Support for Flash Player 10 in Adobe Director

11.5.8.612

Contents

This document provides an overview of the support provided by Adobe Director with respect to Adobe Flash Player 10 and ActionScript 3.

New Properties and Methods

The following methods and properties are new in this release:

- ✓ spriteObjRef.newProxyCallback(#handlerName, {scriptObject})
- ✓ spriteObjRef.stage

spriteObjRef.newProxyCallback (Flash Sprite Method)

The parameters for this method are shown in the following table:

Parameter	**Description**
handlerName	This parameter specifies the Lingo/Javascript handler that should be called when the callback is triggered from the ActionScript 3 script of the flash sprite. This is a required parameter.
scriptObject	This parameter returns a Lingo/Javascript object that can be used to pass in place an ActionScript 3 function object to get a call back in Adobe Director. This is an optional parameter.

The following example creates a callback and passes it to the `addEventListener()` method and then handles the event in Adobe Director.

```
on testAS3ToLingoCallback
      spr = sprite(“AS3Swf”)
      lcbk = spr.newProxyCallback(#mouseUpLingoHandler)

      -- get a reference to the AS3 MouseEvent class
      mouseUpEvt = spr.getVariable("flash.events.MouseEvent", false)

      -- get a reference to a button object present inside the swf
      flbtn = spr.getVariable(“FlashButton”, false)

      -- Pass the lingo callback instead of a AS3 Function object
      flbtn.addEventListener(mouseUpEvt.MOUSE_UP, lcbk)
end

on mouseUpLingoHandler evt
      put “Got the mouseUp callback from Flash As3!!”
      put “You clicked on ” & evt.target.name
end
```

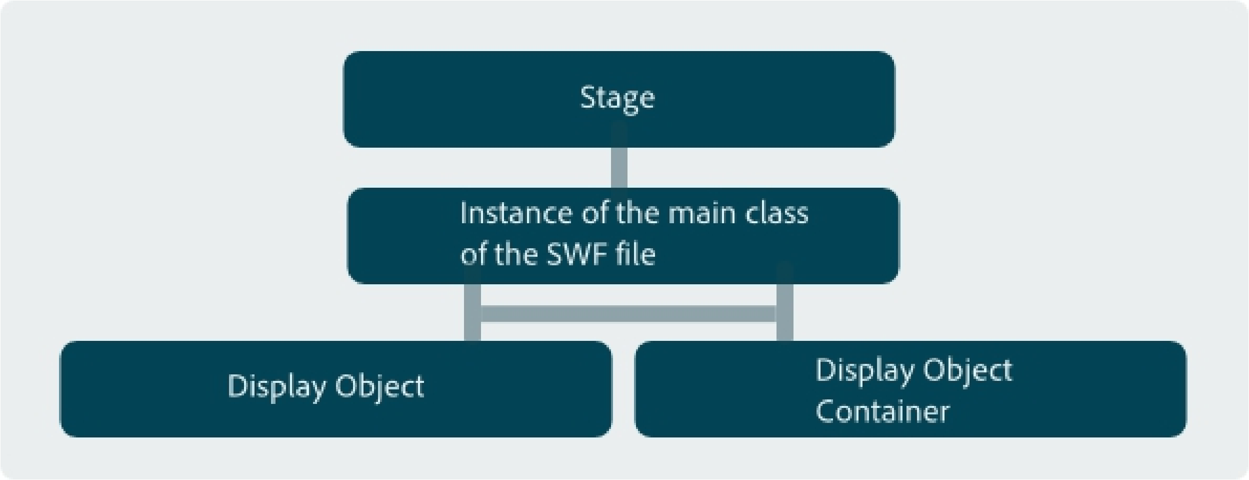
Note - The proxy callback objects should be created and used on the same sprite. It is invalid to create a callback object using one sprite and use it to get callbacks from other sprites' events/invocations.

For example, the following usage is unsupported:

```
cbk1 = sprite1.newProxyCallback(#As3ToLingoCallback)
sprite2.stage.addEventListener(“mouseUp”, cbk1)
-- cbk1 is created for sprite1. It can’t be used on sprite2
```

spriteObjRef.stage (Flash Sprite Property)

The `spriteObjRef.stage` method returns the root display object (stage object) of the given Flash movie. Using the properties and methods of the '`flash.display.DisplayObjectContainer`' on the stage object, you can navigate through the display list of the Flash sprite as shown in the following figure:



For example, the following Lingo code will print the name of the first child of the stage of the Flash sprite “AS3Swf”

```
put( sprite(“AS3Swf”).stage.getChildAt(0).name )
```

Changes to Existing Properties and Methods

The following sections provide an overview of the changes to the existing properties and methods.

spriteObjRef.newObject(objectType {, arg1, arg2 ....})

To create a new ActionScript3 object, the `objectType` passed to `newObject()` function should be a fully qualified class name whose instance needs to be created. For instance, to create an ActionScript 3 `BitmapData` object, you need to pass the `objectType` as “`flash.display.BitmapData`”. Just passing “BitmapData” will simply return <void>.

For example, the following Lingo code creates an instance of the hypothetical class ‘`com.xyz.TestClass`’ defined in the Flash sprite “AS3Swf”. The constructor here takes one string argument and a number.

```
TestClassObj = sprite(“AS3Swf”).newObject("com.xyz.TestClass", "First arg for TestClass", 10)
```

spriteObjRef.getVariable(variableName, {returnValueOrReference})

ActionScript 3 does not support top-level global variables. Any global variable should be declared as static members of some class. To access these variables in Adobe Director, first the `getVariable` method should be called with the ‘variableName’ as a fully qualified name of the class, which the static variable is part of. On the class reference, we can either call the `getVariable` method with the static variable name or simply use the dotted syntax, if the ‘`propDirectAccess`’ is set to true.

For example, to access a static member ‘`helloString`’ of a class `com.xyz.Hello`, the following Lingo code can be used:

```
on putHelloString
      spr = sprite(“AS3Swf”)
      helloClass = spr.getVariable(“com.xyz.Hello”, false)
      put helloClass.getVariable(“helloString”)
  ⚠   --put spr.getVariable(“com.xyz.Hello.helloString”) does not work
end
```

OR

```
on putHelloString
      spr = sprite(“AS3Swf”)
      spr.propDirectAccess = true
      helloClass = spr.getVariable(“com.xyz.Hello”, false)
      put helloClass.helloString
end
```

spriteObjRef.setVariable(variableName, newValue)

In the context of ActionScript 3, `setVariable()` is only supported on a Flash object, not on a spriteObjectRef.

For example,

```
on setHelloString
      spr = sprite(“AS3Swf”)
      helloClass = spr.getVariable(“com.xyz.Hello”, false)
      helloClass.setVariable(“helloString”, “Hello World”)
   ⚠  --put spr.setVariable(“com.xyz.Hello.helloString”, “HW”) does not work
end
```

OR

```
on setHelloString
      spr = sprite(“AS3Swf”)
      spr.propDirectAccess = true
      helloClass = spr.getVariable(“com.xyz.Hello”, false)
      helloClass.helloString = “Hello World”
end
```

Functions not supported in Flash movies with ActionScript 3

The following functions are not supported in Flash Player with ActionScript 3:

Functions	Description
setCallback()	Use the newProxyCallback() method.
setFlashProperty()	Not needed. Use the setVariable() method.
getFlashProperty()	Not needed. Use the getVariable() method.
tellTarget()	Not supported in ActionScript 3