

Visual Paradigm Standard Edition Tutorial

Theodore Norvell

Updated for VP 12.1 winter 2016

With annotations by Andrew Vardy

Conventions: When I say “click” I mean with the left mouse button, unless otherwise indicated. “Drag” means hold down the left mouse button and move the mouse; “drop” means release the mouse button. “Select” means click on the indicated item.

The following tutorial is for Windows. Adjust as needed for OSX and Linux.

1 Class diagrams and basics.

On the start menu, find Visual Paradigm and select it.

The workspace launcher should appear. Select a workspace in place that is regularly backed up.

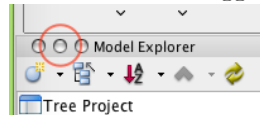
- If the workspace launcher does not appear, use Help >> Switch Workspace.
- If using LabNet, use a folder on your H: drive
- If you use Dropbox or a similar product, consider using a folder that will be automatically backed up.

Create a new project (Project >> New). Name it ‘Tree Project’ with ‘Data type set’ as UML. Save the project to your workspace folder.

Create a new class diagram (Diagram >> New). Call it Tree Classes.

Show the Model Explorer pane (View >> Panes >> Model Explorer).

On the Model Explorer pane, find and click the “Toggle auto-hide” button.



On Macs it is the middle circular button at the top. This should let you see the Model Explorer and the whole of the Tree Classes diagram at the same time.

On the palette to the left of the diagram you should find a tool named Package.

Click on the Package tool then on the diagram.



Name the package **TreePkg**. Note the change in the Model Explorer.

Changes to diagrams result in changes to the model.

In the Model Explorer, double click on the **TreePkg** package and rename it to **tree**. Note the change to the diagram.

Changes to the model can result in changes to diagrams.

So you see that Visual Paradigm is not just a UML diagramming application. It is a database with diagramming capabilities. The diagrams are kept consistent with the database and hence with each other.

Select the diagram of the package and drag one corner to make the package bigger.
Click Class tool and click within the diagram of the package. Name the class **Tree**.



Create two new packages in the diagram below the **tree** package. Call them **window** and **file**.
Draw dependencies from the **tree** package to the **window** package and the **file** package.
Find the **window** package in the Model Explorer. Right-click on it and select Model Element >> Class. Name the class **Window**.
Select the **Window** class in the Model Explorer and drag it on top of the **window** package in the diagram.
Similarly create a **FileSystem** class in the **file** package and place it on the diagram.
Right-click on the **Tree** class in the diagram. Select “Add >> Operation” and type the following three lines — don’t press Enter after the third line:+

```
+Tree( fs : FileSystem, w : Window )
+paint()
+leftMouseClicked( x : int, y : int )
```

To find the right tool (in this case “Generalization”) hover over the various tools available until you find it. Note that for generalization, you click first on the super-class (Handle) and then on the sub-class (e.g. FileHandle).

Add classes **FileHandle**, **DirectoryHandle**, and **Handle** to the **file** package and to the diagram.
Select the Generalization tool. Make **FileHandle** and **DirectoryHandle** generalize **Handle**.

This might be a good time to change the background colour for diagram elements. Select all item in the diagram. Then use Diagram >> Format to change the background colour to something lighter. Be sure to make this the default for the whole project.

Your diagram should now look something like Figure 1.

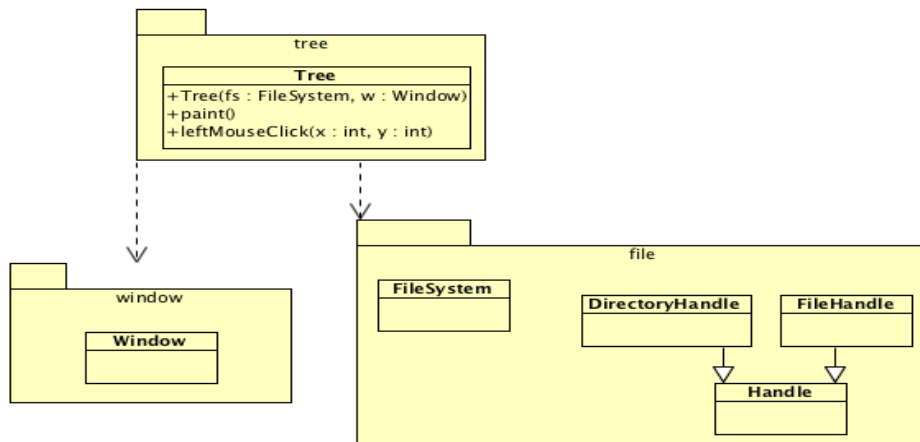


Figure 1:

~~You can hide individual operations, by selecting them and then using Right click >> Selection >> Hide. You can reveal all operations using Right click >> Presentation Options >> Operations >> Show all. Try it.~~

Select the **Tree** class in the diagram and press the Enter key. Note the many properties of a class you can edit. Try adding an attribute.

2 A New Diagram

In the Model Explorer, right-click on **file** package, and select Sub Diagrams /New Diagram and pick Class Diagram.

Drag all four **file** package classes onto the new diagram.

Right click on the **FileSystem** class in the diagram select "Delete"

- In response to 'Are you sure you want to delete "FileSystem"?' click "Yes".
- What happened in the Model View? Nothing! Check the other diagram, what happened there? Nothing.

Deleting from a diagram does not delete from the model..

Drag the **FileSystem** class back to the diagram.

In the Model Explorer delete **FileSystem**. What happened in your two diagrams?

Now recreate the **FileSystem** class and put it back on your diagrams

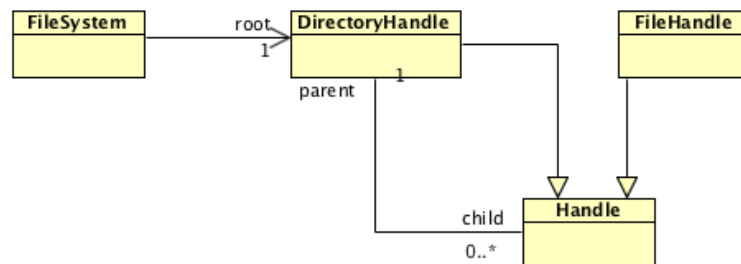
Draw an association from **FileSystem** to **DirectoryHandle**. Right-click on the association near the **DirectoryHandle** end. Give this end a multiplicity of 1 and a role name of "root".

On the other end, set Navigable to "Unspecified".

Draw an association from **DirectoryHandle** to **Handle**. (If need be, put a corner in the line, so you can see it.) Select the association and press "Enter".

Add role names and multiplicities to the roles (i.e., the ends).

After tidying, your diagram should look something like this



3 A sequence diagram

In the model explorer, right-click on the **tree** package, and select Sub Diagrams / New Diagram ... and pick Sequence Diagram.

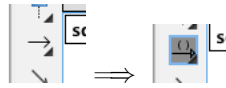
Drag class **Tree** from the model explorer to the diagram. Select "lifeline". Do the same for **Window** and **FileSystem**. Give the three objects names "t", "w" and "fs". Use the lifeline tool to create a fourth object in the diagram. Name the object "client".



Use the Create Message tool to show that the "client" creates the **Tree** object.



For the next few actions you will need the Call Message tool. To reveal it, click on the triangle by Message and the select Call Message.

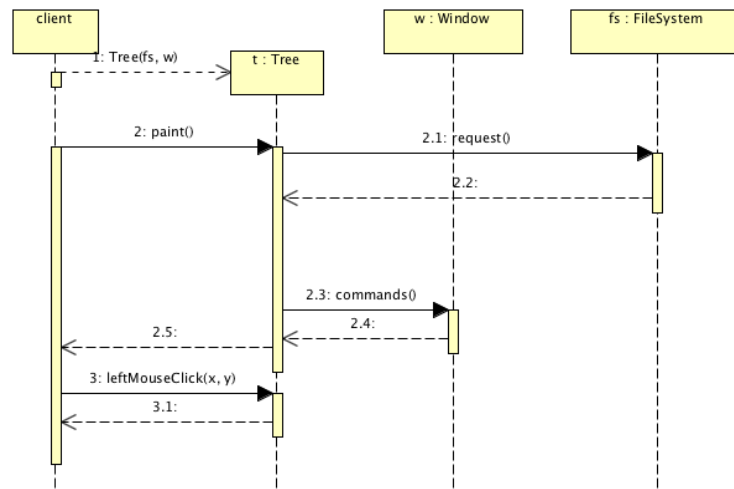


Use the Call Message tool to show that the “client” calls the paint operation of the Tree.

Add a return by selecting the call message and clicking on the little button that appears at its right end.

To suppress the parameter types in messages, right click on some blank area of the diagram and select Presentation Options / Message Display Options /Show Parameter Types in Message Operation Signatures.)

Complete the diagram to look like this



To show that the executions of paint and leftMouseClicked are not overlapping, you may have to use the Split Overlapping Execution command on the context menu.

~~[Note: This diagram shows parameter names rather than arguments. If anyone can figure out how to get the diagram to show arguments instead of parameter names, please let me know!]~~

4 Saving and loading models

Never allow your only copy of a VP or Eclipse workspace exist on a laptop, or even a desktop, for any nonnegligible length of time. Laptops in particular are attractive to thieves and prone to being dropped.