**Requirements**

There is a requirement to store the display reference images per product, so each product can have a separate set of reference images.

Note that currently there is only a single produce being tested (i.e. RM-902) however in the future other products will also be tested.

**Current implementation**

The current implementation stores all reference images in the \reference\_files\images\ directory.

**Proposal**

The proposal is to store the reference images in sub-directories for each product:

\reference\_files\images\RM-902\

\reference\_files\images\RM-903\

etc.

This requirement adds a number of complications:

1. The code that requires this information in order to be able to read or save the reference image names/files has no knowledge of the phone that is currently connected and we should avoid adding additional assembly references (e.g. to FuseLib) from these assemblies.  
     
   The list of reference image names is returned by the ReferenceImageConverter class, which is part of the Granite.Studio.ActivityLibrary.dll.

The directory for the root reference image files directory is obtained from the GranitePaths.ReferenceImagesDirectory, which is part of the GraniteLibrary.dll.

It would be best if we do not add a reference to the FuseLib assembly for both of these assemblies.

1. When executing the DisplayImageComparerActivity executes in the framework. At run-time it therefore needs to obtain the Product RM number in order to look up the image files in the correct sub-directory.
2. We need to cater for the scenario where there is no product connected at all.  
     
   For instance a user could have no Fuse connections and still create a new Studio test, add the DisplayImageComparerActivity and open a display image file (previously saved on the Desktop for instance) and then save the reference image file with a specified reference image name.

For this we may need to use a default product name (e.g. RM-000).

**Proposed implementation overview:**

Add a new string property to the StudioModel for the ProductRM (which has a property changed event). If the main phone connection changes, this property should be updated.

Add a string ProductRM property to the PhoneLiveDataProvider.cs and PhoneDefaultDataProvider.cs (where the default is “RM-000”).

Add an UpdateProductRM(string productRM) method to the PhoneDataProviderFactory.cs.

Add an UpdateProductRM(string productRM) method to the GraniteWorkflowDesigner.xaml.cs.

When the StudioModel. ProductRM property changes the StudioView calls workflowDesigner.UpdateProductRM(productRM) with the product RM value, which in turn updates the data provider (as is done in a similar way for other data, see OnStudioPropertyChanged(object sender, System.ComponentModel.PropertyChangedEventArgs e) in the StudioView.xaml.cs file).

The ReferenceImageConverter class get the PhoneDataProviderFactory. PhoneDataProviderInstance().ProductRM property when constructing the reference images file path, e.g. “\reference\_files\images\RM-902\”

Similarly the DisplayImageComparerActivityDesigner gets the PhoneDataProviderFactory. PhoneDataProviderInstance().ProductRM property when constructing the reference images file path, e.g. “\reference\_files\images\RM-902\”

The Execution of the DisplayImageComparerActivity should be updated so that it obtains the main ProductRM from the framework. This needs to be investigated to see the best way of achieving this.

No changes should be required to GraniteLibrary.dll

Ideally no new project references should be required for any of the Granite Studio assemblies.

**Testing:**

Test switching the main phone and reference phone to see that the DisplayImageComparerActivity refreshes and behaves correctly for the current main product.

When switching phones we may not necessarily have any display reference images for the new phone. The DisplayImageComparerActivity should behave correctly, not throw FileNotFoundException.

We need to ensure this solution will work (as best it can) even if a user deletes all Fuse connections.