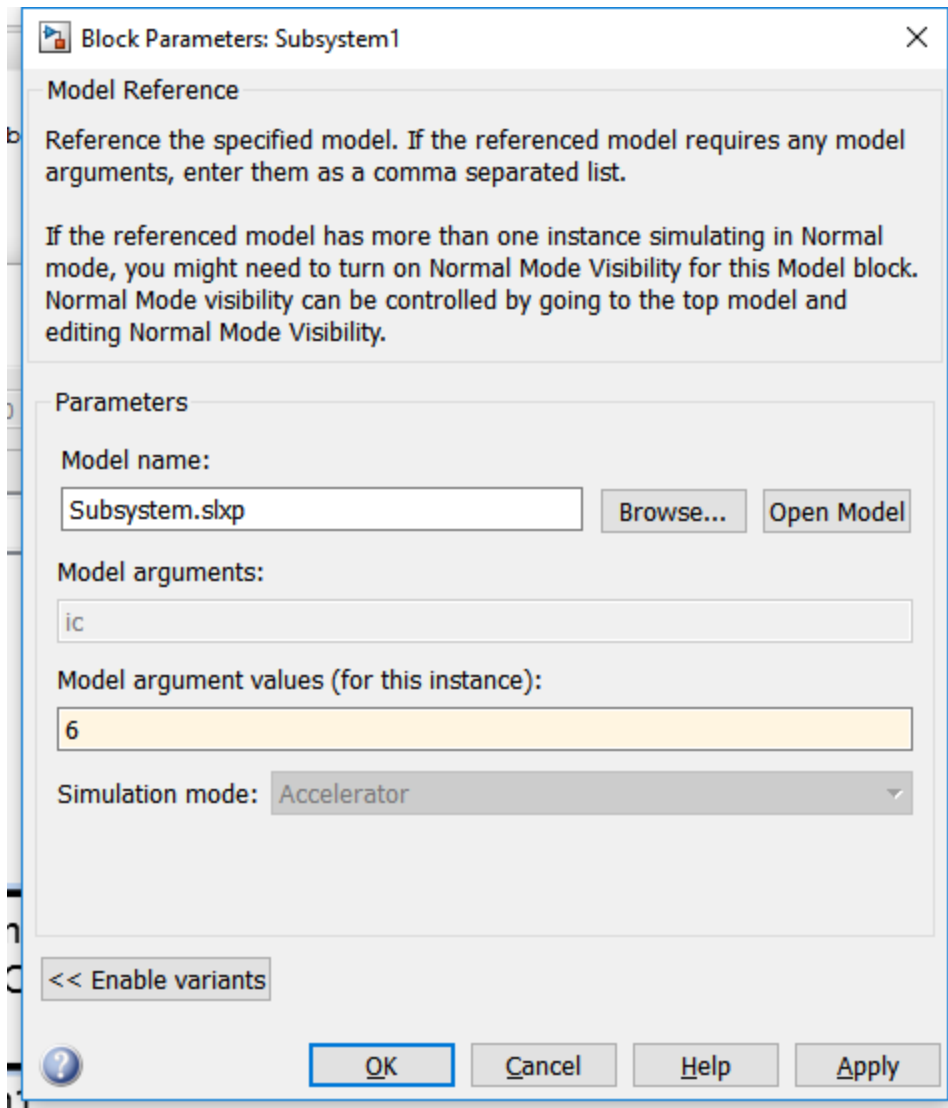
Step 8

Still in “test2.slx”, copy and paste the “Subsystem” model reference block to connect the Constant block and the scope, and delete the bottom set of blocks so that your “test2.slx” looks like this:



Step 9

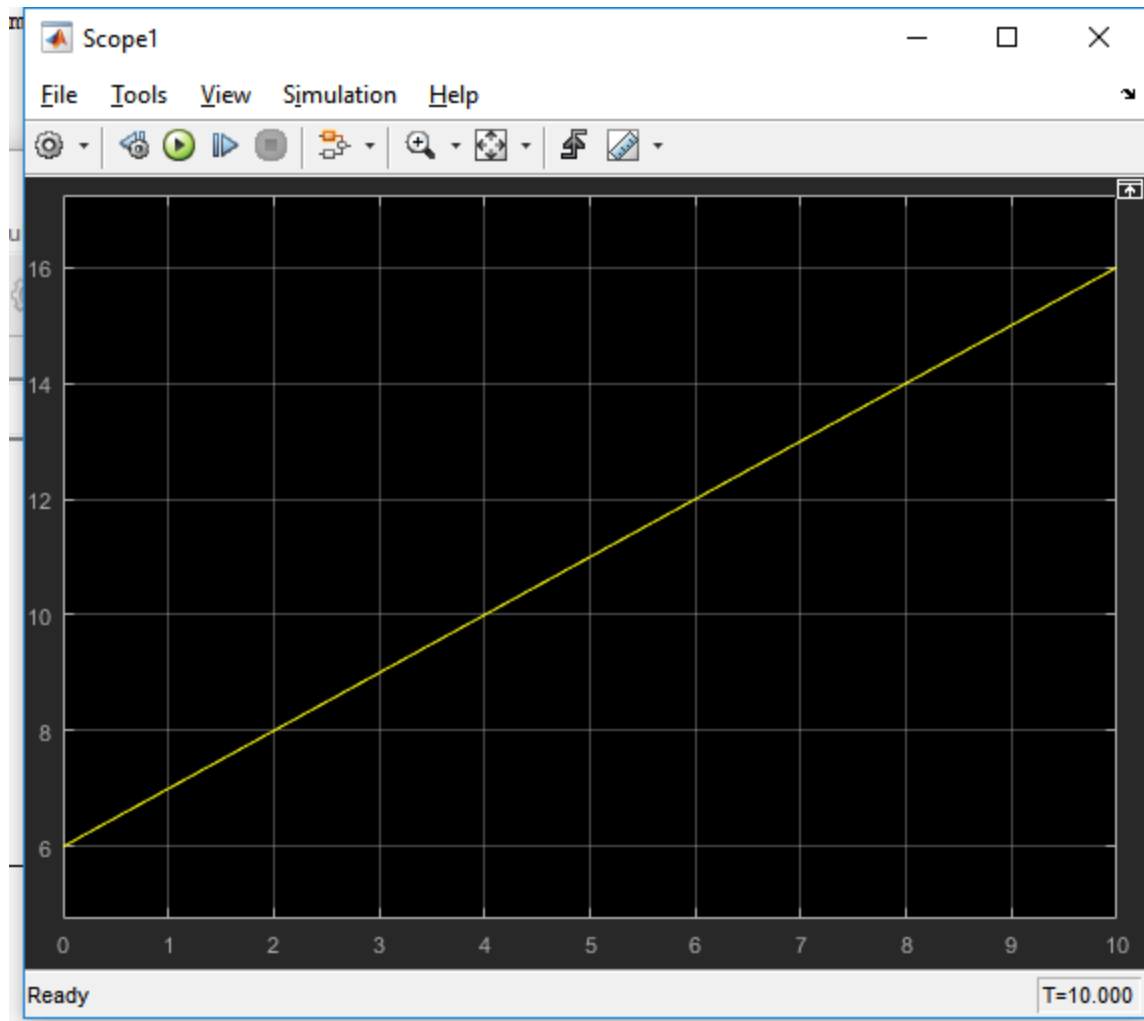
Open up the Block Parameters for the “Subsystem1” block again, and specify a value for “ic” under “Model argument values (for this instance):



Here I specified the value "6". Hit "OK".

Step 10

Save and run the model. You can see the expected result as show below:



Furthermore, you may share this protected model reference with different people, of which each can specify a different value for "ic", and each will obtain a different result. This adds flexibility to the simulation despite each model reference pointing to the same protected model.

In addition, if you double-click on the "Subsystem1" model reference block, a Webview of the inside is displayed, which is the same outcome as if you could check the "read-only view" option in the dialog box. Again, if you received error earlier, please let me know and we can work this out further.