FLEX - APPLICATIONS

Before we start with creating actual *HelloWorld* application using Flash Builder, let us see what are the actual parts of a Flex application. A Flex application consists of following four important parts out of which last part is optional but first three parts are mandatory:

- **Flex Framework Libraries**
- **Client-side code**
- **Public Resources** *HTML/JS/CSS*
- **Server-side code**

Sample locations of different parts of a typical Flex application **HelloWord** will be as shown below:

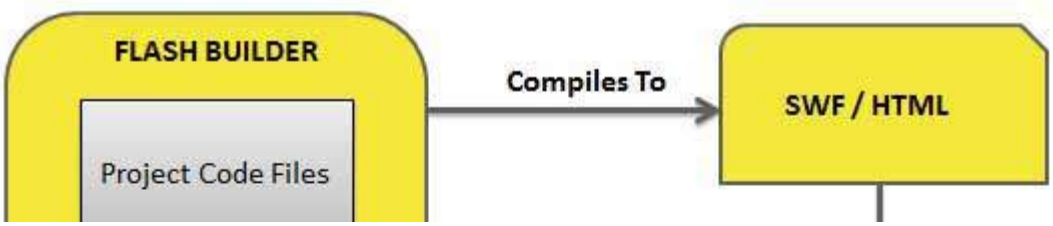
Name	Location
Project root	HelloWorld/
Flex Framework Libraries	Build Path
Public resources	html-template
Client-side code	table table-bordered/com/tutorialspoint/client
Server-side code	table table-bordered/com/tutorialspoint/server

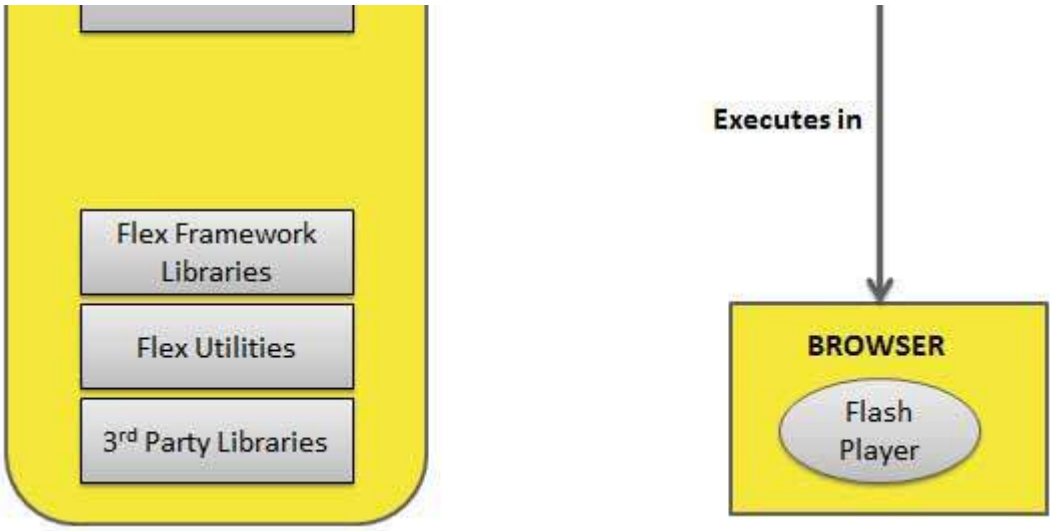
Application Build Process

Flex application required Flex Framework libraries. Flash Builder automatically add the libraries to build path.

When we build our code using Flash Builder, Flash builder will do the following tasks

- Compiles the source code to HelloWorld.swf file.
- Compiles a HelloWorld.html *awrapperfileforswffile* from a file index.template.html stored in html-template folder
- Copies HelloWorld.swf and HelloWorld.html files in target folder, bin-debug.
- Copies swfobject.js, a javascript code responsible to load swf file dynamically in HelloWorld.html in target folder, bin-debug
- Copies framework libraries in form of swf file named frameworks_xxx.swf in target folder, bin-debug
- Copies other flex modules . *swffilessuchassparkskins$_x$xx. swf, textLayout$_x$xx. swf* in target folder.





Application Launch Process

- Open the HelloWorld.html file available in \HelloWorld\bin-debug folder in any web-browser.
- HelloWorld.swf will load automatically and application will start running.

Flex Framework Libraries

Following is the brief detail about few important framework libraries.

In flex libraries are denoted using .swc notation

S.N.	Nodes & Description
1	**playerglobal.swc** This library is specific to FlashPlayer installed on your machine and contains native methods supported by flash player.
2	**textlayout.swc** This library supports the text layout related features.
3	**framework.swc** This is the flex framework library contains the core features of Flex.
4	**mx.swc** This library stores the definations of mx UI controls.
5	**charts.swc** This library supports the charting controls.
6	**spark.swc** This library stores the definations of spark UI controls.

7	**sparkskins.swc** This library supports the skinning of spark UI controls.

Client-side code

Flex application code can be written in MXML and ActionScript.

S.N.	Type & Description
1	**MXML** MXML is an XML markup language that we'll use to lay out user interface components.MXML is compiled into ActionScript during build process.
2	**ActionScript** ActionScript is an object-oriented procedural programming language and is based on the ECMAScript $ECMA-262$ edition 4 draft language specification.

In Flex, we can mix ActionScript and MXML, to do the following:

- Lay out user interface components using MXML tags
- Use MXML to declaratively define nonvisual aspects of an application, such as access to data sources on the server
- Use MXML to create data bindings between user interface components and data sources on the server.
- Use ActionScript to define event listeners inside MXML event attributes.
- Add script blocks using the <mx:Script> tag.
- Include external ActionScript files.
- Import ActionScript classes.
- Create ActionScript components.

Public resources

These are helper files referenced by Flex application, such as Host HTML page, CSS or images located under html-template folder.It contains following files

S.N.	File Name & Description
1	**index.template.html** Host HTML page, with place holders. Flash Builder uses this template to build actual page HelloWorld.html with HelloWorld.swf file.
2	**playerProductInstall.swf**

This is a flash utility to install Flash Player in express mode.

3

swfobject.js

This is the javascript responsible to check version of flash player installed and to load HelloWorld.swf in HelloWorld.html page.

4

html-template/history

This folder contains resources for history management of the application.

HelloWorld.mxml

This is the actual MXML/AS *ActionScript* code written implementing the business logic of the application and that the Flex compiler translates into SWF file which will be executed by flash player in the browser.A sample HelloWorld Entry class will be as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
   xmlns:s="library://ns.adobe.com/flex/spark"
   xmlns:mx="library://ns.adobe.com/flex/mx"
   width="100%" height="100%"
   minWidth="500" minHeight="500"
   initialize="application_initializeHandler(event)">

   <fx:Script>
      <![CDATA[
         import mx.controls.Alert;
         import mx.events.FlexEvent;
         protected function btnClickMe_clickHandler(event:MouseEvent):void
         {
            Alert.show("Hello World!");
         }

         protected function application_initializeHandler(event:FlexEvent):void
         {
            lblHeader.text = "My Hello World Application";
         }
      ]]>
   </fx:Script>
   <s:VGroup horizontalAlign="center" width="100%" height="100%"
   paddingTop="100" gap="50">
      <s:Label />
      <s:Button label="Click Me!"
      click="btnClickMe_clickHandler(event)" />
   </s:VGroup>
</s:Application>
```

Following Table gives the description of all the tags used in the above code script.

S.N.	Node & Description
1	**Application** Defines the Application container that is always the root tag of a Flex application.
2	**Script**

Contains the business logic in ActionScript language.

3

VGroup

Defines a Vertical Grouping Container which can contain Flex UI controls in vertical fashion.

4

Label

Represents a Label control, a very simple user interface component that displays text.

5

Button

Represents a Button control, which can be clicked to do some action.

Server-side code

This is the server side part of your application and its very much optional. If you are not doing any backend processing with-in your application then you do not need this part, but if there is some processing required at backend and your client-side application interact with the server then you will have to develop these components.

Next chapter will make use of all the above mentioned concepts to create HelloWorld application using Flash Builder.