FLEX - DEBUG APPLICATION

Flex provides excellent capability of debugging flex code and Flash Builder 4 has an excellent built-in debugger and debugging perspective support.

- During debug mode, Flex Application runs on Flash Player Debugger version built in Flash Builder 4 which supports debugging capability.
- So developers get an easy and inbuilt debugging configuration in Flash Builder

In this article we'll demonstrate usage of debugging Flex Client code using Flash Builder. We'll do the following tasks

- Set break points in the code and see them in BreakPoint Explorer.
- Step through the code line by line during debugging.
- View the values of variable.
- Inspect the values of all the variables.
- Inspect the value of an expression.
- Display the stack frame for suspended threads.

Debugging Example

Step	Description
1	Create a project with a name *HelloWorld* under a package *com.tutorialspoint.client* as explained in the *Flex - Create Application* chapter.
2	Modify *HelloWorld.mxml* as explained below. Keep rest of the files unchanged.
3	Compile and run the application to make sure business logic is working as per the requirements.

Following is the content of the modified mxml file **src/com.tutorialspoint/HelloWorld.mxml**.

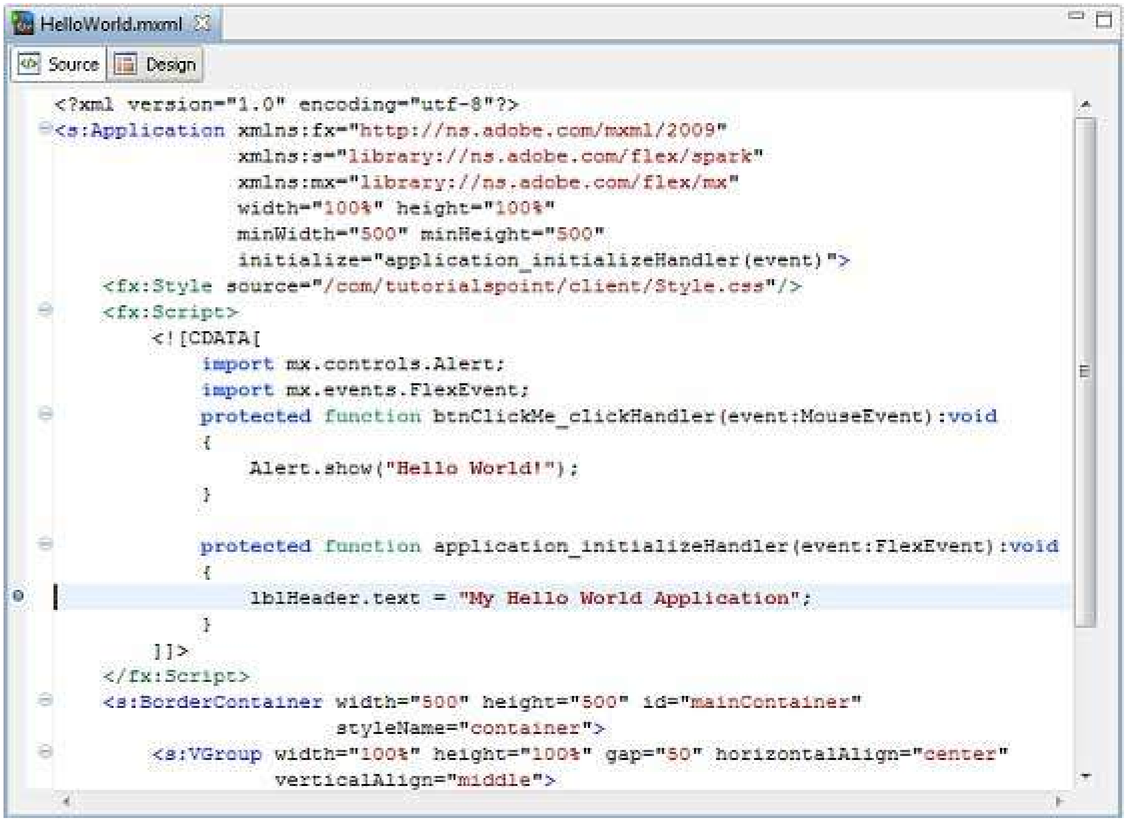
```
<?xml version="1.0" encoding="utf-8"?>
<s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
   xmlns:s="library://ns.adobe.com/flex/spark"
   xmlns:mx="library://ns.adobe.com/flex/mx"
   width="100%" height="100%"
   minWidth="500" minHeight="500"
   initialize="application_initializeHandler(event)">
   <fx:Style source="/com/tutorialspoint/client/Style.css"/>
   <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:BorderContainer width="500" height="500"
      styleName="container">
      <s:VGroup width="100%" height="100%" gap="50" horizontalAlign="center"
         verticalAlign="middle">
         <s:Label
            styleName="heading"/>
         <s:Button label="Click Me!"
            click="btnClickMe_clickHandler(event)" styleName="button" />
      </s:VGroup>
   </s:BorderContainer>
</s:Application>
```

Once you are ready with all the changes done, let us compile in normal mode as we did in Flex - Create Application chapter.

Step 1 - Place BreakPoints

Place a breakpoint on the first line of application_initializeHandler of HelloWorld.mxml

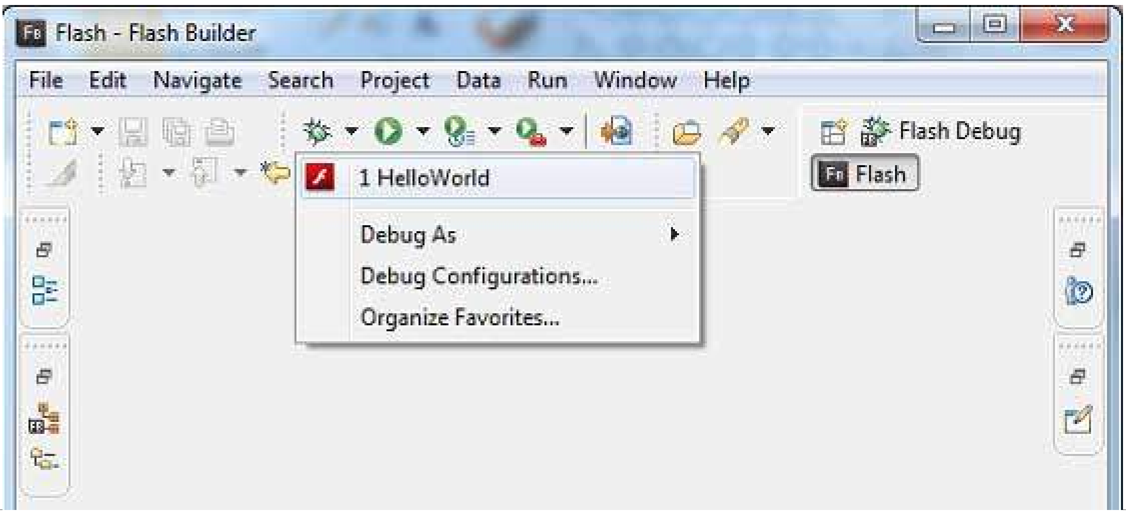


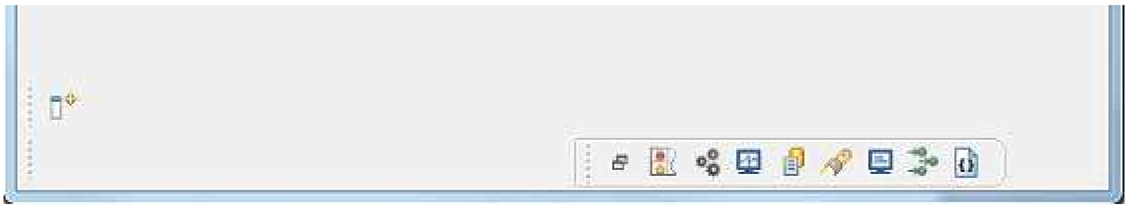
```
<?xml version="1.0" encoding="utf-8"?>
<s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
               xmlns:s="library://ns.adobe.com/flex/spark"
               xmlns:mx="library://ns.adobe.com/flex/mx"
               width="100%" height="100%"
               minWidth="500" minHeight="500"
               initialize="application_initializeHandler(event)">
    <fx:Style source="/com/tutorialspoint/client/Style.css"/>
    <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:BorderContainer width="500" height="500" id="mainContainer"
                       styleName="container">
        <s:VGroup width="100%" height="100%" gap="50" horizontalAlign="center"
                  verticalAlign="middle">
```

Step 2 - Debug Application

Now click on
Debug application menu and select **HelloWorld** application to debug the application.

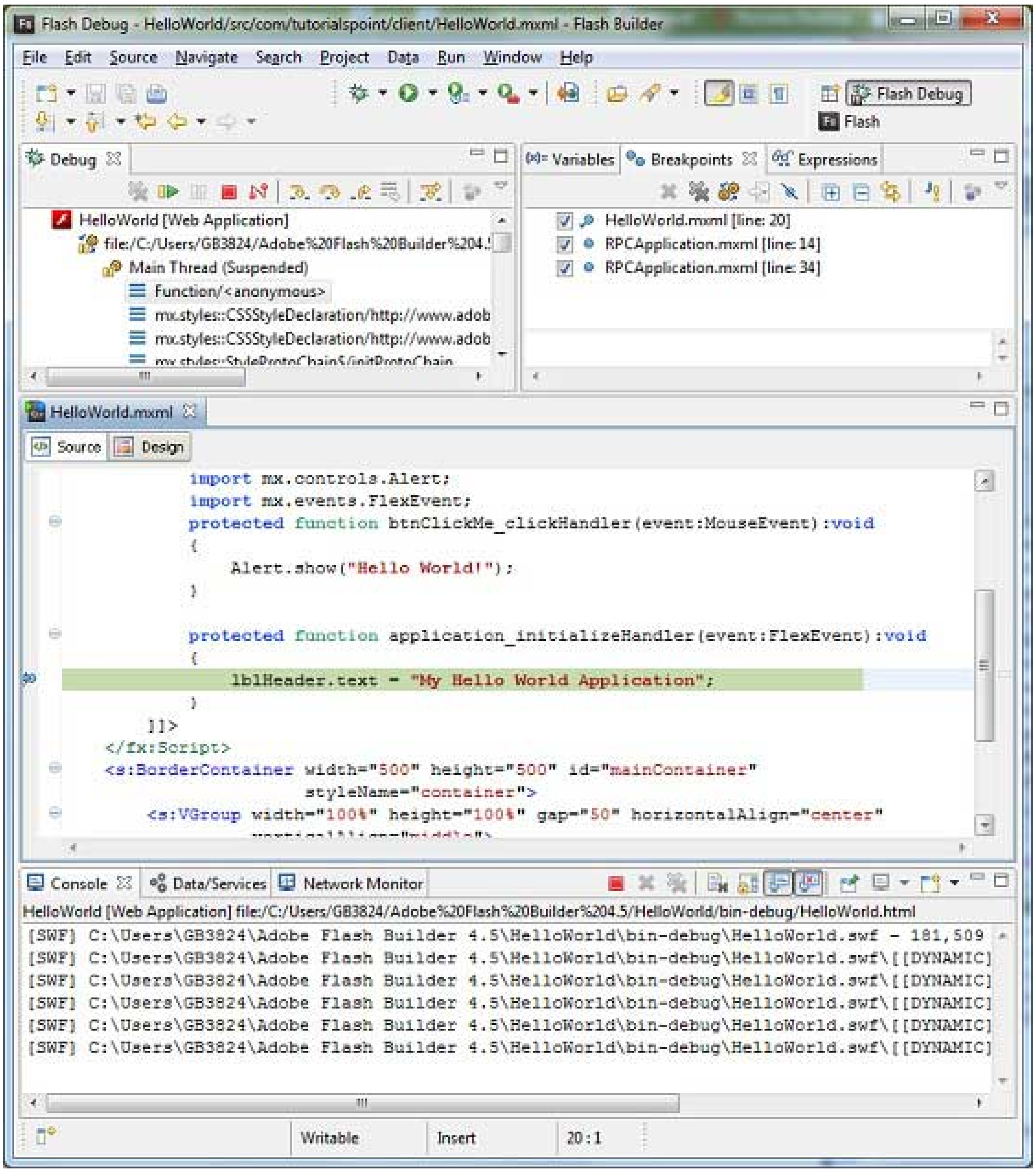




If everything is fine, application will launch in the browser and you will see following debug logs in Flash Builder console

```
[SWF] \HelloWorld\bin-debug\HelloWorld.swf
- 181,509 bytes after decompression
[SWF] \HelloWorld\bin-debug\HelloWorld.swf\[[DYNAMIC]]\1
- 763,122 bytes after decompression
[SWF] \HelloWorld\bin-debug\HelloWorld.swf\[[DYNAMIC]]\2
- 1,221,837 bytes after decompression
[SWF] \HelloWorld\bin-debug\HelloWorld.swf\[[DYNAMIC]]\3
- 1,136,788 bytes after decompression
[SWF] \HelloWorld\bin-debug\HelloWorld.swf\[[DYNAMIC]]\4
- 2,019,570 bytes after decompression
[SWF] \HelloWorld\bin-debug\HelloWorld.swf\[[DYNAMIC]]\5
- 318,334 bytes after decompression
```

As soon as Application launches,you will see the focus on Flash Builder breakpoint as we've placed the breakpoint on first line of application_initializeHandler method.


Flash Debug - HelloWorld/src/com/tutorialspoint/client/HelloWorld.mxml - Flash Builder
File Edit Source Navigate Search Project Data Run Window Help
Flash Debug
Flash
Debug
Variables Breakpoints Expressions
HelloWorld.mxml Source Design
Console Data/Services Network Monitor
Writable Insert 20 : 1

HelloWorld [Web Application]
file:/C:/Users/GB3824/Adobe%20Flash%20Builder%204.5/
Main Thread (Suspended)
Function/<anonymous>
mx.styles::CSSStyleDeclaration/http://www.adob
mx.styles::CSSStyleDeclaration/http://www.adob

HelloWorld.mxml [line: 20]
RPCApplication.mxml [line: 14]
RPCApplication.mxml [line: 34]

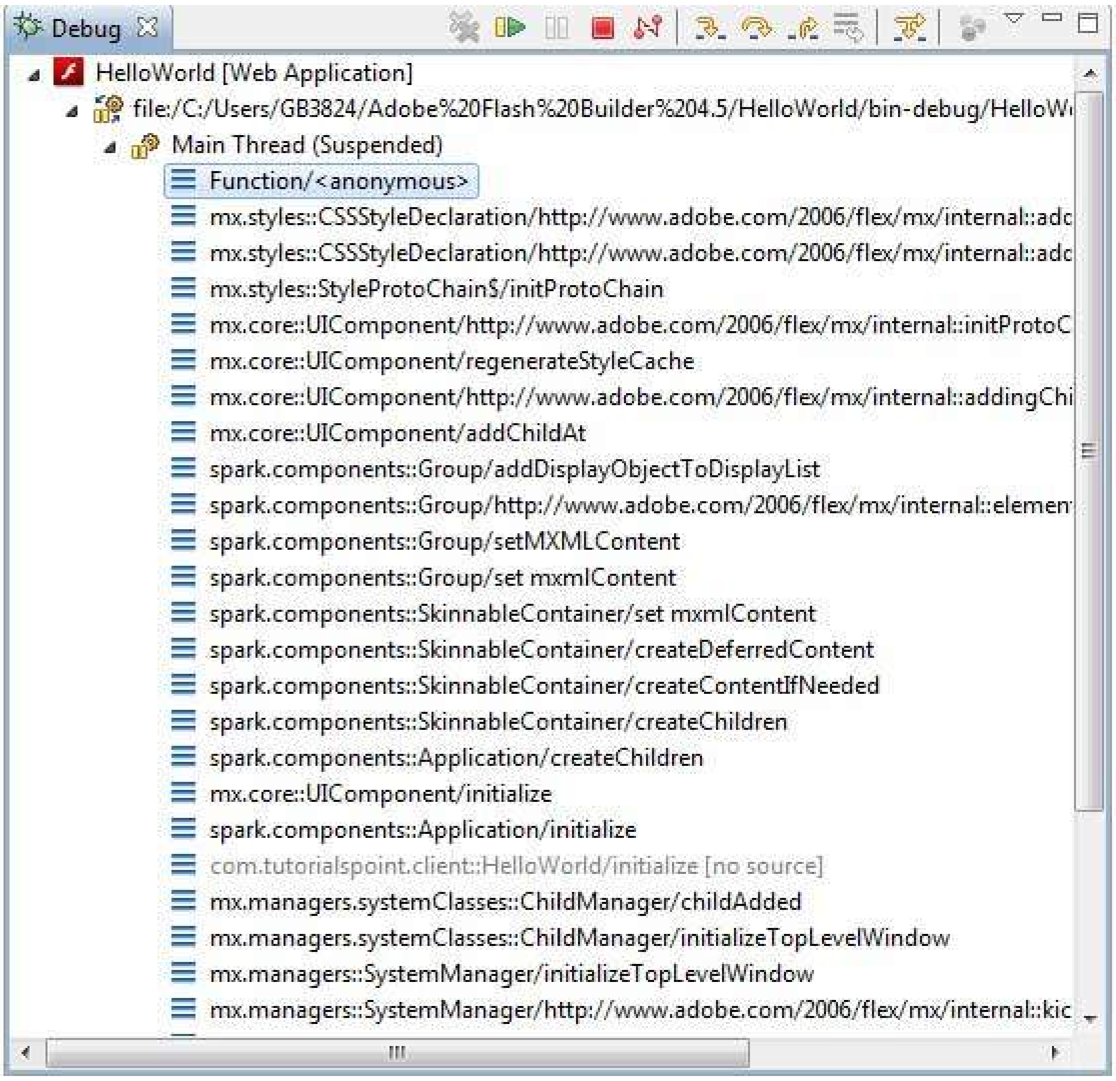
```
            import mx.controls.Alert;
            import mx.events.FlexEvent;
            protected function btnClickMe_clickHandler(event:MouseEvent):void
            {
                Alert.show("Hello World!");
            }

            protected function application_initializeHandler(event:FlexEvent):void
            {
                lblHeader.text = "My Hello World Application";
            }
        ]]>
    </fx:Script>
    <s:BorderContainer width="500" height="500" id="mainContainer"
                       styleName="container">
        <s:VGroup width="100%" height="100%" gap="50" horizontalAlign="center"
```

HelloWorld [Web Application] file:/C:/Users/GB3824/Adobe%20Flash%20Builder%204.5/HelloWorld/bin-debug/HelloWorld.html

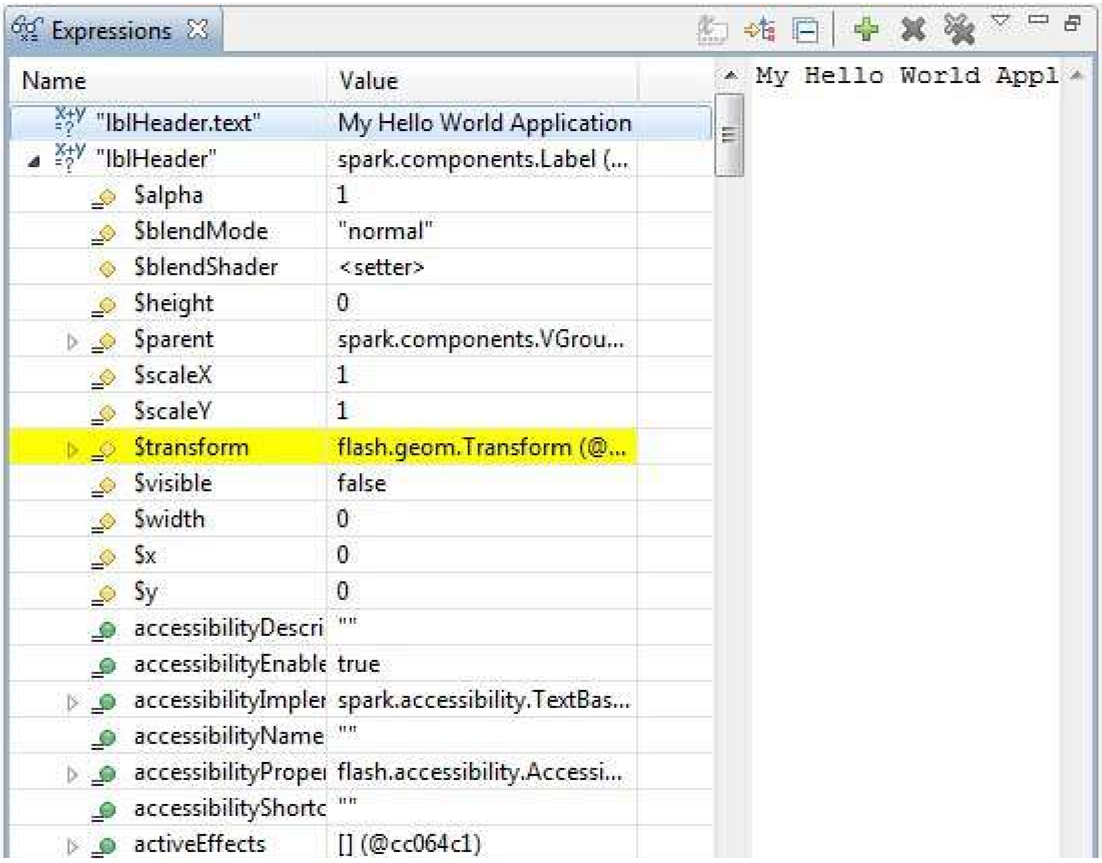
```
[SWF] C:\Users\GB3824\Adobe Flash Builder 4.5\HelloWorld\bin-debug\HelloWorld.swf - 181,509
[SWF] C:\Users\GB3824\Adobe Flash Builder 4.5\HelloWorld\bin-debug\HelloWorld.swf\[[DYNAMIC]
[SWF] C:\Users\GB3824\Adobe Flash Builder 4.5\HelloWorld\bin-debug\HelloWorld.swf\[[DYNAMIC]
[SWF] C:\Users\GB3824\Adobe Flash Builder 4.5\HelloWorld\bin-debug\HelloWorld.swf\[[DYNAMIC]
[SWF] C:\Users\GB3824\Adobe Flash Builder 4.5\HelloWorld\bin-debug\HelloWorld.swf\[[DYNAMIC]
[SWF] C:\Users\GB3824\Adobe Flash Builder 4.5\HelloWorld\bin-debug\HelloWorld.swf\[[DYNAMIC]
```

You can see the stacktrace for suspended threads.



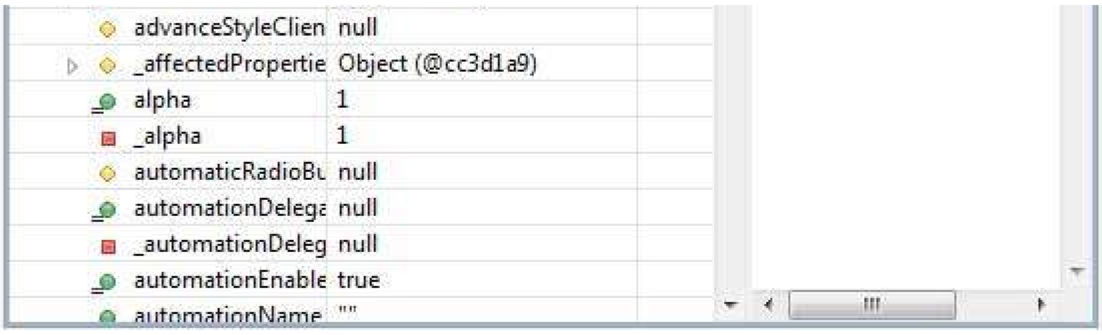
```
HelloWorld [Web Application]
  file:/C:/Users/GB3824/Adobe%20Flash%20Builder%204.5/HelloWorld/bin-debug/HelloW
    Main Thread (Suspended)
      Function/<anonymous>
      mx.styles::CSSStyleDeclaration/http://www.adobe.com/2006/flex/mx/internal::adc
      mx.styles::CSSStyleDeclaration/http://www.adobe.com/2006/flex/mx/internal::adc
      mx.styles::StyleProtoChain$/initProtoChain
      mx.core::UIComponent/http://www.adobe.com/2006/flex/mx/internal::initProtoC
      mx.core::UIComponent/regenerateStyleCache
      mx.core::UIComponent/http://www.adobe.com/2006/flex/mx/internal::addingChi
      mx.core::UIComponent/addChildAt
      spark.components::Group/addDisplayObjectToDisplayList
      spark.components::Group/http://www.adobe.com/2006/flex/mx/internal::elemen
      spark.components::Group/setMXMLContent
      spark.components::Group/set mxmlContent
      spark.components::SkinnableContainer/set mxmlContent
      spark.components::SkinnableContainer/createDeferredContent
      spark.components::SkinnableContainer/createContentIfNeeded
      spark.components::SkinnableContainer/createChildren
      spark.components::Application/createChildren
      mx.core::UIComponent/initialize
      spark.components::Application/initialize
      com.tutorialspoint.client::HelloWorld/initialize [no source]
      mx.managers.systemClasses::ChildManager/childAdded
      mx.managers.systemClasses::ChildManager/initializeTopLevelWindow
      mx.managers::SystemManager/initializeTopLevelWindow
      mx.managers::SystemManager/http://www.adobe.com/2006/flex/mx/internal::kic
```

You can see the values for expressions.



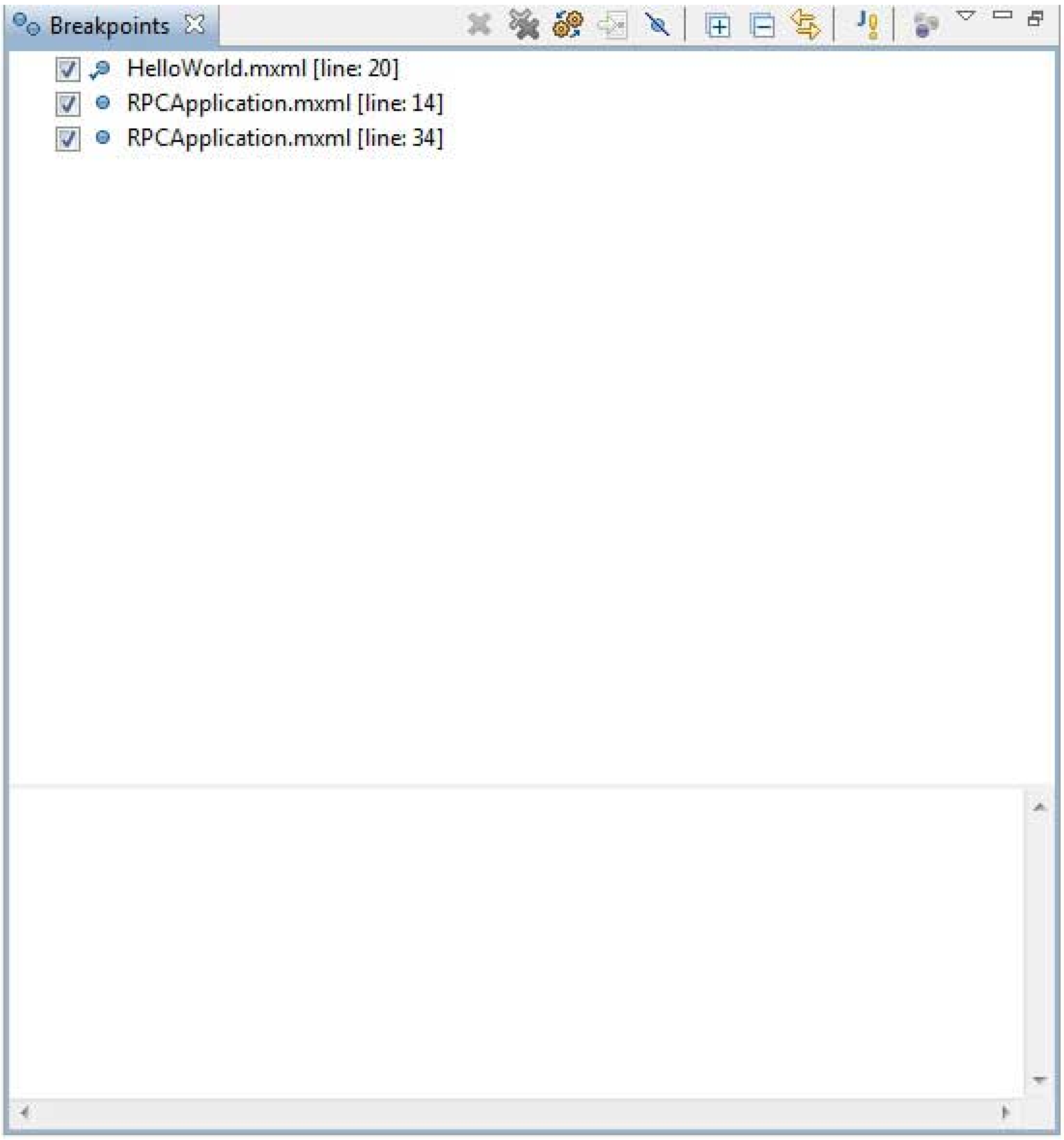
Name	Value
"lblHeader.text"	My Hello World Application
"lblHeader"	spark.components.Label (...
$alpha	1
$blendMode	"normal"
$blendShader	<setter>
$height	0
$parent	spark.components.VGrou...
$scaleX	1
$scaleY	1
$transform	flash.geom.Transform (@...
$visible	false
$width	0
$x	0
$y	0
accessibilityDescri	""
accessibilityEnable	true
accessibilityImpler	spark.accessibility.TextBas...
accessibilityName	""
accessibilityPropei	flash.accessibility.Accessi...
accessibilityShortc	""
activeEffects	[] (@cc064c1)

```
My Hello World Appl
```



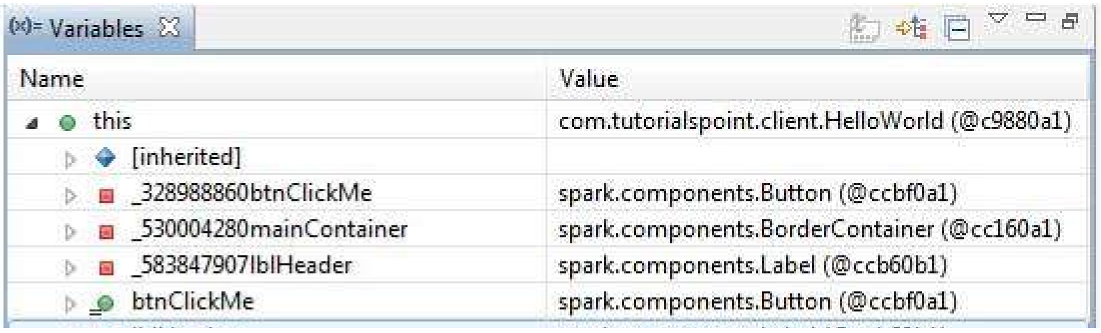
advanceStyleClien	null
_affectedPropertie	Object (@cc3d1a9)
alpha	1
_alpha	1
automaticRadioBu	null
automationDelega	null
_automationDeleg	null
automationEnable	true
automationName	""

You can see the list of breakpoints placed.

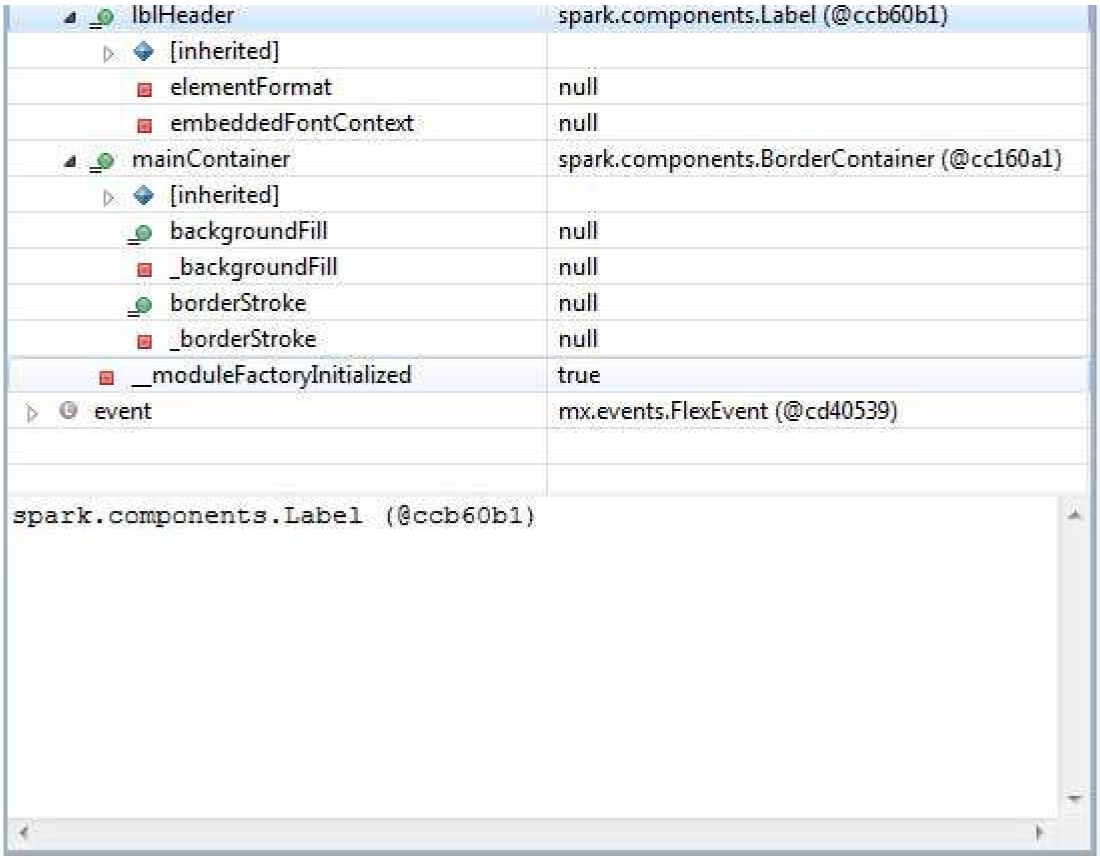


HelloWorld.mxml [line: 20]
RPCApplication.mxml [line: 14]
RPCApplication.mxml [line: 34]

Now keep pressing F6 until you reach the last line of application_initializeHandler method. As reference for function keys, F6 inspects code line by line, F5 steps inside further and F8 will resume the application. Now you can see the list of values of all variables of application_initializeHandler method.



Name	Value
this	com.tutorialspoint.client.HelloWorld (@c9880a1)
[inherited]	
_328988860btnClickMe	spark.components.Button (@ccbf0a1)
_530004280mainContainer	spark.components.BorderContainer (@cc160a1)
_583847907lblHeader	spark.components.Label (@ccb60b1)
btnClickMe	spark.components.Button (@ccbf0a1)



lblHeader	spark.components.Label (@ccb60b1)
[inherited]	
elementFormat	null
embeddedFontContext	null
mainContainer	spark.components.BorderContainer (@cc160a1)
[inherited]	
backgroundFill	null
_backgroundFill	null
borderStroke	null
_borderStroke	null
__moduleFactoryInitialized	true
event	mx.events.FlexEvent (@cd40539)

```
spark.components.Label (@ccb60b1)
```

Now you can see the flex code can be debugged in the same way as a Java Application can be debugged. Place breakpoints to any line and play with debugging capabilities of flex.

Loading [MathJax]/jax/output/HTML-CSS/jax.js