SYS-CON

DEVELOPING PHOTO GALLERIES USING DREAMWEAVER

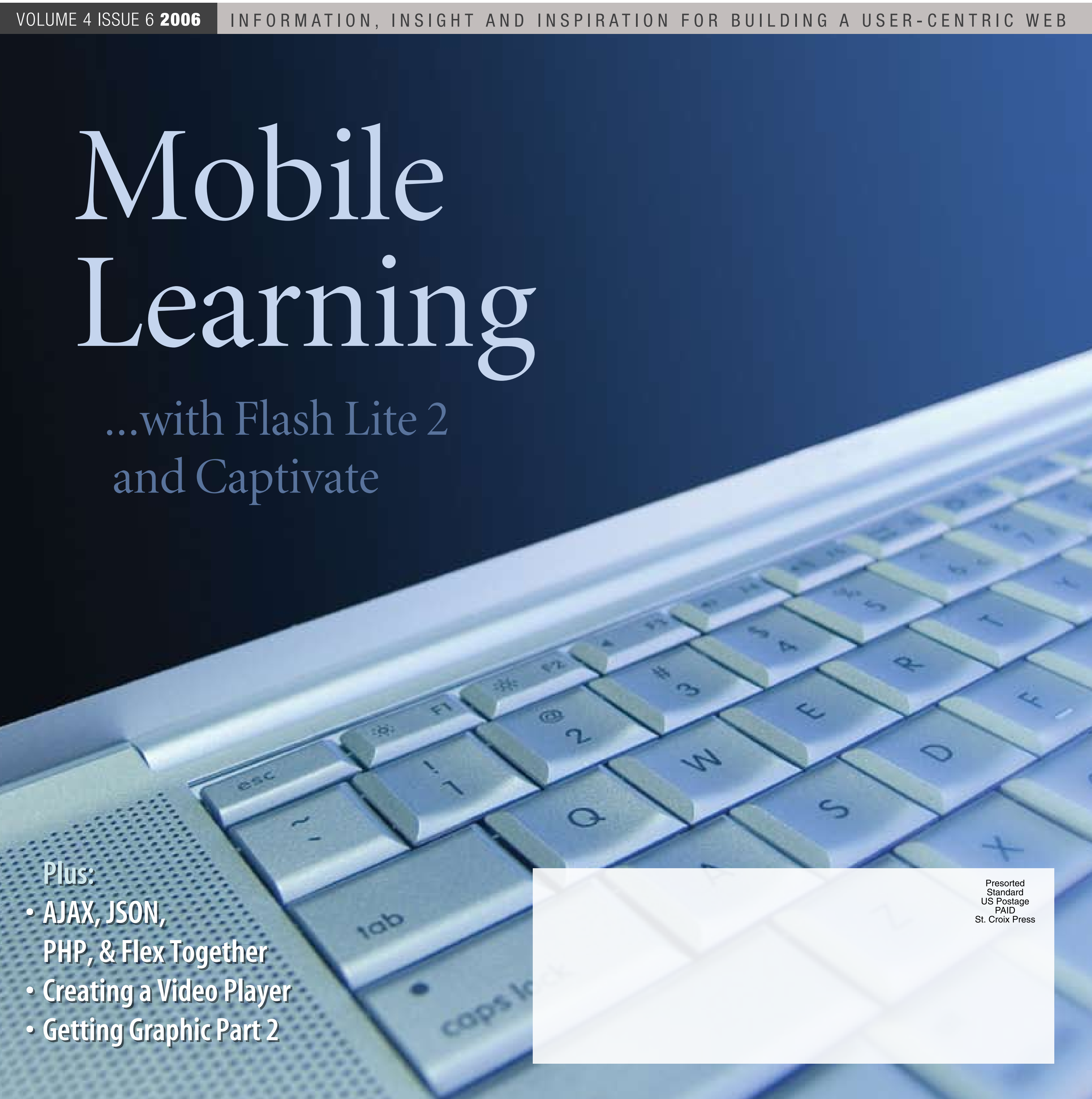
FORMERLY MX DEVELOPER'S JOURNAL

Web Developer's & Designer's Journal

WWW.WEBDDJ.COM

VOLUME 4 ISSUE 6 **2006**

INFORMATION, INSIGHT AND INSPIRATION FOR BUILDING A USER-CENTRIC WEB



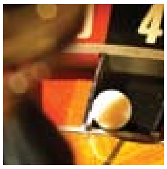
Mobile Learning

...with Flash Lite 2 and Captivate

Plus:
- AJAX, JSON, PHP, & Flex Together
- Creating a Video Player
- Getting Graphic Part 2

YUM!
INTERSPEC
TACULAR

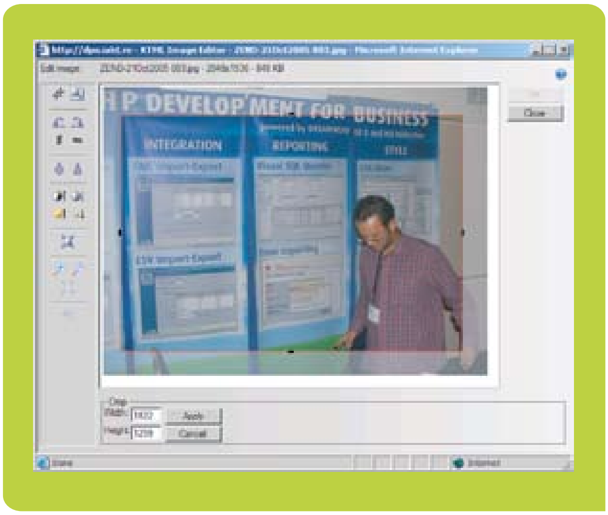
Web Developer’s &
Designer’s Journal
July 2006



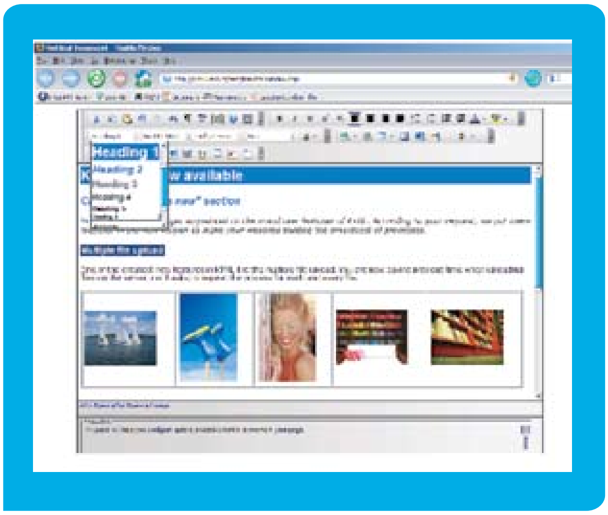
KTML4 Word editing in browser

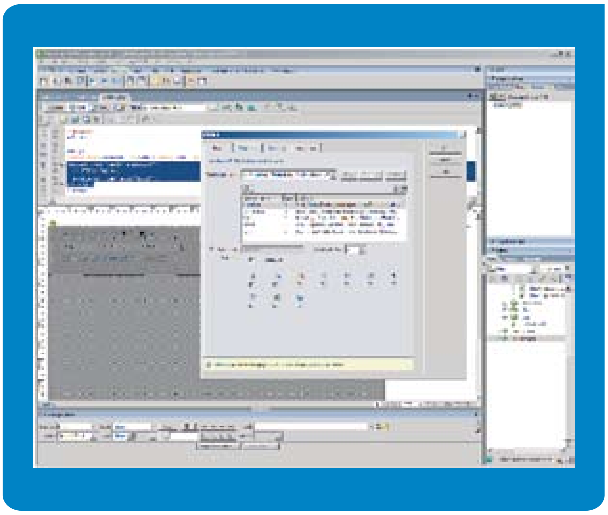
Along with the most affordable **UNLIMITED** licence

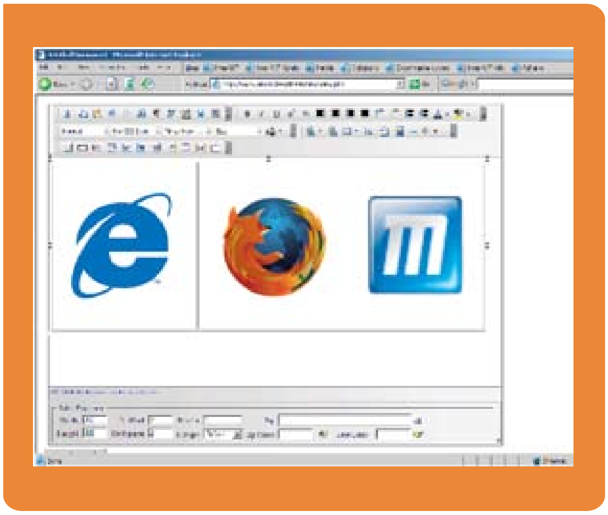
Revolutionary Image Editor



Word-like visual CSS styles







Fast Dreamweaver integration

Instant paste from Word
Incredible speed
Easy to use Word-like toolbars
Improved CSS authoring
Remote File Explorer
XHTML 1.1 compliant

Wide browser compatibility

Multiple file upload at once
HTML Table Editor
Support for multimedia (Flash, Avi)
Documents management (.doc, .pdf)
Page templates
WAI compliant

Please visit **www.interaktonline.com/ktml4/** for details

work smart

Interakt

Web Developer's &
Designer's Journal

Group Publisher Roger Strukhoff
Art Director Louis F. Cuffari

Editorial Board
Dreamweaver Editor
Dave McFarland

Flash Editor
Brian Eubanks

Fireworks Editor
Joyce J. Evans

FreeHand Editor
Louis F. Cuffari

Director Editor
Andrew Phelps

Captivate Editor
Tom Green

International Advisory Board
Norway
Jens Christian Brynildsen

UK
David Hurrows

USA
Joshua Davis, Jon Gay, Craig Goodman,
Phillip Kerman, Danny Mavromatis,
Jesse Nieminen,
Gary Rosenzweig, John Tidwell

Canada
Colin Moock

Editorial
Editor
Nancy Valentine, 201 802-3044
nancy@sys-con.com

Technical Editors
Jesse Warden • Sarge Sargent

To submit a proposal for an article, go to
http://grids.sys-con.com/proposal.

Subscriptions
E-mail: subscribe@sys-con.com
U.S. Toll Free: 888 303-5282
International: 201 802-3012
Fax: 201 782-9600
Cover Price U.S. $5.99
U.S. $29.99 (12 issues/1 year)
Canada/Mexico: $49.99/year
International: $59.99/year
Credit Card, U.S. Banks or Money Orders
Back Issues: $12/each

Editorial and Advertising Offices
Postmaster: Send all address changes to:
SYS-CON Media
135 Chestnut Ridge Rd.
Montvale, NJ 07645

Worldwide Newsstand Distribution
Curtis Circulation Company, New Milford, NJ

List Rental Information
Kevin Collopy: 845 731-2684,
kevin.collopy@edithroman.com,
Frank Cipolla: 845 731-3832,
frank.cipolla@epostdirect.com

Promotional Reprints
Megan Mussa, 201 802-3024
megan@sys-con.com

Web Developer's & Designer's Journal (ISSN#1546-2242) is published monthly (12 times a year) by SYS-CON Publications, Inc., 135 Chestnut Ridge Road, Montvale, NJ 07645.

A True Truism

by Roger Strukhoff

There are a number of dichotomous truisms that become part and parcel of everyday thinking. And one thing that is generally true about them is that they're not true at all. Left-brain versus right-brain thinking, for example. Things that "girls" are good at versus what "boys" are good at. Geekspeak versus artsy-fartsy patois. The unbridgeable gap between art and science, and between science and religion.

Following one truism that is true, i.e., don't talk about politics or religion, I'll move onto the crux of my argument, which is that even the most purely focused disciplines (and minds) encompass a variety of skills, approaches, and thought processes.

The most dedicated and skillful software developers, who live in a world of unyielding grammatical precision, will tell you that there is a mystical (and unprecise) aspect in pushing the limits of their discipline. And the most brilliant designers will certainly tell you that without a solid technical foundation and understanding of tangible truths, they will achieve nothing.

It is in the spirit of this overall gestalt – the idea that Web development and design incorporates skills, approaches, and thought processes from a range of "pure" disciplines – that we continue to bring you the newly revamped Web Developer's & Designer's Journal.

This issue features straightforward advice on "Building CSS-Based Pop-Up Menus," for example, complemented by a review of "How to Develop Photo Galleries Using Dreamweaver." Lovers of alphabet soup will salivate over our feature on AJAX, JSON, PHP (and Flex, of course). Other features tackle subjects such as "Mobile Learning with Flash Lite 2 and Captivate," "Creating a Video Player," and for those who are interested in getting the full Flex experience live and in person, we present to you the "Top 10 Reasons to Attend MAX 2006."

I once heard Sun co-founder and ultra-guru Bill Joy deliver a speech about his latest thoughts on programming. He held up several recent books he had read, all with abstruse titles such as "N-Dimensional Assymetrical Modeling in Distributed Network Paradigms."

Bill is known as one of the truly premier minds in the field of software development. Yet, after giving his book reports, he segued into a conversation about the use (and potential misuse) of technology.

Someone asked him how someone so "technical" could address overall societal matters so thoughtfully. "That's the value of a liberal arts education," he said. "Read the Greeks, learn about the history of civilizations, and incorporate that thinking into everything you do."

Inspired by this enlightened outlook, we thereby present our latest issue. We hope that, through our careful planning process and commitment to presenting the complete megillah, we can enlighten and educate you, so that you can go out and create your best possible work – whether you're left-brained, right-brained, boy, girl, geek, or artiste.

Roger Strukhoff is Editorial Director and Group Publisher of SYS-CON Media, Inc.
roger@sys-con.com

Build CSS Based Pop-up Menus

...without writing any code using Fireworks 8 and Dreamweaver

by Joyce Evans

In this tutorial, you will learn how to build a completely functional pop-up menu using CSS (Cascading Style Sheets) without knowing how to write any code at all! Below is an example of such a menu;

An example of a CSS menu

This tutorial takes only 25 minutes to complete. How to make pop up menus quick and easy

With Fireworks 8 you can now design a decent pop-up menu with an optional rollover. The pop-up menu can even have submenus (several menus deep). You can customize the color, the borders, the size of the menu itself and its positioning. There is a wizard driven command that gives you a ton of control over the look and feel of the menu.

Perhaps you are thinking, "wait, I've already seen this in previous versions of Fireworks". And I'd have to say that you were correct. But there is a major difference now, and that is the ability to make the menus using CSS instead of JavaScript. Not only can you use CSS now but the CSS can be in a separate Stylesheet. The options I believe are turned on by default but be sure to check by going to File | HTML Setup | General and check that the two CSS options are both checked.

Figure 1

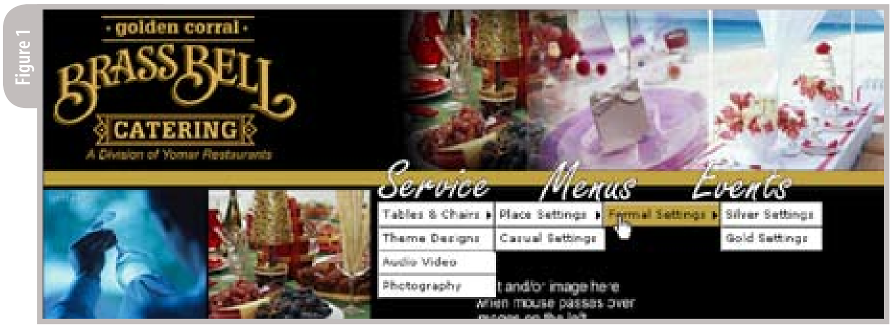
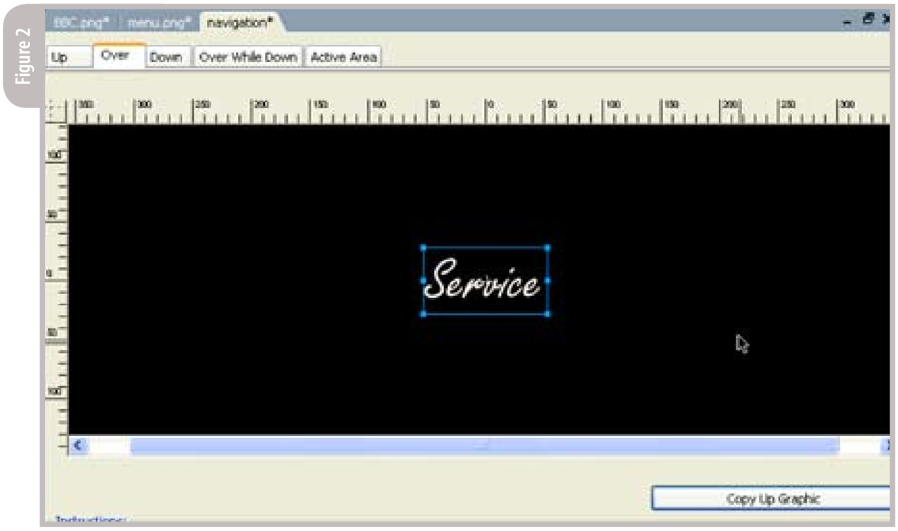


Figure 2



You can also convert pop up menus you made in previous versions of Fireworks to the new CSS based menus. To do this you simply open the menu file, change the HTML setup to use CSS styles and export again.

To begin this tutorial open Fireworks with a new canvas or an existing site. If you'd like to use my file then open BBC.png from the zip file. This is only a mockup for a client and not a completed site.

How to make navigation using symbols

Add text and/or images

The first step is to add text links or image links that you want to use for the navigation. This example uses the font, Freestyle Script, size 52, color white, smooth anti-alias, and the word Service.

Using Symbols

Select your text or image (Service), press F8, name the symbol and select Button and click OK. Press F11 to open the Library. Notice your button symbol is there. Drag the button name from the Library to the canvas—2 times.

Notice that the symbol "copies" have a little arrow in the bottom. This indicates that this button symbol is attached to the original. With a symbol, when you use edit it as seen next, it will change all instances of the symbol.

Editing Symbols

To edit a symbol, double click on Service to open the Button Editor. Click the Over tab and the Copy Up Graphic. Make changes to the over state and repeat for other states if desired. Click Done. All instances will be changed.

Link Names and Linking

I know I said if you edited the symbol all instances would be changed. The exception is that you can change the text on individual buttons. It wouldn't be much use otherwise. To change the names of additional instances of the symbol, select them and change the name in the Text field in the Property inspector. You can also add the link in the link field. Click Preview to see the rollover effects.

Tip: When adding links use relative links (you'll need to know the site structure).

Adding a Pop Up Menu

Adding the menu to the first button

Select Service, go to Modify|Pop-up Menu|Add Pop-up Menu. In the Content dialog box, add menu names and links (# is a null link). To indent a menu item (which is a sub of the menu above it), use the indent button (the second blue icon to the right of the + - icons).

Tip: If you aren't using a symbol then you need to insert a slice to your navigation element prior to adding the Pop-up menu.

Setting the menu Appearance

Select the Appearance tab. Here you set the menu colors, vertical or horizontal menus and HTML or text buttons versus an image. Also choose the button color and the text colors. In the bottom pane you'll see a preview of the menu.

Links and Styles

Select Image from the styles icons for an image link. To add a custom style, export from Fireworks and save in the Fireworks program files. Navigate to the Configuration folder then the Nav Menu folder and save. Your custom style will then show up in the editor.

Advanced Menu Options

Set the cell width and height or choose the automatic option to scale the menu automatically to the text. You can add padding and border effects if necessary. With no border you won't see a line between the links.

TIP: When I want the lines between links and no outside border, I make the border and highlight color the same as the background. This is what I did in the sample project.

Set the Position

In the Position dialog, select the position of the menu. This exercise is using the menu below the trigger. Select the submenu (the one to the side of the submenu) and click Done.

Edit the Position

In your document, select the button slice and you'll see a blue outline for the submenu. Click and drag the blue area to visually place the submenu. Right click on the slice and select Edit Pop-up Menu. Select the Position tab to see the placement change.

Tip: Repeat steps 5-10 for each separate link.

Prepare to optimize and export for use in Dreamweaver

You could optimize and export right from the client document (the comp file) but it would require a lot of work. This menu will be "floating" above the background elements in a DIV so I don't want the yellow bar or the image to show up with the navigation. It helps to know what your final use will be prior to exporting. Plus I like to keep my Fireworks menus in a separate file.

To remove the menus from the comp document, simply select the first menu, press the Shift key and select each of the other completed menus. Select Edit|Copy or Ctrl+C. Choose File|Open|New (black canvas) and paste (Ctrl+V). Save your file.

The sample file in the zip file is named menus.png and only Services is completed.

Optimize the image

Using menus.png, in slice view select the slice over the text and select the Preview tab. Open the Optimize panel (F6). In the Optimize panel select the file format. Since this is text GIF is best.

Set colors and Transparency

Select Index transparency to remove the background color. It will leave a bit of black on the edge

SYS-CON Media

President & CEO
Fuat Kircaali, 201 802-3001
fuat@sys-con.com

Group Publisher
Roger Strukhoff, 201 802-3033
roger@sys-con.com

Advertising
Senior Vice President, Sales & Marketing
Carmen Gonzalez, 201 802-3021
carmen@sys-con.com

Vice President, Sales & Marketing
Miles Silverman , 201 802-3029
miles@sys-con.com

Advertising Sales Director
Robyn Forma, 201 802-3022
robyn@sys-con.com

Advertising Sales Manager
Megan Mussa, 201 802-3023
megan@sys-con.com

Associate Sales Managers
Kerry Mealia, 201 802-3026
kerry@sys-con.com
Lauren Orsi, 201 802-3024
lauren@sys-con.com

Production
Lead Designer
Louis F. Cuffari, 201 802-3035
louis@sys-con.com

Art Director
Alex Botero, 201 802-3031
alex@sys-con.com

Associate Art Director
Tami Beatty, 201 802-3038
tami@sys-con.com

Assistant Art Director
Abraham Addo, 201 802-3037
abraham@sys-con.com

Video Production
Ryan Palmieri, 201 802-3036
ryan@sys-con.com

SYS-CON.COM
Consultant, Information Systems
Robert Diamond, 201 802-3051
robert@sys-con.com

Web Designers
Stephen Kilmurray, 201 802-3053
stephen@sys-con.com
Wayne Uffleman, 201 802-3004
wayne@sys-con.com

Accounting
Financial Analyst
Joan LaRose, 201 802-3081
joan@sys-con.com

Accounts Payable
Betty White, 201 802-3002
betty@sys-con.com

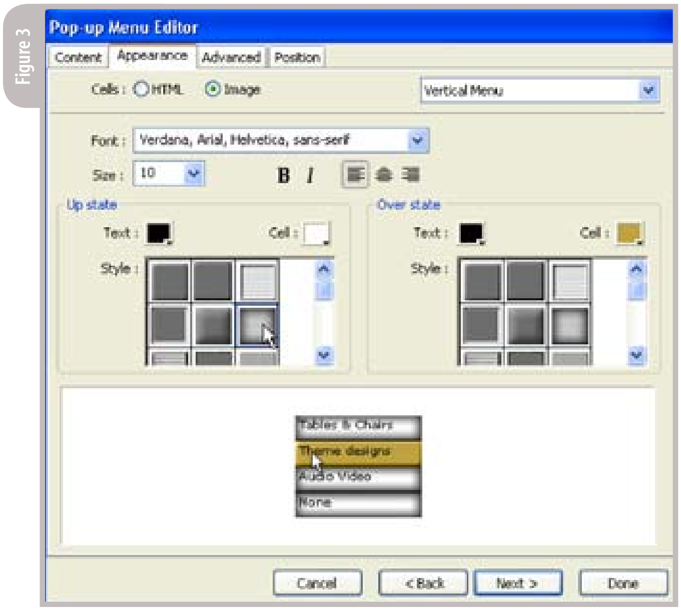
Accounts Receivable
Gail Naples, 201 802-3062
gailn@sys-con.com

Events
Conference Chair
Jeremy Geelan, 201 802-3051
jeremy@sys-con.com

Customer Relations
Circulation Service Coordinators
Edna Earle Russell, 201 802 -3081
edna@sys-con.com

JDJ Store Manager
Brundila Staropoli, 201 802-3000
bruni@sys-con.com

Figure 3



Pop-up Menu Editor

Content | Appearance | Advanced | Position

Cells : ○ HTML ● Image — Vertical Menu

Font : Verdana, Arial, Helvetica, sans-serif

Size : 10

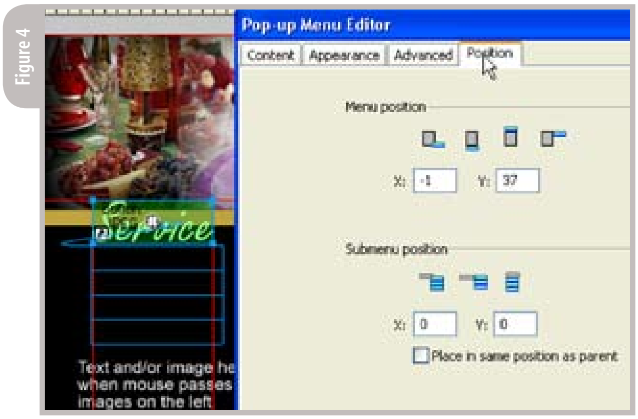
Up state — Text : ■ Cell : □ Style :

Over state — Text : ■ Cell : ■ Style :

Tables & Chairs
Theme designs
Audio Video
None

Cancel | < Back | Next > | Done

Figure 4



Pop-up Menu Editor

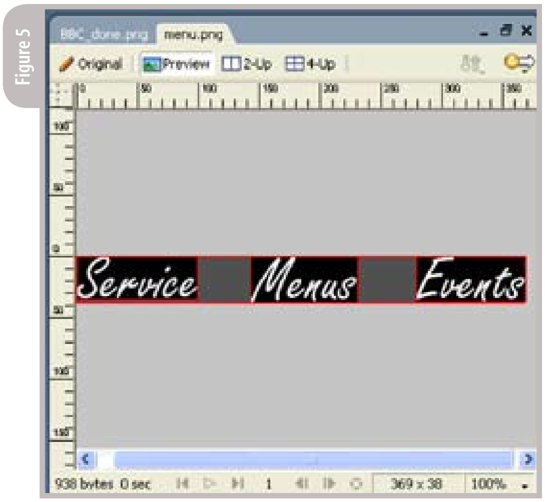
Content | Appearance | Advanced | Position

Menu position — X: -1 Y: 37

Submenu position — X: 0 Y: 0

□ Place in same position as parent

Figure 5



938 bytes 0 sec | 1 | 369 x 38 | 100%

but for our purposes that is fine. Use 64 colors, any less and the anti-aliasing won't be as good. Set Matte to none.

HTML setup

Choose|File|HTML setup. In the General tab be sure the 2 CSS options are checked. In the Table diaglog, select "Space with a single table, no spacers" and select "Use Canvas color". Click OK.

Export the menus

Select File|Export, the export should be set to HTML and Images so you get the CSS menus and the JavaScript. Be sure slices is set to "Export Slices" and do not check "include areas without slices". Export to you root folder.

Files to upload to your server

You should export all your graphics from Fireworks into your sites root folder. The export folder we just completed is in the BBC site. Note the files that got exported. There is a JavaScript file and a CSS menu. The JavaScript file needs to be in every folder that has a page using the menu. If you don't do this, the menus won't work, this is a limitation of the Fireworks pop-up menus. They are CSS based but there is a small bit of JavaScript still utilized. These are not pure CSS menus but a huge improvement over previous versions in Fireworks and Dreamweaver. Also note the images folder. The rollovers and an arrow for the submenus were automatically exported. You may want to rename the images.

Putting the menus into a Dreamweaver layout

Define a Dreamweaver site

Before you start, copy the source.zip to your hard drive.

Go to Site|Manage Sites and click on New site. Name the site and navigate to the sites root folder (BBC). You'll need to also setup the Remote server before you can upload from Dreamweaver.

Setup the FTP

Click on Remote Info (Advanced tab). For Access select FTP. Fill in your host info (URL or IP address) as well as your FTP login information. Click OK when you are finished then Done to close.

For answers call us at 1-866-EDGEWEB

3 3 4 3 9 3 2

When calling your web host for support you want answers, not an annoying song stuck in your head from spending all day on hold. At Edgewebhosting.net, we'll answer your call in two rings or less. There's no annoying on-hold music, no recorded messages or confusing menu merry-go-rounds. And when you call, one of our qualified experts will have the answers you're looking for. Not that you'll need to call us often since our self-healing servers virtually eliminate the potential for problems and automatically resolve most CF, ASP, .NET, SQL, IIS and Linux problems in 60 seconds or less with no human interaction.

Sleep soundly, take a vacation, and be confident knowing your server will be housed in one of the most redundant self-owned datacenters in the world alongside some of the largest sites on the Internet today and kept online and operational by one of the most advanced teams of skilled Gurus, DBAs, Network and Systems Engineers.

By the Numbers:

- 2 Rings or less, live support
- 100% Guarantee
- 99.999% Uptime
- 2.6 GBPS Redundant Internet Fiber Connectivity
- 1st Tier Carrier Neutral Facility
- 24 x 7 Emergency support
- 24 Hour Backup
- Type IV Redundant Datacenter

EDGE
WEB HOSTING

For a new kind of easy listening,
talk to EdgeWebHosting.net

http://edgewebhosting.net

CFDJ READERS' CHOICE AWARD

MXDJ READERS' CHOICE AWARD

2003 - 2006

◦ Shared Hosting ◦ Managed Dedicated Servers ◦ Managed Colocation ◦ Semi-Private Servers
◦ ColdFusion ◦ BlueDragon ◦ ASP ◦ .NET ◦ .Linux ◦ .Java ◦ SQL Server ◦ .MySQL ◦ Self-Healing Servers

“With Fireworks 8 you can now design a decent pop-up menu with an optional rollover”

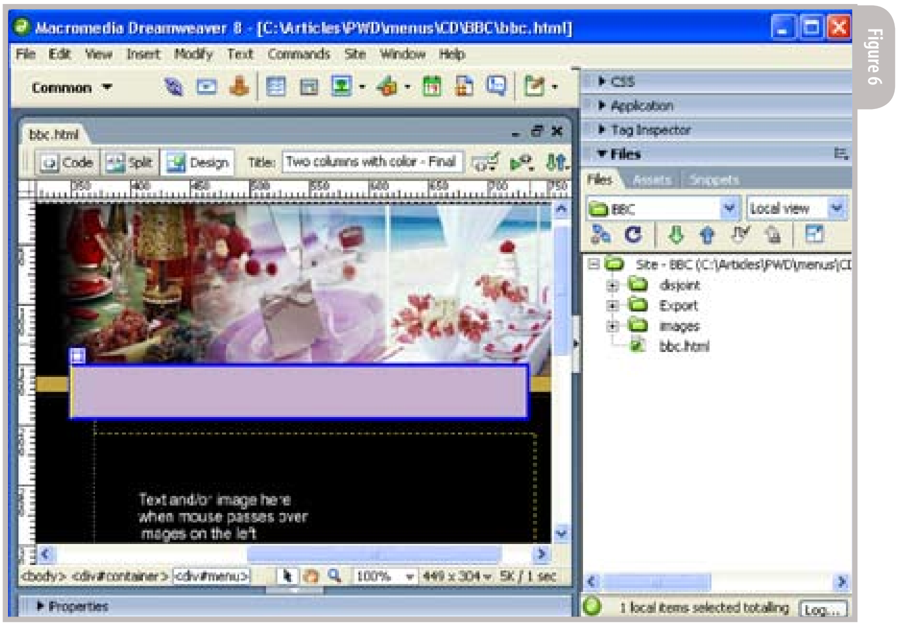
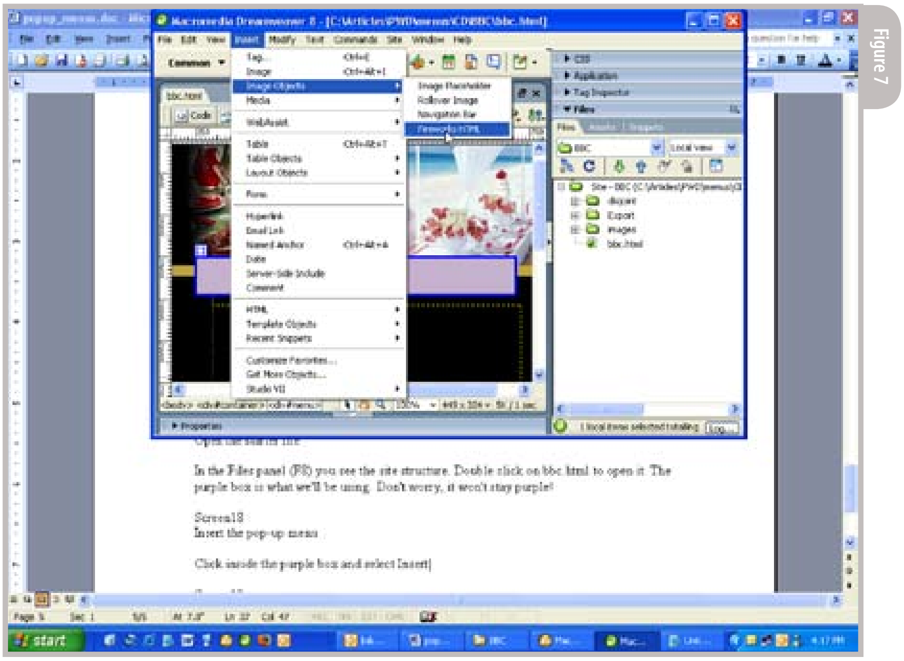

Macromedia Dreamweaver 8 - [C:\Articles\PWD\menus\CD\BBC\bbc.html]
File Edit View Insert Modify Text Commands Site Window Help
Common
bbc.html
Code Split Design Title: Two columns with color - Final
Text and/or image here when mouse passes over images on the left
<body> <div#container> <div#menu>
100% 449 x 304 5K / 1 sec
Properties
CSS
Application
Tag Inspector
Files
Files Assets Snippets
BBC
Local view
Site - BBC (C:\Articles\PWD\menus\CD
disjoint
Export
images
bbc.html
1 local items selected totalling Log...

Figure 6



Open the starter file

In the Files panel (F8) you see the site structure. Double click on bbc.html to open it. The purple box is what we'll be using. Don't worry, it won't stay purple!

Screen18
Insert the pop-up menu

Click inside the purple box and select Insert|

Figure 7

Joyce J. Evans is a training veteran with over 10 years of experience in educational teaching, tutorial development, and Web design. She has spoken at conferences such as Macromedia MAX, TODCON and Seybold. Joyce has authored and co-authored over 15 books including Macromedia Studio MX 2004 Bible, Dreamweaver MX 2004 Complete Course, Web Design Complete Course. Fireworks MX 2004: Zero to Hero and Dreamweaver in 10 simple steps or less. Joyce founded, designed, and now maintains Idea Design (www. ideadesignwebsites.com), a web design studio. Joyce’s current projects include several training CD’s for Dreamweaver 8, Fireworks 8. joyce@joyceevans.com

Open the starter file

In the Files panel (F8) you see the site structure. Double click on bbc.html to open it. The purple box is what we’ll be using. Don’t worry, it won’t stay purple!

Insert the pop-up menu

Click inside the purple box and select Insert|Image Objects|Fireworks HTML. Browse to the Export folder and select menus.png, click OK. The menu will now be in the purple div box.

Edit the CSS

To edit the style that makes the div purple, open the CSS styles panel (Shift+F11) and you’ll see this sites styles. Menu.css was added by the Fireworks insert. Scroll down towards the bottom and select #menu. In the Properties for “#menu” area, click the background color box and click on the Trash can to delete it.

Preview in a browser

Press F12 to preview in your default browser. Test the Services rollover. Do you see your menu? If not then you need to go to the Export folder of the BBC folder and copy the mm_ccc_menu.js file and paste into the BBC folder so it’s with the bbc.html file.

That’s all there is to building a CSS dropdown menu!

How to Develop Photo Galleries Using Dreamweaver

Sharing photos

by Joyce Evans

Time it takes: *15 minutes*

Requirements: *Fireworks 8, Dreamweaver*

What you'll learn:

You'll learn how to build several variations of photo galleries without even touching the code. Two galleries use Dreamweaver extensions and one is a Flash movie—without Flash!

Level: *Beginner to Intermediate*

Everyone is toting a digital camera these days it seems and they have photos to share with you. I'm sure you're not one of those who want to bore family and friends with your gazillion pictures of your dog/cat/snake. Even if you are, you can now do it with some pizzazz and it's easier than you may think. You could send the photos via email but then they'd take forever to arrive and your family probably doesn't know how to open them anyway. So the best solution is a spiffy new photo gallery.

Photo galleries aren't just for displaying personal pictures but they are widespread in the business world as well. The obvious use is for photographers. But there are a lot of less obvious uses as well, such as a portfolio of products, designs, sketches etc. I've built the Flash version to display photos of office staff and to display products. They can even be used in online catalogs. The more you use your imagination, the more uses you'll come up with.

There are so many different solutions to display your images. Some involve a database and others use Flash. Most are quite difficult to build by the average user and require some programming skills and definitely code editing skills. A popular method is to have thumbnails that when clicked display larger images on the same page. This involves a lot of show and hide layers in Dreamweaver and is a bear to edit. An inexpensive alternative I've found to this is an extension sold at DMXzone.com called "Flash Album Generator". It's the easiest to build and edit that I've come across so far.

The projects we are going to build today are all free and are very easy to use and build. These tools are great when you need to build or design a photo gallery in a hurry or on a budget.

Have you seen galleries that take forever to load? How about the ones that the image displays one strip at a time? These are photos that beginners have posted without resizing and worse yet, without optimizing. The tools you'll be using today will resize and optimize—to a degree. It's still best to process your images using Fireworks before building your gallery. A link is provided in the projects for doing batch processes in Fireworks.

In the past year I've had more requests to build photo galleries then ever before and as prices continue to drop and the cameras get smaller and easier to use, the demand will only get higher. If you haven't built a gallery yet, then the time is now. Roll up your sleeves and let's have some fun.

To begin these projects you'll need some photos—go figure.

Build a quick and easy photo gallery

Sceen1: Page setup

Open a new page and save it. Be sure to add your page title now. The Web Photo Album that ships with Dreamweaver has no design templates. But this is a fine alternative if you want to place the gallery into your own designs. Select Commands|Create Web Photo Album

Screen2: Fill in the dialog box

Everything is right here in the same

Joyce Evans is an active Web Designer and Consultant. She has spoken at conferences such as MAX, Seybold, DesignFest etc. She loves to teach and help others get excited about Web Design! www.JoyceJEvans.com

dialog box. Give it a title, subtitle if desired. Navigate to your photos folder and a destination folder. Select the size for the thumbnails and if you want the file names to show. Select the image quality and scale the images if needed. Scale large images and click OK.

Screen3: Batch Process in Fireworks

The thumbnail resizing is actually done with a batch process in Fireworks. If you scaled the large images down at all, then they will also be batch processed to the percentage of the scale. When the batch is done, return to Dreamweaver and click on OK to the Album has been created successfully dialog.

Screen4: The photo album

Press F12 to preview in your default browser. Notice the image names? If you are going to show file names then you'll want to either rename the files or rename in Dreamweaver. Click on any of the thumbnails for a larger view. Navigation forward and back is also provided automatically.

Boxout (mini 3 step): Editing the album

You can dramatically change this album very easily by editing the tables and images. You may want to move the taller images together or change the placement of the photos. It's nicer if you put like images together. If you are using titles, change them and alter the font properties.

box1

The first thing we will do is change the header area.

Select the top table and make the size 580 pixels wide, change the rows to 1 and name it "header". Delete the title. In the Property inspector click on "bg Image" and browse to the background image (on the CD) to insert it. Add an embedded style for the table with a height of 82.

box2

Adjusting the content area to match the size of the header table.

Now select the lower table and make it 580 pixels wide (assuming you used 100px wide thumbnails and 5 columns). The idea is to make the top and lower table the same width.

box3

Change the background color of the content table

With the larger table selected, click in the bk color box and use the dropper to sample a purple color from the flowers. Click your cursor between the 2 tables and delete the extra space. Now preview and see the big difference with very little effort.

Time it takes: 10 minutes
Requirements: Dreamweaver

Downloading and installing a Dreamweaver extension

Screen5: Register at Macromedia.com

From the Help menu, select Dreamweaver exchange. Once there you'll need to create a login and password if this is your first time here. Don't worry, it's free. By registering with Macromedia, you'll also get cool updates and notices when new content has been added to the Developer center.

Screen6: Go to the Dreamweaver Exchange

Once logged in you'll see a Category menu top left. In the right column you can search by highest rated, newest or most downloaded extensions. Note all the availability, download, rating and product information of each extension. At the top right, click on the Search Exchanges link.

Screen7: Find a specific extension type

Type in "photo album", select Dreamweaver exchange and click on Search. Check out some of the extensions, some are very good. Scroll down to the Web Photo Album 2.2 for Dreamweaver MX and 2004 (by Macromedia). Click on the name for more information. Click the Download button to get the extension.

Screen8: Install an extension

Now that you have the extension, you'll need to install it. I keep all Dreamweaver extensions in the same folder using subfolders. Locate the WebPhotoAlbum226.mxp file you just downloaded and double-click on it. Accept all the screens and your extension will be installed automatically. If Dreamweaver is open, you'll need to close it and re-open it.

Screen9: Extension Manager

Let's take a quick look at the Extension Manager. You can either open it from Programs|Macromedia|Extension Manager or from Dreamweaver. In Dreamweaver select Commands|Manage Extensions. You can click on installed items to turn them off or you could install from here. Note that the other Macromedia application extensions are all controlled from here.

Screen10: Define a site

You'll need to define a site or open the one you are putting the photo gallery into. Select Site|Manage Sites|Add New|Site. Name it, navigate to the sites folder and setup remote information to upload your files. The step by step instructions for defining a site can be found in last months article or from the Dreamweaver Help files.

Time it takes: 20 minutes
Requirements: Fireworks 8, Dreamweaver

Build a Photo Gallery with a Dreamweaver Extension

Screen 11: No page needed

With your defined site open, select Commands|Web Photo Album 2.2. The dialog window that opens gives you 2 different layout options. The first one is like the one we just built so select the "Create a Photo Album that includes only navigation pages" option. Click on Next.

Screen12: Choosing a design

Select "Create Page from Template" and check out all the options. I'm selecting Floral Yellow. You can even select to go to the Exchange for more templates. Click Next. Click Add Folder and navigate to your photos folder. Select and remove any images you don't want. Click Next

Screen13: Finishing the album build

Browse to the destination folder, give it a title and select the navigation style you want. I'm going to scale to 30% since they are large images. You can choose

> “There are so many different solutions to display your images. **Some involve a database and others use Flash”**

to add captions or not. I’m not going to. Click Next then Finish. The images will again process in Fireworks. Click OK, isn’t that a nice gallery?

Screen14

Try another variation. Repeat only this time select the first option with the thumbnail page. I used the green floral and changed the navigation to no loop. Click on any image to see the larger view and the caption which was added in Fireworks. Each image as it was processed prompted for a caption.

Screen15: Check the image size and resolution

I recommend if you use either of these two extensions that you batch process the larger images yourself using Fireworks. The extensions resize the images but do not change the resolution. Camera resolution may be 300 dpi which is far too much for the web.

Screen16a: Batch Processing in Fireworks

Open File|Batch Process and navigate to your folder of images. You can then add some or all of the images and click Next. Follow the instructions throughout the wizard. You can size, optimize and even apply commands to an entire folder of images.

More info: If you are new to batch processing, I have a movie tutorial here: http://www.joycejevans.com/tutorials/Fireworks/FWMX/Batch/batch.html

That will demonstrate step by step how to do it.

Time it takes: 20 minutes
Requirements: Dreamweaver

Build a Flash gallery in Dreamweaver

Screen16: Setting up the page

Open a new document, save and give it a page title. Go to the Insert bar and select Flash Elements. Or select the tab, depending on your view. I know it says Elements plural, but there is only one element called the “Flash Viewer” I believe the intent was that more would be developed but here we are on the second year since it was added and it’s still all alone.

Screen17: Add the Flash Element

Click on the Flash Element. The “Save Flash Element” dialog box opens. Navigate to where you’d like to save the generated Flash movie. Name it and save. You will now notice a gray box added to your document. Notice the Property inspector, how it’s changed. You’ll be able to preview right here in Dreamweaver.

Screen18: Flash Element Panel

Look in the Panels area, there is a new panel added “Flash Element”. You’ll see all the properties available. This is where you’ll edit the movie file to display your photos. You should have your images resized prior to inserting. Select the Flash Element, in the Property inspector change the dimensions to just a little larger than your largest image.

Boxout annotation. (I don’t know how much space this will take so I may need to do some editing when I return from vacation. I’m winging this)

Use image annotations.gif (but screen18.tif cropped please)

1. ***bgColor:*** Click in the bgColor box and select a color. This is the background of the Flash movie not of your document.
2. ***imageLinks*** (I have a screeshot if it can be fit in): You can add links that correspond to each image if you’d like. Click the little pencil icon to open the Image Links Array dialog. Add links in the same order as image URLs.
3. ***imageURLs:*** Click the pencil icon, enter the URL of each image. Click in the first name and click the yellow folder to navigate to your image files. Click the + button to add more image links.
4. ***showControls/slideAutoPlay:*** These two kind of go together. You can have the movie AutoPlay with or without navigation controls.
5. ***slideDelay:*** The number you enter determines how much time elapses before the next slide is seen. You’ll probably need to test a few times to get the timing just right.
6. ***transitionsType:*** There are 11 types of transitions, you’ll want to test them all! Or you can choose to have no transition at all.

Screen19 (depending on room left I can expand or contract these last couple of steps)

Test the movie

Select the Flash Element and in the Property inspector click on Play. For me a delay of 5 is too much so I’m changing it to 3. Stop the movie, change the transition, then play again to test. Repeat as often as needed.

Screen20: Preview in a Browser

Press F12 to view in your default browser. This Flash Element can be used in any design. Just be sure to size the images and viewer to fit in the available space.

Resources

- Free expert tutorials, movies and reviews for books and products. www.JoyceJEvans.com:
- New Macromedia resource connects you to help and forums www.macromedia.com/community/
- Free download of many Macromedia extensions (some for purchase). http://www.macromedia.com/cfusion/exchange/

AJAX, JSON, PHP, and Flex Together

Building a Google-finance type site

by Mike Potter

We've all seen Google finance, and the great job that it does at mixing HTML content with Flash content, as seen on company stock quotes. Google has done a great job at using Flash where it makes the most sense, in the graphs for company quotes, with HTML where it makes sense (linking to news items). Today, I'll show you how to build your own Google-finance type site, using a combination of Ajax, JSON, PHP and Flex / Flash. And, best of all, we'll do it all for free.

In this example, we'll grab data from a Feedburner feed, and then show the hits to that feed in a graph, over time. We'll bring in the RSS information from our blog feed, and when we click on the items in the graph, they'll highlight the RSS feed items for that day.

To complete this tutorial, you'll need the following software / accounts:
1.Note: You can perform this same tutorial without Flex Charting. You would need to create your own Flex component, and you could do that with the SDK. So, technically this tutorial could be accomplished for $0. In my case, to reduce development time, I'm using the Flex Charting components, which cost $249.

The first part to this tutorial is to install all the above software. I won't go through that here, if you have problems please comment in my blog.

Believe it or not, there actually are pieces of Adobe software that I don't use, feel free to use Photoshop to design the appearance of the webpage, Lightroom to hold photos of pepople you will put on the HTML page, Premiere to do video editing, ColdFusion can replace the PHP part fairly easily etc... I leave it up to you to extend this tutorial to make use of every piece of Adobe software.

Once we've got everything installed, we'll start by hooking up our PHP backend to Spry. The reason that we need the PHP backend is because Ajax applications cannot load data from outside sources: we need to load data from the same source as the Ajax application. So, we've created a small PHP file that will go to Feedburner, get the stats and then print the output. That PHP file is very small and very simple:

```
<?php
$fpURL = ‘http://api.feedburner.com/
awareness/1.0/GetFeedData?uri=adobe/
mpotter&dates=2006-07-01,2006-07-17’;
$handle = fopen($fpURL, “r”);
while (!feof($handle)) {
$strOutData .= fread($handle, 8192);
}
fclose($handle);
header(‘Content-type: text/xml’);
echo $strOutData;
?>
```

Any server side language could do something similar.

And here's the JavaScript code for Spry to call and load that PHP file:

```
var dsFeedburner = new Spry.Data.
XMLDataSet(“getdata.php”, “/rsp/feed/
entry”);
```

When the HTML page loads, it will call getdata.php, and populate the Spry dsFeedburner data source with data from that file. Here's a sample of the XML content that gets output from that PHP file.

```
<?xml version=”1.0” encoding=”UTF-8”?>
<rsp stat=”ok”>
<!--This information is part of the
FeedBurner Awareness API. If you want
to hide this information, you may do
so via your FeedBurner Account.-->
<feed id=”412263” uri=”adobe/mpotter”>
<entry date=”2006-07-01” circula-
tion=”0” hits=”0”/>
<entry date=”2006-07-02” circula-
tion=”0” hits=”0”/>
...
```

The text "/rsp/feed/entry" is simply an xpath expression to get to each <entry... item in the XML file.

So, now we've got the data. To populate a table with data, we do the following:

```
<table border=”1” spry:
region=”dsFeedburner”>
<tr>
<th onClick=”dsFeedburner.sort(‘@
date’);”>Date</th>
<th onClick=”dsFeedburner.sort(‘@
hits’);”>Hits</th>
</tr>
<tr onclick=”alert({@date});” spry:
repeat=”dsFeedburner” spry:select=”Sel
ectedFeedburnerItem” spry:hover=”Hover
FeedburnerItem”> <td>{@date}</td>
<td>{@hits}</td>
</tr>
</table>
```

We repeat each table row (<tr>) with the spry:repeat=”dsFeedburner” attribute. We set the selected item’s CSS class to “SelectedFeedburnerItem”, and the hover state to “HoverFeedburnerItem”. You can edit the style for those items with simple CSS, in the HTML file (or an external CSS file, whichever you prefer.)

OK, so if we run that, then we should see a repeatable table showing dates and hits to our items. Pretty good, now let’s hook that up to a graph component that we made in Flex.

Its important to note here that Flex can be used to create a number of components. Its possible, with a little work, that you could create a component that exactly matches the graphing component that Google uses, with a date slider / selector at the top, above the graph. I won’t do that here, for the sake of simplicity, but it is possible. Or, you could create an Ajax media browser that plays videos in Flash. The possibilities are limited only by your imagination.

So, let’s go ahead and create our Flex graph... Here’s the MXML code:

```
<?xml version=”1.0” encoding=”utf-8”?>
<mx:Application xmlns:mx=”http://www.
adobe.com/2006/mxml” layout=”absolute”
width=”400” height=”400” backgroundGra
dientColors=”[#ffffff, #ffffff]”>
<fab:FABridge xmlns:fab=”bridge.*” />
<mx:Script>
<![CDATA[
import mx.charts.events.
ChartItemEvent;
import mx.collections.ArrayCollection;
import com.adobe.serialization.json.
JSON;
public function bindJSONToChart(
JSONString:String ):void {
var arr:Array = (JSON.
decode(JSONString) as Array);
linechart.dataProvider = new
ArrayCollection(arr);
}
public function
chartClicked(clickEvent:
ChartItemEvent):void
{
if( ExternalInterface.available )
ExternalInterface.call(“chartClicked”,
clickEvent.hitData.item);
}
]]>
</mx:Script>
<mx:LineChart id=”linechart”
paddingLeft=”5” paddingRight=”5”
showDataTips=”true” left=”10”
right=”10” top=”10” bottom=”10” itemCl
ick=”callEI(event)”>
<mx:horizontalAxis>
<mx:CategoryAxis categoryField=”@date”
displayName=”Date”/>
</mx:horizontalAxis>
<mx:series>
<mx:LineSeries yField=”@hits”
displayName=”Hits”/>
</mx:series>
</mx:LineChart>
</mx:Application>
```

Everything should be fairly self-explanatory. The only lines that may be unfamiliar are:

```
<fab:FABridge xmlns:fab=”bridge.*” />
which is needed for the Flex / Ajax
bridge to function properly and:
public function callEI(func:String,
clickEvent:ChartItemEvent):void
{
if( ExternalInterface.available )
ExternalInterface.call(“chartClicked”,
clickEvent.hitData.item);
}
which is needed to call back to the
page that contains the Flash file, in
our case the Ajaxed HTML page.
OK, so we’ve got our Flex application
built, we’ve got our Ajax file, now we
need to pass data from the Ajax page
to the Flex application. Here’s how
you do that in JavaScript:
var obs = new Object;
obs.onPostLoad = function(notifier,
data)
{
Spry.Debug.trace(“obs.onPostLoad
called!”);
};
var notifierData;
obs.onDataChanged = function(notifier,
data)
{
notifierData = notifier.data;
try {
with( FABridge.example.root()) {
bindJSONToChart( jsmin( toJsonString(
notifier.data ) ) );
}
}
catch( e )
{
var initCallback = function( )
{
with( FABridge.example.root() ) {
bindJSONToChart( jsmin( toJsonString(
notifierData ) ) );
}
}
FABridge.addInitializationCallback(“ex
ample”,initCallback);
//alert( “Error in onDataChanged”+e );
}
Spry.Debug.trace(toJsonString( noti-
fier.data ));
};
dsFeedburner.addObserver(obs);
```

There’s lots going on here. First, we create an observer on the data’s “onDataChanged” method, so every time the data is changed, we send new data to the Flex application. Although we only use this once, when the page is loaded, you could use this if you had say a drop down list of a number of your feeds, and wanted the data source to change when the drop down changed. When you changed your Spry data source, the Flex graph would refresh because of this observer.

Then, we hack around a bit. First of all, we need to convert the data to JSON format:

```
toJsonString( notifier.data )
```

We also need to minimize it, because Flex’s JSON Library doesn’t like new lines or carriage returns in that data (this took me a long time to figure out!). So, we run:

```
jsmin( toJsonString( notifier.data ) )
Finally, we pas the result of that
function to the method in our MXML
file, bindJSONToChart():
bindJSONToChart( jsmin( toJsonString(
notifier.data ) ) );
I’m lazy and I’ve made that all one
line in the code.
```

Now, it will do that fine if the .swf file has loaded. However, sometimes the data gets returned before the .swf file is actually loaded. I found this out when running the application locally. So, I’ve wrapped

Software / Account	Why do we need it?	Cost?	Where do I get it?
Feedburner with API access turned on	*Data to populate an Ajax page and Flash graph.*	*$0*	*www.feedburner.com*
PHP	*Acts as a proxy for our Ajax application.*	*$0 / Open Source*	*www.php.net*
	Connects to Feedburner and prints out XML.	*Spry Framework for Ajax*	*Create the Ajax page*
$0 / Open Source	*labs.adobe.com*		
Flex and Flex charting	*Create the .swf file that will show the graph and call back to JavaScript*	*Flex SDK: Free*	
Flex Charting: Free trial.	*www.adobe.com/go/flex*		
Flex / Ajax bridge	*Call ActionScript (Flash) from JavaScript*	*$0 / Open Source*	*labs.adobe.com*
ActionScript 3 Library for JSON	*Parse the JSON that we're passing to Flex*	*$0 / Open Source*	*labs.adobe.com*
JSON and JSMin Library	*Remove new lines and carriage returns in JSON*	*$0 / Open Source*	

> "Its important to note here that Flex can be **used to create a number of components"**

all this in a try / catch statement, and added an initialization function on the swf file, so that if its not loaded, when it loads it will run this function and populate the graph with data from the Ajax call.

There you go. When you load the HTML page, Spry will load that PHP file using an Ajax call, the PHP file will connect to Feedburner, get the XML data, print it out. Spry will read that in, bind it to the HTML elements on the page, then call the Flex application, passing it the data in JSON format. The Flex application will read in the data and display it in a Flex chart. Here are a few tips and tricks that I've found when building this out:

1. I found it easiest to modify the HTML that gets output by Flex Builder, rather than to reference the built swf file. If you do that, be sure to modify index.template.html in the html-template folder of your Flex project, rather than the .html files in the bin/ directory of your Flex project. The HTML files in bin/ get overwritten when you save and re-build your Flex application, and if you modify those, rather than index.template.html, you'll lose your changes.
2. I started building out the Flex graph using JavaScript and the Flex / Ajax bridge. I don't recommend that. Build out your entire Flex component in Flex Builder, then simply write functions to pass data to it from HTML. I think that's easier than trying to build Flex components using JavaScript.
3. Similarly, write functions that closely couple your Flex application to your HTML page. For instance, you'll notice that in my MXML file, I call the chartClicked JS function using:if(ExternalInterface.available)

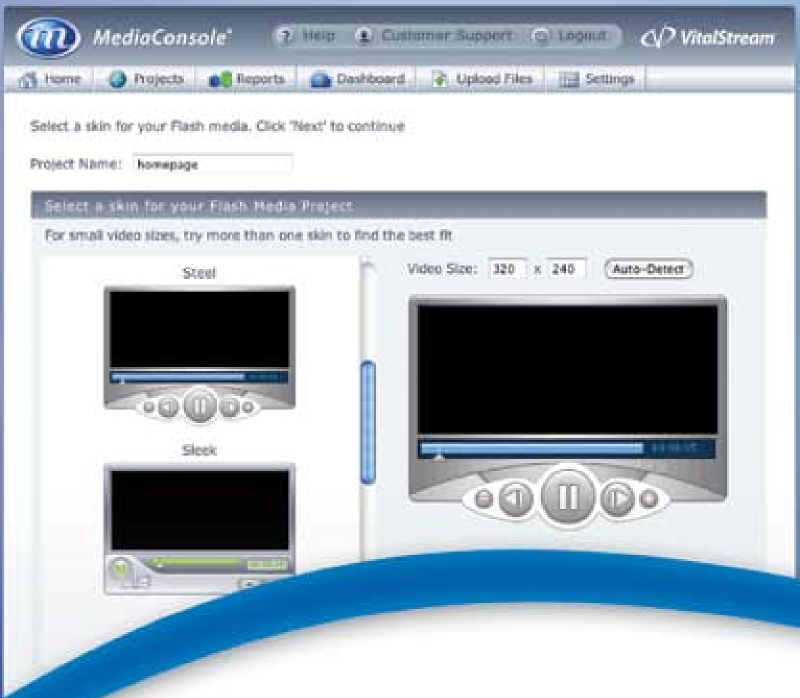
```
ExternalInterface.call("chartClicked", clickEvent.hit-
Data.item);
```

I could have attached an observer to the lineChart instead, using the Flex / Ajax bridge, but I find it easier to get the data items and debug the application in Flex Builder, rather than trying to do that in JavaScript on the

Innovative Tools

To Manage, Deliver and Track Streaming Media On the Internet

MediaConsole



Select a skin for your Flash media. Click 'Next' to continue

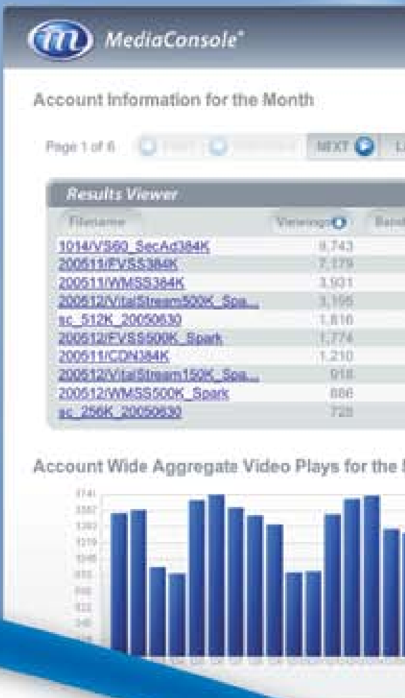
Project Name: homepage

Select a skin for your Flash Media Project

For small video sizes, try more than one skin to find the best fit

Video Size: 320 x 240

Real-time



Account Information for the Month

Page 1 of 6

Filename	Viewings
1014/VS60_SecAd384K	9,743
200511/FVSS384K	7,179
200511/WMSS384K	3,931
200512/VitalStream500K_Spa...	3,195
sc_512K_20050630	1,816
200512/FVSS500K_Spark	1,774
200511/CDN384K	1,210
200512/VitalStream150K_Spa...	918
200512/WMSS500K_Spark	886
sc_256K_20050630	728

Account Wide Aggregate Video Plays for the

Using the Right Tools For the Job Is Critical to The Success of Your Business

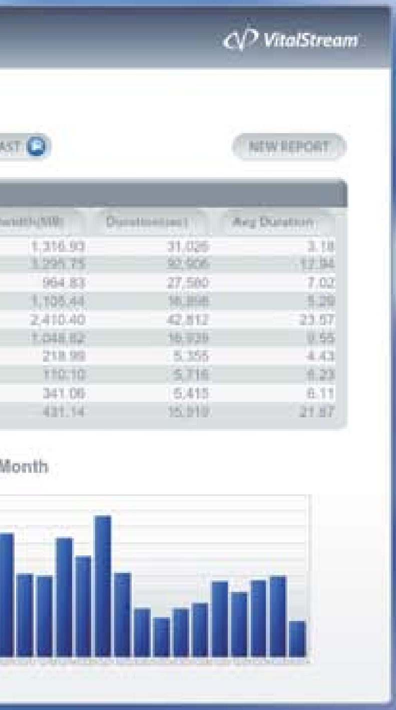
VitalStream reliably delivers streaming media to large global audiences using an award-winning content delivery network that powers all of our services. These services come with innovative tools that support and enable a variety of business models for online media distribution including advertising, subscriptions, pay-per-view and more.

Introducing the New VitalStream Media Player for Flash

Continuing its commitment to innovation, VitalStream is proud to announce the new Media Player for Flash to help speed up your workflow by allowing you to test and play streaming Flash video locally on your computer before you upload the files to our servers.

Download this player for free at http://www.vitalstream.com/tools/mxdj.asp

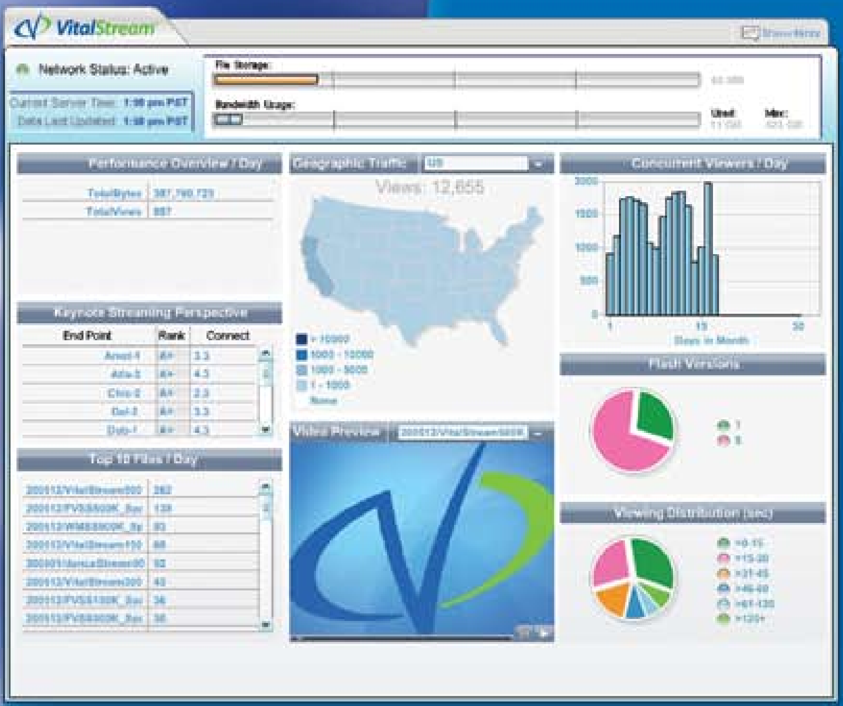
Reporting



width(MB)	Duration(sec)	Avg Duration
1,316.93	31,026	3.18
3,295.75	92,906	12.94
964.83	27,580	7.02
1,105.44	16,898	5.29
2,410.40	42,812	23.57
1,048.82	16,939	0.55
218.99	5,355	4.43
110.10	5,716	6.23
341.06	5,415	6.11
431.14	15,919	21.87

Month

Reporting Dashboard



TotalBytes	387,760,723
TotalViews	887

End Point	Rank	Connect
Anaol-1	A+	3.3
Atla-1	A+	4.3
Chic-2	A+	2.3
Dal-2	A+	3.3
Dub-1	A+	4.3

200512/VitalStream500	282
200512/FVSS500K_Bsc	139
200512/WMBS500K_Bp	93
200512/VitalStream150	68
200501/JamesStream00	52
200512/VitalStream300	43
200512/FVSS150K_Bsc	36
200512/FVSS500K_Bsc	36

To learn more about VitalStream, call (800) 254-7554 or visit www.vitalstream.com

VitalStream®

The World Leader In Audio and Video Streaming

Creating a Video Player

...using the state design pattern and ActionScript 3.0 - Part 1

by William B. Sanders

When you create a Flash Media Server 2 application, you typically place emphasis on optimizing the quality of the communications. That is certainly as it should be. Likewise, quality object-oriented programming (OOP) is another priority. One standard in OOP is design patterns—abstract concepts for solving recurring problems using designs that optimize OOP when you apply them appropriately. The seminal work in design patterns is Design Patterns: Elements of Reusable Object-Oriented Software by Erich Gamma, Richard Helm Ralph Johnson, and John Vlissides, affectionately known as "The Gang of Four" or simply GoF.

One design pattern in particular, the state design pattern (or SDP), focuses on the different states in an application, transitions between states, and the different behaviors within a state. A simple Flash video player application, for example, has two states: Stop and Play. In the Stop state, the video is not playing; in the Play state, a video is playing. Furthermore, the player transitions from the Stop state to the Play state using a method that changes the application's state. Likewise, it transitions from Play to Stop using a different transition and method to make it happen.

An interface holds the transitions, and each state implements the transitions as methods that are unique to that state. Each method is implemented differently depending on the context of its use. For example, the startPlay() method would do one thing in the Stop state and something entirely different the Play state, even though startPlay() is part of both states. To understand and appreciate the value of the SDP, it helps to understand something about state machines.

This article begins with a simple two-state application that plays and stops playing an FLV file. It requires only Flash Player 9 and ActionScript 3.0, which you can download from the links below. The initial application introduces the basics of a state machine and the state design pattern.

The application is then expanded into a more robust one using the same state structure and incorporating Flash Media Server 2. This illustrates both the expandibility of an application using a design pattern and the process of incorporating Flash Media Server 2 into that design using ActionScript 3.0.

State machines and statecharts

A state machine is the conceptual model of states you would be using in an application with the SDP; the actual application is the state engine. So if my video player application is designed around key states, that design represents the state machine. However, one need not worry about which is which because they're used differently and interchangeably in the literature on state machines. The important point to keep in mind is the idea of states and their differing contextual behavior.

Rather than beginning with the usual diagrams associated with design patterns, I'll start with a statechart. At its most basic level, a statechart is an illustration of an application's states and transitions. As such, it is a model for the state machine and engine. Taking a simple video player application, you can see the Play and Stop states. When the application is first run, the application enters the Stop state and can transition only to the Play state (see Figure 1).

The line going from the black dot to the Stop state shows the Application Not Running state. For all intents and purposes, however, assume that the starting point is the Stop state. This could be illustrated in a hierarchical state with Application Running and Application Not Running states, or you could even place the whole hierarchy into Computer On and Computer Off states, but that's not too useful because you aren't coding to those states.

Before I discuss getting from one state to another, consider what each state can actually do. In the Stop state, I can initiate only the Play state. That is, I cannot stop in the Stop state because I'm already stopped. By the same token, if I'm in the Play state, the only thing I can do is transition to the Stop state.

Transitions

The transitions in a state machine are the actions that change states. In the simple statechart, the line from Stop to Play would be a startPlay() method of some sort; and from Play to Stop, it would be a stopPlay() method. As more states are added, you might find that you cannot transition directly from one state to another. Rather, you have to go through a series of states to get where you want to go. As you will see further on, if you're in the Stop state, you cannot go directly to the Pause state. You have to go first to the Play state before going to the Pause state.

Triggers

Finally, to initiate a transition, you need some kind of trigger. A trigger is any event that initiates a transition from one state to another. Usually we think of some kind of user action as a trigger, such as a mouse movement or button click. However, in simulations certain states can be triggered by ongoing conditions, such as running out of simulated fuel, draining a simulated battery, or a collision with an object. Likewise, triggers are subject to contexts and should work only in the appropriate contexts to initiate a state. So while you might use a Play button to initiate the Play state from the Stop state, it should not trigger a Play state from the Play state.

Often triggers are placed along with the transitions on the statecharts. This helps identify the trigger events and the transitions they trigger. Figure 2 shows the statechart updated to include both the triggers and transitions they initiate.

If you're interested in more information about using state engines, statecharts, and the more general aspects of working with Flash and states, see Flash MX for Interactive Simulation by Jonathan Kaye and David Castillo (Thomson, 2003). Although it goes back a couple generations of Flash, the book is timeless in its concepts and shows some very smooth device simulations.

State design pattern

Fortunately, one of the original design patterns that GoF described is the State pattern. Closely resembling the Strategy pattern, the State pattern is used when an application's behavior depends on changing states at runtime or has complex conditional statements that branch depending on a current state (see Figure 3). When the internal states change, an object alters its behavior when designed using the State pattern.

Using the SDP, all of the behaviors (methods) for a single state are placed into single objects (concrete states), and all transition behaviors for the application (state machine) are placed into a single interface. Each state object implements the interface in a fashion appropriate for the state. Because of this structure, no conditional statements are required to branch differentially depending on the current state. Rather than writing complex conditional statements, the individual state objects define how the methods are to behave for that state.

For example, with a two-state machine (Play and Stop) the following pseudo code could direct the state behavior to start playing the video depending on the state machine's current state:

```
function doPlay():void{
   if(state == Play)
   {
      trace(“You’re already play-
ing.”);
   }
   else if (state == Stop)
   {
      trace(“Go to the Play state.”);
   }
}
```

Note: By the way, an important but small difference between ActionScript 2.0 and ActionScript 3.0 is that all void special types are in lowercase. In ActionScript 2.0 the first letter was in caps: Void. Watch out for that!

With a couple of states, that's not too difficult. As you add states, however, things get more complicated and you're swimming in a sea of conditional statements that all have to work in sync.

The alternative is to set up "contextual" behavior using a State pattern. For example, the following code has two different objects with different implementations of behaviors from an interface:

```
//Interface
interface State
{
   function startPlay():void;
   function stopPlay():void;
}
//Play State object
class PlayState implements State
{
   public function startPlay():void
   {
      trace(“You’re already playing”);
   }
   public function stopPlay():void
   {
      trace(“Go to the Stop state.”);
   }
}
//Stop State object
class StopState implements State
{
   public function startPlay():void
   {
      trace(“Go to the Play state.”);
   }
   public function stopPlay():void
   {
      trace(“You’re already stopped”);
   }
}
```

As you can see, the behaviors (methods) have different implementations in the different states. When you add more states, all you need to do is add their transitional behaviors to the interface and create a new concrete state (class) that implements them. Each new behavior needs to be added to the existing state classes.

Context manager in a state design pattern

To manage the states and their transitions, you need some kind of management object—something to keep track of everything in the state machine. In Figure 3 the Context box is the abstraction of the state engine. The context manages the different states that make up the state machine and contain the different states. Figure 4 shows a more concrete representation of what needs to be transformed.

Creating a context class

In looking at the example of creating a simple video player, we need a context that will serve to get and set the different states. So the next phase will be to look at a class (object) that does just that. Let's first take a look at the class, and then you'll see what's going on:

```
class VideoWorks
{
   var playState:State;
   var stopState:State;
   var state:State;
```

Figure 1

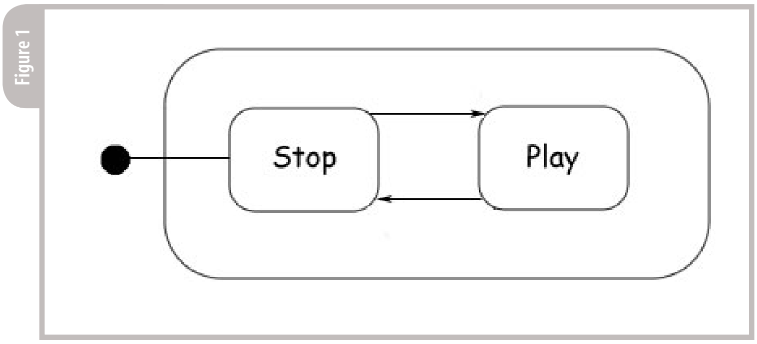
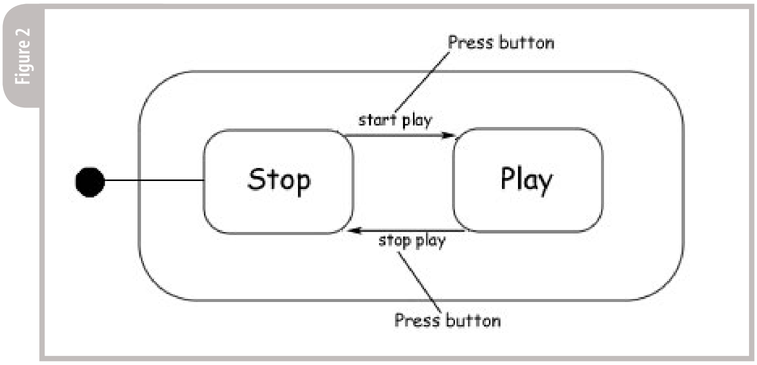


Figure 2



```
06      public function VideoWorks()
07      {
08         trace(“Video Player is
On”);
09         playState = new
PlayState(this);
10         stopState = new
StopState(this);
11         state = stopState;
12      }
13      public function startPlay():
void
14      {
15         state.startPlay();
16      }
17      public function stopPlay():void
18      {
19         state.stopPlay();
20      }
21      public function setState(state:
State):void
22      {
23         trace(“A new state is
set”);
24         this.state = state;
25      }
26      public function getState():
State
27      {
28         return state;
29      }
30      public function getPlayState():
State
31      {
32         return this.playState;
33      }
34      public function getStopState()
:State
35      {
36         return this.stopState;
37      }
38     }
```

Initially, in lines 3–11, the script instantiates three State objects—one of each of the two you designed (PlayState and StopState), and one (state) that acts as a variable to hold the current state. Because the state machine begins in the Stop state, the state variable is assigned the Stop state. (This works just like your car in the morning before you change it from the Off state to the On state.)

Next, the two behaviors from the State interface are specified in terms of the current state's context (lines 13–20). Although you're going to have to add some code to the two state classes for it to work with the context class, for now think of what will happen in the two different states when those behaviors are executed. For example, in the Play state the startPlay() method doesn't do anything but in the Stop state it switches to the Play state.

Finally, add the getter and setter methods (lines 21–38). You need a total of six methods—a set and get function for each of the three state instances. The setters return nothing and the getters return a State object.

Completing a state class

To get everything working, you need to revise the state classes to include the reference to the context: VideoWorks. Also, because you're working with ActionScript 3.0, all of the classes and interface need to be in a package container.

Note: All files are grouped into a single unreferenced folder for purposes of simplicity. However, typically you would be using multiple folders for organizing your files. Each package would import the appropriate related files. But by using a single folder, this application reduces a layer of comlexity.

Save the following code as State.as:

```
package
{
//State Machine Interface
  interface State
  {
    function startPlay():void;
    function stopPlay():void;
  }
}
Save the following code as PlayState.
as:

package
{
  //Play State
  class PlayState implements State
  {
    var videoWorks:VideoWorks;
    public function
PlayState(videoWorks:VideoWorks)
    {
      trace(“--Play State--”);
      this.videoWorks=videoWorks;
    }
    public function startPlay():void
    {
      trace(“You’re already playing”);
    }
    public function stopPlay():void
    {
```

Welcome to the Future of Video on the Web!

REGISTER NOW!
www.iTVcon.com

CALL FOR PAPERS NOW OPEN!

LIVE SIMULCAST!
AROUND THE WORLD ON SYS-CON.TV

iTVCON.COM
INTERNET TV CONFERENCE & EXPO 2006

Coming in 2006 to New York City!

> "Internet TV is wide open, it is global, and in true 'Web 2.0' spirit it is a direct-to-consumer opportunity!"

For More Information, Call 201-802-3023 or Email itvcon@sys-con.com

Welcome to the Future!

Did you already purchase your ".tv" domain name? You can't afford not to add Internet TV to your Website in 2006!

2005 was the year of streaming video and the birth of **Internet TV**, the long-awaited convergence of television and the Internet. Now that broadband is available to more than 100 million households worldwide, every corporate Website and every media company must now provide video content to remain competitive, not to mention live and interactive video Webinars and on-demand Webcasts.

20 years ago the advent of desktop publishing tools opened the doors for the creation of some of today's well-known traditional print media companies as well as revolutionized corporate print communications. Today, with maturing digital video production, the advent of fully featured PVRs, and significant advances in streaming video technologies, **Internet TV** is here to stay and grow and will be a critical part of every Website and every business in the years to come.

It will also very rapidly become a huge challenge to network and cable television stations: **Internet TV** is about to change forever the $300BN television industry, too.

The Internet killed most of print media (even though many publishers don't realize it yet), Google killed traditional advertising models, and **Internet TV** will revolutionize television the way we watch it today. You need to be part of this change!

Jeremy Geelan

Jeremy Geelan
Conference Chair, iTVCon.com
jeremy@sys-con.com

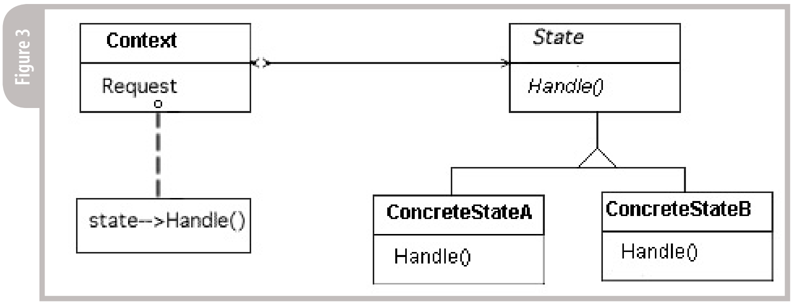
PRODUCED BY
SYS-CON EVENTS

List of Topics:

- > Advertising Models for Video-on-demand (VOD)
- > Internet TV Commercials
- > Mastering Adobe Flash Video
- > How to Harness Open Media Formats (DVB, etc)
- > Multicasting
- > Extending Internet TV to Windows CE-based Devices
- > Live Polling During Webcasts
- > Video Press Releases
- > Pay-Per-View
- > Screencasting
- > Video Search & Search Optimization
- > Syndication of Video Assets
- > V-Blogs & Videoblogging
- > Choosing Your PVR
- > Product Placement in Video Content
- > UK Perspective: BBC's "Dirac Project"
- > Case Study: SuperSun, Hong Kong

Track 1	Corporate marketing, advertising, product and brand managers
Track 2	Software programmers, developers, Website owners and operators
Track 3	Advertising agencies, advertisers and video content producers
Track 4	Print and online content providers, representatives from traditional media companies, print and online magazine and newspaper publishers, network and cable television business managers

Figure 3



```
trace("Stop playing.");
      videoWorks.setState(videoWorks.
getStopState());
    }
  }
```

Save the following code as StopState.as:

```
package
{
  //Stop State;
  class StopState implements State
  {
    var videoWorks:VideoWorks;
    public function
StopState(videoWorks:VideoWorks)
    {
      trace("--Stop State--");
      this.videoWorks=videoWorks;
    }
    public function startPlay():void
    {
      trace("Begin playing");
      videoWorks.setState(videoWorks.
getPlayState());
    }
    public function stopPlay():void
    {
      trace("You're already stopped");
    }
  }
}
```

Save the following code as VideoWorks.as:

```
package
{
  //Context class
  class VideoWorks
  {
    var playState:State;
    var stopState:State;
    var state:State;
    public function VideoWorks()
    {
      trace("Video Player is On");
playState = new PlayState(this);
      stopState = new StopState(this);
      state=stopState;
    }
    public function startPlay():void
    {
      state.startPlay();
    }
    public function stopPlay():void
    {
      state.stopPlay();
    }
    public function setState(state:
State):void
    {
      trace("A new state is set");
      this.state=state;
    }
    public function getState():State
    {
      return state;
    }
    public function getPlayState():
State
    {
      return this.playState;
    }
    public function getStopState():
State
    {
      return this.stopState;
    }
  }
}
```

To test the state engine completely, you need to test each state. By calling each state from a different state as well as from within itself, you can see the contextual nature of the state machine. The following steps show you how:

In order to use the Stage and execute a test of the program in ActionScript 3.0, you need to use the Display class. In general you can use either a sprite or a movie clip, but because you're not using a Timeline here, use a sprite. Open a new ActionScript file and enter the following code as TestState.as:

```
package
{
  //Test states
  import flash.display.Sprite;
  public class TestState extends
Sprite
  {
    public function TestState():void
    {
      var test:VideoWorks = new
VideoWorks();
      test.startPlay();
      test.startPlay();
      test.stopPlay();
      test.stopPlay();
    }
  }
}
```

Open a new Flash document and display the Properties panel. Click on the Stage and, in the Properties panel, type TestState in the Document Class text window.

Test the movie by pressing Control+Enter (Command+Return) just as you would for any application test. In the Output window, you should see the following:

```
Video Player is On
--Play State--
--Stop State--
Begin playing
A new state is set
You're already playing
Stop playing.
A new state is set
You're already stopped
```

Both the VideoWorks class and the PlayState and StopState classes include a trace() statement to indicate their instantiation, and appear as soon as you test the script. Because the initial state is Stop, when the script calls the first startPlay() method, you will see "Begin playing" in the output window, along with the message, "A new state is set."

However, note that when the same state is called a second time, the state responds with the message, "You're already playing," and no state transition occurs. However, as soon as the script calls the first stopPlay() method, it notes both that the playing has stopped and a new state is set. The second call of stopPlay(), though, responds with “You’re already stopped.”

The advantage of making each state self-aware becomes clear when you consider the alternative. For example, if I knock together an application that I want to use for playing FLV files, what happens if I click the Play button when it’s already playing? It starts playing the video all over again because it has no idea what state it’s in. However, if I click Play and have a state machine design, the application knows it’s in a Play state and will require the user to transition first to the Stop state before starting the Play over again. Of course, you can design the state machine to restart playing if that’s what you want to happen on all Play commands.

The point is that you can design the states to do exactly what you want—not what they will do automatically if left unconsidered.

From the conceptual to the functional

All you’ve seen so far has been the output of trace() statements to help understand how a state design pattern and state machine works. To add something useful, you need to include a reference to both a NetStream() object and a string for referencing a FLV file. However, you need only a string reference for playing the video because you can stop it simply by closing the NetStream() instance.

Below is an update to the VideoWorks.as file. Importantly, I’ve added the NetConnection and NetStream classes from the flash.net package because both classes have been included in the user package created. While the new script has set up the state machine to actually play and stop a video, all of the trace() statements have been left in place:

```
package
{
  import flash.net.NetStream;
  interface State
  {
    function startPlay(ns:
NetStream,flv:String):void;
    function stopPlay(ns:NetStream):
void;
  }
}
```

Save the following code as PlayState.as:

```
package
{
  import flash.net.NetStream;
  //Play State

  class PlayState implements State
  {
    var videoWorks:VideoWorks;
    public function
PlayState(videoWorks:VideoWorks)
    {
      trace(“--Play State--”);
      this.videoWorks=videoWorks;
    }
    public function startPlay(ns:
NetStream,flv:String):void
    {
      trace(“You’re already playing”);
    }
    public function stopPlay(ns:
NetStream):void
    {
      ns.close();
      trace(“Stop playing.”);
      videoWorks.setState(videoWorks.
getStopState());
    }
  }
}
```

Save the following code as StopState.as:

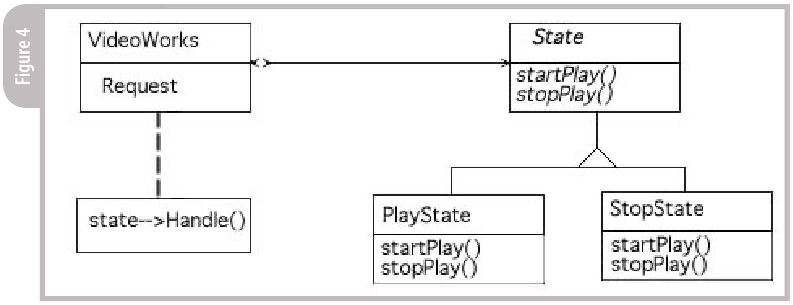
```
package {
  import flash.net.NetStream;

class StopState implements State
{
  var videoWorks:VideoWorks;
  public function
StopState(videoWorks:VideoWorks)
  {
    trace(“--Stop State--”);
    this.videoWorks = videoWorks;
  }
  public function startPlay(ns:
NetStream,flv:String):void
  {
    ns.play(flv);
    trace(“Begin playing”);
    videoWorks.setState(videoWorks.
getPlayState());
  }
  public function stopPlay(ns:
NetStream):void
  {
    trace(“You’re already stopped”);
  }
  }
}
```

Save the following code as VideoWorks.as:

```
package
{
  import flash.net.NetStream;
  class VideoWorks
  {
    var playState:State;
    var stopState:State;
    var state:State;
    public function VideoWorks()
    {
      trace(“Video Player is on”);
      playState = new PlayState(this);
      stopState = new StopState(this);
      state=stopState;
    }
    public function startPlay(ns,flv):
void
    {
      state.startPlay(ns,flv);
    }
    public function stopPlay(ns):void
    {
      state.stopPlay(ns);
    }
    public function setState(state:
State):void
    {
      trace(“A new state is set”);
      this.state=state;
    }
    public function getState():State
    {
      return state;
    }
    public function getPlayState():
State
    {
      return this.playState;
    }
    public function getStopState():
State
    {
      return this.stopState;
    }
  }
```

Figure 4



```
}
```

The FLA that tests the state engine is a hybrid one. At this writing, no ActionScript 3.0 or Flash 9 UI components are available, so it uses the new ActionScript 3.0 SimpleButton class to create buttons for starting and stopping the video play. Like most classes, this one is set up for reuse whenever an application requires a button labeled with text. Thus, before writing the test code in the Flash authoring environment, create the following class in an ActionScript file and save it as NetBtn.as in the same folder as the VideoWorks.as file:

```
package
{
  import flash.display.Sprite;
  import flash.display.SimpleButton;
  import flash.display.Shape;
  import flash.text.TextFormat;
  import flash.text.TextField;
  import flash.text.TextFieldAutoSize;
  public class NetBtn extends
SimpleButton
  {
    public function NetBtn(txt:String)
    {
      upState = new BtnState(0xffff00,
txt);
      downState = new
BtnState(0x00aaff, txt);
      overState= new BtnState
(0x00cc77,txt);
      hitTestState=upState;
    }
  }
  class BtnState extends Sprite
  {
    public var btnLabel:TextField;
    public function BtnState(color:
uint,btnLabelText:String)
    {
      btnLabel = new TextField();
btnLabel.text=btnLabelText;
      btnLabel.x=5;
      btnLabel.autoSize=TextFieldAuto
Size.LEFT;
      var format:TextFormat = new
TextFormat(“Verdana”);
      format.size=12;
      btnLabel.setTextFormat(format);
      var btnWidth:Number=btnLabel.
textWidth + 10;
      var bkground:Shape = new
Shape();
      bkground.graphics.
beginFill(color);
      bkground.graphics.
lineStyle(2,0x00cc77);
      bkground.graphics.drawRoundRect(
0,0,btnWidth,18,8);
      addChild(bkground);
      addChild(btnLabel);
    }
  }
}
```

Keep NetBtn.as handy; it will be used to provide buttons in all of the developing examples.

Staging and coding the application

The final step is to create a script for bringing everything to the Stage. The first step is to build a testing class as you did with the abstract state machine. Open up a new ActionScript file and add the following code (saving it as TestVid.as):

```
package
{
  import flash.display.Sprite;
  import flash.net.NetConnection;
  import flash.net.NetStream;
  import flash.media.Video;
  import flash.text.TextField;
  import flash.text.TextFieldType;
  import flash.events.MouseEvent;
  public class TestVid extends Sprite
  {
    public function TestVid()
    {
      var nc:NetConnection=new
NetConnection();
      nc.connect(null);
      var ns:NetStream=new
NetStream(nc);
      var vid:Video=new
Video(320,240);
      this.addChild(vid);
      vid.x=(550/2)-(320/2);
      vid.y=50;
      var vidTest:VideoWorks=new
VideoWorks();
      //Play and Stop Buttons
      var playBtn:NetBtn=new
NetBtn(“Play”);
      this.addChild(playBtn);
      playBtn.x=(550/2)-50;
      playBtn.y=300;
      var stopBtn:NetBtn=new
NetBtn(“Stop”);
      this.addChild(stopBtn);
      stopBtn.x=(550/2)+50;
      stopBtn.y=300;
      //Add Event Listeners
      playBtn.addEventListener(MouseEv
ent.CLICK,doPlay);
      stopBtn.addEventListener(MouseEv
ent.CLICK,doStop);
      //Add the text field
      var flv_txt:TextField= new
TextField();
      flv_txt.border=true;
      flv_txt.background=true;
      flv_txt.backgroundColor=0xccc
ccc;
      flv_txt.type=TextFieldType.
INPUT;
      flv_txt.x=(550/2)-45;
      flv_txt.y=10;
      flv_txt.width=90;
      flv_txt.height=14;
      this.addChild(flv_txt);
      //Start and stop play
      var flv:String;
      function doPlay(e:MouseEvent):
void
      {
         if (flv_txt.text != “” &&
flv_txt.text != “Provide file name”)
         {
           flv_txt.textColor=0x000000;
           flv=flv_txt.text + “.flv”;
           vidTest.startPlay(ns,flv);
```

```
          vid.attachNetStream(ns);
        } else
        {
          flv_txt.textColor=0xcc0000;
          flv_txt.text=”Provide file
name”;
        }
      }
      function doStop(e:MouseEvent):void
      {
        vidTest.stopPlay(ns);
        vid.clear();
      }

//The following code gets rid of an error
statement in the //Public Alpha
//In the final version, this won’t be
necessary
//For good measure you can see how to
find the length of
//your FLV file with the metadata.dura-
tion property

      function getMeta(mdata:Object):void
      {
        trace(mdata.duration);
      }
      var dummy:Object=new Object();
      ns.client=dummy;
      dummy.onMetaData=getMeta;
    }
  }
}
```

As you can see, the code includes importing several different packages. In using ActionScript 3.0, you will find that it is crucial to import the right packages for your applications. By avoiding the use of the wildcard character (*), the application imports only exactly what you need and no more, thereby keeping the overhead down.

Open a new Flash document and save it in the same folder as all of the other files for this application. In the Document Class window of the Property inspector, type TestVid and save the file.

To test actual the application, you will need an FLV file. You can convert an existing video file (AVI or MOV format) or use any FLV file on hand. Place the file in the same folder as the application. Figure 5 shows what your initial state machine looks like when completed.

The UI is simple and relates to the transitions: Stop and Start (playing video). Furthermore, you can see the relationship between the video playing and the trace() statement showing what happens when you click the button. For example, if you click Start and the video is already playing, nothing new occurs because the startPlay() function in the Play state does nothing other than offering a trace() statement to the effect that you’re already playing. As an added bonus, you get to see the length of your FLV file from the metadata function.

Adding states and hierarchical state machines

A fundamental feature of virtually all design patterns is their ability to expand and accept change. The kind of change you’re expecting in an application to some extent determines the type of design pattern you select. In this particular application, you are adding states.

The first state to add to the state machine is a Pause state. This state exists only in the Play state; you cannot get there directly from the Stop state. Once in the Play state, the user can turn the Pause state on and off. A hierarchical state diagram depicts this new state accurately (see Figure 6).

The hierarchy is a simple one. The first level is the Play and Stop states. Within the Play state is the Pause and No Pause states.

Because the Pause function is a toggle between the Play and Pause states, the No Pause state is exactly the same as the Play state. So rather than creating “pause start” and “pause stop” functions, you can establish a Do Pause behavior that acts differently in either state. In the Pause state, the Do Pause behavior returns to the default Play state; in the Play state, it goes to the Pause state.

ActionScript 3.0 has two different options for creating a Pause state. First, you can create controls around NetStream.pause() and NetStream.resume() that stand as two different NetStream methods. In previous versions of ActionScript, only the pause() method was available and it worked as a toggle. Second, you could use the new method NetStream.togglePause(). This new method is actually just the old method with a new name. Sticking with the statechart depicted in Figure 6, this application uses the togglePause() method.

Creating a Flash Media Server 2 application

Now that the structure can support a simple FLV playback system, the next step is to add two more states and see if the state machine can be adapted to a Flash Media Server 2 application. To keep the focus on the design pattern, only two new states will be added to the Play, Stop, and Pause states: Record and Append.

Making the change from a Flash application to a Flash Media Server (FMS) application requires key changes in the FLA script to include a connection to the server, as well as adding Camera and Microphone objects. Other than that, adding the Record and Append states is relatively simple.

The first task when working with state machine models is to update the model. Figure 7 shows the addition of Append and Record. The original three states are pretty much the same as before. Note that the Stop state is the central one for all transitions except for the Play-Pause toggle. To change from any state except Pause, the transition must go first to the Stop state.

As noted at the outset, statecharts make it easy to see required program changes. By adding the new states—Pause, Append, and Record—all of the other states and context need to be changed as well. However, you don’t need to change a huge number of conditional statements. The testing application in the Actions panel needs changes as well, but because that code is more like a user of the state machine rather than an actual part of the state machine, it will be handled separately.

William B. Sanders is currently a professor of Sociology and Interactive Information Technology at the University of Hartford in West Hartford, Connecticut, of which he was a founding faculty member. He has published over 40 computer books, including eight on Flash, ActionScript, or Flash Communication Server. Bill founded Sandlight Productions in 1984 as a computer book and software publication company, which has evolved over the past 22 years into an Internet development company specializing in Flash, ActionScript, and Flash Communication Server applications, along with e-business strategies. He has worked with computers ever since his alma mater, the University of California at Santa Barbara, was one of only four nodes on ARPANET, along with UCLA, Stanford University, and the University of Utah. When he’s not writing, he updates his websites at sandlight.com and iit.hartford.edu. Bill recently published an instructional video on using Flash Media Server 2 for authorized training provider Train Simple.

Mobile Learning

...with Flash Lite 2 and Captivate

By Marco Casario

in just a few years long-distance learning, or Electronic Learning (e-learning), has become a reality. The problem with distance and having to be in a certain place at a certain time has been eliminated with the use of the personal computer and the Web.

esc
F1
F2
F3
F4
F5
~
`
!
1
@
2
#
3
$
4
%
5
6
Q
W
E
R
T
Y
A
S
D
F
Z
X
C
tab
caps lock

Figure 1

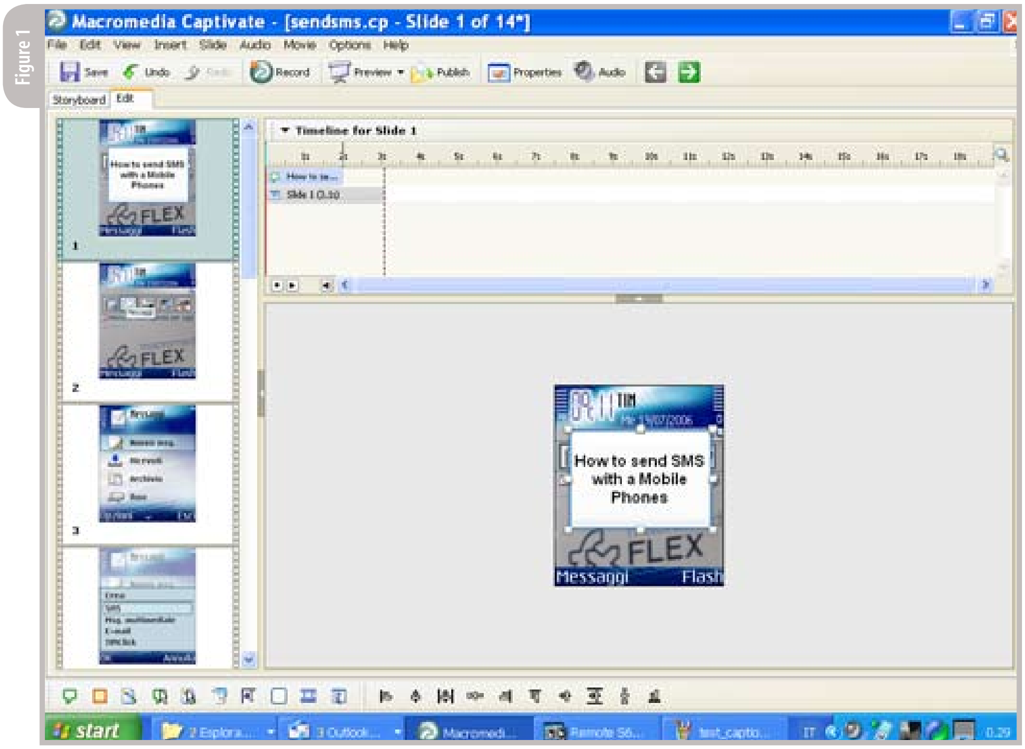
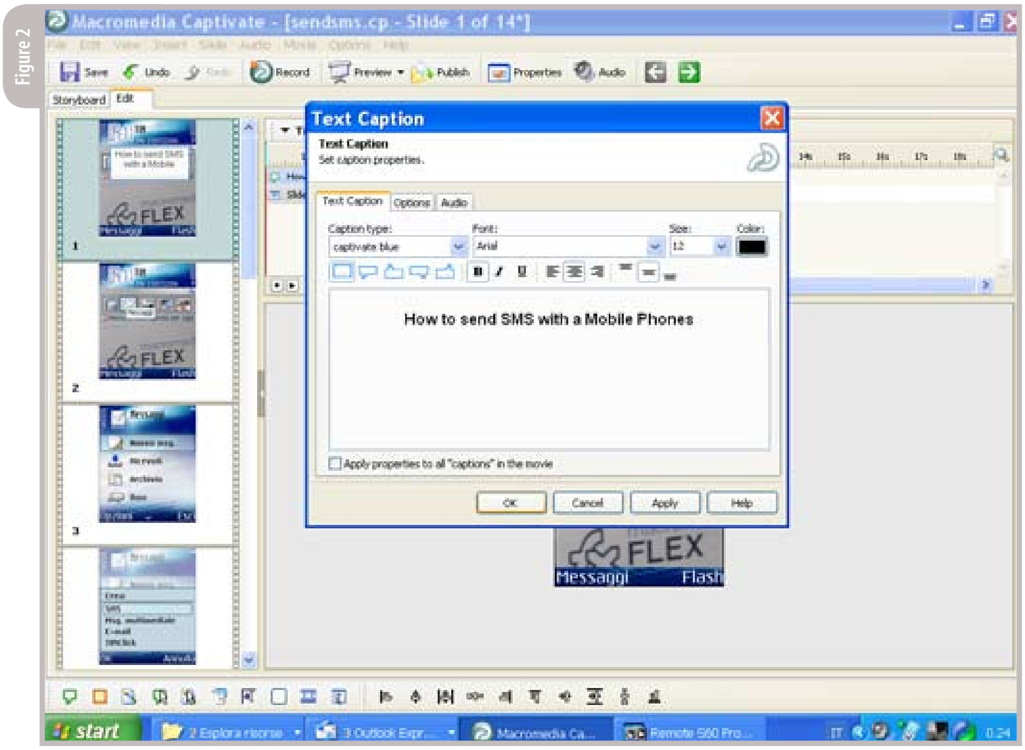

Macromedia Captivate - [sendsms.cp - Slide 1 of 14*]
File Edit View Insert Slide Audio Movie Options Help
Save
Undo
Redo
Record
Preview
Publish
Properties
Audio
Storyboard
Edit
Timeline for Slide 1
How to se...
Slide 1 (3.10)
How to send SMS with a Mobile Phones
FLEX
Messaggi
Flash
start

Figure 2



Text Caption

Set caption properties.

Caption type: captivate blue

Font: Arial

Size: 12

How to send SMS with a Mobile Phones

☐ Apply properties to all "captions" in the movie

Nowadays, hundreds of millions of people are able to receive information as they walk along the street – with a mobile phone.

The use of the mobile phone as a device to deliver content for distance learning, Mobile Learning (m-learning), has passed the experimental stage and received excellent feedback from tutors and students.

The mobile's CPU power is growing monthly. With increasing processor speed and larger memories in which to store data, the mobile phone is truly a portable PC, leaving little to be desired from its big brother, insomuch as it can be used to carry out complex programs, load multimedia elements such as audio and video, speed up a browser, and surf the Web.

In the technological process of the mobile phone, Adobe Flash Lite, the player that enables us to enjoy the content created by Flash, has taken an enormous step forward. We all know the characteristics that have made Flash famous on the Web and we have also seen and used the excellent didactic content created with this technology, which lends itself so successfully as an instrument for learning, thanks to its interactivity and multi-media.

Now the second version of Flash Lite offers an efficacious technology for creating content for the mobile device, providing support for ActionScript 2, the possibility of launching videos in 3GP format, loading and parsing XML files, saving data on the mobile, and loading external images and sounds.

Flash Lite's popularity is growing daily and a team of developers has created a variety of Flash content for mobile devices: games, utility, RSS reader, screensaver, animated wallpapers, etc. To see what can be done with Flash, the starting point is certainly the Flash Lite Exchange (http://www.adobe.com/cfusion/exchange/index.cfm?view=sn310), which picks up many Flash Lite applications.

The greater limits of the diffusion of Flash applications are shown by its penetration, which increases exponentially thanks to the new mobile devices that are coming out (especially, but not limited to, Nokia S60 and Sony Ericsson) and have already embedded the player.

Adobe has other software that enables long-distance teaching and the creation of didactic material, for example, Captivate, Director, and Authorware.

Adobe Captivate, in particular, is the best for creating learning objects in the form of interactive tutorials with audio comments, the simulation of software obtained with the desktop recording, and quizzes and interactive content that allow the tracing of data (SCORM and AICC compliant). This program is for creating, in a speedy and efficacious manner, software simulations using the desktop recording features and the recording and synchronization of the audio, and support of the content, but this is not all. It is, in fact, possible to make this content interactive and to ask the user to participate actively in certain operations, for example, by repeating a determined operation or sequence of clicks that has been explained to the user and to receive feedback and a grade. Activities like this and quizzes with questions and answers can be traced in any LMS (Learning Management System), AICC, or SCORM compliant. (For more information on this topic, read "An Overview of Tracking Options in Captivate" on the Adobe Developer Center [http://www.adobe.com/devnet/captivate/articles/lms_output.html].)

Used separately, Captivate and Flash Lite produce excellent work. In this article we shall try to see what the possible integrations between these two instruments are and if it is possible to effectively create Flash Lite's learning objects for the mobile phone.

Learning Object with Captivate for Flash Lite

In creating a learning object for a device such as the mobile phone using Captivate, we must bear in mind the target on which we are going to visualize the movie. Apart from setting some options in the exportation phase, which allows the compatibility with Flash Lite 2, we must first consider the reduced dimensions of the screen and the different measures that vary from despotic to device. There is an article on my blog that examines more thoroughly: "Nokia S60 platform UI screen resolutions for Flash Lite" (http://casario.blogs.com/mmworld/2006/05/nokia_s60_platf.html) and for this example a screen resolution of 170x208 pixels will be used. This is the classic S60 UI resolution supported from the first S60 devices, the Nokia7650

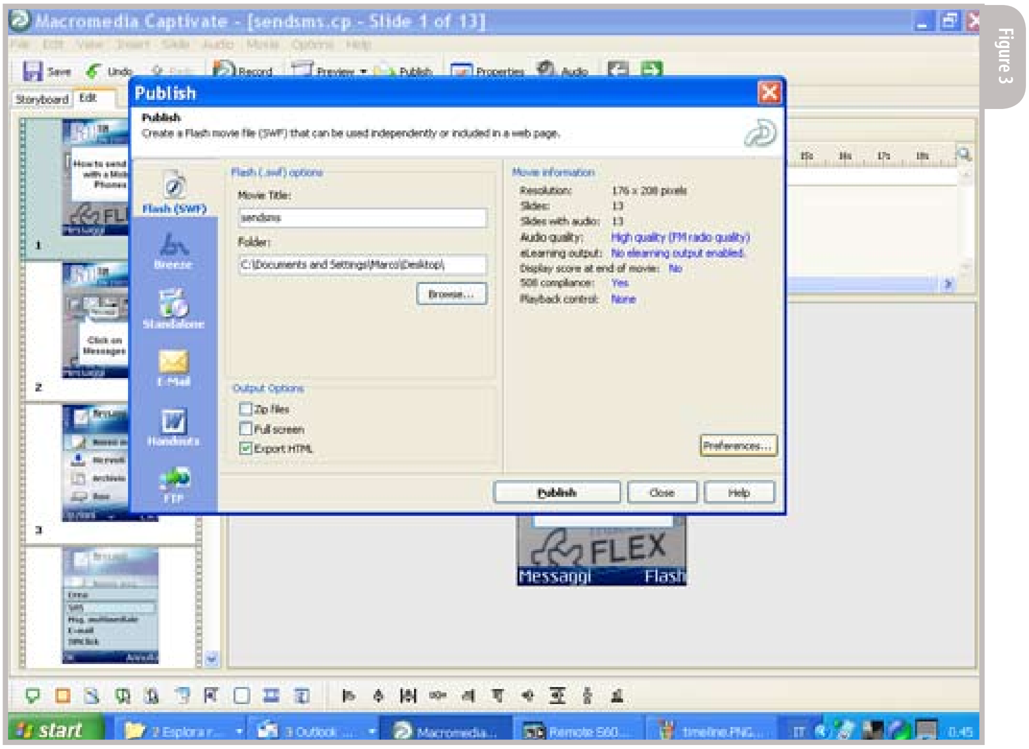


Figure 3

Macromedia Captivate - [sendsms.cp - Slide 1 of 13]

Publish

Publish
Create a Flash movie file (SWF) that can be used independently or included in a web page.

Flash (.swf) options
Movie Title: sendsms
Folder: C:\Documents and Settings\Marco\Desktop\

Output Options
□ Zip files
□ Full screen
☑ Export HTML

Movie information
Resolution: 176 x 208 pixels
Slides: 13
Slides with audio: 13
Audio quality: High quality (FM radio quality)
eLearning output: No elearning output enabled.
Display score at end of movie: No
508 compliance: Yes
Playback control: None

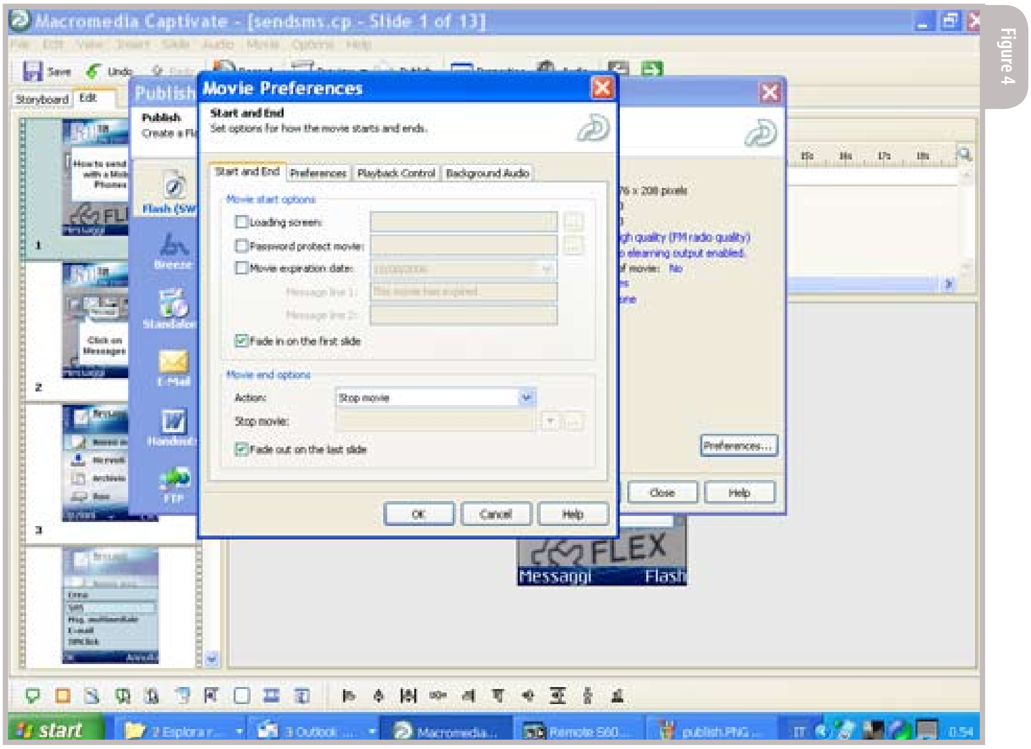


Figure 4

Macromedia Captivate - [sendsms.cp - Slide 1 of 13]

Movie Preferences

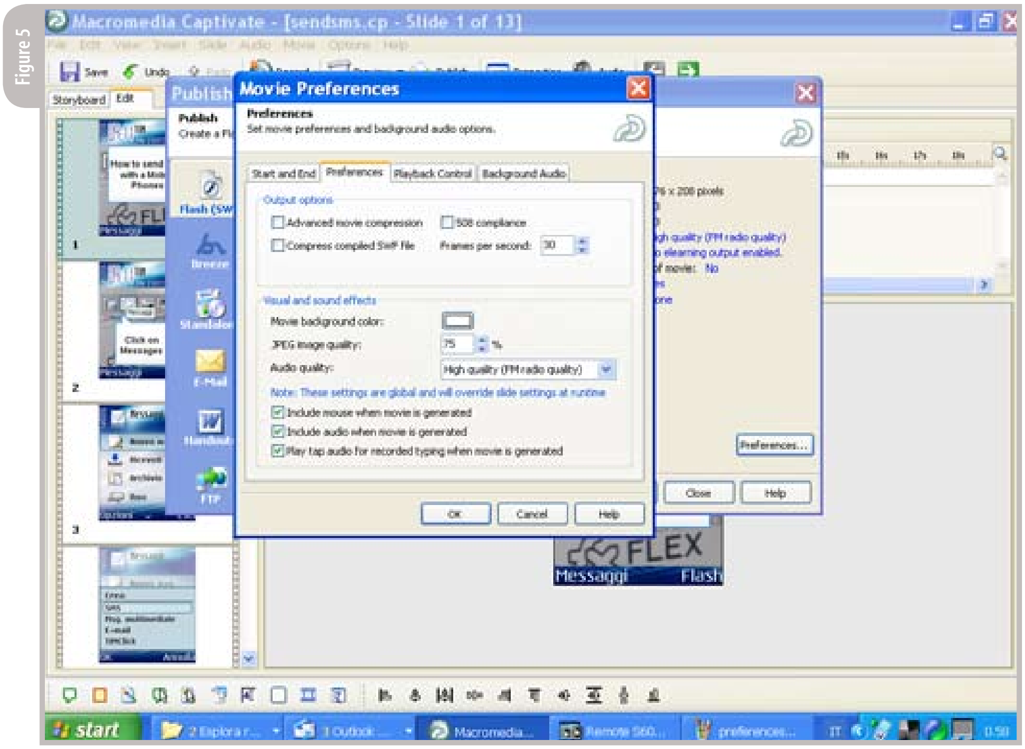
Start and End
Set options for how the movie starts and ends.

Start and End | Preferences | Playback Control | Background Audio

Movie start options
□ Loading screen:
□ Password protect movie:
□ Movie expiration date:
Message line 1: This movie has expired.
Message line 2:
☑ Fade in on the first slide

Movie end options
Action: Stop movie
Stop movie:
☑ Fade out on the last slide

Figure 5



Movie Preferences

Preferences
Set movie preferences and background audio options.

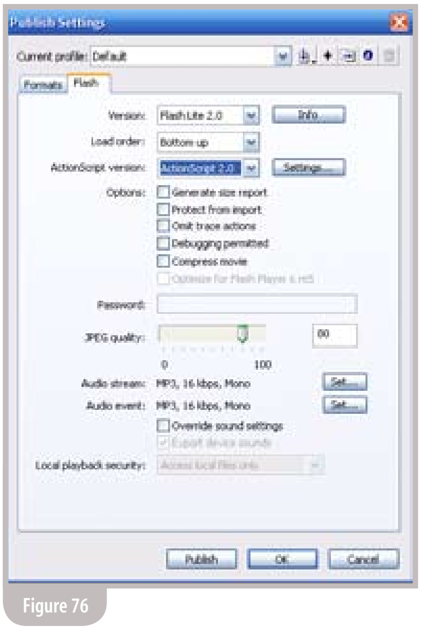
Start and End | Preferences | Playback Control | Background Audio

Output options
□ Advanced movie compression
□ 508 compliance
□ Compress compiled SWF file
Frames per second: 30

Visual and sound effects
Movie background color:
JPEG image quality: 75 %
Audio quality: High quality (FM radio quality)
Note: These settings are global and will override slide settings at runtime
☑ Include mouse when movie is generated
☑ Include audio when movie is generated
☑ Play tap audio for recorded typing when movie is generated

OK | Cancel | Help

“Flash Lite’s popularity is growing daily and a team of developers **has created a variety of Flash content for mobile devices”**



Publish Settings

Current profile: Default

Formats | Flash

Version: Flash Lite 2.0 | Info...
Load order: Bottom up
ActionScript version: ActionScript 2.0 | Settings...
Options:
□ Generate size report
□ Protect from import
□ Omit trace actions
□ Debugging permitted
□ Compress movie
□ Optimize for Flash Player 6 r65
Password:
JPEG quality: 80
0 100
Audio stream: MP3, 16 kbps, Mono | Set...
Audio event: MP3, 16 kbps, Mono | Set...
□ Override sound settings
☑ Export device sounds
Local playback security: Access local files only

Publish | OK | Cancel

Figure 76

phone through to the latest Nokia3250 music phone. To have a complete list of S60 phones Nokia has created a site dedicated to these models: S60.com

Creating a tutorial with Captivate is easy and amusing. On opening the program you’ll see the Set Up, which shows the basic operations you can carry out.

Under the option Record new movie, select Record or create new movie. Now the recording stage begins and the window to select the kind of movie you want to create opens.

Since we have to create content for a particular device, we have to choose the precise size and position for the recording window. The successive display will allow us to insert the dimensions of the recording rectangle, 176x208 pixels, and to position it on the screen.

In the example provided with this article, you will find a recorded tutorial demonstrating how to send an SMS message using a common mobile device. (To download the example code, go to the online version of this article at http://webddj.sys-con.com.) For this demonstration, the Remote S60 mobile application, which clones the screen of the mobile phone on the PC desktop, will be used. Remote S60 helps create screenshots and even allows you to record AVI movies from the screen of your mobile phone.

Before starting with the registration, deselect the Record Narration check box. By clicking Record, the recording will start automatically and can be stopped by pressing the DEL key on the PC. When the recording is finished, Captivate will process the recording and open the slides captured inside its IDE.

The final result of the recorded example is in the file sendsms.cp and is made up of 14 slides. Captivate allows us to edit and modify the captured images that it has automatically carried out. To edit we must go to the Edit tab on the right of the Storyboard tab. This will display all the slides on the left, and on the right the details of the selected slides and a Timeline (see Figure 1). You can work on this to impose the viewing times of the various topics.

The software allows us to insert Text Captions among the various images of Captivate to support the captured images.

From Captivate’s Insert menu select

Text Caption to create a text box. It is possible to choose from different types of graphics and to set dimensions and a type font (see Figure 2). A Text Caption of the Captivate Blue type with black Arial font, dimension 12px bold, will be inserted in the example.

When the editing phase is finished, the movie can be exported. After several tries, I managed to understand the options that have to be disabled in order for the movie to be used by Flash Lite 2. Pay attention to the following operations.

To export a movie, go to the file menu and select Publish. A window will open, allowing you to choose the type of exportation. For the example, the movie will be exported in Flash (swf). Let's name the movie and decide on which folder to save it in. In the lower right of the window, Publish is the Preferences key that allows access to the advanced options of exportation to which we can make the changes necessary to render the movie compatible with Flash Lite 2 (see Figure 3).

On the first tab of the Preferences window, Start and End, disable the option Loading Screen that automatically runs a preloader that is awaiting the loading of the entire movie (see Figure 4).

The next tab, Preferences, is the one that contains the options most compatible with Flash Lite. Uncheck Advanced movie compression, 508 compliance, Compress compiled SWF file, Include Breeze Metadata (this option is present only if you have Breeze Presenter installed on the machine) (see Figure 5).

The game is over. Click on OK and confirm all the options. The final result is sendsms_final.swf.

With this setting it's impossible to have any type of e-learning output or to track any data generated by the users. This information is contained in metadata inside the file SWF that Flash Lite 2 did not like.

Load Captivate Movies into Flash Lite

The sendsms_final.swf file that you just created is now ready to be copied onto the mobile device and sent.

Make sure your mobile is compatible with Flash Lite 2 and that the player has already been installed. It is possible to see the compatibility of the devices on Adobe's Web site, under the Mobile section (www.adobe.com/mobile) and download Flash Lite 2 from Adobe's store for $10 U.S. (https://store1.adobe.com/cfusion/store/index.cfm?store=OLS-US&view=ols_prod&category=/Applications/FlashLite2).

Use the Bluetooth connection or the lead USB with such software as Nokia PC Suite to transfer the SWF file onto the mobile. Once it has been copied, select it to send to the Flash Player and see it loaded.

The only inconvenience is that the movie will not load in Full Screen; you will be forced to choose the option of the Soft Key on the left.

Let's create a short stub movie that consists of a simple Flash Lite 2 shell with a button that sends the movie.

We shall open the Flash 8 Professional (make sure that the Flash Lite 2 Update for Flash Professional 8 is installed) and from the Start Page create a new project for the mobile by selecting the line Global Phones from Create from Template and choosing Flash Lite1.1-Symbian Series 60 Template.

A new project will be created and the dimensions of the stage are set at 176x208 pixels, those of the screen of a modelS60 mobile device.

We'll now create a button-type movie clip on the first frame of the Timeline and name it load_btn. Then we insert the following ActionScript code on the first frame at the level ActionScript on the Timeline:

```
load_btn.onPress=function()
}
            loadMovie("sendsns_final.
swf",1):
}
```

The final code on the first frame will be the following:

```
fscommanda(FullScreen",true);
_focusrect=false;
load_btn.onPress=function(){
           loadMovie("sendsms_final.
swf",1):
}
stop();
```

Save the file and name it main.fla.

The only thing left to do is to export the movie, launching Publish Settings under the File menu, make sure that the version from the exported file is set to Flash Lite 2 as in Figure 6.

Before copying the two files (main.swf and sendswf_final.swf) into the mobile, test the movie inside the Flash 8 IDE. From Commands menu item select Test Movie (or Command/Contrast-Return shortcut). The Flash 8 Mobile Emulator will be launched to simulate the movie as it will appear on the mobile device.

The last step is to copy the files onto your device and run them (see Figure 10).

Flash 8 Mobile Emulator Limits

If you wish to use Flash 8 to develop Flash Lite content, when you hit Command/Control-Return to test the movie, load the mobile emulator, which you can use to run and debug mobile applications in context, on your mobile handset of choice and see how the content behaves. One of the features I love is the ability to rotate the content on the emulated mobile screen 90 CW or CCW (from View menu). It's a very helpful tool because now you can test movies without physically moving them to your phone.

Don't forget that it's just an emulator and it can't:

- Simulate FSCommand2 control
- Simulate some handset attributes
- Simulate performance
- Simulate the mobile memory consumption

The Future of Mobile Learning

Mobile learning is a new frontier in long-distance learning. As in all new fields there are still a lot of things to discover and try out.

The idea of being able to optimize our time and of always having a device at hand that enables us to study at any moment during the day is indeed very interesting.

The topic becomes even more interesting the moment in which it is possible to use mobile technology, such as Flash Lite, integrated with software, such as Captivate or Flex, that allows us to create rich and interactive content and to involve the learner.

As Benjamin Franklin said: Tell me and I forget. Teach me and I remember. Involve me and I learn.

Marco Casario is CEO of Comtaste, a company devoted to develop Rich Internet Applications on the Web and for mobile devices. He collaborates intensively with Adobe Italy as a speaker at conferences and as a consultant for Flash, Flex, and Flash Lite. Learn more about Marco Casario at his blog http://casario.blogs.com. m.casario@comtaste.com

Rich Internet Applications: AJAX,

SANTA CLARA SILICON VALLEY

SYS-CON Events is proud to announce the first-ever AjaxWorld Conference & Expo 2006!

The world-beating Conference program will provide developers and IT managers alike with comprehensive information and insight into the biggest paradigm shift in website design, development, and deployment since the invention of the World Wide Web itself a decade ago.

The terms on everyone's lips this year include "AJAX," "Web 2.0" and "Rich Internet Applications." All of these themes play an integral role at AjaxWorld. So, anyone involved with business-critical web applications that recognize the importance of the user experience needs to attend this uniquely timely conference – especially the web designers and developers building those experiences, and those who manage them.

CALL FOR PAPERS NOW OPEN!

We are interested in receiving original speaking proposals for this event from i-Technology professionals. Speakers will be chosen from the co-existing worlds of both commercial software and open source. Delegates will be interested learn about a wide range of RIA topics that can help them achieve business value.

Flash, Web 2.0 and Beyond...

REGISTER TODAY AND SAVE!

October 3-4, 2006

Santa Clara Convention Center
Hyatt Regency Silicon Valley
Santa Clara, CA

To Register
Call 201-802-3020 or
Visit www.AjaxWorldExpo.com

May 7-8, 2007

First International AjaxWorld Europe
Amsterdam, Netherlands

"Over the two information-packed days, delegates will receive four days' worth of education!"

Early Bird*
(Register Before August 31, 2006) **$1,495****
See website or call for group discounts

Special Discounts*
(Register a Second Person) **$1,395****
See website or call for group discounts

(5 Delegates from same Company) **$1,295/ea.****
See website or call for group discounts

On-Demand Online Access
(Any Event) **$695**

*Golden Pass access includes Breakfast, Lunch and Coffee Breaks, Conference T-Shirt, Collectible Lap-Top Bag and Complete On-Demand Archives of sessions in 7 DVDs!

**OFFER SUBJECT TO CHANGE WITHOUT NOTICE, PLEASE SEE WEBSITE FOR UP-TO-DATE PRICING

"It Was The Best AJAX Education Opportunity Anywhere in the World!" —John Hamilton

Topics Include...

Themes:
> Improving Web-based Customer
> Interaction
> AJAX for the Enterprise
> RIA Best Practices
> Web 2.0 – Why Does It Matter?
> Emerging Standards
> Open Source RIA Libraries
> Leveraging Streaming Video

Technologies:
> AJAX
> The Flash Platform
> The Flex 2 Framework & Flex Builder 2
> Microsoft's approaches: ASP.NET, Atlas, XAML with Avalon
> JackBe, openLaszlo
> JavaServer Faces and AJAX
> Nexaweb
> TIBCO General Interface

Verticals:
> Education
> Transport
> Retail
> Entertainment
> Financial Sector
> Homeland Security

GROUP DISCOUNTS AVAILABLE:
— 5 Delegates from Same Company — for only $995 (each)
— Register a Second Person — for only $1195

Hurry! Limited Seating
This Conference Will Sell-Out!

LIVE SIMULCAST!
AROUND THE WORLD ON SYS-CON.TV

Receive FREE WebCast Archives of Entire Conference!

The best news for this year's conference delegates is that your "Golden Pass" registration now gives you full access to all conference sessions. We will mail you the complete content from all the conference sessions in seven convenient DVDs after the live event takes place.

► **This on-demand archives set is sold separately for $995**

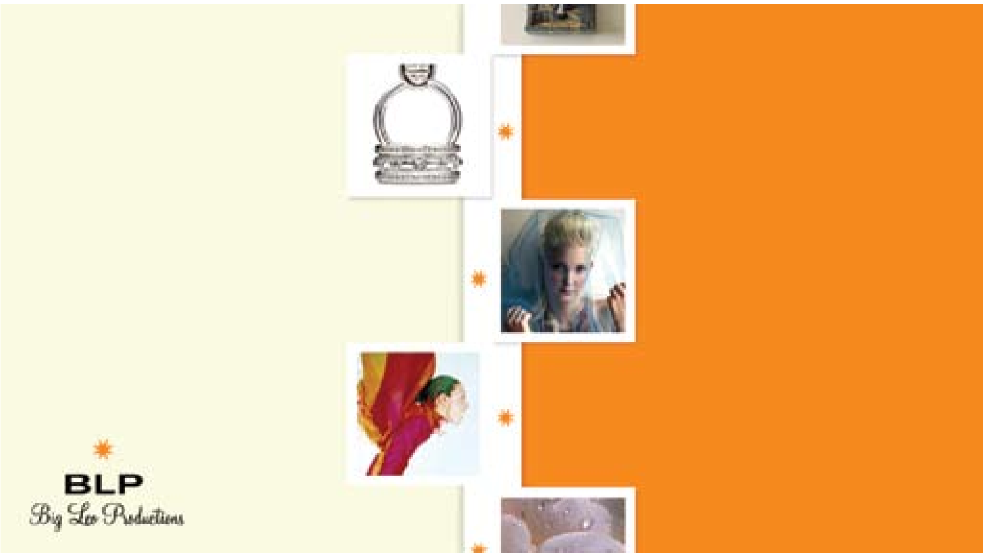
HYATT
HYATT REGECNY SILICON VALLEY

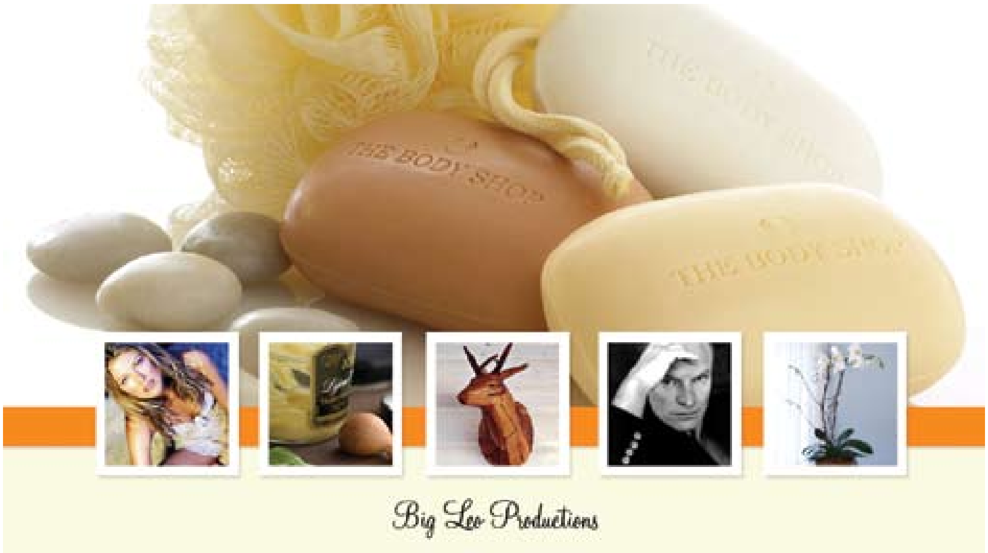
SYS-CON EVENTS
For more great events visit www.EVENTS.SYS-CON.com
VISIT WWW.AJAXWORLDEXPO.COM FOR THE MOST COMPLETE UP-TO-DATE INFORMATION

Edge Lab

Edge Lab (www.edgelabinc.com) is a multi-disciplinary design studio that specializes in branding, and visual identities for interactive and print properties. Founded by Creative Director Lora Appleton in 2001, Edge Lab focus is on guiding clients to create deeper connections with their desired audiences and unify their communications through a variety of media. Top-echelon lifestyle, cultural and media clients like the Smithsonian Institution, Big Leo Productions, Hollywood Records, Discovery Network, and The National Hypertension Association have all turned to Edge Lab for identities and branding concepts that fuse culturally resonant imagery, design, interactivity and communication. "My team of designers works to discover what makes audiences connect with visual mediums,"Appleton says. "We like to unearth key branding messages and create distinctive visual voices— whether we are designing and developing high profile print campaigns or more intimate projects with smaller organizations."

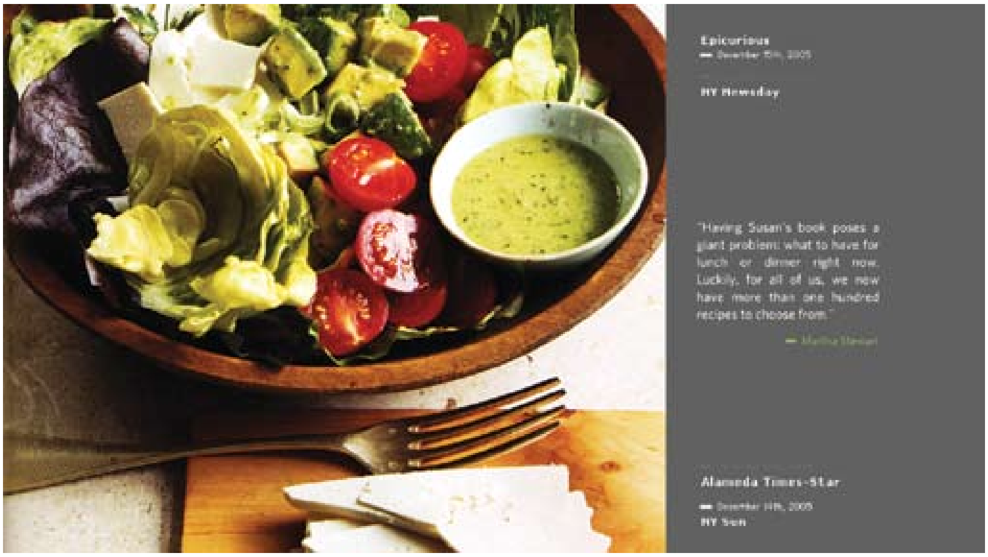
Powerful ideas synthesized with strategic web development is a hallmark in all of Edge Lab's work. Discovering new technologies and marrying them with engaging visual elements, and developing easy-to- use content management systems for clients to have hands-on access to growing their online vehicles, Edge pushes the limits of design and educates clients to be bolder with their online properties. When developing NYC based photo agency Big Leo Production's (www.bigleoproductions.com) website,





Edge Lab designed a flash platform interactive, semi-dynamic navigation system and embedded a full deep link component to feature the extensive collection of editorial and advertising photo galleries. This enables ad agencies and art buyers to easily navigate and view the impressive skill of their talent roster. The deeplink system enables Big Leo to link potential customers direct to individual images, allowing their audience to go straight to their desired image.

Edge Lab launched the National Hypertension Association website, (www.nathypertension.org) to engage their member base as well as connect with scientists around the world and promote effective communication among medical non-profits. "That's really what we're all about," reflects Appleton, "building and creating interesting brands and connecting them with their audience." A philosophy that has brought Edge no small amount of attention from the creative world. In addition to having two websites: (www.jjappleton.com & www.lloydlandesman.com) featured in the latest TASCHEN book, ICONS: Web Design Music Sites, Edge was recently awarded an American Design Award for Chef and Author, Susan Spungen's website, (www.themoderncook.com) a companion website to her Harper Collin's cookbook, "Recipes: A Collection for the Modern Cook". From interactive websites to large scale print campaigns, all brands are developed with the core Edge Lab philosophy: the fusion of dynamic ides, cutting edge technology, and compelling graphics.



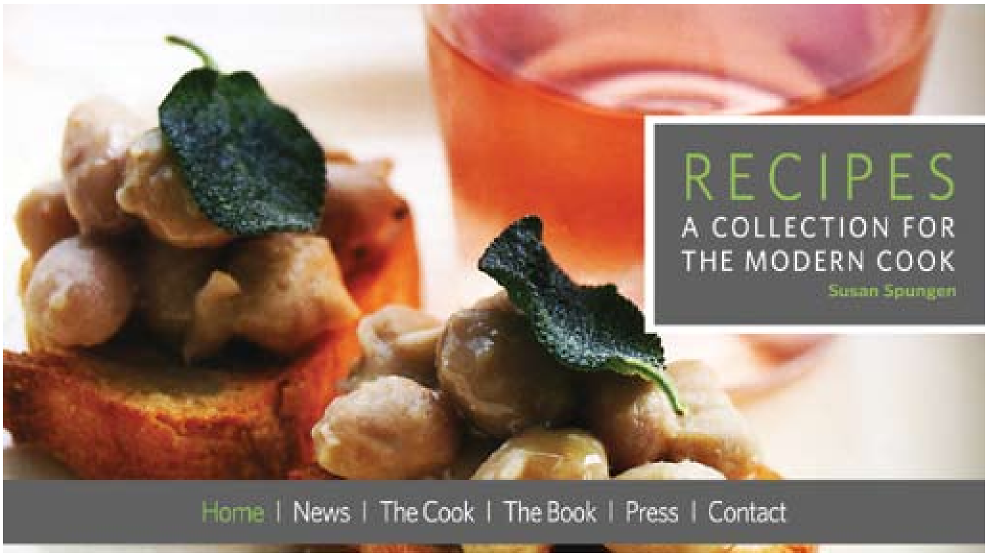
Epicurious
— December 15th, 2005

NY Newsday

"Having Susan's book poses a giant problem: what to have for lunch or dinner right now. Luckily, for all of us, we now have more than one hundred recipes to choose from."

— Martha Stewart

Alameda Times-Star
— December 14th, 2005
NY Sun



Freedom
Interactive Design

We approached the opportunity to create a piece for Adobe's Motion Design Center with great enthusiasm. We have advanced skills in several disciplines. We decided to develop a motion graphics piece that would give us a chance to combine illustration, animation and full-motion video.

Our team brainstormed for 20-30 minutes before our Senior Art Director, Matt Sundstrom, struck gold. He had the idea to get out on the streets behind a green-screen sandwich board, film our antics, and then come back and do intense post-production work in our shop.

Within days we tested our approach and knew we were onto something original, exciting, and challenging. We plotted our locations on a New York City subway map and shot our footage over the course of a couple weeks.

We used After Effects to motion-track the corners of our sandwich board, in many cases, working frame by frame. We also designed immersive motion graphics "reels" of several of our projects to super-imposed onto the tracked areas.

Once the videos began taking shape, we started to focus on the interface. We wanted a look and feel that would be similar to our own website. We took a screenshot of a map of Manhattan and used it as a background template to create our collage of photography, illustration and animation.

Within 2 months, our piece began to take shape. Once it launched, we received a huge amount of traffic and attention from around the world. People from Japan and Brazil emailed us to find out how much those "portable monitors" cost. Others wanted to know how long the batteries last, and how much the boards themselves cost! www.wouldyoulikeawebsite.com

RICHOMME INC

Antique Furniture and Object Restoration

Ian Nicolson & Mary McGowan

Graphicmachine

graphicmachine began as a response to the idea of making use of technology to create beautiful things. Everything that we do tries to find the balance between the function, design and commercial viability of the piece at hand. (Of course, a decent budget never hurts either.) However, it's more than the vacuum of a project that informs our response to the work of our clients. New York has always been such a strong character in the design that we produce. The often cluttered sidewalks, subways and lampposts provide a unique graphic language of odd juxtapositions and interesting relationships, which bring both confusion and new understanding. It's these reminders that help us understand how important it is to design with clarity so that a project doesn't get lost in the noise of its surroundings.

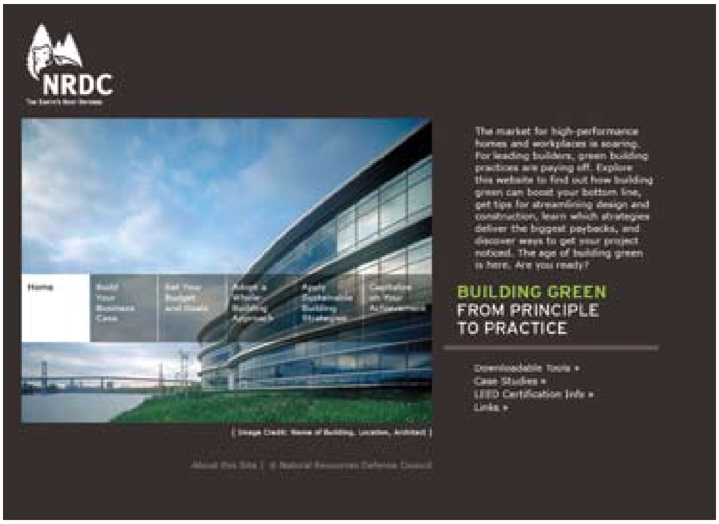
Each project that we are showing here is our effort to create that clearness for our clients. These pieces are the result of our mutual efforts to convey their work, services or products in the most effective manner.

Sullivan NYC

We are a strategic marketing and design firm that works with leading companies to help them communicate more effectively at the pivotal point of decision.

We combine expertise in marketing strategy, graphic design, copywriting, Web development, and functional communications to develop communications that align and influence people as they interact with products, services, businesses, and brands. The principals who lead the firm have 20 years of client-side experience in Financial Services, Media and Entertainment, Healthcare, and Not- for-Profit.

Our success springs from our anthropological approach that gets to the heart of what will make customers say yes, a relentless focus on our customer's customer, and the ability to create tangible materials and not get mired in endless strategy. This combination creates deep, lasting relationships with clients, strategic vision for their organizations, and compelling materials for their audiences.


NRDC
The Earth's Best Defense
Home
Build Your Business Case
Set Your Budget and Goals
Adopt a Whole Building Approach
Apply Sustainable Building Strategies
Capitalize on Your Achievements

The market for high-performance homes and workplaces is soaring. For leading builders, green building practices are paying off. Explore this website to find out how building green can boost your bottom line, get tips for streamlining design and construction, learn which strategies deliver the biggest paybacks, and discover ways to get your project noticed. The age of building green is here. Are you ready?

BUILDING GREEN
FROM PRINCIPLE
TO PRACTICE

Downloadable Tools »
Case Studies »
LEED Certification Info »
Links »

[Image Credit: Name of Building, Location, Architect]

About this Site | © Natural Resources Defense Council

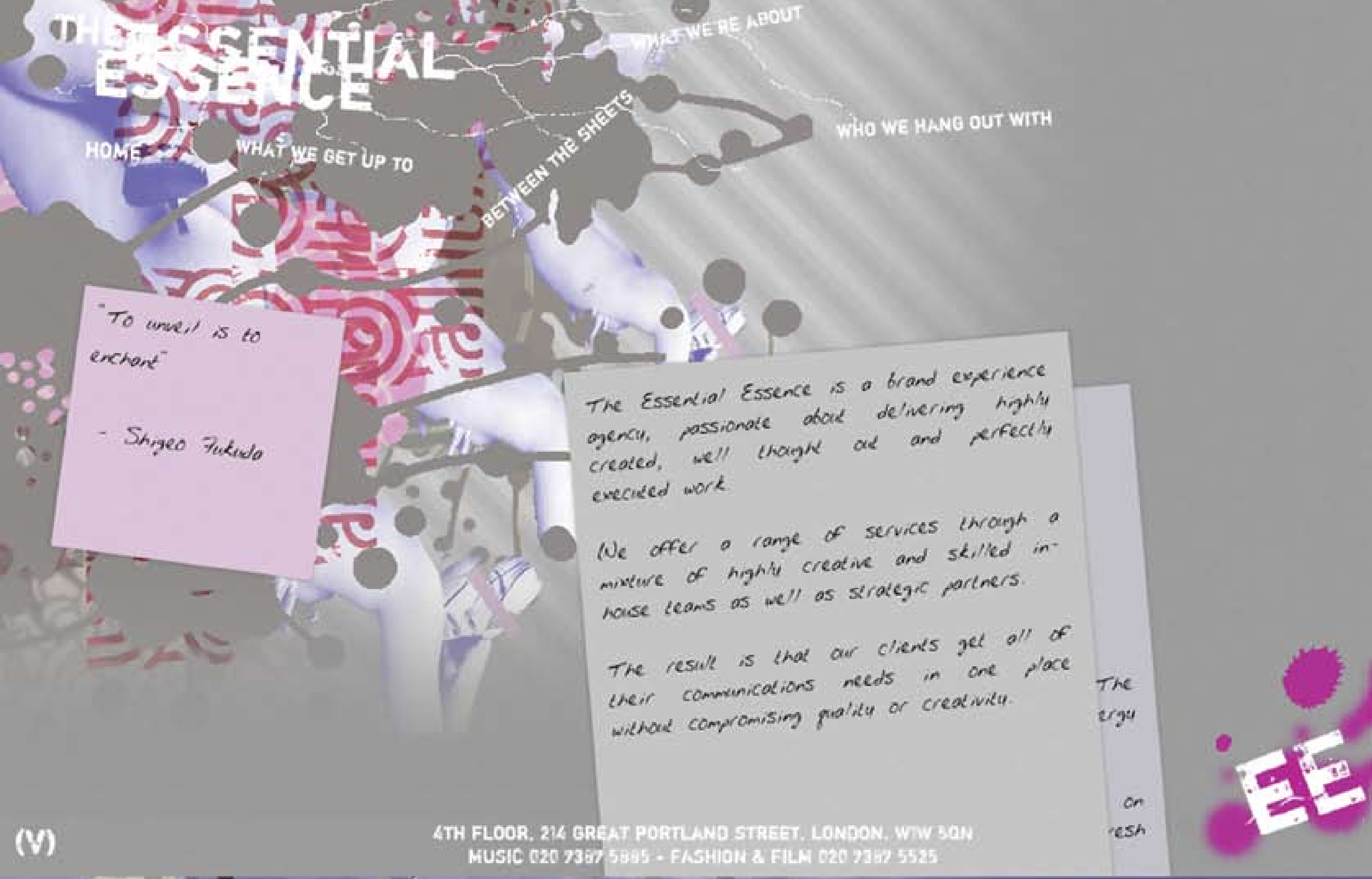
C
BUILDING GREEN FROM PRINCIPLE TO PRA
Build Your Business Case
Set Your Budget and Goals
Adopt a Whole-Building Approach
Apply Sustainable Building Strategies

Version Industries

We just opened a second office in Manhattan, New York after starting out in London a few years ago. Back then rich content – and Flash in particular – was really starting to come into its own and it became our express intention to use it to push the envelope of what the internet could do. Given this aim of doing large-scale creative work, the move to New York was inevitable.

The Flash website that Hi-ReS! did for the film Requiem For A Dream in 2000 became the first major focus of our interest in website design. Unlike so many of its contemporaries at the time, it presented abstract ideas rather than just text and images – successfully blending the cinematic elements of film with web design. For us this changed the game and opened the door to comparatively low-budget creative collaborations with just about anyone in the world, no matter what their profession.

From this point on we've been about doing something new with every project and in some way giving our clients more a piece of work backed by a strong creative idea than just a bare-bones functional design. We look for work which brings together a variety of different media – film, music, visual art – and we believe the reason why Version Industries works is that we all bring such different ingredients to the table: Giles is a coder with a desire to write some day, Gavin is a prolific electronic musician and Caspar, a wannabe filmmaker, is the son of two fine artists. We've yet to realise our individual personal goals, but you can be sure that when they arrive, they'll be designed beautifully and have a great website to back them up. www.versionindustries.com



"To unveil is to enchant"

- Shigeo Fukuda

The Essential Essence is a brand experience agency, passionate about delivering highly created, well thought out and perfectly executed work.

We offer a range of services through a mixture of highly creative and skilled in-house teams as well as strategic partners.

The result is that our clients get all of their communications needs in one place without compromising quality or creativity.

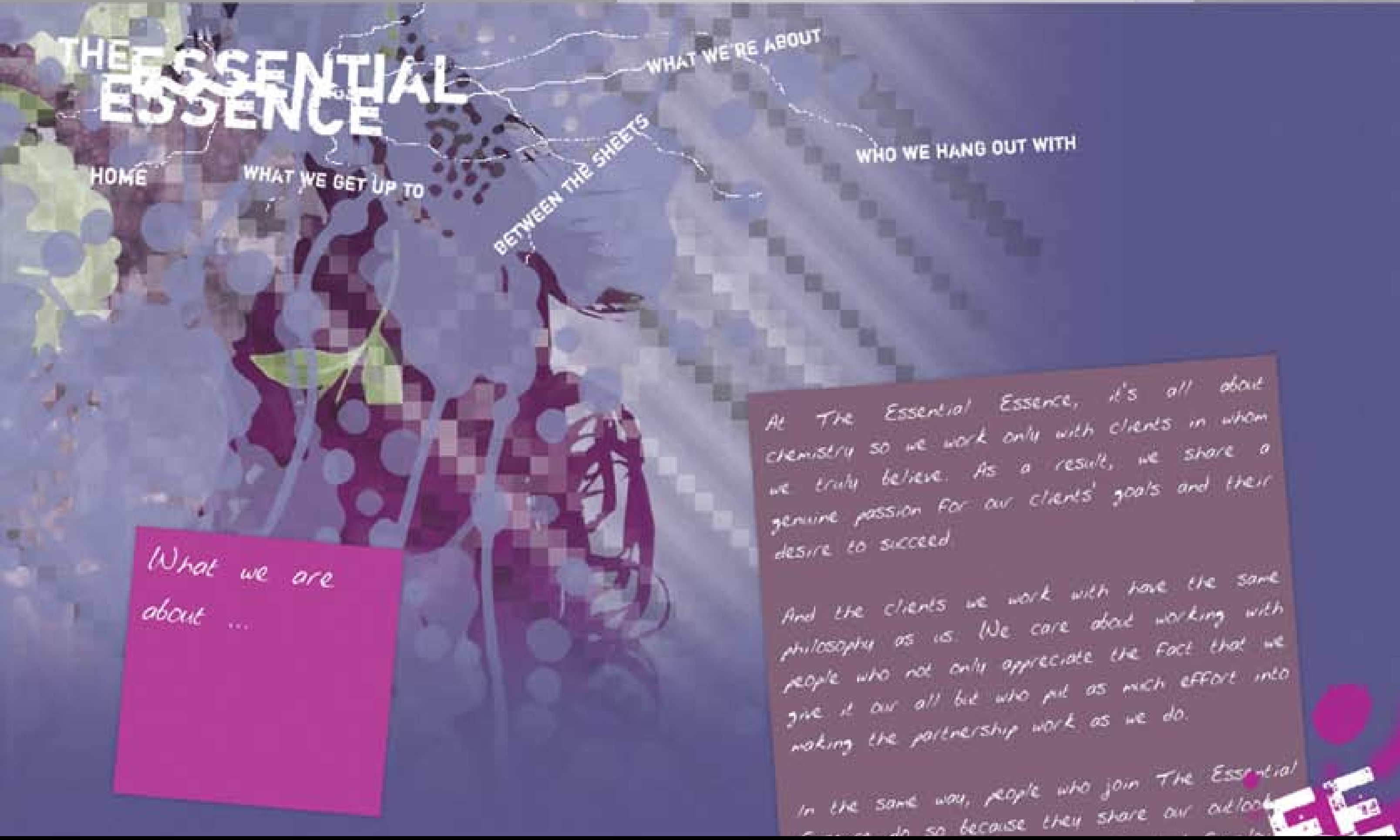
The

ergy

on

resh

4TH FLOOR, 214 GREAT PORTLAND STREET, LONDON, W1W 5QN
MUSIC 020 7387 5885 - FASHION & FILM 020 7387 5525



What we are about ...

At The Essential Essence, it's all about chemistry so we work only with clients in whom we truly believe. As a result, we share a genuine passion for our clients' goals and their desire to succeed.

And the clients we work with have the same philosophy as us. We care about working with people who not only appreciate the fact that we give it our all but who put as much effort into making the partnership work as we do.

In the same way, people who join The Essential Essence do so because they share our outlook

Top 10 Reasons to Attend MAX 2006

...in Las Vegas

by Ben Forta

Ben Forta is the Adobe senior product evangelist and the author of numerous books, including ColdFusion Web Application Construction Kit and its sequel Advanced ColdFusion Application Development, as well as books on SQL, JavaServer Pages, WAP, Windows development, and more. Ben co-authored the official ColdFusion training material, the certification tests and Macromedia Press study guides for those tests, and now spends a considerable amount of time lecturing, speaking, and writing about application development worldwide. Visit Ben's blog to read his regular postings on ColdFusion and more.

it's been more than half a year now since Macromedia and Adobe combined, and during this time we've been hard at work leveraging the strengths of each of these companies to forge the new Adobe. One of Macromedia's greatest strengths was its user community and the resultant relationship between company and customer. Each and every year, since 1999, this relationship culminated in the event we affectionately call MAX.

For those of you not yet fortunate enough to attend MAX (or DevCon, as it was known earlier on), try and picture several thousand of your peers spending three intense days learning, interacting, networking, talking shop, and just having fun. MAX, the official company conference, is your chance to sharpen your development skills, learn about new products, spend time with product engineering teams, and more.

From the opening keynote to the numerous hands-on sessions, community lounge and Cyber Café, sneak peeks, annual "special event," exhibit hall, informal birds-of-a-feather sessions, sponsor lunches, and much more—MAX attendees find themselves immersed in products and technologies, and emerge the better for the experience several days later.

Recognizing a good thing when they see it, Adobe is continuing the MAX tradition, maintaining not just the MAX name but also the MAX experience. MAX 2006 promises to be the biggest MAX ever. More tracks, more sessions (over 30 more than last year), more products and technologies, more attendees...more reasons for you to attend MAX this October 23–26 at the magnificent Venetian in Las Vegas, Nevada.

Just in case you remain unconvinced, here are my own top 10 reasons to attend MAX 2006 in Las Vegas:

The main reason to attend MAX is simply this: Be it web design, application development, creative work, building intelligent documents, or what have you, you rely on Adobe products to do what you do. This is your chance to learn new skills, discover new opportunities, and prepare for the future. It's also your chance to participate in the Fourth Annual MAX Awards.

Networking, schmoozing, chatting, late-night bar sessions. Whatever your style, you'll find hundreds of like-minded developers and designers who'd love to share and hear from you. For many people, this ability to associate faces with e-mail addresses is the highlight of MAX.

Spend time with the individuals responsible for the products you use and rely on. Like a feature? Tell them. Have a suggestion? Great, they love feedback. Even complaints (should you actually have any) are accepted and appreciated.

Flex 2 is hot! Avail yourself of the most Flex 2 information and expertise ever concentrated in a single place.

Over 100 unique sessions on a variety of products (Adobe Flex, Acrobat, Photoshop, ColdFusion, Flash, Dreamweaver, LiveCycle, and many more). Whether your interests are more code-centric or whether you are into design and visuals, you'll find sessions targeted specifically to you.

Don't just listen. Do! Take advantage of 90-minute hands-on training sessions on a variety of products and technologies, all taught by certified instructors and product experts. You'll learn by doing, writing code, manipulating design, and more.

"Sneak peeks" mean that MAX attendees are typically the first to see the stuff that our product teams are cooking up. Our engineers love showing off features planned for new versions of your favorite software, or futuristic skunkworks projects that may or may not ever see the light of day, or just about whatever else tickles their fancy. This is live and unscripted—our version of reality entertainment—and it's just for MAX attendees.

The "special event." No, I won't spill the beans but consider this: Previous special events included private, exclusive access to Universal Studios in Orlando, taking over the Olympic Oval in Salt Lake City, Mardi Gras World in New Orleans, and Disney's California Adventure in Anaheim. 'Nuff said!

Shirts, bags, and other apparel and goodies. While I can't promise that you'll be more confident and desirable, you will definitely look the part. The well-dressed MAX attendee is a sight to behold. (Those of us who've been attending year after year have saved fortunes on our clothing budgets.)

MAX is more than a conference, it's an experience. MAX 2006 is shaping up to be the biggest and best yet. If you want to register or just learn more about the event, visit adobe.com/events/max/.

Advertising Index

Advertiser	URL	Phone	Page
Adobe	adobe.com/go/designer		2,3
InterAKT	www.interaktonline.com/ktml4/		6
CFDynamics	www.cfdynamics.com	866-233-9626	5
EdgeWebHosting	http://edgewebhosting.net	1866-334-3932	11
CommunityMX	www.communitymx.com		13
Intermedia.net	www.intermedia.net	888-379-7729	17
Hosting.com	www.hosting.com		21
Vitalstream	www.vitalstream.com	800-254-7554	22,23
AJAXWorld	www.AjaxWorldExpo.com	201-802-3020	38, 39
Stream57	www.stream57.com	201-909-2550 x 1012	52
HostMySite	www.hostmysite.com		53