## FlowgraphManager changes

* On first load (when the flowgraph table in the PAK is empty), FlowgraphManager should perform “first time setup”. This goes through each Composite and checks to see if it is vanilla or not.
  + If the composite is not vanilla, flowgraphs should be disabled. Disabling flowgraphs returns the old way of navigating links, and does not display the nodegraph.
  + If the composite is vanilla, the layout is written to the PAK from the pre-defined table of layouts. This information in the PAK is then what is used from then on as it should always be in sync with the content in the Composite.
    - This means the “vanilla” list in FlowgraphManager is never directly pulled from when looking for layouts, only the “custom” list. (might want to rename this)

## Handling Multiple Flowgraphs

* The CompositeDisplay should manage the current active Flowgraphs.
  + When a composite is loaded, check FlowgraphManager to see if flowgraphs are supported – if not, revert to the old layout (as detailed above).
  + If flowgraphs are supported, create a Flowgraph document for each flowgraph
* Flowgraphs can be created via a button in the top bar, or by right clicking on the tabs.
* Flowgraphs can be deleted by right clicking on the tabs.
* The entity inspector should show all instances of nodes for the entity (need to think about the best way to do this). When you select a node, it should take you to the correct flowgraph and node position.

## Saving and compiling

* Flowgraph layouts should be saved and compiled:
  + Before the Commands is saved
  + When the Composite in CompositeDisplay changes
    - When the new Composite is set, if there was previously one active, save all Flowgraphs before changing and closing them

I can’t think of an instance where you wouldn’t want to save the layout AND compile at the same time, so this functionality should probably be merged together to avoid diverging.

I should disable right clicking context menu when it’s over a link

Need to make sure flowgraph names are unique within composite