MX
developer's journal

volume 3 issue 5 2005 www.mxdj.com

THE LEADING MAGAZINE FOR
MACROMEDIA MX DEVELOPMENT,
DESIGN, & PRODUCTION TOOLS

captivateprimer

Tips, Tricks & Techniques

DREAMWEAVER
Using Image Viewer Flash Element
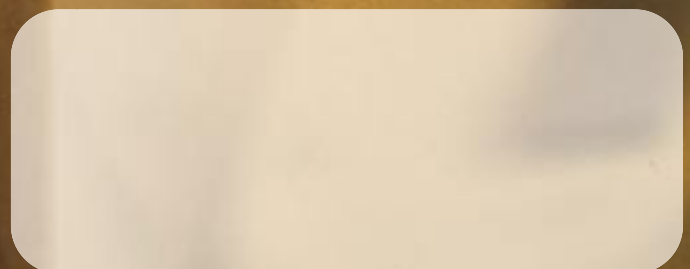
FLASH
ProgressBar Component Part 2

FIREWORKS
Repurposing Web Design Comps

COLDFUSION
Enterprise Manager

We figured it was about time that
web video stopped looking like web video.

This is a scene from one of the world's best websites, with video playing next to traditional web animation. The look of full-screen video, without ugly web players or pop-ups.

To tour the site—and see the future of video—visit:

macromedia.com/go/video5

ARE YOU READY FOR

MX7

WE ARE

MX
developer's journal

MX
developer's journal

Marketing 101 for Independent Software Contractors

by charles e. brown

as many of you know, I do Macromedia seminars around the world. A fair number of participants are independent contrac-tors. The number one question I get asked isn't about the latest programming techniques in Flash or how to develop dynamic pages in Dreamweaver. The number one question is: How do I market and charge for my services?

I always hear a collective gasp when I say that I usually charge a minimum of $250,000. I also make that the wake-up call to the participants. There are a large number of facts to support my position.

For starters, get rid of the "hourly-rate" mentality.

Think about any number of sports figures today. Why do team owners pay them tens of millions of dollars? The answer is quite simple: because those team owners are making considerably more as a result of the high-paid player.

Imagine for a second that you are called in to build the reservation system for ABC Hotel. ABC Hotel projects that it will receive $5 million in additional annu-al bookings as a result the site. Because of your skills, you can build the system in 100 hours. Considering that the average consultant is charging about $75 an hour, conventional thinking will dictate a fee of $7,500. Just think about that: charging $7,500 so that someone else can make $5 million.

Would charging something like $250,000, or even $500,000, be out of line? Of course not! Yet I see stuff like that happen everyday.

The answer is never an hourly rate, but what is it worth to someone.

As a society, we tend to be more con-cerned about the price than the worth. Something is only expensive if the rate of return is a low percentage compared to what we paid. A MacDonald's food fran-chise may seem prohibitively expensive.

However, when you find out that you will make that investment back within a year or two, with a considerable profit, sud-denly the price seems irrelevant.

Bottom line: think in terms of per-centage rather than price.

The next question is how we position ourselves to get the large enterprise sites. The process will usually go something like this:

Get your name out there by publish-ing a lot. Publishing will not get you work directly; instead, consider it your letter of introduction. It has been my experience that when you publish, people are more willing to speak with you and, in some cases, will even seek you out.

Along with publishing, attend trade shows such as MAX and FlashForward. Go to the event well-armed with busi-ness cards; be aggressive about both collecting and handing-out cards. Once again, if you have already published, that is a great door-opener.

Finally, enterprise websites will not come to you directly. Affiliate yourself with major consulting firms such as PriceWaterhouse, and so on. Again, you will see them at the trade shows and, if you have published, the door is open for you to market.

If I had to break the whole thing down to one word, it would be: network-ing.

In many respects, this is just as impor-tant as keeping your technical skills a bit ahead of the curve. If you do your job correctly, you will soon get a great repu-tation and will be a sought-after, high-paid consultant.

As a final bit of advice: always look toward the future and never to the past. Try to get a feel as to where the market is going and then try to get there before anyone else.

Each day offers us new opportunities: learn to take advantage of them.

Charles E. Brown is the editor-in-chief of MX Developer's Journal. He is the coauthor of Fireworks MX, Zero to Hero *and the auther of* Beginning Dreamweaver MX. *He also contributed to* The Macromedia Studio MX Bible. *Charles is a senior trainer for FMC on the MX product family.*
charles@sys-con.com

Media Types and Their Uses

There are presently 10 recognized media types in CSS 2
by stephanie sullivan

these days, not all users will be viewing your web site on a computer screen. Using Media types can open up a whole host of opportunities for restyling your content to cater to a variety of different devices. Media types can also open up a couple of cans of worms if you're still supporting archaic browsers, but we'll talk about that in a bit.

If you've begun to use Cascading Style Sheets for styling your screen documents, you've seen how powerful CSS really is. You may have even ventured into the realm of creating a print style sheet to facilitate a user's ability to print your web pages. If so, you already know that you had to specify the target medium (print) using the media attribute on the *<link>* element. Or maybe you just copied something a friend told you to use and you really have no idea what you did. Likely the element in the head of your document looked something like this:

```
<link rel="stylesheet" type="text/css"
media="print" href="print.css">
```

Tip: If you have not yet created a print style sheet, see Adrian Senior and Stephanie Sullivan's article in Macromedia's DevNet Center for a behind-the-scenes look at the creation of the Community MX print style sheet (http://www.macromedia.com/devnet/mx/dreamweaver/articles/screen_to_print.html)'

Recognized Media Types

There are presently 10 recognized media types in CSS 2; *all, aural, braille, embossed, handheld, print, projection, screen, tty,* and *tv*. CSS 2.1, which is now a Candidate W3C Recommendation, switches *speech* for *aural* in this list. These media types allow you to indicate to specific user devices, different methods of styling and presenting your content. As the list of possible devices grows, it is likely that more will be added. The names chosen for these media types are not case sensitive and should be fairly self-explanatory, but in case they're confusing, Table 1 is the list according to the W3C site:

Due to the variety of unique characteristics each media has, each media type may have certain properties that are unique to it. Some examples are:

- *page-break-before* - applies only to paged media
- *pause-before* - applies only to spoken media

These unique properties allow you more control within each media itself.

Media Type Tips to Keep in Mind

First, it is imperative to know which browsers and devices you are supporting. Take a look at your client's current browser statistics, or request detailed information regarding their desired browser and device support. This information determines the method you choose to specify your media-dependent style sheets.

Browser support for some of the media types can be tough. Netscape 4 doesn't support anything but *screen*, and even that may be buggy. Luckily though, the other media types won't choke it. IE4

table 1

Media Type	Use
all	Suitable for all devices.
aural (CSS 2)	Intended for speech synthesizers. See the section on aural style sheets for details.
speech (CSS 2.1)	Intended for speech synthesizers. Note: CSS2 had a similar media type called 'aural' for this purpose. See the appendix on aural style sheets for details.
braille	Intended for braille tactile feedback devices.
embossed	Intended for paged braille printers.
handheld	Intended for handheld devices (typically small screen, monochrome, limited bandwidth).
print	Intended for paged, opaque material and for documents viewed on screen in print preview mode. Please consult the section on paged media for information about formatting issues that are specific to paged media.
projection	Intended for projected presentations, for example projectors or print to transparencies. Please consult the section on paged media for information about formatting issues that are specific to paged media.
screen	Intended primarily for color computer screens.
tty	Intended for media using a fixed-pitch character grid, such as teletypes, terminals, or portable devices with limited display capabilities. Authors should not use pixel units with the "tty" media type.
tv	Intended for television-type devices (low resolution, color, limited-scrollability screens, sound available).

Note: See the CSS 2.1 specification for Recognized media types for more information on the differences between CSS 2 and CSS 2.1 (http://www.w3.org/TR/CSS21/media.html#media-types).

can read *all*, *screen*, and *print*, so if that browser must be supported, you can't use the same tricks you use for NN4. Personally, I'm doing less and less for those old gals. I want them to receive the same information in a neat logical order, but outside that, unless a client's stats show me otherwise, the styling they see will be minimal.

My main goal for legacy browsers is to make sure they don't choke on the web pages." For more on CSS support in old browsers, RichInStyle.com has a CSS support chart (http://www.richinstyle.com/bugs/table.html) with information on browser support for the *screen*, *all* and *print* media types, among other CSS properties. The chart doesn't include any version-6+ browsers, but is good for keeping track of elderly browser support.

All

According to the CSS2 recommendation, if a media type is not specified, it should default to the *all* media type. Most browsers follow this guidance. The *all* media type can be used for properties that are available to all media devices. This sheet will start the CSS cascade, and from there, the specific devices, if in compliance, will continue by parsing their specific style sheet. You should note however, that older versions of Opera used media screen as their default display.

Screen

Screen is the most commonly used media type. At this time, the majority of assistive technology products available read the *screen* style sheets instead of *aural* ones. I hope that they'll catch up sooner rather than later. Remember though, that style sheets given the *screen* media type will not be used for any of the other devices, including printers.

Handheld

If you're creating a CSS document for *handheld* media, remember that you're creating not only for a small screen, but likely also a monochromatic one, with very little bandwidth or memory available. Sometimes the connection may be less than optimal as well. Be sure to use the *{display:none}* property to remove any extraneous page information to keep

the display simple. Give them, "just the facts!" Here are some tips from the AcIS Handheld Report:

• Use very few, low-resolution, or low color depth images
• Use *alt* text for all images
• Keep individual pages short
• Use very little text formatting
• Keep page formatting flexible to fit many screen shapes and sizes

For more tips, see Optimizing Web Pages for Handheld Devices at the Palm Power web site (http://www.palmpower.com/issuesprint/issue199902/avant-gotips.html). Also be aware that Opera currently supports several media types including handheld.

Projection

This one is pretty nifty in my opinion. I first saw it in action at the fifth TODCON web conference event a couple years ago during Eric Meyer's presentation and later used it in my own. Opera still uses this media type to the best advantage. You can create a presentation for viewing either full screen or on an overhead projector.

This page on OperaShow (http://www.opera.com/support/tutorials/operashow/index.dml) gives you full instructions, but in a nutshell, you create your presentation as one long HTML document and then create the page breaks. This could even be used for an existing web site where the screen display looks completely different from the one needed or wanted for the projection device. There are some interesting styling possibilities that are available with this method. For example, you can place a watermark-type background image that will appear on each projected page. Simply set the image positioning value to static in the body element in your CSS document. Or, you can give an interesting bordered heading style to the *<h1>* element that will be at the top of each page. In action, this one looks like magic — or even PowerPoint!

Print

As mentioned before, NN4 only understands CSS documents with either no media type specified or the *screen* media type. Thus, if you want NN4 to see any styling, you'll need to use one

of these methods. Since IE4 will read the style sheet when the media type is specified as *all*, if it's on your list of browsers to support, and you've written some complex X/HTML, you may need to hide your *all* media style sheet by importing it. If you opt to use no media type at all, NN4 will read the styles, and the property values in that document will be inherited by the *print* style sheet through the cascade. However, as mentioned above, older versions of Opera will erroneously view the CSS document given no media type as *screen*, and will not "send it on" for printing. You must choose your battles.

Older Mac versions of Opera had a bug (apparently fixed in more recent versions) when using point (pt) sizes for print. Due to its improper rendering of 96dpi in print, as well as on screen, Opera's text printed larger than other browsers.

Aural/Speech

The CSS2 recommendations for Aural style sheets are on the W3C site (http://www.w3.org/TR/REC-CSS2/aural.html & http://www.w3.org/TR/CSS21/aural.html), however, much debate exists about whether to implement the aural media type within your web site. The reasons against it are based on the fact that most assistive technology that reads aloud contains its own preferences and controls. Thus, its user has likely set the voice, tone, volume, etc. that they want to listen to your web page in. Most say their preference is to listen with their own settings and that aural style sheets could be disconcerting and even annoying. At this time, none of the major screen reader technologies have integrated aural media support in their product. The hope of many is that the ability to render assistance to the sight-impaired, through the aural media type, could be integrated without being intrusive. Are there any developer's listening?

Note: If you'd like to see a list of Screen Readers and how they deal with the aural media type, linked or imported style sheets and some accessibility methods, view this page of the CSS-Discuss Wiki, researched and assembled by Bob Easton (http://css-discuss.incutio.com/?page=ScreenreaderVisibility).

Stephanie Sullivan is the founder of VioletSky Design, a successful web development company, and coming soon, W3Conversions. She is a partner at Community MX, a web development site geared toward Macromedia products. Along with writing here at MXDJ, Community MX, Macromedia's DevNet center and other web sites, Stephanie co-authored Dreamweaver MX 2004 Magic. She has done presentations at both TODCON 2004 & 2005. design@violetsky.net

Linking the CSS to your Document

Attaching a CSS document that does not use a media type is no different than attaching one that does. You can either use *<link>* with a media attribute, or *@import* with a media type or types. However, there's a third option for specifying media type as well. The *@media* rule can be used to specify several target media types within one style sheet, whether linked, imported, or embedded. So, if you had very few selectors you needed to change between media types, you can place them in your *all* style sheet and then put them in *@media* rules within that sheet like this:

```
@media print {
  body { font: 11pt/130% Times New Roman, serif; }
}
@media screen {
  body { font: 100%/130% Verdana, Arial, sans-serif; }
}
@media handheld {
  body { font: 10px Verdana, Geneva, sans-serif; }
}
```

Using the *@import* rules instead of the *@media* rules can significantly reduce the download times for different devices, since the user agents can ignore the rules that don't apply. So all would be fine in our little world — except that Internet Explorer for Windows, even the most recent version of IE6, does not recognize media types placed on an *@import* style sheet. Is there hope in IE7? Time will tell. A little web site called w3development.de has some excellent charts available showing tests with most every browser imaginable and varying versions of the *@import* rule (http://w3development. de/css/assigning_external_css/style.html). If you're weighing your options, these are worth a look. Several older browsers will not recognize the *@media* rule including: Netscape 4.x, Mac IE 5.0 and Win IE 4. w3development.de also provides a chart showing more complete browser support for the *@media* rule (http://w3development.de/css/assigning_external_css/at_media.html).

Using Shorthand

It is also possible to specify more than one media type when linking your CSS document. Thus, if you want projection and screen to use the same sheet, it can be written like this:

```
<link rel="stylesheet" type="text/css"
 media="screen, projection"  href="screen.css">
```

Similarly, *@media* rules can have more than one media type specified:

```
@media  screen, projection  {
  body { font: 100%/130% Verdana, Arial, sans-serif; }
}
```

Multiple media types should be separated by commas, and the nested sets of curly brackets are the correct method of writing *@media* rules.

In Conclusion

I hope this article has given you a good overview of media types and their possible uses. As the web matures, and older browsers wane and die out, media types will open a whole world of possibilities. For now, pay attention to the specifications of your job (the required supported browsers), weigh your options, and present something usable (but perhaps not as beautiful) to everyone. 〰

CommonSpot™
Efficient Content Management

fast. easy. affordable.

- 100 % browser-based
- Content object architecture
- Template driven pages
- 50+ standard elements
- Extensible via ColdFusion
- Content reuse
- Content scheduling
- Flexible workflow
- Granular security
- CSS support
- 508 compliance
- Personalization
- Replication
- Custom metadata
- Custom authentication
- Static site generation
- Multilanguage support

With CommonSpot Content Server you get it all. CommonSpot's exceptional blend of rapid deployment, ease of use, customization and scalability make it the leading ColdFusion content management solution.

Our rich Web publishing framework empowers developers with out-of-the-box features such as template-driven pages, custom content objects, granular security, flexible workflow and powerful ColdFusion integration hooks (just to name a few), allowing you to rapidly build and efficiently deploy dynamic, high performance Web sites.

For larger implementations, CommonSpot scales efficiently, delivering enterprise-level capabilities like replication, static content generation, multi-site clustering, personalization and custom authentication, across a diverse set of platforms.

For the past six years, PaperThin has been a leader in the ColdFusion community, and CommonSpot has been the solution of choice for organizations of all sizes, including AFL-CIO, Boeing, Kaiser Family Foundation, Ohio University, PGA.com and hundreds of others. CommonSpot's sophisticated feature set and affordable pricing are an unbeatable combination.

Call us today at **800.940.3087** to schedule a live demonstration of your site running under CommonSpot, or visit **www.paperthin.com** to learn more.

Paper|Thin
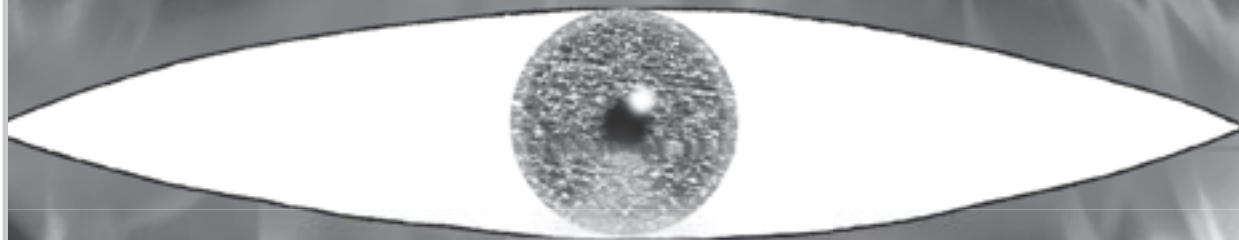
Using Image Viewer

Flash Element

by ranjan datta

Flash Elements were introduced in Dreamweaver MX 2004. Flash Elements allow you to quickly add Flash content to your page from within Dreamweaver. Currently the only flash element that comes packaged with Dreamweaver is the Image Viewer. The Image Viewer quickly assembles a highly customizable album. This tutorial will show you how.

First Things First

First, define a web site. If you don't know how to define a site.. Check the Dreamweaver MX help files. To do this go to HELP/GETTING STARTED AND TUTORIALS click the CONTENTS tab and choose "Quick Site Setup" and read the section titled "Define a Dreamweaver site". This is required to handle the image paths better. Open a New Document (Crtl+N). Change your Insert bar to 'Flash Elements' as shown below.

You would now see the Image Viewer Icon on the Insert Bar. When you click on the icon, you are prompted to save the Flash file. Name your file and save it.

You will see the swf inserted in Dreamweaver. Also you will see the property inspector for the Image Viewer.

Click the Play Button to see the image viewer. You will notice that since we have not specified our images, the image viewer currently shows only the navigation elements.

Let us now specify our images. To do so open your "Tag Inspector (F9)"

To specify your images click the edit icon on imageURLs. Click the folder icon to browse to your images one by one. Click the +/- icons to add or remove an image.

You may want to link the Images to a URL. Click the edit icon on imageLinks.

You can either browse to a link within your site of specify an absolute URL.

You may also add captions by clicking the edit icon on imageCaptions.

Since our images are 335 x 600 px, We can specify the width and height of our flash element add some space for borders and frames in the Properties Inspector. Since our images are 335 x 600 px, We can specify the width and height of our flash element add some space for borders and frames in the Properties Inspector.

figure 1
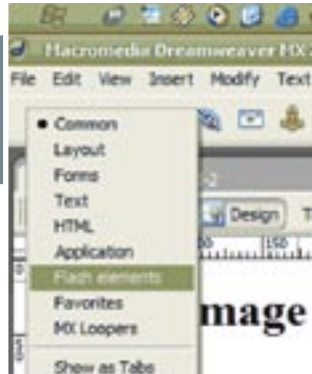
figure 2
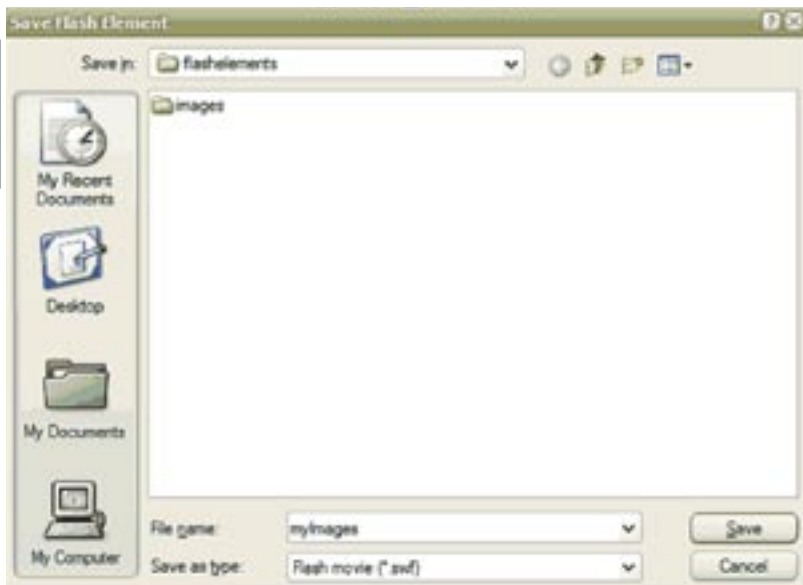
figure 3
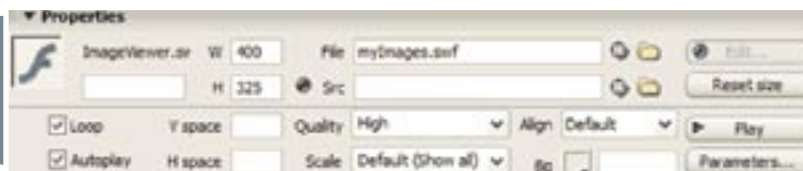
figure 4
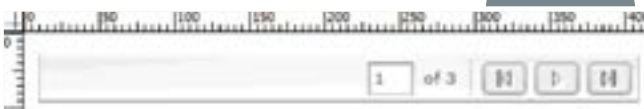
Spice It Up

You can spice up the image viewer as follows:

Background Color
Select a color using the color picker on bgColor

Captions
captionColor, captionFont, captionSize style your captions

Frame
To show a frame around your image viewer select frameShow as 'Yes', frame-Color and frameThickness further style the frame.

Transitions
Select a transitionsType from the drop down

Title
title, titleColor, titleFont, titleSize style the title of your Image show

Controls
showControls will show or hide controls, slideAutoPlay, slideDelay and slideLoop will further control the way your image viewer behaves. 〰

Ranjan Datta is a Mechanical Engineer by qualification. He took to the Internet in 1998 and has been using Dreamweaver since version 3.0. He has written online tutorials and has also created an impressive array of Dreamweaver extensions. You can find out more about his work by visiting his web site at www.dreamlettes.net. ranjan@comp-act.ws

figure 5

▼ Tag <object>	
Flash element	
bgColor	☐ #FFFFFF
captionColor	■ #333333
captionFont	Verdana
captionSize	10
frameColor	■ #333333
frameShow	(No)
frameThickness	2
imageCaptions	[]
imageLinks	['http://macromedia...
imageLinkTarget	_blank
imageURLs	['img1.jpg','img2.jpg',..
showControls	(Yes)
slideAutoPlay	(No)
slideDelay	5
slideLoop	(No)
title	
titleColor	■ #333333
titleFont	Verdana
titleSize	10
transitionsType	Random

figure 6

Edit 'imageURLs' Array

+ −

Value
'assets/00000014.JPG'
img2.jpg
'img3.jpg'

OK
Cancel
Help

figure 7

imageLinks	macromedia.com/1
imageLinkTarget	_blank
imageURLs	[a Edit Array Values...

figure 8

Edit 'imageLinks' Array

+ −

Value
'http://gthelp.com/'
'http://dreamlettes.net/'
'http://ranjan.ws/'

OK
Cancel
Help

figure 9

▼ Properties

ImageViewer.sv W 480
H 640

figure 10

▼ Tag <object>	
Flash element	
bgColor	■ #636E3A
captionColor	☐ #E5E4E2
captionFont	Verdana
captionSize	12
frameColor	■ #6C6E60
frameShow	(Yes)
frameThickness	5
imageCaptions	['Puerto Rico Fortress','Puerto Rico ...
imageLinks	['http://gthelp.com/','http://dreamle...
imageLinkTarget	_blank
imageURLs	['assets/00000014.JPG','assets/000...
showControls	(Yes)
slideAutoPlay	(Yes)
slideDelay	3
slideLoop	(Yes)
title	Puerto Rico!!
titleColor	■ #6C6E60
titleFont	Narkisim
titleSize	16
transitionsType	Random

the premier coldfusion technical conference
CFUnited

June 29 - July 1, 2005
Washington DC Area
7th Annual ColdFusion Conference

3 full days!
New Venue
75% Bigger

www.cfunited.com

CFUN has become the premier CF specific event, and Michael Smith and his team deserve all sorts of praise for their hard work in pulling it all off yet again.

Ben Forta

"...this event really is the best gathering in the world for people developing or managing CF systems. It's here that we can understand what happened, hear what's happening, and learn what's going to happen. You can't beat it."

Chuck Hoffman

"Great place to network yourself and pick up new techinques and ideas. Also to meet ones peers, and see what the future holds for all those involved with ColdFusion"

Daniel Gregorio

"Introductions to the latest developing techniques. Get inspiration in new ways to develop projects. Ge"erally, to "re-kindle" your cf "fire" by being part of a group excited by and interested in cf development."

Kathleen Ballard

Tree Component – Horizontal Scrolling

A Problem and a Solution
by guy watson

I've been playing around with the Tree Component in great depth and have found a few discrepancies which I feel are major pit holes. After a lot of searching, it was also clear that many other developers have come across the same issues. My main gripe: the tree component's support for horizontal scrolling is not very useful. I expected it to work like the ScrollPane's horizontal scrollbar, but it doesn't.

Horizontal Scrolling: The Problem

Now I have a little application that allows you to add and remove nodes to a tree, whooppee. Ok, so what happens if the user adds enough levels to the tree such that they cannot be seen because they run outside of the tree components boundaries? Enter, horizontal scrolling.

The first thing I do is checkout a property which the Tree component inherits from the List component: Tree.hScrollPolicy

```
It has two possible string values "on"
or "off" – great, so no "auto."
```

Ok, so hopefully if I just set it to "on" then it should enable the scrollbar when it is needed so that users can scroll them in and out of view. Wrong. What it actually does, is show a disabled scrollbar at the bottom of the component. Luckily, thanks to good documentation I was quickly led to another property of the Tree component that's inherited from the List component:

```
Tree.maxHPosition
```

What this property does is specify the number of pixels that the scrollbar will horizontally scroll.   In practice setting this value displays an enabled scrollbar that scrolls left and right 'x' amount of pixels even if it isnt needed. So, for the horizontal scrollbar to be useful, you actually need to update the property every time you add, remove or replace an item in the Tree to reflect the width of the widest item – which is a pain to say the least.

What makes it worse is that, when it comes to vertical scrolling, the Tree component automatically updates the size of the vertical scrollbar for you by default, unless you turn the vertical scrollbar off:

```
tree_mc.vScrollPolicy="off"
```

So, why not the horizontal scrollbar? I'm not really sure, but I think it has something to do with the component engineers worrying about the speed of the component.

Adding code to find the widest row in the tree wouldn't be hard, however it may become very slow code if you were to have 10,000 rows in the tree and every time you added, removed or replaced an item, that code would have to run again to determine the widest row and then update the scrollbars size. Not a problem if you are only adding one row to the tree, because all the code has to do is check the new row's width and compare it with the current largest row. However, it will be a problem if you are to add say 10,000 rows at once, by changing the dataProvider of a tree, because then you would have to loop through each of those new rows, and looping is very slow, when you have so many iterations. To add to that, you would only want to detect the width of the rows in the tree that are visible, i.e., their parent is open, not just those that are inside the viewable area. Which means you would also have to write

code to get a list of all the visible rows and then use that list to determine the widest row. This also means that you would have to execute this code every time a Tree node is opened or closed.

So, I guess what I am saying is that maybe the engineers actually had a valid reason to not include an auto updating horizontal scrollbar. But then I would say, how often does somebody need a Tree component that will display 10,000 rows? Not often, I would imagine, and even if it was high priority to support large datasets, I'm sure there are ways to optimize those routines for speed.

The Solution

My findings were, that there isn't really a proper solution for this, it could get very messy, simply because of the way the Tree is rendered. Let me explain.

An instance of the Tree Component only ever contains "x" amount of row movieclips, where "x" is equal to the Tree.rowCount. So, if I specify Tree.rowCount to be 7 (which is the default), then there will be seven row movieclips created inside the component when it is rendered. This causes problems because ideally we need a visual representation of each row in the Tree to determine its width. In the end I decided to start writing my own Tree component that works how I would expect a Tree component to work.

Guy Watson is the managing director of FlashGuru LTD, a Flash development consultancy company based in London, England. Guy has delivered presentations at various Flash industry events worldwide and has also won numerous industry awards for his work. He maintains the popular Flash resource website, FlashGuru's Knowledge Base - www. flashguru.co.uk guy@flashguru.co.uk

Intermedia.NET...

Now serving the hottest ColdFusion.

At Intermedia.NET we go beyond the industry standard by supporting the hottest new Coldfusion software, offering power like never before. For nearly a decade, we've been providing reliable, secure hosting to thousands of companies across the globe. We can do the same for you.

Intermedia.NET's premier hosting services include:

- ColdFusion MX hosting
- Custom tag registration
- Competitive plans
- Verity collections search engine
- Security sandboxes
- Guaranteed service levels

**Unprecedented power, unmatched reputation...
Intermedia.NET is your hosting solution.**

Call us at: **1.888.379.7729**
e-mail us at: **sales@intermedia.NET**
Visit us at: **www.intermedia.NET**

INTERMEDIA.NET

Cartoons By the People For the People

How flash is democratizing animation
by larry clow

When Disney closed the doors of its Orlando, Florida animation studio in January 2004, many saw it as the death knell for traditional hand-drawn animation. Of the 18 big-studio animated films to be released this year, only a handful will be done in 2D animation and of those, only a couple will even make it into theaters, according to KeyFrame-Online.com. In 2001, the first Oscar for an animated film was given out and the 3D animated Shrek took the prize.

Since then, only one 2D animated movie snagged the award, 2002's Spirited Away. But like its hand-drawn predecessor, 3D animation is still expensive and requires copious amounts of computer power and animators. While big-budget 3D animated movies are the most visible step in animation's evolution, they only make up a small portion of the cartoons out there. Some of the most innovative, interesting animation is being created using Flash.

A Brief Flash History

The ubiquity of Flash can hardly be overstated – as anyone who has set up a new computer can attest, you can't go three steps on the Web without being prompted to download the Flash Player for your virgin browser in order to properly view the Flash objects embedded in the Web.

The object in question might be as tiny as a button or an ad embedded in a normal HTML page, or as elaborate as an animation or entire site constructed wholly in Flash. Once you've installed the player, though, the Flash becomes a seamless part of the Web, and you never

think about it again.

But if you were to wave a magic wand that removed all the Flash objects from the Web, cyberspace would suddenly look like Swiss cheese. Great swaths of it would go black. There are one million developers using Flash now – a number that doesn't include the thousands of amateur animators, ranging in age from 15 to 50 years old, who are creating their own cartoons in bedrooms or small studios across the country.

Flash made its first appearance in 1995 as FutureSplash and was used to deliver animated content over narrow-band Internet connections. Macromedia bought the company; since then, it's evolved into a complete Web application development program that can be used to do everything from creating animated shorts and interfaces for databases to creating interactive websites with MP3 audio and full motion video. There are over 515 million computers using Macromedia's Flash player, according to the company's Web site.

Flash Animation: Starting to Hit the Big Time

Long a staple of college kids and bored cubical dwellers who pass links to Flash sites around in e-mails, Flash animation is starting to hit the big time. Gregg and Evan Spiridellis, the creators of JibJab.com, were thrust into the national spotlight just before the 2004 presidential election with their Flash short "This Land," a parody of the song "This Land is Your Land" featuring animated versions of George W. Bush and John Kerry. There was a deluge of media coverage, with the political toon getting airtime on all the cable news networks and other big

media outlets, as well as making its way to e-mail inboxes and message boards across the Web. After the election, the Spiridellis brothers followed up on their success with "Second Term," a celebration of Bush's victory sung to the tune of "She'll Be Comin' Round the Mountain."

But the JibJab toons are only the tip of the growing cartoon iceberg that Web surfers are careening into on a daily basis. Legions of amateur auteurs are using Flash to drive the animation revolution. Like a tiny digital snowflake, each Flash short is unique. Unrestrained by the bureaucracies and focus groups of big-budget studios, the political hang-ups of network censors and the shortsightedness of fickle advertisers, Flash artists are free to produce a dazzling array of short films and series as diverse as the individual minds they sprang from.

Consider Salad Fingers, a sickly-green human-vegetable hybrid with a British accent who loves to fondle rusty spoons and mutilate himself. Or Nylon Futura, a spunky redhead whose good looks halt an alien invasion. And then there's Bitey of Brackenwood, a mischievous goat-man who duels with a witch after breaking one of her flowerpots. The inhabitants of Flash Nation are a strange lot, but it's a story as old as the Web – put the means of production in the hands of the people, and they will make weird, weird stuff.

For their creators, traditional, hand-drawn animation is a memory, a cumbersome, expensive dinosaur that can't keep up with ever-changing technology. Flash is the future, they say, an emerging medium that, already enjoying a ubiquitous presence on the Web, is poised to take over TV, DVD and everything else.

Cartoons for the People

Tom Fulp, creator of Newgrounds.com, a popular Flash animation repository on the Web, said his site has 700,000 registered users, many of whom are frequently submitting original Flash content. Newgrounds receives more than 200 submissions a day. The shorts on Newgrounds usually run about three minutes in length and cover the gamut of genres, from "Comissioner Kong," a slickly animated neo-noir that casts a monkey as a police commissioner under siege, to the ridiculously offensive "Birds 'n Bees," a take-off of '50s sex-ed films that extols the joys of abortion. Users vote on each short; a high score will keep the file on the site, while a low user score will send the piece to the trash. Previously, the only reward for having a high user score was visibility on Newgrounds; now, Fulp said he's starting a monthly selection of the top five submissions, with each winner receiving $250, a T-shirt and an award certificate.

"I have always believed in Flash as an emerging medium," Fulp said in an e-mail interview. "Every year the submissions to Newgrounds continue to improve."

Games produced using Flash are also popular on Newgrounds. Fulp's game "Alien Hominid" has been played more than a million times since appearing on the site in 2002. Last year, it was redeveloped and made the crossover to the PlayStation 2 console.

Newgrounds started as a fanzine for the NeoGeo video game system in the early '90s. It wasn't until 1998 that the current form of Newgrounds hit the Web, just in time for the dot-com boom. Investors were dumping loads of money into sites like IceBox.com and others that were offering original Flash content.

"During the boom, a lot of big companies were investing millions in Web entertainment and not making a dime in return," Fulp said. "I know of several sites that spent over $20 million and never saw anywhere near the traffic of Newgrounds, which was being run from my college apartment."

One Flash series that got its start on Newgrounds is "College University." Created by Mike and Andy Parker, "CU" chronicles the adventures of Mike and Parks, two freshmen at the academically questionable College University, a place

where monkeys teach monkey physics and Optimus Prime (of "Transformers" fame) is head of security. The series is like a cross between "The Simpsons" and "Family Guy," with heavy doses of smart-ass pop culture humor and non-sequiturs.

"Andy came up with the idea to base it in a college setting, and with obvious influences from 'The Simpsons,' it grew from there. We originally wrote a script for a 30-minute TV pilot, but when I discovered Flash, we split the script up, and I started adapting it for the Web," said Mike Parker in an e-mail interview.

"CU" premiered on Newgrounds in 2000 and moved to its own site shortly afterwards. Mike Parker said the site has since logged four million hits and has 4,000 registered users.

"Our diehard fans are the best," Parker said. "They support everything we do and swarm like fire ants when anybody tries to badmouth the show. Without the killing, of course."

Despite the popularity, "CU" is still very much a small operation. It takes about three months to produce an episode. Both brothers write and provide voices, while Mike animates the show in his free time.

"I've got my day job that pays the bills and fuels my appetite for fine china," he said. "I do side freelance and side projects here and there for extra income, but nothing too involved because it ends up taking time away from CU."

It's a sentiment expressed by other Flash artists. Kirk Millett is a self-taught animator whose short film "Souvenir" appeared in a number of short film and Flash film festivals last year. It was selected as a runner-up in 2004's "The Greatest Story Never Told" contest. Millett is also the creator of "Nylon Futura," an action-adventure series that chronicles the exploits of a sexy, flame-haired heroine.

"Making money with Web content is still a tough racket. It has to compete with TV, DVDs, movies ... and then there's the whole broadband issue," Millet said in an e-mail. "On the upside, it's still the easiest and most affordable way to market your movie ideas. I do make a living off my contract work, and it's mostly animation, but it's not all glamorous stuff-a lot of ad banners and product demos."

While most shorts don't achieve

the same kind of fame that JibJab did, Flash artists say the medium is no longer confined to a small cult following on the Web.

"Flash has made it to the mainstream," said Nicholas Da Silva, creator of FlashTV.com and "The Greatest Story Never Told" contest. Some recent mainstream successes for Flash animation include "Lil' Pimp," which started out as a series on the Internet and was recently released as a full-length animated DVD. There are also a handful of cartoons on TV that are animated with Flash, including "Mucha Lucha," a series about a family of Mexican pro-wrestlers on the WB network, the ill-fated Comedy Central series "Kid Notorious," and Cartoon Network's "Atomic Betty" and "Foster's Home For Imaginary Friends." Other cross-over successes include Mondo Media's "Happy Tree Friends," which have spawned a line of DVDs and action figures and the short film "Karma Ghost," which won the viewer's choice award for animation at the 2002 Sundance Online Film Festival.

"With Flash, you're able to produce something at half the cost," Da Silva said. "The cost of the software isn't as expensive as high end software for animation. One or two people can have a studio at home and crank out their own stories."

Da Silva launched Flash TV in 2000 as a place to reach out to Flash artists. In 2004, the site sponsored the first "TGSNT" competition. The contest received 256 entries from across the globe, Da Silva said. This year, entries were down to 110, which Da Silva attributes to the $30 entry fee he had to charge to keep the contest going. Other contests, like FlashForward, are also entirely devoted to original Flash shorts.

The versatility of Flash itself means the future of animation lies far beyond the Web, according to Da Silva. "We can actually take a story, storyboard it, bring it into Flash, animate it, export it to the Web, export the same file to DVD as well as broadcast TV, (and then) bring it down in size for wireless phones," he said.

Building a Killer App

Along with its creative applications, Flash animation also has a great deal of commercial use, said Jeremy Clough, creative director for Hatchling Studios in Portsmouth, New Hampshire. Hatchling

Newgrounds
www.newgrounds.com

One of the biggest Flash portals on the Web, Newgrounds receives more than 200 Flash submissions a day. Newgrounds has everything from games like "Samurai Asshole" to a series of shorts that pit Mr. T against a variety of opponents including Godzilla and Hitler.

Flash TV
www.iwantmyflashtv.com

Flash TV is like the older, more gainfully employed brother of Newgrounds. Birthplace of "The Greatest Story Never Told" (www.tgsnt.com), Flash TV features work by professional Flash artists, broken down by genre. Standouts include Hong Kong action-style "Deadly Thirst" and "Xombie," a post-apocalyptic tale with a zombie hero.

College University
www.collegeuniv.com

Fans of "The Simpsons" and "Family Guy" will appreciate "College University." "CU" follows the misadventures of Parks and Mike, two freshmen at College University. The two friends face the usual college pitfalls of angry lemurs, ornery monkey professors, and the watchful eye of campus security chief Optimus Prime. "CU" recently had former wrestler and current Slim Jim spokesman Randy "Macho Man" Savage stop by for a guest appearance.

The Happy Tree Friends
www.happytreefriends.com

If you like your animals cute and cuddly and your violence outrageous and gruesome, check out the Happy Tree Friends. Join Lumpy, Sniffles and Flippy the Green Beret bear as they meet a variety of untimely, bloody demises, each more spectacular than the last. Key episode: "Flippin' Burgers." Flippy visits a fast food joint and starts having Vietnam flashbacks. The ketchup and blood flows freely as Flippy massacres the restaurants' patrons with condiment bottles and other weapons.

Homestar Runner
www.homestarrunner.com

Home of Strong Bad, The Cheat and an assortment of cartoon misfits, Homestar Runner's most popular feature is the series of "Strong Bad Email" shorts, where the Lucha Libre-masked character answers fans' e-mail questions. Old-school video game fans should watch the "Peasant's Quest" preview, a dead-on send-up of old computer RPGs.

Kirk Millett
www.kirkmillett.com

Check out Millett's site for the full series of Nylon Futura cartoons, as well as "Souvenir," a neo-noir tale of a shipwreck survivor taken aboard a mysterious submarine. "Souvenir" was selected as the runner-up in 2004's "The Greatest Story Never Told" contest.

specializes in 3D animation and Web development using Flash.

Hatchling's clients have included Proctor and Gamble and Gigunda Group, a marketing firm that specializes in "experiential marketing."

"(The majority) of the audience out there are not analytical, they're visual; Flash is just a great way to represent information in a flow," he said. "Flash is the bridge between static information and animation."

This means using Flash for everything from advertising tool bars to interactive product demos. Flash can integrate video and sound onto a Web site and can be used to build Web sites with or without HTML.

Will Powley, owner of Mad*Pow Media Solutions in Portsmouth, said about half the design and Web development work he does uses Flash. Some of Powley's clients include American Express,

Timberland and textbook publisher McGraw Hill.

"We do visual explanations, whether it be for marketing, doing Web sites, doing systems, or building applications, so Flash is about a huge chunk of our business," he said.

One of the attractions of Flash as a development tool is that users can create and modify extensions for the program, Powley said. This means Flash developers can essentially custom-

ize limited portions of the program to fit their development needs.

"It's really helped the application grow," Powley said. "With the release of Flash MX Professional ... Macromedia is saying 'We don't want to have it just for animation, we want it to build any kind of Web application you can ever imagine.'"

In the future, Powley sees even simple systems, like hotel reservation programs, being built in Flash. "The interface is completely customized and branded. From a systems standpoint, that's the very next level, creating very, very complex Web applications using Flash that could never be dreamed about before," he said.

The Death of Cell Animation?

But with everyone becoming small-scale Walt Disneys in the comfort of their homes, what's the future of old-school cell animation?

"It's dead. Gone," Clough. said. "You can do everything in Flash, without the overhead," he added. "One person could be a studio, two people can develop a great concept." Flash eliminated obstacles like enormous file sizes and large frame rates that kept animation off the Web, Clough said.

Millett isn't so quick to write cell animation's obituary yet.

"Flash is just another animation tool. It's what you make of it. I don't think hand-drawn art of any kind will ever die. There will always be new toys, but there's no substitute for the fundamentals," Millett said.

As Flash artists become more and more sophisticated, Fulp said they will get closer to replicating pen and ink drawings.

"A Flash animation done with a drawing pad has the potential to look as good as something drawn with ink," he said. "Cell animation is expensive; Flash opens the doors for many more independent artists to create their masterpieces without a big budget."

Though there's no agreement on the future of traditional animation, there is an almost unanimous opinion that Flash is democratizing cartoons.

"I think Flash is changing animation by allowing everybody to give it a try. Before Flash really came along, if you wanted to do some animation, you'd better have a lot of paper and free time," Parker said. "Now anybody with a computer can give it a shot. I don't think it takes anything away from the art of traditional animation, but just allows more people a new way to express themselves."

For Da Silva, the future of animation lies entirely in Flash.

"It empowers the little people to tell their story. The technology out there also allows us to present it online to the masses," he said.

The members of Flash Nation are just getting started, though.

"The Web is taking more and more eyeballs from television and it will continue to do so," Fulp said. "We are a new generation of artists and are still in the process of getting organized."

Larry Clow is staff writer for The Wire (www.wirenh.com), a free weekly newspaper distributed at more than 200 locations up and down the New Hampshire seacoast, from Salisbury to Rochester, from Exeter to Durham and Kittery, and all around Portsmouth and everywhere between. lclow@wirenh.com

FLASHFORWARD
& FLASH® FILM FESTIVAL

**JULY 6-8 2005 NEW YORK CITY
NEW YORKER HOTEL**

THE WORLD'S PREMIER FLASH EVENT

The 14th Flashforward conference and Flash® Film Festival, the largest and longest running Flash user conference in the world, shares the latest in design, development, education and inspiration.

FLASHFORWARD 2005 FEATURES:
- 6 In-Depth Flash Workshops
- 25 One-Hour Seminars
- 20 Q&A Sessions
- 17 "Ask the Experts" Sessions
- 11 Technology Showcase Seminars
- Flash® Film Festival
- Macromedia Keynote
- Exhibition Area
- Meet the Speakers and Flash Authors Reception
- Networking Receptions
- Exclusive Conference Workbook with Speaker Notes

GOLD SPONSOR

macromedia®

SILVER SPONSOR

VitalStream®

BRONZE SPONSORS

introNetworks

macromedia®
PRESS

METALIQ

New Riders

Peachpit Press

MEDIA SPONSORS

ANIMATION

MX
developer's journal

REGISTRATION:
$599 through June 3
$699 through July 1
Additional $50 Macromedia User Group discount
$799 at the door

ADDITIONAL INFORMATION:
- Call toll free 1.877.4.FLASH.4
- www.flashforwardconference.com
- 1-303-593-1977 outside the US and Canada

Skinning the V2 ProgressBar Component

Part 2

In Part One of this two-part article ("Skinning the V2 ProgressBar component ," MXDJ vol 3 issue 5) we did what we set out to do, but I just can't bear to leave the skinning-a-Button issue unresolved.

by colin moock

The warning at the beginning of last month's article pointed out that the Button component is surprisingly difficult to skin because its skin is made up of code, not graphics. As a last exercise, then, let's take a quick look at an alternative approach: skinning SimpleButton instead of Button.

The SimpleButton class is the superclass of Button, and is accordingly less complex. Specifically, SimpleButton instances cannot be resized and do not have text labels – which makes them easy to skin. Follow these steps to skin a SimpleButton instance:

1. Create a new Flash document (.fla file).
2. Drag a Button instance to the stage and then delete it. This adds the Button component to the document Library.
3. Create four movie clips, named up, over, down, and icon, with Linkage Identifiers set to "up", "over", "down", and "icon", respectively.
4. In the up movie clip, create a graphic that represents the button in an idle state.
5. In the over movie clip, create a graphic that represents the button when the mouse pointer is above it.
6. In the down movie clip, create a graphic that represents the button when it is pressed.
7. In the icon movie clip, create a graphic that will appear in the middle of the button in all states.
8. Select Frame 1, Layer 1.
9. Add the following code to the Actions panel:

```
import mx.controls.*;

var initObj = {falseUpSkin:"up",
            falseDownSkin:"down",
            falseOverSkin:"over"};

initObj.falseUpIcon   = "icon";
initObj.falseDownIcon = "icon";
initObj.falseOverIcon = "icon";
initObj.trueUpIcon    = "icon";
initObj.trueDownIcon  = "icon";
initObj.trueOverIcon  = "icon";

// Cast the object returned by create-
ClassObject() to SimpleButton.
var sButton:SimpleButton =
SimpleButton(
                      this.
createClassObject(SimpleButton,
                      "sButtonInstance",

                      0,

                      initObj));
sButton.move(250, 200);
```

10. To export the movie, choose Control > Test Movie.

In the preceding steps, we created the SimpleButton instance with code (via createClassObject()) because there is no SimpleButton visual component included in the Flash MX 2004 Components panel. It is theoretically possible to subclass SimpleButton and use the subclass to create a component that could be added to the Components panel, but doing so is outside the scope of this article. For information on creating components, see "Further Study," below.

The SimpleButton class is not the only alternative to the Button component. The CustomButton class can also be skinned with graphical movie clips. CustomButton offers the additional features of resizability and a text label. Note, however, that a CustomButton instance can be resized along one axis only – either horizontally or vertically. Accordingly, the CustomButton skin is structured much like the ProgressBar skin; it has two end caps for the edges of the button and a middle section that is stretched between the caps. You can use the SimpleButton skinning technique described above to skin a CustomButton instance. Of course, you'll have to use the CustomButton skin properties instead of the SimpleButton skin properties. For reference, here is a list of CustomButton's skin properties:

- falseUpSkinL
- falseUpSkinM
- falseUpSkinR
- falseDownSkinL
- falseDownSkinM
- falseDownSkinR
- falseOverSkinL
- falseOverSkinM
- falseOverSkinR
- trueUpSkinL
- trueUpSkinM
- trueUpSkinR
- trueDownSkinL
- trueDownSkinM
- trueDownSkinR
- trueOverSkinL
- trueOverSkinM
- trueOverSkinR
- falseDisabledSkinL
- falseDisabledSkinM
- falseDisabledSkinR
- trueDisabledSkinL
- trueDisabledSkinM
- trueDisabledSkinR

In the preceding property names, "false" means that the button is not in a "toggled on" state, while "true" means that the button is in a "toggled on" state. Non-toggle buttons (i.e., normal buttons) use the "false" skins only. The letter "L" means left cap, "M" means middle, stretching section, and "R" means right cap.

Note, however, that rather than relying on the above list of CustomButton skin properties you should always consult the actual source code for the CustomButton class to verify which skin properties it supports. Different versions of the class may support different properties. The source code also contains a brief description for each skin element. The source file for CustomButton is located here:

- Windows: \Program Files\Macromedia\ Flash MX 2004\[LANGUAGE_CODE]\ First Run\Classes\mx\controls\ CustomButton.as
- Macintosh: HD/Applications/ Macromedia Flash MX 2004/First Run/ Classes/mx/controls/CustomButton.as

Normally you would be able to find more information in the Macromedia Flash documentation, but neither the CustomButton class nor the SimpleButton class are included in the current Macromedia Flash MX 2004 documentation. See also Nigel Pegg's blog post (In the preceding steps, we created the SimpleButton instance with code (via createClassObject()) because there is no SimpleButton visual component included in the Flash MX 2004 Components panel. It is theoretically possible to subclass SimpleButton and use the subclass to create a component that could be added to the Components panel, but doing so is outside the scope of this article. For information on creating components, see "Further Study," below.

The SimpleButton class is not the only alternative to the Button component. The CustomButton class can also be skinned with graphical movie clips. CustomButton

> "CustomButton offers the additional features of resizability and a text label. Note, however, that a CustomButton instance can be resized along one axis only – either horizontally or vertically"

offers the additional features of resizability and a text label. Note, however, that a CustomButton instance can be resized along one axis only – either horizontally or vertically. Accordingly, the CustomButton skin is structured much like the ProgressBar skin; it has two end caps for the edges of the button and a middle section that is stretched between the caps. You can use the SimpleButton skinning technique described above to skin a CustomButton instance. Of course, you'll have to use the CustomButton skin properties instead of the SimpleButton skin properties. For reference, here is a list of CustomButton's skin properties:

- falseUpSkinL
- falseUpSkinM
- falseUpSkinR
- falseDownSkinL
- falseDownSkinM
- falseDownSkinR
- falseOverSkinL
- falseOverSkinM
- falseOverSkinR
- trueUpSkinL
- trueUpSkinM
- trueUpSkinR
- trueDownSkinL
- trueDownSkinM
- trueDownSkinR
- trueOverSkinL
- trueOverSkinM
- trueOverSkinR
- falseDisabledSkinL
- falseDisabledSkinM
- falseDisabledSkinR
- trueDisabledSkinL
- trueDisabledSkinM
- trueDisabledSkinR

In the preceding property names, "false" means that the button is not in a "toggled on" state, while "true" means that the button is in a "toggled on" state. Non-toggle buttons (i.e., normal buttons) use the "false" skins only. The letter "L" means left cap, "M" means middle, stretching section, and "R" means right cap.

Note, however, that rather than relying on the above list of CustomButton skin properties you should always consult the actual source code for the CustomButton class to verify which skin properties it supports. Different versions of the class may support different properties. The source code also contains a brief description for each skin element. The source file for CustomButton is located here:

- Windows: \Program Files\Macromedia\ Flash MX 2004\[LANGUAGE_CODE]\ First Run\Classes\mx\controls\ CustomButton.as
- Macintosh: HD/Applications/ Macromedia Flash MX 2004/First Run/ Classes/mx/controls/CustomButton.as

Normally you would be able to find more information in the Macromedia Flash documentation, but neither the CustomButton class nor the SimpleButton class are included in the current Macromedia Flash MX 2004 documentation. See also Nigel Pegg's blog post (In the preceding steps, we created the SimpleButton instance with code (via createClassObject()) because there is no SimpleButton visual component included in the Flash MX 2004 Components panel. It is theoretically possible to subclass

SimpleButton and use the subclass to create a component that could be added to the Components panel, but doing so is outside the scope of this article. For information on creating components, see Further Study, below.

The SimpleButton class is not the only alternative to the Button component. The CustomButton class can also be skinned with graphical movie clips. CustomButton offers the additional features of resizability and a text label. Note, however, that a CustomButton instance can be resized along one axis only--either horizontally or vertically. Accordingly, the CustomButton skin is structured much like the ProgressBar skin; it has two end caps for the edges of the button and a middle section that is stretched between the caps. You can use the SimpleButton skinning technique described above to skin a CustomButton instance. Of course, you'll have to use the CustomButton skin properties instead of the SimpleButton skin properties. For reference, here is a list of CustomButton's skin properties:

- falseUpSkinL
- falseUpSkinM
- falseUpSkinR
- falseDownSkinL
- falseDownSkinM
- falseDownSkinR
- falseOverSkinL
- falseOverSkinM
- falseOverSkinR
- trueUpSkinL
- trueUpSkinM
- trueUpSkinR
- trueDownSkinL
- trueDownSkinM
- trueDownSkinR

- trueOverSkinL
- trueOverSkinM
- trueOverSkinR
- falseDisabledSkinL
- falseDisabledSkinM
- falseDisabledSkinR
- trueDisabledSkinL
- trueDisabledSkinM
- trueDisabledSkinR

In the preceding property names, "false" means that the button is not in a "toggled on" state, while "true" means that the button is in a "toggled on" state. Non-toggle buttons (i.e., normal buttons) use the "false" skins only. The letter "L" means left cap, "M" means middle, stretching section, and "R" means right cap.

Note, however, that rather than relying on the above list of CustomButton skin properties you should always consult the actual source code for the CustomButton class to verify which skin properties it supports. Different versions of the class may support different properties. The source code also contains a brief description for each skin element. The source file for CustomButton is located here:

- Windows: \Program Files\Macromedia\ Flash MX 2004\[LANGUAGE_CODE]\ First Run\Classes\mx\controls\ CustomButton.as
- Macintosh: HD/Applications/ Macromedia Flash MX 2004/First Run/ Classes/mx/controls/CustomButton.as

Normally you would be able to find more information in the Macromedia Flash documentation, but neither the CustomButton class nor the SimpleButton class are included in the current Macromedia Flash MX 2004 documentation. See also Nigel Pegg's blog post (http://www.markme.com/nigel/ archives/004644.cfm) on the subject of skinning buttons. Nigel is one of the original creators of the V2 component set.

Further Study

For more information on skinning and customizing the V2 components, see the following resources:
- Skinning the Flash MX 2004 Components (http://www.macrome- dia.com/devnet/mx/flash/articles/skin- ning_2004_print.html)

- Flash MX 2004 Version 7.2 updat- ed Help: Using Components > Customizing Components > About Skinning Components
- Building and Testing Components in Macromedia Flash MX 2004 (http:// www.macromedia.com/devnet/mx/ flash/articles/buildtest_comp.html)
- Flash MX 2004 Version 7.2 updated Help: Using Components > Creating Components

Colin Moock is an independent web guru with a passion for networked creativity and expression. He is author of the world-renowned guides to Flash programming, "ActionScript for Flash MX: The Definitive Guide" (O'Reilly & Associates, 2003, 2001) and "Essential ActionScript 2.0" (O'Reilly & Associates, 2004). A web professional since 1995, Moock runs one of the web's most venerable Flash developer sites, www.moock.org. He spends most of his time pursuing his cardinal interest, multiuser application development, and working on Unity, moock.org's complete commercial framework for creating and deploying multiuser applications for Macromedia Flash."

The NumericStepper Component

One of flash's hidden gems
by steven grosvenor

perhaps one of the easiest components to overlook in Flash MX 2004 is tucked away in the UI Components section of the Components Panel, and carries the uninteresting name NumericStepper. Perhaps you're developing experimental effects and require on-screen controls to alter dynamic feedback within those effects. Maybe you're building complex forms that allow users to quickly select numerical data. Either way, it's time to cast your eye over the NumericStepper component.

Let's look at an example. All the dynamics of the movie pictured below are controlled with instances of the NumericStepper.

NumericStepper is an extremely valuable component, but at its most basic, it's designed to step up or down through a series of numbers incrementally. There are several properties of the component that we can get and set. These extremely useful properties, which we'll explore in this article, are as follows:

- minimum: the minimum value of the component
- maximum: the maximum value of the component nextValue: the next incremental value of the component (read-only)
- previousValue: the previous incremental value of the component (read-only)
- stepSize: the incremental change value for the component
- value: the currently selected value of the component

Potential uses for the NumericStepper component include:

- used as form elements to facilitate rapid numeric value selection
- used as date selection elements

(months, years etc)
- used in experimental applications to control complex values

The main difference between the NumericStepper and ComboBox components, apart from the methods, properties and events, is to do with ease of use.

While you can accomplish similar effects using the ComboBox component, the NumericStepper component allows you to increment up or down a selection rapidly via the 'up' and 'down' arrows, or the up and down keys on the keyboard. Using the ComboBox, you need to click the ComboBox, wait for it to display the options, then select the desired value from a dropdown.

The NumericStepper solution is obviously much quicker than the ComboBox, and its capabilities allow NumericStepper to be implemented and used easily and effectively in some unique applications.

Numeric Stepper Basics

To get a feel for this component, let's add a single instance of NumericStepper to the stage. We'll then populate some dynamic text boxes with a couple of the NumericStepper component's properties.

The source FLA file for this example is called NumericStepperBasics.fla and can be found in the downloadable code archive accompanying the online version of this article at http://mxdj.sys-con.com. Setting the Stage

1. Start by creating a new Flash document. Accept the default width, height and frame rate settings.
2. Rename the default layer Actions and add another layer beneath it called Components. Drag an instance of the NumericStepper component from

the UI Components section of the Components panel into the first frame of the Components layer. Name the instance StepperMain.
3. Select the NumericStepper component instance and add the following values within the Property Inspector: maximum:100, minimum:0; stepSize:5, value:10.

Note: Here, we've added the component's property values via the Property Inspector, but we can also add these values programmatically via ActionScript if we wish (more on this later).

4. Create two dynamic text boxes one above the other. Name the instances previousTextBox and nextTextBox.

When we click the NumericStepper component, these text boxes will be populated with the next and previous values of the component. We'll add the ActionScript that achieves this in a moment.

5. Add relevant labels to the component and text boxes if you wish, as shown in the above screenshot.

Let's move on. Now, we'll add the ActionScript that allows us to see the component in action...

Adding the ActionScript

1. With the first frame of the Actions layer selected, add the following code within the Actions Panel:

```
var palletteArray = new Array ();
palleteArray = ["0x7FAAD4",
"0xFFDF7F"];
function setTheme() {
 _global.style.setStyle("themeColor",
palleteArray[0]);
 _global.style.setStyle("backgroundCol
```

```
or", palleteArray[1]);
 _global.style.setStyle("fontSize",
11);
 _global.style.setStyle("fontWeight",
"bold");
 _global.style.setStyle("color", pal-
leteArray[0]);
}
setTheme(); // Called at startup

var stepsListener = new Object();
stepsListener.change = function() {
 populateTextBoxes();
};
stepperMain.addEventListener("change",
stepsListener);

function populateTextBoxes() {
 if ((stepperMain.value<stepperMain.
maximum) && (stepperMain.
value>stepperMain.minimum)) {
    nextTextBox.text = stepperMain.
nextValue;
    previousTextBox.text = stepperMain.
previousValue;
 }
}
```

The first thing that we are doing here is creating a new array called 'palletteArray', which we populate with a couple of color codes that we've selected for our interface: a nice pastel yellow (#7FAAD4) and a complementary blue (#FFDF7F).

Note: Creating an array of colors makes it much easier to switch those colors if we need to: we simply change the values of different array members. This solution allows us to avoid having to search through the code, changing individual hexadecimal values as we find them.

We then have a function called setTheme(), which is called at startup. It sets the theme for the NumericStepper component (or any other Flash MX 2004 component on the stage) using the palletteArray array we set up earlier.

We then create a new listener object that listens for a change in the value of the NumericStepper component. When a change is detected, the populateTextBoxes() function is called. It pushes the nextValue and previousValue properties of the component to the relevant dynamic text boxes.

2. Save the Flash document to a location of your choice and preview your work in Flash.

As you can see, clicking on the NumericStepper instance increments or decrements the values of the component, and pushes the next and previous values into the dynamic text areas.

This example gave us a nice simple introduction to the component. We can now build on these foundations by creating more complex creations for use in your projects.

Date Selection

The source FLA file for this example is NumericStepperDates.fla.

This example is a little different as it uses the NumericStepper component to create simple date controls within the application. These date controls might be used within an online application form, or a Flash-driven blog-searching application. In this example, we'll just send the output of a function to the Output Panel.

Setting the Stage

We'll use some of the same core code that we used in the previous example and build upon that to create a different outcome; also in this example we will populate the NumericStepper component using ActionScript rather than hard-coding its contents via the Property Inspector.

1. Create a new Flash document with default properties and rename the default layer Actions. Below it, add a further layer called Components.
2. Drag 2 instances of the NumericStepper component from the UI Components section of the Components Panel into the first frame of the Components layer. Name the instances StepperYear and StepperMonth.
3. Add relevant text labels above each component (e.g. Select Year, Select Month).

Adding the ActionScript

It's now time to populate the components with the values we want for the month and year. When the values in either of the component instances change, the selected date is pushed to the Output Panel.

1. Select the first frame of the 'Actions' layer and add the following code

figure 1
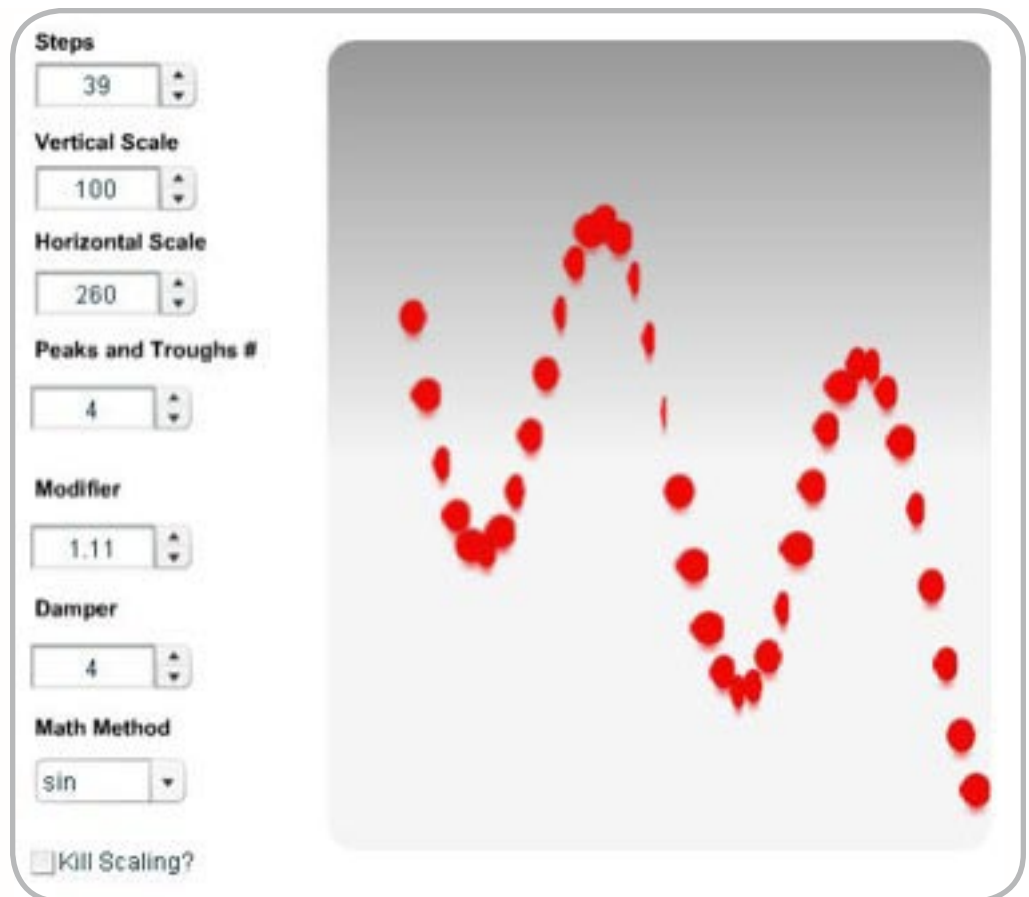
within the Actions Panel:

Note: Code highlighted in bold has altered from the previous example.

```
var monthArray = new Array ();
monthArray = ["January", "February",
"March", "April", "May", "June",
"July", "August", "September",
"October", "November", "December"];

var palletteArray = new Array();
palleteArray = ["0x7FAAD4",
"0xFFDF7F"];
function setTheme() {
 _global.style.setStyle("themeColor",
palleteArray[0]);
 _global.style.setStyle("backgroundCol
or", palleteArray[1]);
 _global.style.setStyle("fontSize",
11);
 _global.style.setStyle("fontWeight",
"bold");
 _global.style.setStyle("color", pal-
leteArray[0]);
}
setTheme();

var stepsListener = new Object();
stepsListener.change = function() {
 pushDate();
};
stepperYear.addEventListener("change",
stepsListener);
stepperMonth.addEventListener("change"
, stepsListener);

function pushDate() {
 trace("The selected year is: "+step-
perYear.value+
   "; the selected Month is: "+monthA
rray[stepperMonth.value-1]);
}
function populateYearStepper() {
 var now = new Date();
 stepperYear.minimum = now.getUTCFul-
lYear();
 stepperYear.maximum = now.getUTCFul-
lYear()+10;
 stepperYear.stepSize = 1;
 stepperYear.value = now.getUTCFul-
lYear();
}
populateYearStepper();
function populateMonthStepper() {
 var now = new Date();
 stepperMonth.minimum = 1;
 stepperMonth.maximum = 12;
 stepperMonth.stepSize = 1;
 stepperMonth.value = now.getUTC-
Month()+1;
}
populateMonthStepper();
```

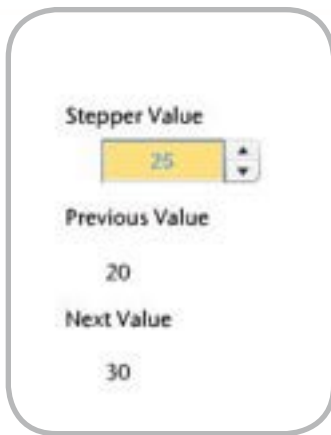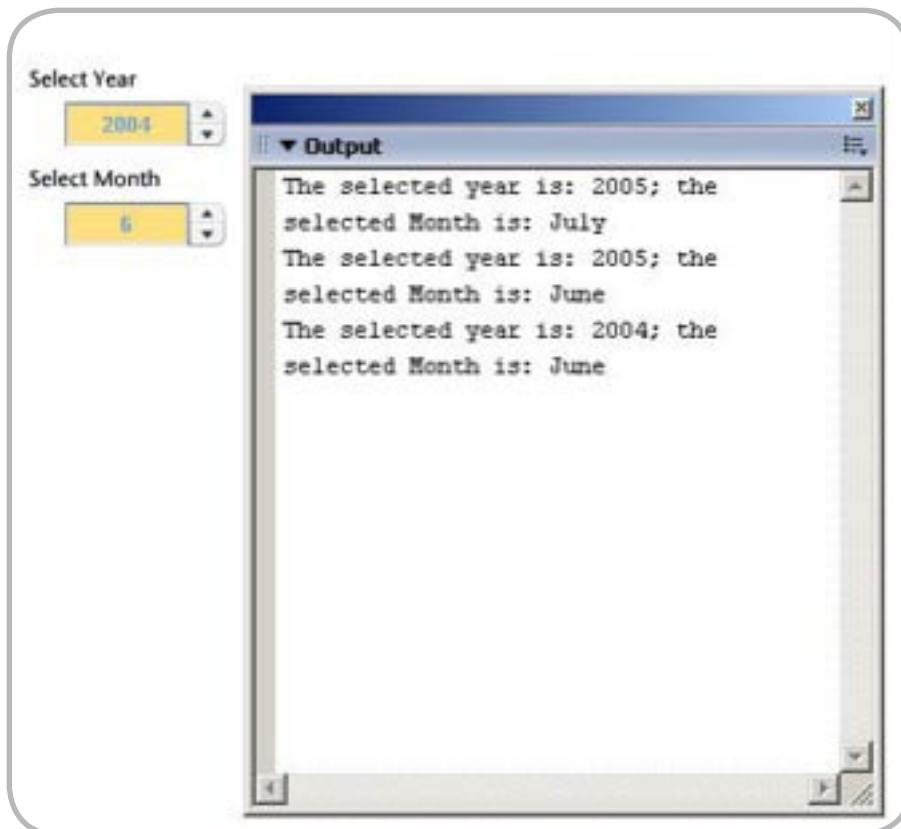Initially, we create a new array called monthArray, which holds the names of the months of the year; we will use this later in the example to get month names from the pushDate() function.

Note that in this example, we tie both instances of the component to the same listener object. Whenever the value of either instance changes, the pushDate() function is called.

```
1.stepperYear.addEventListener("change
", stepsListener);
stepperMonth.addEventListener("change"
, stepsListener);
```
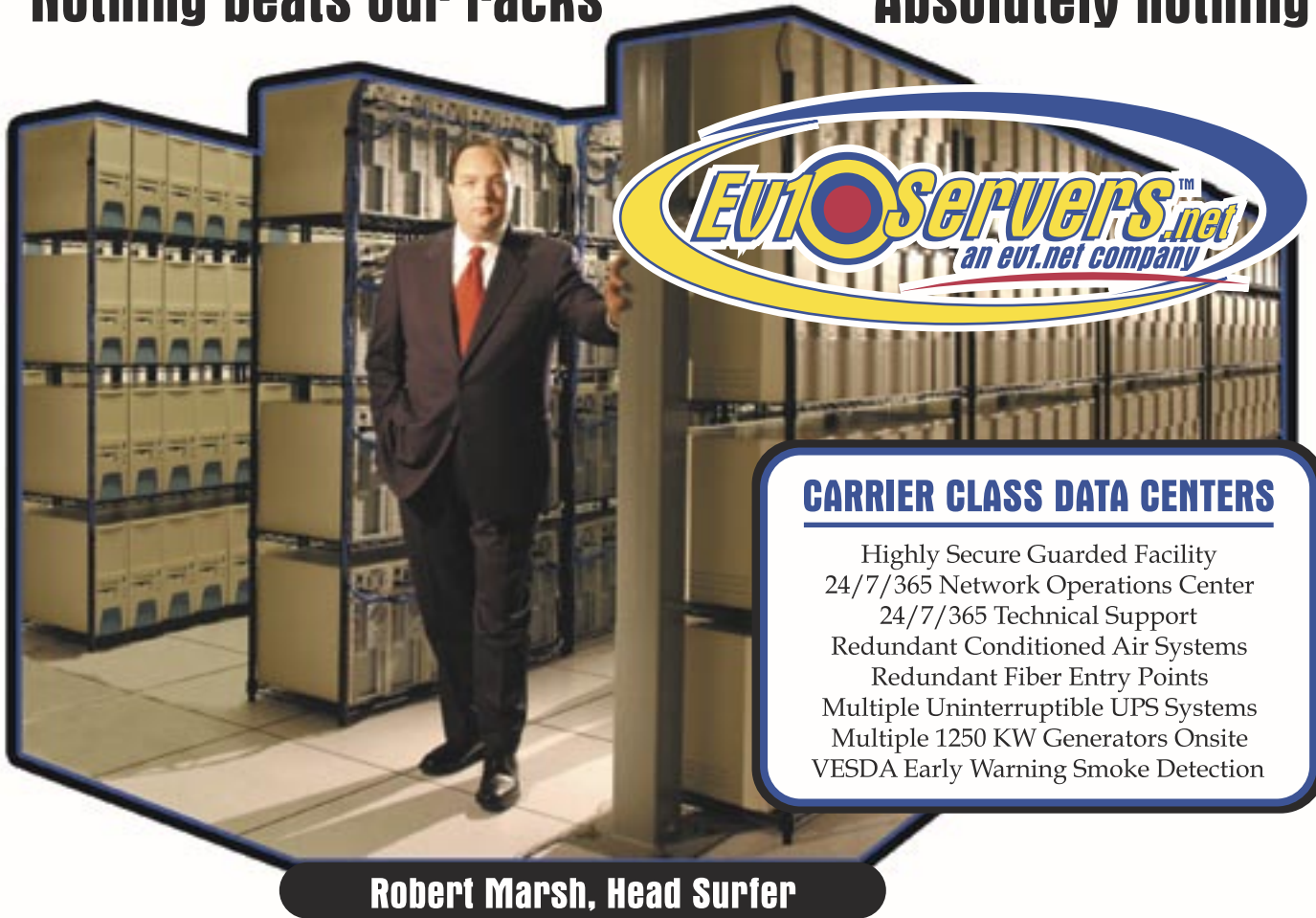
We have two functions in this example (populateYearStepper(), and populateMonthStepper()), both of which are called at startup. They dynamically populate the stepperYear and stepperMonth NumericStepper instances, respectively.

The populateYearStepper() function creates a new date object, then pushes the minimum, maximum, stepSize and

figure 2

Stepper Value

25

Previous Value

20

Next Value

30

figure 3

Select Year

2004

Select Month

6

Output

The selected year is: 2005; the
selected Month is: July
The selected year is: 2005; the
selected Month is: June
The selected year is: 2004; the
selected Month is: June

value properties into the stepperYear component instance using various iterations of the getUTCFullYear() method. This produces a minimum value, which is the current year, and a maximum value, which is the current year plus 10. It sets the step size to 1 and the initial value of the NumericStepper instance to the current year.

A similar process occurs with the populateMonthStepper() function, which populates the stepperMonth instance with the month numbers (1-12). It then sets the initial value to the current month using the following code:

```
now.getUTCMonth()+1
```

The getUTCMonth() method returns an integer between 0 and 11, indicating the month value in the Date object (January will appear as 0, December as 11, etc.). We add 1 to this to obtain the number we wish to display in the NumericStepper.

When either of the values of the NumericStepper instances change, the pushDate() function is called. We use this function to trace out the selected date. The year is simple enough to trace out, as we can easily get the value of the stepperYear NumericStepper instance using stepperYear.value.. But, in order to convert the zero-based month value into a month name, we need to look up its position within the monthArray array we initiated earlier.

```
trace("The selected year is: "+step-
```

```
perYear.value+
 "; the selected Month is: "+monthArra
y[stepperMonth.value-1]);
```

2. Save your Flash document and preview your work.

When you click on either of the NumericStepper instances, the push-Date() function is called and the selected month and year are traced out to the Output Panel in a convenient manner. There are multiple iterations and uses of the NumericStepper component that can be derived from this example, so I suggest that you experiment and see how you can apply this component in your own projects.

Experimental Control

As we've already mentioned, another interesting application of the NumericStepper component is to use it as a control for experimental effects, such as the one shown in the screenshot at the beginning of this tutorial.

In this example, we'll use the component dynamically to create multiple dynamic iterations of text fields via the createEmptyMovieClip() and createTextField() method.

The value property of the NumericStepper instance is retrieved and used to iterate through a loop, which creates a series of text fields. On each subsequent change of the component's value, the screen is redrawn with the number of text fields reflecting the value within the component.

The source FLA file for this example is called NumericStepperDynamicText.fla and can be found in the code archive for this article.

Note: The code archive includes another example of this FLA in which the dynamic creations are placed randomly on the stage. The file's named NumericStepperDynamicRandomText.fla

While this example shows how to dynamically control the creation and destruction of Movie Clips and text fields, I'm going to skim through the dynamic creation code. Instead, I'll concentrate on the logic flow through the example so you can get on with dissecting the example and modifying it for your needs.

Setting the Stage

1. Create a new Flash document with default properties and rename the default layer Actions. Add below it a layer called Components.
2. Drag an instance of the NumericStepper component from the UI Components section of the Components Panel into the first frame of the Components layer. Name the instance StepperMain.
3. With the NumericStepper component instance selected, add the following values within the Property Inspector: maximum:100, minimum:0; stepSize:5, value:10.
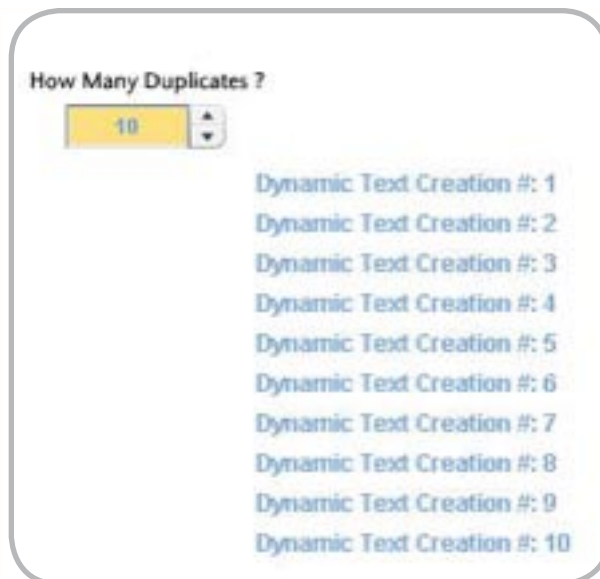
From a setup point of view, that's all we need to do. The rest of the example is controlled via ActionScript.

Adding the ActionScript

1. Select the first frame of the Actions layer and add the following code within the Actions Panel:

```
var palletteArray = new Array();
palleteArray = ["0x7FAAD4",
"0xFFDF7F"];
function setTheme() {
 _global.style.setStyle("themeColor",
palleteArray[0]);
 _global.style.setStyle("backgroundCol
or", palleteArray[1]);
 _global.style.setStyle("fontSize",
11);
 _global.style.setStyle("fontWeight",
"bold");
 _global.style.setStyle("color", pal-
leteArray[0]);
```


figure 4

How Many Duplicates ?

10

Dynamic Text Creation #: 1
Dynamic Text Creation #: 2
Dynamic Text Creation #: 3
Dynamic Text Creation #: 4
Dynamic Text Creation #: 5
Dynamic Text Creation #: 6
Dynamic Text Creation #: 7
Dynamic Text Creation #: 8
Dynamic Text Creation #: 9
Dynamic Text Creation #: 10

```
}
setTheme();

var stepsListener = new Object();
stepsListener.change = function() {
 removeDeadClips();
 createTextBoxes();
};
stepperMain.addEventListener("change",
stepsListener);

function createTextBoxes() {
 if (stepperMain.value>stepperMain.
minimum) {
    var yPos = 30;
    var yStart = stepperMain._y;
    var yStep = 20;
    var xStart = 150;
    var textMC = _root.createEmptyMovie
Clip("textMC", 10);
    for (var i=1; i<=stepperMain.value;
i++) {
       var field = textMC.createTex
tField("sampleText"+i, i, xStart,
yStart+yPos, 200, 25);
       field.multiline = true;
       field.wordWrap = true;
       field.border = false;
       //Format Text
       formatText(field, i);
       var yPos += yStep;
   }
 }
}
createTextBoxes();

function formatText(field, number) {
 var textFormatter = new TextFormat();
 textFormatter.color = palleteAr-
ray[0];
 textFormatter.bullet = false;
 textFormatter.underline = false;
 textFormatter.font = "_sans";
 textFormatter.size = 12;
```

```
 textFormatter.bold = true;
 field.text = "Dynamic Text Creation
#: "+number;
 field.setTextFormat(textFormatter);
}

function removeDeadClips() {
 for (z in _root) {
    if (typeof (_root[z]) == "mov-
ieclip") {
      _root[z].removeMovieClip();
    }
 }
}
```

As before, we set up the change event handler. This time, it fires two functions (removeDeadClips() and createTextBoxes()) when the value of the NumericStepper component changes.

After setTheme(), the first function to be called at startup is createTextBoxes(), which takes the initial value of the stepperMain instance, and creates an empty movie clip. It then iterates through a for loop using the stepperMain.value property as the limiting value.

As we iterate through the for loop, a series of text fields are created and formatted using the formatText() function. This sets general properties of the text field, including color from the palletteArray array, and adds some simple dynamic text.

When the stepperMain NumericStepper instance value is changed, the removeDeadClips() function is called from the event handler. It removes the empty movie clip we created to hold the text fields.

Note: While only a single Movie Clip is created dynamically on the stage, and we could simply remove it

using removeMovieClip(), I chose to use the function in this example. This approach provides extra benefits in that, if I wish to create multiple Movie Clips later on, I can simply alter the function to suit my needs, and point it at a particular level or nested Movie Clip instance.

After the stage has been cleared up, we call the createTextField() function again, to create the empty Movie Clip and dynamic text fields.

2. Save your Flash document and preview your work.

It's as simple as falling off a bike! This example shows the basics of using the component as a starting point that triggers dynamic object creation.

Just the Beginning...

While the examples presented in this article have shown a few different uses for the NumericStepper component, the possibilities really are without limits. Use your imagination and the skills you've developed here, and you can create some interesting applications of the component. ∞

Steven Grosvenor is founder of Phireworx.com. His job title changes more often than his socks, but we're sure he does something important for a Managed Internet Security company in Manchester. A Ph.D. in Molecular Virology has served Steven well in the web industry, bearing no relevance at all, but this hasn't stopped him progressing so deep into code that nobody else understands. When not worked to death, his wife and young children ensure he can relax at home. stevengrosvenor@hotmail.com

Repurposing Fireworks Web Design Comps for Extra Dollars

Adapt Your Design Comps, Use PowerPoint Design Templates
by joanne watkins

have you ever wondered what to do with all those web site design comps - even the ones the clients rejected? Are you looking for new ideas and ways to use your design skills with Fireworks? It's time to break free of the browser and HTML restrictions, unleash your creativity, and let the graphic power of Fireworks take you in a new direction: PowerPoint Design Templates. It only takes a few modifications and enhancements to the original PNG files, export as a PNG, and import into PowerPoint. Even Fireworks-generated button images import into PowerPoint to use as customized bullet points, action buttons, and graphic design elements.

When you create a web site design, you consider several criteria such as monitor resolution, wide format, on-screen color, and navigation between pages. These also apply when you design a PowerPoint presentation. In a PowerPoint presentation the action buttons help navigate from one slide to another and from one presentation to another. You can also add hyperlinks to web sites. As you begin to explore the similarities, you can start to envision that it will take just a few changes to adapt your Fireworks web design comps and turn them into a PowerPoint Design.

Basic Web Site Design Comp

A design comp is a simple layout of a web site page that consists of basic shapes. You start with a canvas sized for a web page, add wire frames (simple basic shapes), develop a color scheme, and add a few basic elements.

Now, let's examine the process to create a simple design comp for a fictitious web site for a plant nursery company called JQ Plants and Gardens:

1. In Fireworks, start with a new canvas and the appropriate dimensions for a web page.
2. Now draw some rectangles and other basic shapes to block in your design (see Figure 1).
3. Develop a color scheme and add color to the basic shapes (see Figure 2).

This basic block approach serves as the basic design comp for the web page. The basic page can be modified for different pages of the site. For example, to create an interesting home page add a graphic to the top and add a 50% Grass texture to the main color block (see Figure 3).

In the next stage of the design comp, add more basic elements such as the corporate logo and the navigation button images (see Figures 4 and 5). Now the design comp is ready to transform to a design for PowerPoint.

Modify and Enhance for On Screen Presentation

At this point in the process, you can add artistic enhancements for the on screen presentation without browser or HMTL restrictions and make some slight modifications to the layout. Be creative and have fun.

1. Increase the canvas size for a full screen presentation and if needed adjust the placement of the graphic elements. The example canvas size is 1024 by 768 pixels.
2. Add enhancements such as a changing the solid rectangle for the side bar and the top horizontal rectangle to a gradi-

ent with less than 100% opacity.
3. Add a vertical gradient rectangle that gives it a three dimensional look (see Figures 6 and 7).

Export as PNG

When your design is complete, choose Optimize from the Window menu to prepare your file for the PNG export (see Figure 8). The numbers, 8, 24, and 32 refer to the color bit depth. Generally you would choose either 24 or 32 as most all modern projection systems can display 24bit. The PNG 32 includes the extra color channel to handle transparencies. The PNG export format is a good choice to use with PowerPoint for better presentation performance and the small file size. Now the file is ready to export using the Images only option in the Export dialog window. Be sure to save your original enhanced Fireworks native PNG files if you need to make any changes.

PowerPoint

For the fictitious company, JQ Plants and Gardens, the enhanced design comp for the home page will become the Title Slide template and the other enhanced design comp will become the Slide Template. Please note that the example in this article uses Windows XP PowerPoint 2003.

1. In PowerPoint, open a Blank Presentation and choose the Slide Master View.
2. On the Slide Master View Toolbar, click the Insert New Title Master button.
3. On the Title Slide Master select Background from the Format menu.
4. In the Background window, select Fill Effects from the pop-up menu (see Figure 9).

figure 1

figure 2

figure 3

figure 4

figure 5

figure 6

figure 7

Joanne Watkins is a featured an instructor for Hewlett Packard's Online Learning Center. She is also an Associate Professor in the Applied Graphic Design Technology department of Collin Community College and teaches in the Senior Adult Education Program of the Business Studies Division at Brookhaven College in Dallas, Texas. Previously at Macromedia she served as a DevNet technical editor, web producer, technical support engineer and technical editor for nine Macromedia Press Books about Fireworks and FreeHand. joannew@imagetechinfo.com

5. On the Picture tab, click select picture and locate the exported PNG file. Click OK. Click Apply.
6. Make adjustments for the position of the text placeholders.
7. Repeat the same process for the Slide Master (see Figure 10).
8. Close the Master Slide view.

When the design is complete, you can save the file as a PowerPoint template (.pot) in the Templates folder. Now the new template will appear in the Apply a design template window of the Task Pane (see Figure 11). Note: If you do not

see your design appear (the list is in alphabetical order) either browse to the file or quit the program and restart.

The new template for the fictitious company is now ready to use for an eye catching presentation that has the same look as their web site (see Figures 12 and 13).

As you explore this new direction from Fireworks web design comps to PowerPoint design templates, here are a few additional PowerPoint tips:

• For Best results use the exported PNG as a background image.
• Use your button images created in Fireworks for customized bullets and other graphics.
• To make a bounding box area of an imported PNG graphic transparent, click the opaque area with the Set Transparent tool located on the Picture toolbar.
• To assign PowerPoint actions over buttons of the imported PNG background image, select a blank action button from the Action Button menu located with the AutoShapes, draw a rectangle over the image, and change the fill and line to none.

• You can add hyperlinks to both imported graphics and ones created in PowerPoint that to go web pages.
• Add Fireworks animated GIFs to the presentation.

With these ideas, a few steps, and tips, you can turn those Fireworks web design comps into PowerPoint Design Templates for additional money makers or as an extra bonus for your web site clients. Now, go look for those comps, unleash your creativity with the power of Fireworks, and create unique, one of a kind PowerPoint templates.
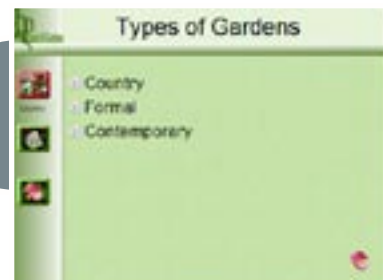
figure 8

figure 9

figure 10
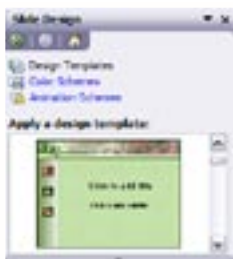
figure 11

figure 12

figure 13

Introducing the ColdFusion Enterprise Manager

New features in CFMX 7 for High Availability, Deployment and Multiple Instances
by tim buntel & geoff green

Wouldn't it be nice to get all the high-availability and scalability benefits of a cluster of multiple servers without the need to purchase multiple machines? How about being able to manage a number of isolated, performance tuned applications on one piece of hardware instead of taking up many machines in your server room?

ColdFusion MX 7 Enterprise Edition introduces the new Enterprise Manager, which will do those things and more. It makes load balancing, application isolation, and deployment management as easy as creating a datasource in the ColdFusion Administrator. In this article, we'll introduce you to the Enterprise Manager and explain how it will make managing and deploying ColdFusion applications easier and more powerful than ever before.

Background

Before ColdFusion 4.0, no high availability (HA) features were built into ColdFusion. One could use a hardware load-balancer to distribute requests to web servers, but this was an expensive option back in 1996. ClusterCats, introduced with ColdFusion 4.0, provided clustering and load-balancing right out of the box for customers who purchased ColdFusion Enterprise. With the introduction of ColdFusion MX, ColdFusion was once again without built-in HA features. The move to Java/J2EE brought many benefits, yet precluded the continued use of ClusterCats.

One of the main benefits of the move to J2EE is the ability of ColdFusion to take advantage of features provided by the underlying app server. Since ColdFusion is a certified J2EE application, it can run on any certified and many non-certified J2EE application servers. The two most prominent of these are BEA Weblogic and IBM Websphere. Both provide robust HA features and multiple instance support. They are, however, an additional expense above the price of ColdFusion itself and will require you to become familiar with their proprietary administrative functions.

As most of you reading the article already know, the underlying app server for ColdFusion is JRun, Macromedia's popular and easy-to-use J2EE server. JRun is a full-featured J2EE app server with clustering and multi-instance support. However, if you're not familiar with J2EE or the management/configuration of JRun, getting up and running with multi-instances and clustering may be a difficult task. In addition to a good understanding of JRun you will need to understand the basics of J2EE and the Java Runtime Environment (JRE). For the average ColdFusion user, there are several layers between you and a HA environment.

Our goal with ColdFusion MX 7 was to reintroduce HA features in a way that was intuitive and easy for the typical ColdFusion customer. The Enterprise Manager, found in the CF Administrator of ColdFusion MX 7 Enterprise, provides an easy-to-use GUI for configuring multiple servers and for managing clusters. ColdFusion users can get all the benefits of clustering and multiple instances right from the comfortable confines of the CF Admin without the need for any J2EE or JRE expertise.

Multiple ColdFusion Instances

Multiple instances (sometimes known as *multiple servers*) refer to J2EE servers' ability to deploy multiple copies of an application server to a single computer. Technically, a new JVM (Java Virtual Machine) is deployed per server instance. The benefit is simply that each new application server has its own process to use and a failure of one server cannot affect other servers; this is usually referred to as *process isolation*. In ColdFusion MX 6.1 Enterprise, it was possible to configure multiple ColdFusion servers, though it was a highly manual process and required the use of the JMC and knowledge of J2EE. With ColdFusion MX 7 creating multiple ColdFusion servers on a single computer is done via a simple interface in the CF Administrator. Each new ColdFusion server has its own CF Admin and is configured separately just as if this were a copy of ColdFusion running on a completely different physical computer. A single machine can now host any number of individual ColdFusion servers, bound only by processing power and RAM. Many new deployment options are available where individual applications, which were once directories, can now be deployed as separate processes. Clustering can also be used across physical machines as well as across instances deployed on the same machine or any combination of physical and logical servers.

Deployment Scenario

Until ColdFusion MX 6.1 Enterprise, a computer could have only one instance of the ColdFusion server process. This means that the file system was the only method for logically separating applications. Moreover, this type of partitioning gave you no flexibility with respect to

the runtime aspects of the ColdFusion server process. It was really just a way to keep an application's templates and other resources together in a directory structure, and there was no way to control an application's access to the ColdFusion runtime. With 6.1, it was possible to configure JRun to host multiple instance of ColdFusion, and with ColdFusion MX 7 this is even further simplified thanks to the new ColdFusion Enterprise Manager.

It's easy to imagine that many ColdFusion servers sitting on corporate intranets host a great number of applications, each with their own users and load demands. If you are running ColdFusion MX or earlier, there is no way to ensure that one busy app on a server doesn't use all of the available resources. If ColdFusion is configured for 10 simultaneous requests and one app is so busy that those threads are always servicing clients of that app, then other apps will seem slow or non-responsive. To fix this situation before the release of ColdFusion MX 7, it would have required a new piece of server hardware and a separate ColdFusion license. With multiple ColdFusion instances, this is an easy problem to solve without the added expense and maintenance of multiple physical servers. Simply create one ColdFusion instance with 10 available threads for the busy app and another ColdFusion instance to host all of the lower demand apps. Now all apps on the server will appear responsive and spikes in demand in one app won't affect the availability of others. As demand grows in particular sections of your website, you can easily create new instances to meet this demand and get a lot more value out of a single piece of server hardware and a single ColdFusion MX 7 Enterprise license.

HA in Detail

Below you'll find a high-level description of some of the HA features available in ColdFusion MX 7 Enterprise. This isn't meant to be an exhaustive analysis of how to use these features in a particular environment. However, it should help you understand the basics of HA computing. The multi-instance feature alone should be enough to justify the move to ColdFusion MX 7 Enterprise, especially for sites that host multiple applications on a single physical server. And if your

server environment hosts mission-critical applications where 100 percent uptime is the goal, then the clustering features of ColdFusion MX 7 are a must-have.

Load Balancing

Load balancing is the simpler of the two HA methods. It refers to the act of distributing requests across *n* number of computers, based on an algorithm for determining load. This can be as simple as round-robin (which isn't technically *load* balanced) where requests are sent to each computer in turn or more complex methods based on, for example, CPU utilization. In this case, some method for reporting load on a CPU has to be communicated back to the load-balancing software. The clustering features in the ColdFusion Enterprise Manager can be configured to act as a simple load balancer, handing out requests to the cluster members. There are three algorithms to choose from: round robin, weighted round robin, and random round robin. Strictly speaking these are not load balanced. However you can use weighted round robin (WRR) to simulate a more complex load-balancer. WRR allows you to apply a weight to each machine in the cluster so that in a mixed hardware environment, where some machines may be more powerful than others, the cluster can be configured to send a proportionally higher volume of requests to those more robust machines. Random round robin simply routes requests randomly among the cluster participants.

Clustering

Load balancing is not usually "application aware." Its job is simply to route incoming requests. The point at which your cluster becomes application aware, and not just a simple load balancer, is when you configure your cluster to replicate session data. The ColdFusion Enterprise Manager's Cluster Manager takes care of copying session data to each node in the cluster so clients of your application will not experience any change in the event of a server failure. Sessions can move freely among members of the cluster or be configured to be "sticky" where each particular client will be routed to the same node for the duration of the session. With either

sticky sessions or not, replicated data will always be up to date on every node at the moment it is requested in your app. Another thing to note is that there are *no changes in the way you use the session object in your CFML code to take advantage of session replication*. The Cluster Manager works under the covers and is entirely invisible to your application. An important aspect of session replication to keep in mind, however, is that since this feature is not part of the J2EE spec the behavior of session replication is not consistent across the range of app servers that support it. Some, like WebSphere, explicitly document the fact that they will only replicate instances of classes found in the JDK. This means that if you are developing a ColdFusion application using the session object that may at some point be deployed on an app server other than ColdFusion's underlying JRun, you should be conservative about the type of objects you store in session.

WARs, EARs and Sourceless Deployment

A ColdFusion Application contains an application's resources, such as ColdFusion Markup Language files (CFM), ColdFusion Components (CFC), and static resources such as HTML pages and image files. The application may also require configuration information such as details about data sources. Managing and deploying an application with all of those resources and settings, for instance moving an application from a development environment to production, can be difficult. The J2EE specification has a solution; a WebARchive (WAR) file. This is a single file used to package an application for easy deployment and it is the standard way to deploy applications in J2EE environments. The archive contains all of the application's logic, resources, and the ColdFusion runtime required to run it.

ColdFusion MX 7 automates the task of creating these J2EE archives. All versions of ColdFusion allow you - from the ColdFusion Administrator - to create a WAR or EAR file from a specified set of application directories. You can configure data sources to be included in the archive, chose to include or exclude the ColdFusion Administrator in the packaged application, include or exclude

> "Our goal with ColdFusion MX 7 was to reintroduce HA features in a way that was intuitive and easy for the typical ColdFusion customer"

external components (such as COM support), and more. And just like the Enterprise Manager, the J2EE packaging feature of ColdFusion is as simple as creating a datasource.

ColdFusion MX 7 also supports packaging and deploying ColdFusion applications as Enterprise Archives (EAR files) – another Java 2 Enterprise Edition (J2EE) application archive format. Whether your choose EAR or WAR is entirely driven by the deployment environment (i.e. Tomcat is a servlet engine and only accepts WARs).

It's important to point out that you can *create* a J2EE package of an application with any version of ColdFusion (Developer, Standard, or Enterprise), but they must be *deployed* onto a server running ColdFusion Enterprise edition. When you create the package, you will need to enter a valid licensed Enterprise serial number for the server on which the application will be deployed. If you leave the serial number out, the application will be deployed in Developer mode (IP restricted).

Another new option we developed for ColdFusion MX 7 is the ability to write a ColdFusion application and deploy it without distributing the unencrypted, human-readable source code. This is presented as an option when you are creating your EAR or WAR file. If this box is checked, the CFML that makes up your ColdFusion application will not be included in the archive, but your application will still run. This gives your intellectual property more protection than was ever possible before in ColdFusion. The Java byte code is not human readable. Even if the byte code were decompiled, it would still only result in Java source code, which is still very far from CFML.

This is achieved by compiling the ColdFusion files to Java byte code. If you expand the archive you will still see your applications directory structure and even all of the files that make up your application - images, style sheets, etc. However, if you open any of the CFML templates you will see that they do not contain CFML, but instead contain Java byte code. Give it a try; you'll see byte code completely obscures your application logic.

Summary

ColdFusion has had features for many years that allowed you to create and deploy high performance scalable web applications in even the most demanding environments. However, with ColdFusion MX 7, we've made many of the high availability, scalability, and management features more powerful and much easier to use. If you aren't familiar with J2EE, you can still take advantage of many of its HA strengths by using the new ColdFusion Enterprise Manager. If you are in a standardized J2EE organization, you can use the new J2EE package and deployment features to make ColdFusion fit more easily alongside any other Java application. And, in either case, the job of managing and deploying these important applications is done in the kind of intuitive and highly productive ways that you've come to expect only from ColdFusion.

Tim Buntel is senior product manager for ColdFusion at Macromedia. tbuntel@macromedia.com

Geoff Greene is Director of Software Engineering, Bullhorn. ggreene@bullhorn.com

captivate**primer**

Tips, Tricks & Techniques

Now that Captivate has been on the street for a few months and I have been out there "yacking it up," the product has moved from novelty to production tool in rather short order. Once that happens, there are the subsequent, "How do I...?" questions that inevitably crop up. This article deals with a five of the more common ones that I have encountered and shows you how to deal with them.

by tom green

They are:
1. Fitting the application interface to be captured into a defined capture area.
2. Adding a Voiceover script to a Captivate movie
3. Moving the Control bar off of the captured interface
4. Creating a custom Control bar
5. Adding streaming video to Captivate.

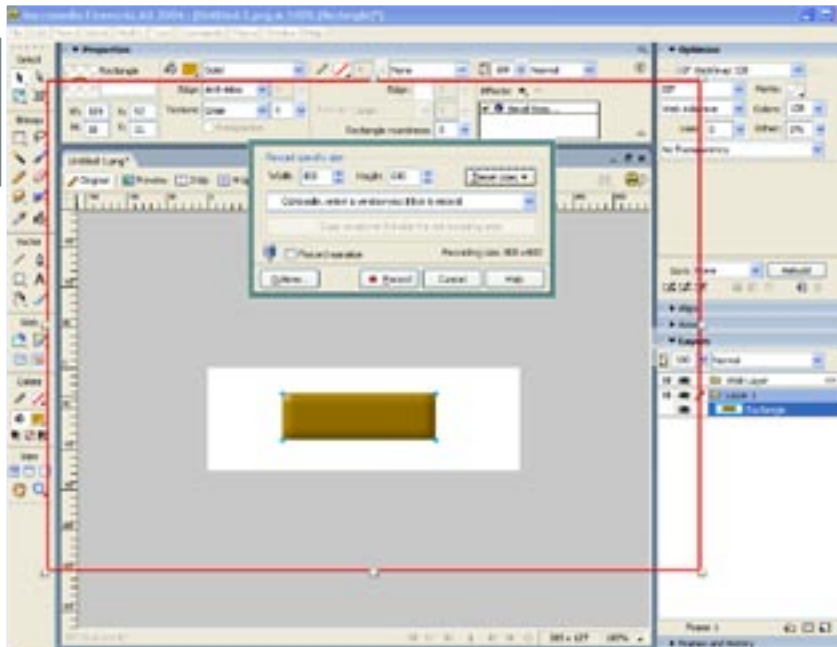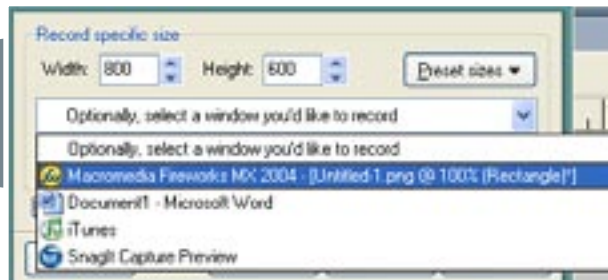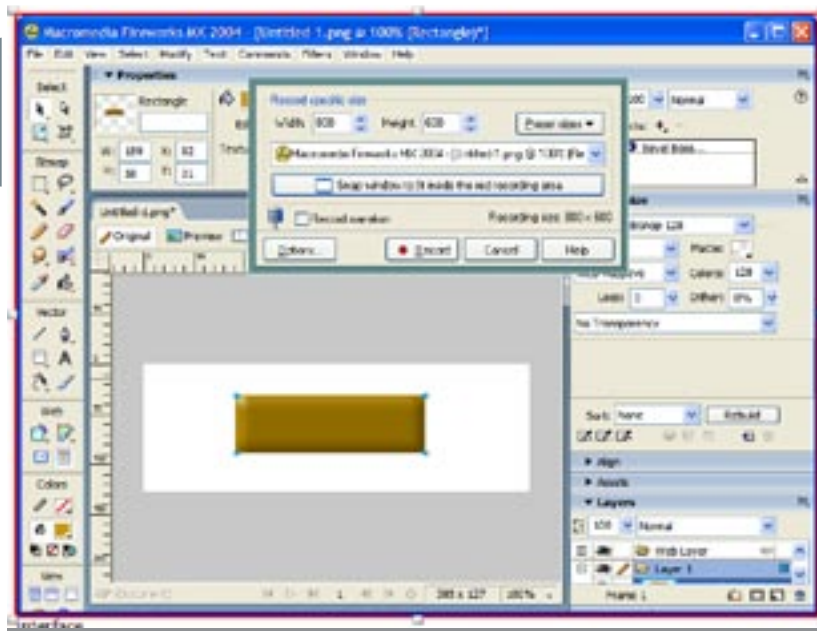Here, then, is a collection of Captivate Tips, Tricks and Techniques.

figure 1

figure 2

figure 3

#1: How Do I Fit an Application's Interface into that Red Capture Area?

This is one of more common questions I am asked. Inevitably, you want to capture to a size that won't result in a huge file. The easiest way of managing file size it to keep the physical capture area to a size that is smaller than the interface that is currently open. Here's how:

1. When Captivate opens, select Custom Size from the New Movie options dialog box.
2. As you can see in Figure 1 there is a lot more Fireworks interface than there is capture area. You can change the size of the capture area by either selecting a preset from the pop down menu or moving the handles in or out to decrease or increase the capture area.
3. Select the application you will be capturing from the list in the Record dialog box, In Figure 2, I select Fireworks.
4. Click the "Snap window to fit inside the red recording area" button and the Fireworks interface will shrink to fit the dimensions of the recording area.

#2: I Can't "Wing It" with Audio. How Can I Create a Script That I Can Read While I Record the Audio?

Very few of us have the talent that allows us to record a "voiceover" narration when we click the "Record" button in the Audio section of Captivate. In this case you have a couple of choices. The first is to break out your word processor and simply type out the script for each slide. But there is another method:

1. Open your Captivate movie.
2. It doesn't me slide thumbnail to open the slide's context menu.
3. Select "Properties" from the pop down to open the Slide Properties dialog box. Click the Notes button to open the Slide Notes input window.
4. Enter the text you will be using for the narration, Figure 4, and click OK to close the Slide Notes window. Don't close the Slide properties. Dialog box just yet.
5. Connect your microphone and click the Audio tab in the slide properties dialog box to open the slide's Audio properties.
6. Click the "Record New" button to open the "Record Audio" dialog box.
7. Select "View Script" and click the Notes

radio button. Your narration will now appear in the script area, as shown in Figure 5, and you can read the narration. Clicking the "Captions" button will simply place the caption text in the script area.

#3: The Stupid Control Bar Always Covers Areas of the Interface in the Movie. Is There a Way to Get it Out of the Way?

Absolutely. It can be frustrating to have the control bar cover critical buttons or information in the slide. Though you can move it to various areas of the screen, the chances of it covering something important are almost 100%. Here's how to contend with this one:

1. Select Movie> Resize movie to open the "Movie Resize" dialog box.
2. Deselect Maintain Aspect Ratio. If you leave this selected the movie will distort.
3. Enter a new height value. I have found most of the controllers are 30 pixels high. In this case I will add 40 pixels to the height value. This will light up the "If movie size is larger..." area.
4. Select "Keep movie the same size and fill background with color".
5. Select a background color if you don't want to go with the white default.
6. Select "Top Left" from the "Position Movie" pop down menu.
7. Deselect "Rescale captions, highlight boxes and other objects." Though not necessary, I always do this as a sort of insurance.
8. If your values match those shown in Figure 6, click the "Finish" button and preview your movie. The controller will be sitting under the capture area.

#4: I Love All the Controllers That Ship with Captivate But I Have a Problem: My Client Wants to Use a Custom Controller. Can I Do This?

Though the process is a bit complicated, the answer is a resounding "Yes." What you will need to do is to create all of the pieces for the controller in an image editing application, such as Fireworks and also a composite image that Captivate will use to construct the controller. Before you get going, keep in mind the rules are rather rigorous:

- All buttons must be the same width and height.

- All buttons can only have an up and down state. There is no "over" state in Captivate.
- All elements, including the composite image, must be saved as 8-bit BMP graphics.
- Playback controllers can only have the following buttons:
  - Play
  - Back
  - Forward
  - Pause
  - Exit
  - Rewind
  - Info

- Playback controllers have two copies of all of the elements in the controller plus a composite, flattened image of the controller.
- All buttons must use the following naming convention: controlstylename-buttonnamebuttonstate.fileextension. For example, if I create a style named MXDJGray then the play button for this controller would have two buttons
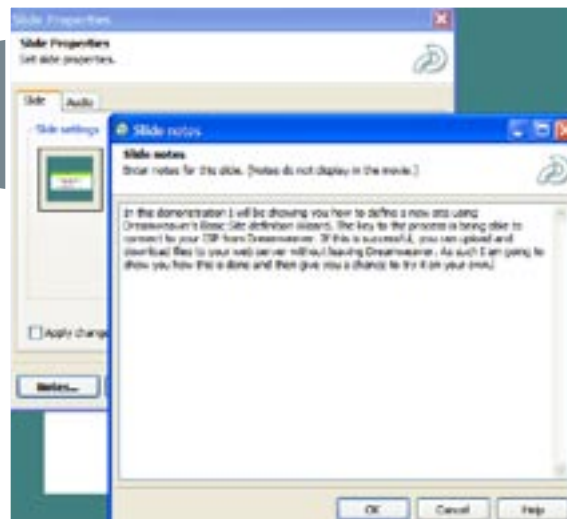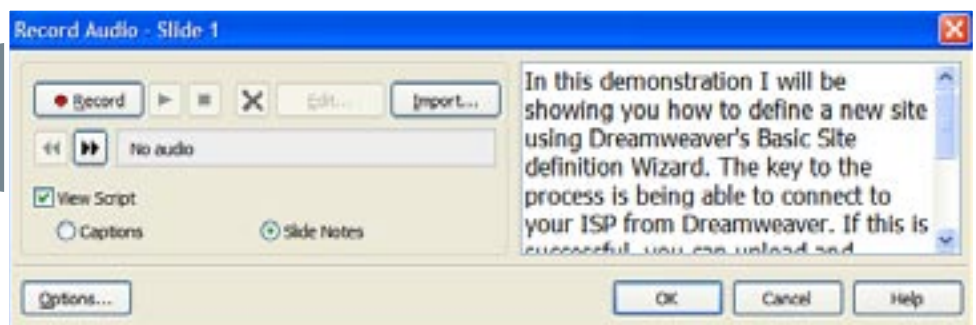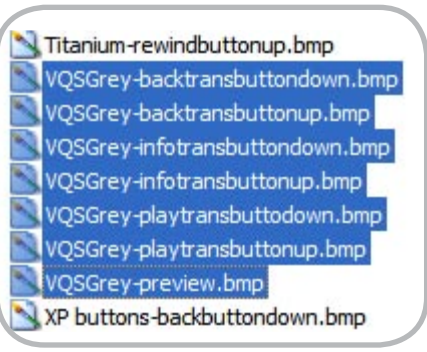

figure 4

figure 5

figure 6

named MXDJGray-playbuttonup.bmp and MXDJGray-playbuttondown.bmp. The composite image, showing the controller with the buttons would be named MXDJGray-preview.bmp.

- To remove the solid background stage color for the buttons or other pieces, add the word "trans" to the file name. For example, the Play button in the above example is circular. This means there will be a bit of the stage color showing when the image is saved as a BMP. In this case the image would be named, MXDJGray-playtransbuttonup.bmp

Here's how:

1. Use the layers in Fireworks MX 2004 or other image editor create the buttons and other pieces for the controller.
2. Select each of the pieces- two button states for each button and the base for the controller- and copy and paste them into separate documents.
3. Save each element of the controller using the naming conventions outlined above.
4. Open the folder where all of the button elements just created are stored and copy all of the files in the folder.
5. Navigate to C: \Programs\ Macromedia\Captivate\Gallery\ Playback Controls. When you have the folder open, paste the files into the folder as shown in .
6. Open the Captivate movie that will use the new controller.
7. Select Movie>Preferences to open the Movie Preferences dialog box.
8. Select the Playback Controls tab and select the controller from the Styles pop down as shown in Figure 8.
9. Click OK and the new controller will appear in the Preview window when you test the movie.

#5: I Really Like the App But Surely There Must Be a Way to Add Video to the Captivate Movie Without Embedding the FLV Into the .SWF?

This is one of those undocumented features of the application. The standard response is that video can be added if you import it directly into the Captivate movie or create an FLV and embed it into a Flash .swf that is subsequently placed in Captivate. Both options result in a rather hefty .swf that will require the entire video to load into the browser. If the video is long, your user is in store for a rather long wait in front of his or her monitor.

Still, there is a place for embedded video. If the Captivate movie is destined for desktop playback or CD playback, using an executable, then feel free to "bulk up" the movie.

For those of you wishing to use video in a web-based Captivate movie, here's how to have the best of both worlds – small file size and quick playback. Simply stream the video from your web server, using a progressive download, into the Captivate movie.

You can do this because streaming

video was introduced the Flash in version 6 of the Flash Player. This just happens to be the Player version used by Captivate. Before you jump in though, be aware that you can't use ActionScript 2, introduced in Flash MX 2004, to code the playback and the best playback results are obtained if the video is short...less than four minutes in length.

Follow these steps to add a streaming video to your Captivate movie:

1. Before you start, create a folder in the local copy of your site and place the FLV file to be used in that folder.
2. Open Flash MX 2004 and create a new movie. Add two new layers and name them "Actions", "Controls" and "Video."
3. Open the Library and select "Video" from the Library pop-down menu. A small camera named "Embedded video" will appear in your library.
4. Drag the Video from the Library to the Video layer on the Flash stage and give it the instance name of "myVideo_vid".
5. Select the video instance on the stage and, in the Property Inspector set the width and height values to match those of the video. In the case of Figure 9, the video being used is 320 pixels wide and 240 pixels high.
5. Select Window>Other Panels>Common Libraries> Buttons to open the Buttons collection that ships with Flash.
6. In the Library that opens, open the Circle buttons folder and drag a copy of the Play and the Stop buttons to the Controls Layer.
7. Select the Play button and give it the instance name of "play_btn". Name the stop button "stop_btn".
8. You don't need a huge stage. Click once on the stage and click the Stage Size button on the Property Inspector to open the Document Properties dialog box. If you click the Contents button and then click OK the stage will shrink to fit the objects on the stage.
9. Select the Actions layer and press the F9 key to open the ActionScript panel. When it opens enter the code shown in Figure 10. Be sure to enter the name of your FLV file to exactly match the file name in the folder.
10. Select File>Publish Settings and when the Publish Settings dialog box opens, click the Formats tab.
11. Deselect the HTML option and click the folder in the Flash .swf area to

MX

open the "Select Publish Destination" dialog box.

12. Navigate to the folder containing the FLV file, name the file and click Save.
13. Click the Flash tab to open the Flash publish settings.
14. Select Flash Player 6 from the version pop down and ActionScript 1.0 from the ActionScript version pop down menu. If you don't do this, Captivate will kick out an alert telling you it can't use the file when you try to place the .swf in the Captivate movie.
15. Click the Publish button to publish the .swf file. When that process finishes, save the .Fla file and quit Flash.

You can also publish the file and then press Control>Enter (PC) or Command-Return (Mac) to test the file. If you can't get the video to play be sure the instance

names are spelled correctly, that the .swf is sitting in the same folder as the FLV and that the name entered for the FLV, precisely matches the one sitting in the folder.

Now that the Flash file is completed, it now has to be added to the Captivate movie. Here's how:

1. Open the Captivate movie where the video will be placed and open the slide where the video will appear.
2. Select Insert>Animation and when the "Open" dialog box appears, navigate to the folder containing the Flash .swf. Select the .swf and click the "Open" button.
3. When the "New Animation" dialog box opens, select "No Transition" from the Effect pop down.
4. Click OK and the Animation, as shown in Figure 11 will appear on the slide.
5. Preview the movie and the video should play. If the slide ends before the video, don't extend the timeline. Instead add a "Next" button to the slide. In this way the movie won't move forward until the button is clicked.

figure 11

```
1   nc= new NetConnection();
2   nc.connect(null);
3
4   ns= new NetStream(nc);
5
6   myVideo_vid.attachVideo(ns);
7   ns.pause("Crockard.flv");
8
9   stop_btn.onRelease= function() {
10      ns.pause("Crockard.flv");
11  }
12
13  play_btn.onRelease = function() {
14      ns.play("Crockard.flv");
15  }
16
```

6. Publish the .swf to the same folder as the FLV, Save the movie and quit Captivate.

From here you can place the Captivate .swf in a Dreamweaver page and test it. If it works, you can then upload the file to your web server as shown in Figure 12. The file is posted to the address shown in the browser if you want to see it for yourself.

There you have it, the answers to some of the more common "How Do I...?" questions around Captivate. In this article I covered a rather broad spectrum of techniques ranging from the simple – fitting an interface into a Capture area – to the complex – creating a custom control bar and adding streaming video to a Captivate movie.

I will be spending some time in the next few months "yacking up" Captivate and will be back later this year with another article that answers the more common "How do I...?" questions that will be asked. See you then.

Tom Green describes himself as 'teacher, author, chief cook and bottle washer.' He is an instructor at Humber College's School of Media Studies in Toronto, and is also the author of 'Building Web Sites with Macromedia Studio MX' and 'Building Dynamic Web Sites with Macromedia Studio MX 2004,' both published by New Riders. tgreen17@cogeco.ca

figure 12
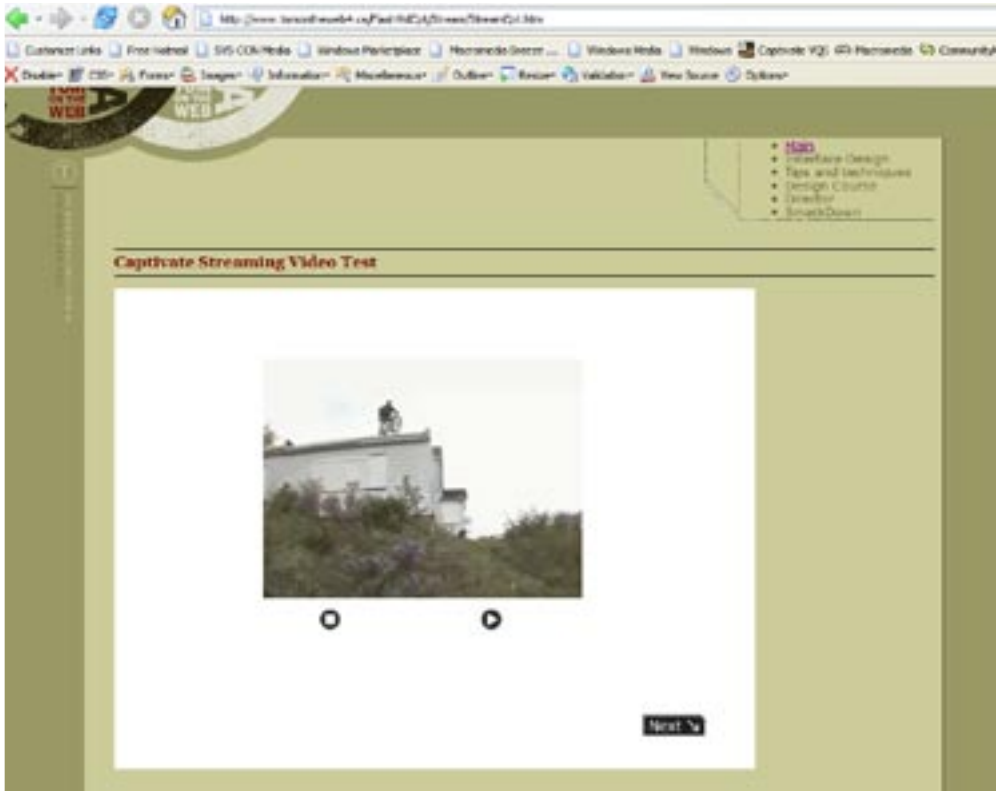
figure 13
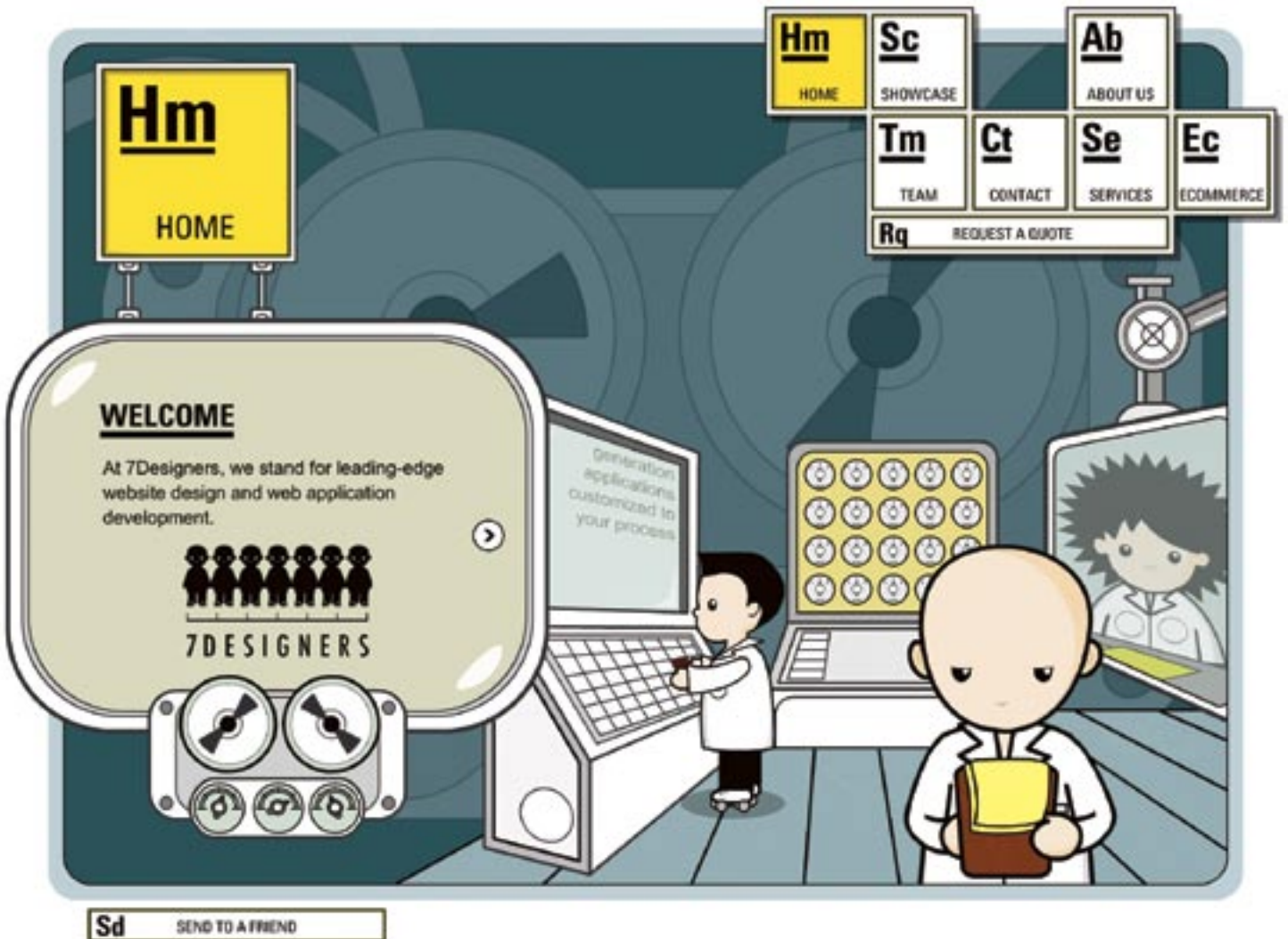

Captivate Streaming Video Test
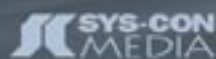
7Designers

7Designers is the creative development division of EcommercePartners, a comprehensive technology and marketing services provider. Since our inception in 1997 we have delivered Internet and e-commerce solutions that generate new revenue opportunities and streamline existing operational systems for our clients.

Through our proprietary development process and market-tested software, 7Designers provides a full spectrum of cost-effective services designed to help our clients succeed online, including Web site development, custom application development, and online marketing.

7Designers focuses on the needs of the individual client. Every assignment is equally important to us – from the single online advertisement to the largest e-commerce application development – you're guaranteed our highest level of service.

Our staff is composed of experts in visual design, interface design, Internet development and programming, as well as search engine optimization and Internet strategy.

Contact us to discuss how we can help you identify potential clients, target them profitably, and provide a superior customer service experience.

macromedia

There are workarounds for a mediocre intranet,
but they still waste time and money.

Meet the Macromedia Web Publishing System. Unlike the average CMS,
this works. By allowing business users to publish content, IT bottlenecks
are eliminated. Users can simply point and click to update specific pages
or sections–while other areas remain protected. So everyone in your
organization can share critical information quickly and easily, and you have
an intranet that actually lives up to its potential.

Learn how to make your intranet work for you, not the other way around:
macromedia.com/go/webupdate