Errata for 647-1 Moroney (corrected in the 2nd printing)

Page	Original Text
1	This chapter will bring you up-to-date on web application technology from the dawn of computing history to today, putting Ajax and Atlas in context. It's the beginning of what I hope will be a fun and informative ride.
	Corrected Text
	This chapter will bring you up-to-date on web application technology from the dawn of computing history to today, putting Ajax and Atlas in context. It's the beginning of what I hope will be a fun and informative ride. This book is based on the July 2006 Community technical preview of the Atlas SDK, available on http://atlas.asp.net.
13	**Original Text**
	To create this client, start a new Visual Studio 2005 web site, and edit the default WebForm1.aspx content to match Listing 1-1.
	Corrected Text
	To create this client, start a new Visual Studio 2005 web site, and edit the default Default.aspx content to match Listing 1-1.
	Original Text

	`<%@ Page language="c#" Codebehind="WebForm1.aspx.cs" AutoEventWireup="false"`
	
	`Inherits="Atlas1_1.WebForm1" %>`

	Corrected Text

	`<%@ Page language="c#" Codebehind="Default.aspx.cs" AutoEventWireup="false"`
	
	`Inherits="Atlas1_1.Default" %>`

14	**Original Text**
	`function updateTotal() {` **`frm = document.forms[0];`** **`url="WebForm2.aspx?A=" + frm.elements['A'].value +`** **`"&B=" + frm.elements['B'].value;`** **`xmlHttp.open("GET",url,true);`** **`xmlHttp.onreadystatechange=doUpdate;`** **`xmlHttp.send();`** **`return false;`** **`}`**
	Corrected Text
	`function updateTotal() {` **`frm = document.forms[0];`** **`url="Default2.aspx?A=" + frm.elements['A'].value +`** **`"&B=" + frm.elements['B'].value;`** **`xmlHttp.open("GET",url,true);`** **`xmlHttp.onreadystatechange=doUpdate;`** **`xmlHttp.send();`** **`return false;`** **`}`**
	Original Text
	This function then takes the values of A and B from their formelements and uses them to build the URL to WebForm2.aspx, which will look something like WebForm2.aspx?A=8&B=3.
	Corrected Text
	This function then takes the values of A and B from their form elements and uses them to build the URL to WebForm2.aspx, which will look something like Default2.aspx?A=8&B=3.
15	**Original Text**

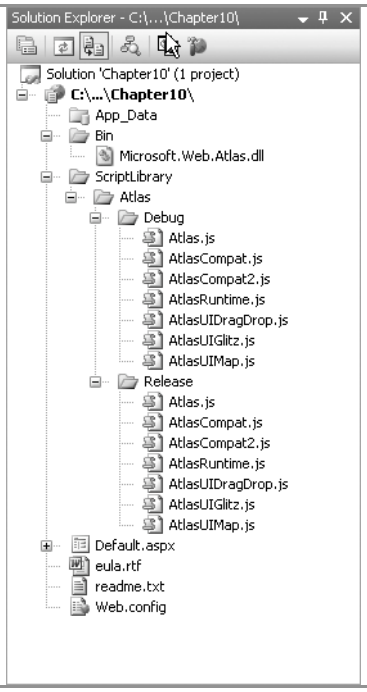
	To get this application to work, add a new C# web form to the project, and keep the default name of WebForm2.aspx. In the page designer, delete all the default HTML so that the page contains just the ASPX declaration: <%@ Page language="c#" Codebehind="WebForm2.aspx.cs" AutoEventWireup="false" Inherits="Atlas1_1.WebForm2" %>
	Corrected Text
	To get this application to work, add a new C# web form to the project, and keep the default name of Default2.aspx. In the page designer, delete all the default HTML so that the page contains just the ASPX declaration: <%@ Page language="c#" Codebehind="Default2.aspx.cs" AutoEventWireup="false" Inherits="Default2" %>
	Original Text
	This handles the request from WebForm1.aspx, getting the values of A and B, and writes them back to the response buffer. The XMLHttpRequest object within WebForm1.aspx then handles the communication and asynchronously updates the Returned Total box.
	Corrected Text
	This handles the request from Default.aspx, getting the values of A and B, and writes them back to the response buffer. The XMLHttpRequest object within Default.aspx then handles the communication and asynchronously updates the Returned Total box.
29	**Original Text**

	Here is an example of programmatically constructing Atlas elements using JavaScript:
	
	Var myTextBox = new Web.UI.TextBox(document.getElement('TextBox1'));
	myTextBox.initialize();
	
	var myLabel = new Web.UI.Label(document.getElement('Label1'));
	
	var myBinding = new Web.Binding();
	myBinding.set_dataContext(myTextBox);
	myBinding.set_dataPath('text');
	myBinding.set_property('text');
	myBinding.set_direction(Web.BindingDirection.In);
	
	myLabel.get_bindings().add(myBinding);
	
	myLabel.Initialize();

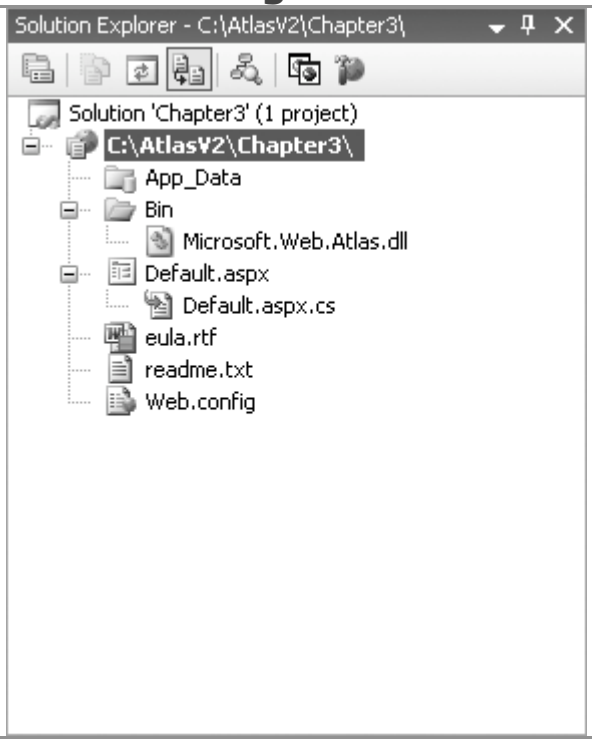
	Corrected Text

	Here is an example of programmatically constructing Atlas elements using JavaScript:
	
	Var myTextBox = new Sys.UI.TextBox(document.getElement('TextBox1'));
	myTextBox.initialize();
	
	var myLabel = new Sys.UI.Label(document.getElement('Label1'));
	var myBinding = new Sys.Binding();
	myBinding.set_dataContext(myTextBox);
	myBinding.set_dataPath('text');
	myBinding.set_property('text');
	myBinding.set_direction(Web.BindingDirection.In);
	
	myLabel.get_bindings().add(myBinding);
	
	myLabel.Initialize();

34	**Original Figure**



Corrected Figure



Original Figure

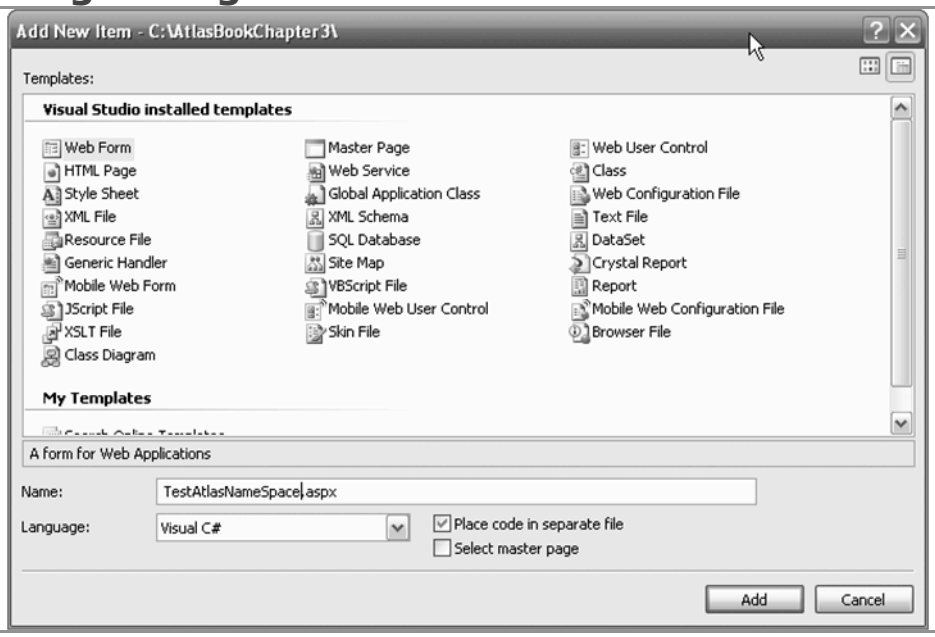
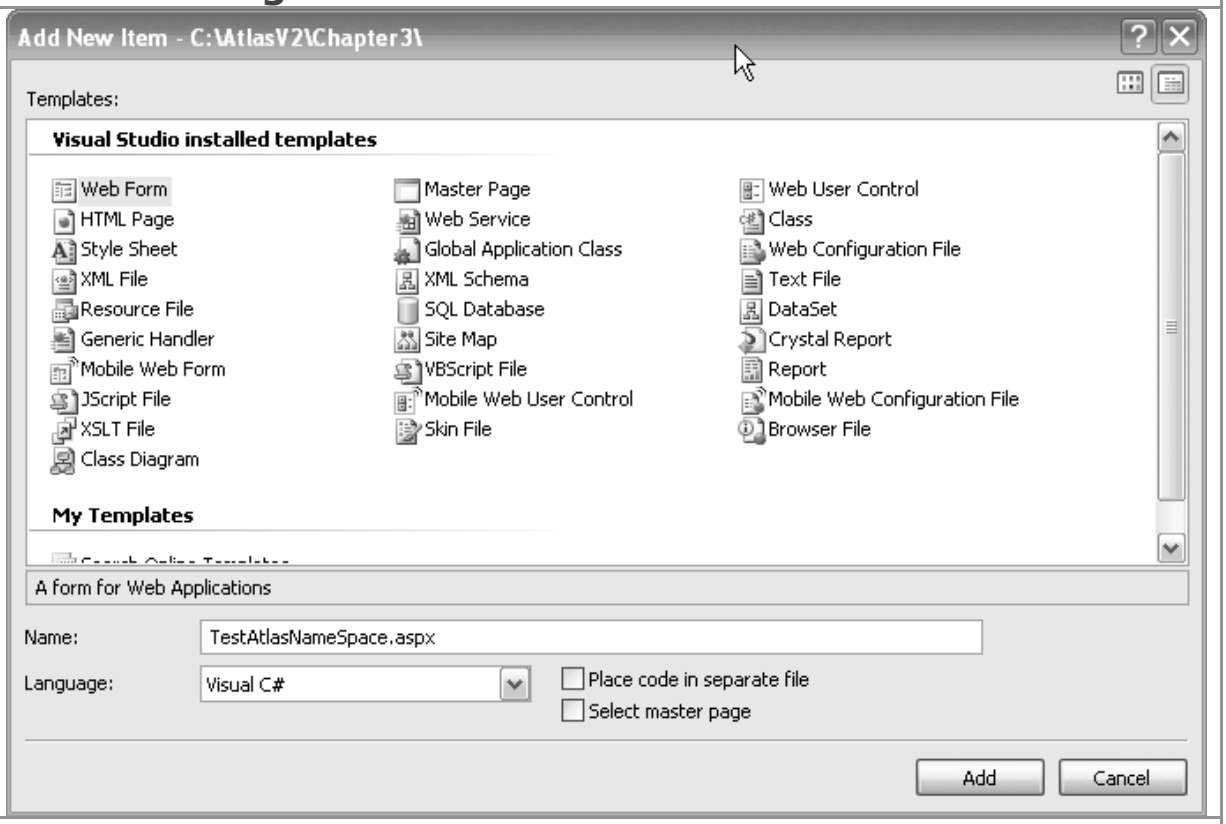
AtlasBook.js

```
// JScript File

Type.registerNamespace('AtlasBook');

AtlasBook.Car = function(strMake, strModel, strYear) {
    var m_Make = strMake;
    var m_Model = strModel;
    var m_Year = strYear;

    this.getMake = function() {
        return m_Make;
    }

    this.getModel = function() {
        return m_Model;
    }

    this.getMakeandModel = function() {
        return m_Make + ' ' + m_Model;
    }

    this.getYear = function() {
        return m_Year;
    }

    this.dispose = function() {
        alert('bye ');
    }
}
Type.registerClass('AtlasBook.Car', null, Web.IDisposable);
```

Corrected Figure

AtlasBook.js

```
// JScript File

Type.registerNamespace("AtlasBook");

AtlasBook.IStickShift = function() {
    this.setGear = Function.abstractMethod;
    this.getGear = Function.abstractMethod;
    this.getGearCount = Function.abstractMethod;
}
AtlasBook.IStickShift.registerInterface('AtlasBook.IStickShift');


AtlasBook.Car = function(strMake, strModel, strYear) {
    var m_Make = strMake;
    var m_Model = strModel;
    var m_Year = strYear;

    this.getMake = function() {
        return m_Make;
    }

    this.getModel = function() {
        return m_Model;
    }

    this.getMakeandModel = function() {
        return m_Make + ' ' + m_Model;
    }

    this.getYear = function() {
        return m_Year;
    }

    this.dispose = function() {
        alert('bye ' + this.getName());
    }
}
AtlasBook.Car.registerAbstractClass('AtlasBook.Car');
```

38	**Original Figure**
	
	Corrected Figure
	
40	**Original Text**
	Click the Browse button, and look in your application directory for the C:\Program Files\ Microsoft ASP.NET\Atlas directory. Find the file called Microsoft.Web.Atlas.dll, and choose it.
	Corrected Text

	Click the Browse button, and look in your application directory for the C:\Program Files\ Microsoft ASP.NET\Atlas directory. In this you will have a version folder (v2.0.50727 for the July CTP), and this contains an 'Atlas' subfolder.
43	**Original Text**
	Listing 3-1. ***Creating a Car Namespace***
	Corrected Text
	Listing 3-2. ***Creating a Car Namespace***
44	**Original Text**

	`this.getYear = function() {`
	`return m_Year;`
	`}`
	`this.dispose = function() {`
	`alert('bye ' + this.getName());`
	`}`
	`}`
	`Type.registerClass('AtlasBook.Car', null, Web.IDisposable);`

	Corrected Text

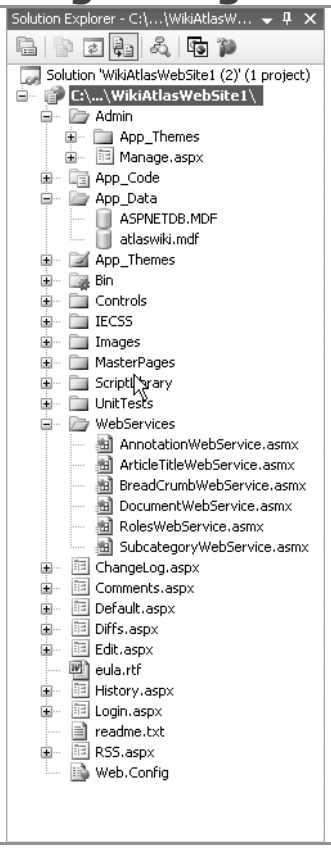
	`this.getModel = function() {`
	`return m_Model;`
	`}`
	`this.getMakeandModel = function() {`
	`return m_Make + ' ' + m_Model;`
	`}`
	`this.getYear = function() {`
	`return m_Year;`
	`}`
	`this.dispose = function() {`
	`alert('bye ' + this.getName());`
	`}`
	`}`
	`AtlasBook.Car.registerAbstractClass('AtlasBook.Car');`

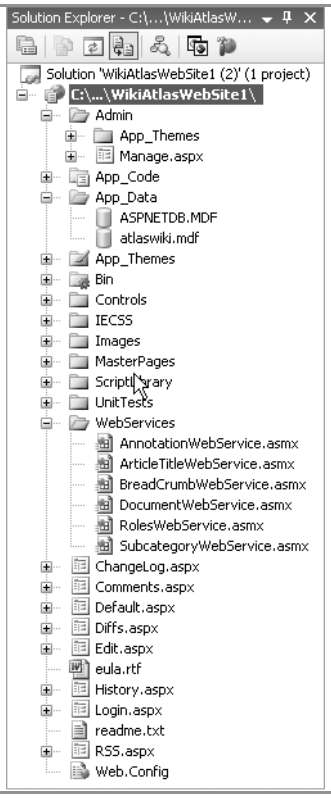
	Original Text
	Finally, the class is registered to the namespace using the Type.registerClass command. This takes three parameters, the first being the fully qualified name of the class, the second being the base type of the class (so if it inherits from another type, you would specify that here), and the last being the base interface of the class. You will learn more about inheritance and interfaces in the next sections. In this case, the base class is null, as Type doesn't inherit from any base class, and the interface is Web.IDisposable, which is an interface defined in Atlas.js that defines that the class will call the dispose method (if implemented) when it is being destroyed.

	Corrected Text
	Finally, the class is registered to the namespace using the AtlasBook.Car.registerAbstract-Class command. This takes a single parameter––the fully qualified name of the class.
45	**Original Text**
	`Type.registerClass('AtlasBook.SUV', AtlasBook.Car, Web.IDisposable);`
	Corrected Text
	`AtlasBook.SUV.registerClass('AtlasBook.SUV', AtlasBook.Car);`
	Original Text
	`Type.registerClass('AtlasBook.SUV', AtlasBook.Car, Web.IDisposable);`
	Corrected Text
	`AtlasBook.SUV.initializeBase(this,[strMake,strModel,strYear]);`
46	**Original Text**
	`Type.registerInterface('AtlasBook.IStickShift');`
	Corrected Text
	`AtlasBook.IStickShift.registerInterface('AtlasBook.IStickShift');`
47	**Original Text**
	`Type.registerClass('AtlasBook.SportsCar',` `AtlasBook.Car, AtlasBook.IStickShift);`
	Corrected Text
	`AtlasBook.SportsCar.registerAbstractClass('AtlasBook.SportsCar',` `AtlasBook.Car, AtlasBook.IStickShift);`
	Original Text
	You can see from the Type.registerClass call that the sports car is being subclassed from a Car class and implements the AtlasBook.IStickShift interface.
	Corrected Text
	You can see from the registerAbstractClass call that the sports car is being subclassed from a Car class and implements the AtlasBook.IStickShift interface.
	Original Text
	`Type.registerClass('AtlasBook.CheapSportsCar',` `AtlasBook.Car, Web.IDisposable);`
	Corrected Text
	`AtlasBook.CheapSportsCar.registerClass('AtlasBook.CheapSportsCar',` `AtlasBook.Car);`
50	**Original Text**
	Label the three test fields as Make:, Model:, and Year:, and call them txtMake, txtModel, and txtYear, respectively. Give the button the text Get Value.
	Corrected Text

	Label the three text fields as Make:, Model:, and Year:, and call them txtMake, txtModel, and txtYear, respectively. Give the button the text Get Value.
72	**Original Text**
	<script type="text/xml-script"> <page xmlns:script="http://schemas.microsoft.com/xml-script/2005"> <components> <control targetElement="panel" cssClass="start" /> <button targetElement="hideButton"> <click> <setProperty target="panel" property="visible" value="false" /> </click> </button>
	Corrected Text
	<script type="text/xml-script"> <page xmlns:script="http://schemas.microsoft.com/xml-script/2005"> <components> <control id="panel" cssClass="start" /> <button id="hideButton"> <click> <setProperty target="panel" property="visible" value="false" /> </click> </button>
73	**Original Text**
	<button targetElement="showButton"> <click> <setProperty target="panel" property="visible" value="true" /> </click> </button> <button targetElement="disableButton"> <click> <setProperty target="panel" property="enabled" value="false" /> </click> </button> <button targetElement="enableButton"> <click> <setProperty target="panel" property="enabled" value="true" /> </click> </button> <button targetElement="largeButton"> <click> <invokeMethod target="panel" method="removeCssClass"> <parameters className="small"/> </invokeMethod> <invokeMethod target="panel" method="addCssClass"> <parameters className="large"/> </invokeMethod> </click> </button> <button targetElement="smallButton"> <click> <invokeMethod target="panel" method="removeCssClass"> <parameters className="large"/> </invokeMethod> <invokeMethod target="panel" method="addCssClass"> <parameters className="small"/> </invokeMethod> </click> </button> </components> </page> </script>

	Corrected Text
	<button id="showButton"> <click> <setProperty target="panel" property="visible" value="true" /> </click> </button> <button id="disableButton"> <click> <setProperty target="panel" property="enabled" value="false" /> </click> </button> <button id="enableButton"> <click> <setProperty target="panel" property="enabled" value="true" /> </click> </button> <button id="largeButton"> <click> <invokeMethod target="panel" method="removeCssClass"> <parameters className="small"/> </invokeMethod> <invokeMethod target="panel" method="addCssClass"> <parameters className="large"/> </invokeMethod> </click> </button> <button id="smallButton"> <click> <invokeMethod target="panel" method="removeCssClass"> <parameters className="large"/> </invokeMethod> <invokeMethod target="panel" method="addCssClass"> <parameters className="small"/> </invokeMethod> </click> </button> </components> </page> </script>
	Original Text
	<control targetElement="panel" cssClass="start" /> This targetElement property specifies the ID of the raw HTML element at which this control is pointing. This is equivalent to the script you saw earlier:
	Corrected Text
	<control id="panel" cssClass="start" /> This id property specifies the ID of the raw HTML element at which this control is pointing. This is equivalent to the script you saw earlier:
	Original Text
	<button targetElement="hideButton">
	Corrected Text
	<button id="hideButton">
75	**Original Text**
	<button targetElement="largeButton">
	Corrected Text

	<button id="largeButton">
126	**Original Text**
	This creates a solution containing references to the Atlas binaries as well as the Atlas script libraries. For more on this, see Chapter 3.
	Corrected Text
	This creates a solution containing everything you need for an Atlas web site including the Atlas binaries and required support files, including the settings to web.config that allow for Atlas Web services. For more on this, see Chapter 3.
180	**Original Text**
	The Atlas wiki application is a reference application that demonstrates what a fully featured ASP.NET 2.0 application looks like and how it can be enhanced using Atlas.
	Corrected Text
	The Atlas wiki application is a reference application that demonstrates what a fully featured ASP.NET 2.0 application looks like and how it can be enhanced using Atlas. The Wiki Application was developed and demonstrated on the April CTP of Atlas. It is not presently available for the most recent CTP (July 2006), but may still be downloaded from http://go.microsoft.com/fwlink/?LinkId=56430. While it isn't supported on the most recent CTP, it is still worth investigating to learn how a real-world style application may be built using ASP.NET with the Atlas toolkit.
	Original Figure
	
	Corrected Figure

	
194	**Original Text**
	<listView targetElement="results5"
	Corrected Figure
	<listView id="results5"
195	**Original Text**
	<itemTemplate> <template layoutElement="resultsItemTemplate5"> <label targetElement="resultsItemDelimiter5"> <bindings> <binding dataPath="CategoryID" property="text" transform="XFormBreadCrumbDelimeter" /> </bindings> </label> <hyperLink targetElement="resultsItemLink5"> <bindings> <binding dataPath="CategoryID" property="navigateURL" transform="XFormSubcategoryLink" /> </bindings> </hyperLink> <label targetElement="resultsItemLinkText5"> <bindings> <binding dataPath="Title" property="text"/> </bindings> </label> </template> </itemTemplate> </listView>
	Corrected Text

```
<itemTemplate>
<template layoutElement="resultsItemTemplate5">
<label id="resultsItemDelimiter5">
<bindings>
<binding dataPath="CategoryID"
property="text"
transform="XFormBreadCrumbDelimeter" />
</bindings>
</label>
<hyperLink id="resultsItemLink5">
<bindings>
<binding dataPath="CategoryID"
property="navigateURL"
transform="XFormSubcategoryLink" />
</bindings>
</hyperLink>
<label id="resultsItemLinkText5">
<bindings>
<binding dataPath="Title" property="text"/>
</bindings>
</label>
</template>
</itemTemplate>
</listView>
```

236

Original Text

You achieve this fade using Web.UI.FadeAnimation. Tied to the button's OnClick event is the following JavaScript:

```
function FadeIn()
{
var a = new Web.UI.FadeAnimation();
a.set_target(Web.Application.findObject('Description'));
a.set_effect(Web.UI.FadeEffect.FadeIn);
a.play();
}
```

Here an instance of a new Web.UI.FadeAnimation is created and gets tied to the Description object. The effect is set to Web.UI.FadeEffect.FadeIn, and when the play method is called, the animation trips, and the layer fades in. In a similar manner, you can fade out a layer using Web.UI.FadeEffect.FadeOut like this:

```
function FadeOut()
{
var a = new Web.UI.FadeAnimation();
a.set_target(Web.Application.findObject('Description2'));
a.set_effect(Web.UI.FadeEffect.FadeOut);
a.play();
}
```

You can see the effect of this in Figure 9-5 and Figure 9-6.

Corrected Text

You achieve this fade using Sys.UI.FadeAnimation. Tied to the button's OnClick event is the following JavaScript:

```
function FadeIn()
{
var a = new Sys.UI.FadeAnimation();
a.set_target(Sys.Application.findObject('Description'));
a.set_effect(Sys.UI.FadeEffect.FadeIn);
a.play();
}
```

	Here an instance of a new Sys.UI.FadeAnimation is created and gets tied to the Description object. The effect is set to Sys.UI.FadeEffect.FadeIn, and when the play method is called, the animation trips, and the layer fades in. In a similar manner, you can fade out a layer using Sys.UI.FadeEffect.FadeOut like this:
	
	function FadeOut()
	{
	var a = new Sys.UI.FadeAnimation();
	a.set_target(Sys.Application.findObject('Description2'));
	a.set_effect(Sys.UI.FadeEffect.FadeOut);
	a.play();
	}
	
	You can see the effect of this in Figure 9-5 and Figure 9-6.

241	**Original Text**
	Here you create an instance of a Web.UI.Image control using the <image> tag and point it at the underlying HTML element called i.
	Corrected Text
	Here you create an instance of a Sys.UI.Image control using the <image> tag and point it at the underlying HTML element called i.