

Adding custom Chrome to your AIR application

Posted At : 6 April 2009 12:23 | Posted By : Shaun McCran
Related Categories: AIR, RIA

Whilst creating my last AIR application I found that the standard 'Chrome' that is provided by the OS just didn't match the application look and feel at all.

After a little searching I found that there are a few key elements in your application that you need to change to remove the standard operating system Chrome, and stop Flex builder from replacing it with its own.

Firstly in your *application-app.xml* document look for the 'systemChrome' xml value. Setting this to none will disable the operating system Chrome. As we are using the 'WindowedApplication' Flex builder will automatically start using its own Chrome framework, so we need to turn that off too.

```
type of system chrome to use (either "standard" or "none"). Optional. Default stan
    <systemChrome>none</systemChrome>
ie window is transparent. Only applicable when systemChrome is none. Optional. Defa
    <transparent>true</transparent>
```

This is done in the WindowedApplication code, set showFlexChrome="false" and that will disable Flex from using its default Chrome.

```
1 <?xml version="1.0" encoding="utf-8"?>
  xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute" initialize="init()" ;
```

Now that we have completely turned it all off, we need to build our own. In this application I am using a canvas with rounded corners that is 15

pixels larger than the application canvas, to give the impression of a border all the way around. Inside that canvas I've added two controls.

```
Label text="_" styleName="controls" toolTip="Minimize" x="173" y="-2" click="onMinimize";
Label text="X" styleName="controls" toolTip="Close" x="184" y="1" click="onClose();
```

These simply replicate the functionality that the minimise and close buttons give a user on a standard window. The functions they call are:

```
public function onMinimize():void
{
    stage.nativeWindow.minimize();
}

public function onClose():void
{
    stage.nativeWindow.close();
}
```

And with that you fully customise the look and feel of your applications Chrome.