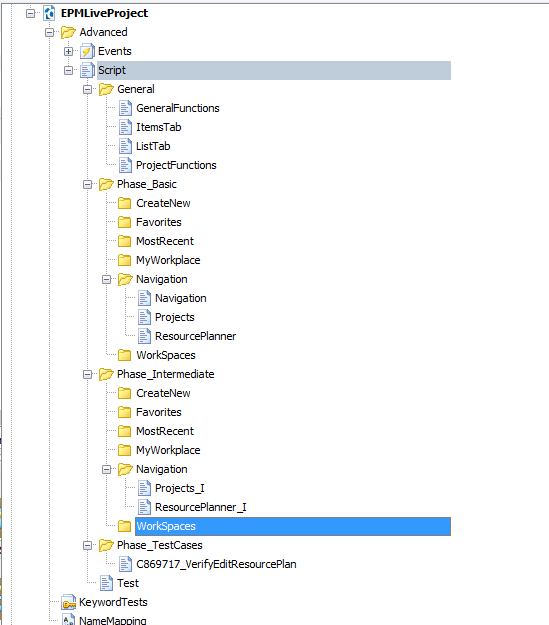
**EPM Live Automation**

Structure of the Project:



In the above tree hierarchy, under the EPMLiveProject Script there are 4 folders viz. General, Phase\_Basic, Phase\_Intermediate and Phase\_TestCases.

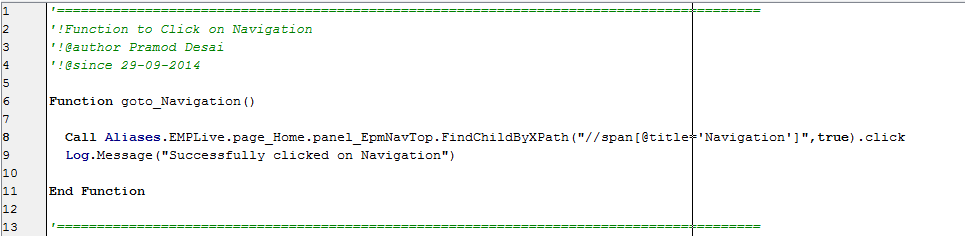
**General :** In this folder ProjectFunctions script contains some of the sub-routines that are required in the project such as opening the browser, logging in into the application etc. No further addition of routines must be done into this script. In this folder only the generic functions script can be added.

GeneralFunctions Script contains the basic general steps that are needed.

ItemsTab and ListTab script contains the functions/routines that contains the actions performed with the itms that are present in these tabs.

**Phase\_Basic :** In this folder there are 7 sub-folders viz. CreateNew, Favorites, MostRecent, MyWorkplace, Navigation, WorkSpaces and Settings. In the application left panel contains the links to these. And whichever basic steps that is to be included can be added in the respective folder by creating script(only if not created).

This folders' scripts contains only basic steps in every sub-routine (One action per sub-routine). Script name must be precise, relative to the routines that are to be added and every word must start with capital letter.

Example: 

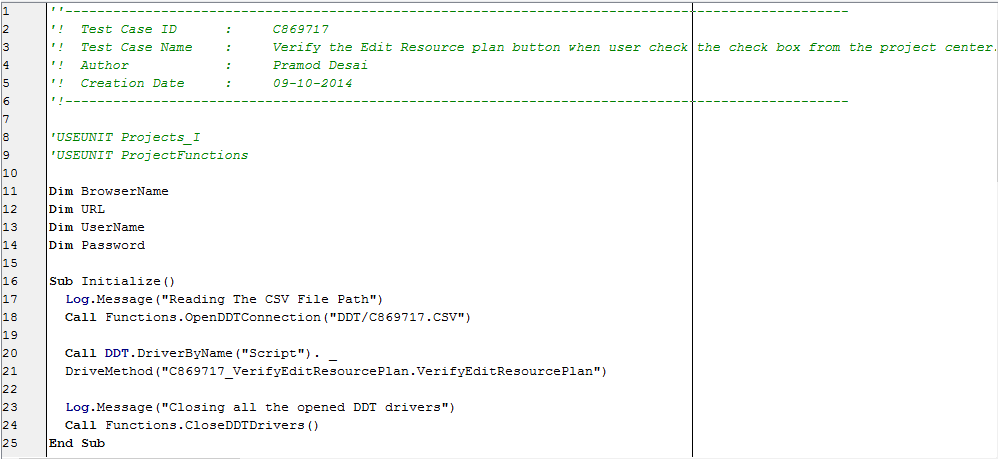
**Phase\_Intermediate:** This folder also contains 7 sub-folders as in Phase\_Basic. Every sub-routine in this, is the set of basic routines that are created in the Phase\_Basic level and the assertions that are required after every action. While creating the Phase\_Intermediate routines, we must take into consideration all the results(perspective) of the particular action and then can create. Even if we are modifying any routines, it should not affect the other TCs that are already using this routine.

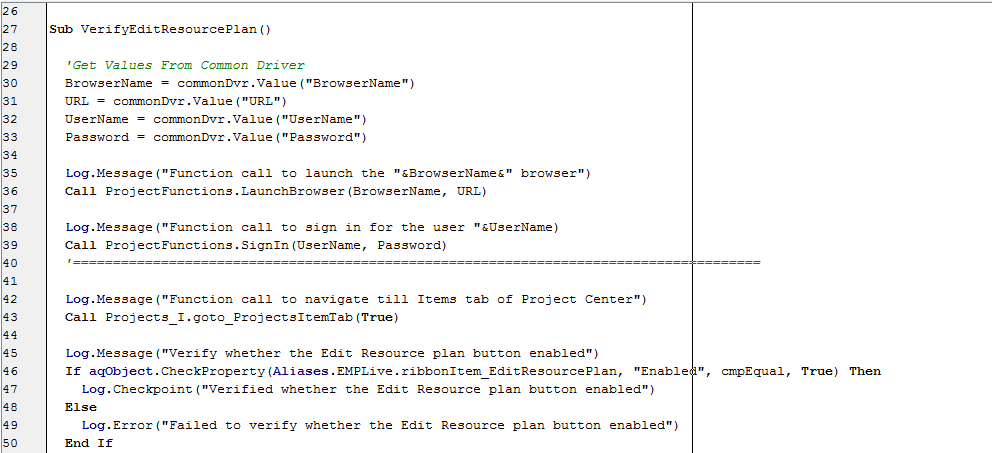
Script name must be precise, relative to the routines that are to be added and every word must start with capital letter and suffix with the underscore and I.

Example: FileName\_I

**Phase\_TestCases:** This folder contains the TCs. In this folder we will be adding the TCs.

Global View of the Test Script :







The above example script shows the conventions that we follow in EPM Live Automation.  
  
Line 1-6 : Contains the Test Case Id, Name (One Line description), Author of the script and the date it has been created. It is the mandatory part in every TC with the format that is shown above.  
  
Line 8-9 : Next part is library files inclusion. Here we can include the library names from which we are importing the functions for the script.

Line 11-14 : This is the variable declaration part. In this we will be declaring the variables that we are using in the script. Usually these are the CSV variables.

Line 16-25 : This is the Initialize routine where we will be opening the DDT connection, and calling the TC from its script name followed by dot and the routine name.

Example : 

Here C869717.CSV is the CSV file name used for the script.



Here C869717\_VerifyEditResourcePlan is the file name and VerifyEditResourcePlan is the main routine name where the actual script executes. Naming convention for the Script name is, TCId\_ProperNameForScript.

And routine name should be precise, relative to TC, meaningful and every word must start with a capital letter. And the same applies for Script name too.



Every Initialize() routine that contains OpenDDTConnection() call, must end with CloseDDTConnection() call.

Line 27-57 : This is the main routine where the TC execution starts.

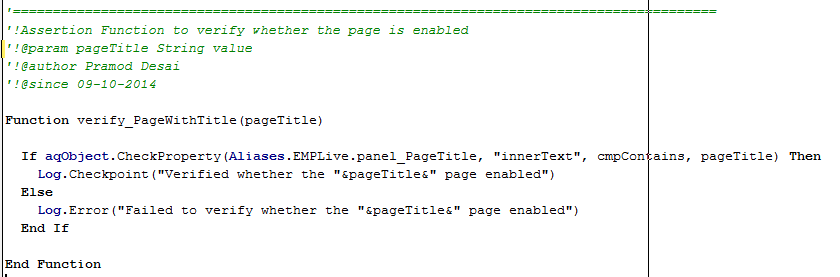
Line 29-33 : Importing the values from CSV. And these variables must be declared globally in the script.

Line : 35-39 : These are the pre-requisite to start the TC execution. And this pre-requisite must be separated with the line.

Line 42-50 : It contains the TC steps and the assertions. This is the main part in the main routine.  
 Line 54 : This is the call to clean-up function where all the actions performed must be reverted to default environment. This must be separated with the line.

Line 60-65 : Clean-up part(Reverting Actions) must be included in the CleanUp() function.

Other Sub-routines :



Above we can see example function. Before creating functions/sub-routines in the other scripts or in TC script, 1. There must be one liner explanation.

2. If any parameters are using in the function, then we must mention the param name and the type of value that accepts.

3. We must mention the author of the function.

4. The date on which it is created must be mentioned.

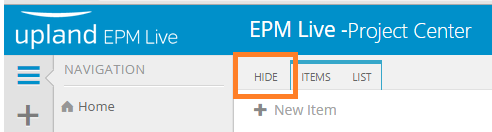
Every function/sub-routine in the script must be separated by a comment line and a blank line.

The above function is the **assertion function** so it **starts with verify word** and underscore followed by a proper name.

Namemapping Conventions :

Unnecessary mappings must be avoided.

Example : textnode\_HideTab



Above shown is an example of namemapping an object. We can see Hide as a tab but actually it is a text node. So namemapping of such objects goes like this,

actualObjectType\_NameVisibleObjectType.

Otherwise, just a object type underscore followed by a name of the object.  
Namemapping must be Camel casing.

**Before Starting Automation can go through these articles :**

**Requirements for Web Testing**

http://support.smartbear.com/viewarticle/56111/

**Preparing Web Browsers:**

http://support.smartbear.com/viewarticle/56974/

**For cross browsing overview :**

http://support.smartbear.com/viewarticle/55299/

**Creating Cross Browser Tests:**

http://support.smartbear.com/viewarticle/62867/

**Considerations for Web Testing :**

http://support.smartbear.com/viewarticle/55654/