

# MX

## developer's journal

november 2003 www.mxdj.com



FOR MACROMEDIA MX  
DEVELOPERS & DESIGNERS

**SPECIAL  
PREMIER  
ISSUE**

RETAILERS PLEASE  
DISPLAY UNTIL  
FEBRUARY 28, 2004

 **FLASH**  
From Design to Reality

 **FIREWORKS**  
Turbocharge Your Toolbar

 **FREEHAND**  
Vector Vortex

 **COLDFUSION**  
ColdFusion in the Trenches

 **DIRECTOR**  
Lingo: stickmancan



cascade  
in  style  
with  
dreamweaver



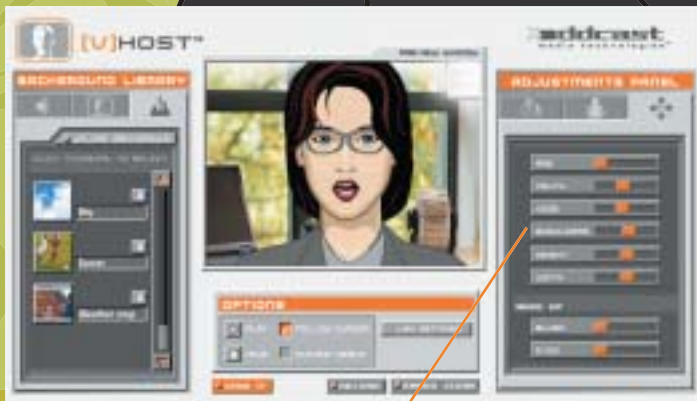
\$5.99US \$6.99CAN

34>



0 71486 02978 6

# CREATE TALKING CHARACTERS IN A FLASH.



## Main Features:

- Export as swf
- Lip Synchronization
- Dynamic Text-To-Speech (with no Plug-in)
- Record by mic or telephone.
- Advanced ActionScript and JavaScript APIs.
- Advanced content management and workflow features
- Usage and performance reports
- Personification features:
  - Set Age
  - Set Color (Skin, Eyes, hair)
  - Set Size (Shoulders, head, nose, mouth)
  - Set Clothing
  - Set Accessories
  - Set Background
  - Set Audio

**oddcast**  
media technologies™

The [V]Host Studio is a tool that enables professionals to design and deploy interactive talking characters in minutes. Output as a .swf file, the characters can be embedded in any Flash movie. Record a voice or use our Text-To-Speech features to read dynamic text, in 14 different languages, then watch the characters breath, talk and come to life. Non-programmers can learn to use the [V]HostT studio in minutes. Developers can use our ActionScript and JavaScript API's to make the character interact with users through wide range of actions and behaviors.

Winner of the following awards:

- European Excellence in Digital Media Award 2003
- Silver Medal, Spin Awards 2003
- Bronze Medal, I.D. Magazine, Interactive Design 2003
- Best Application award, FlashForward 2002

[www.oddcast.com](http://www.oddcast.com)

**Call (212) 375-6290**

**for developer's kit**

# AMERICAN MEDIA

<http://www.americanmediatraining.com>

800 278 7876

# TRAINING

Authorized training partner since 1995

Experienced instruction from beginner to expert level

Clients include movie studios, government agencies, fortune 500, military

The first step to certification

Always competitive pricing

Free technical support

Custom courses - tailored to suit

On-Site

Los Angeles

Denver

Las Vegas

Worldwide

Flash

ColdFusion

Dreamweaver

Fireworks

Contribute

Authorware

Director

Coursebuilder



## Cascade in Style

*Dreamweaver MX 2004's new CSS abilities really shine*  
by stephanie sullivan



## From Design to Reality

*Taking a designed interface and making it work in Flash using application states*  
by jesse r. warden



## Turbocharge Your Toolbar

*Creating navigation bars in Fireworks*  
by charles e. brown

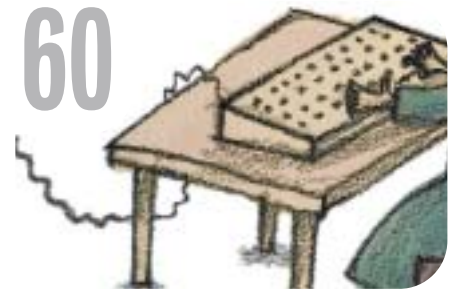
16



38



60



12

## A First Look at Dreamweaver MX 2004

*Meeting the standards that Web developers look for*  
by dave mcfarland

30

## Pushing Data out to the Flash Client with XMLSockets

*An essential for games, presentations, and RIAs*  
by matthew david

66

## The Fireworks Fuzzy Edge Quick Knock Mask

*Keep your edges in line*  
by linda rathgeber

22

## Can You Be Seen... or Heard?

*Make sure your Web site can be found by everyone*  
by bruce lawson

32

## The Flash MX 2004 Professional Data Architecture

*Crafting rich Internet applications*  
by owen van dijk

23

## Dreamweaver Extension of the Month

*Pure Upload*  
by dave mcfarland

48

## Customizing Components in Flash MX 2004

*Style and themes: powerful new tools*  
by peter elst

52

## Flash Remoting vs Flash Player with SOAP

*Flash Remoting offers stability, performance*  
by joe orbman

56

## The Executable Internet

*ASP.NET and Macromedia's Flash Remoting combine to offer a compelling rich-client interface*  
by ryan moore



### Vector Vortex

*FreeHand, Fireworks, and Flash:  
a combination that will save you  
time and money*  
by tom green



### ColdFusion in the Trenches

*CFMX is RX for midwest health  
care organization*  
by ben forta



### Lingo: Stickmancan

*Mixing Flash and Director*  
by niqui merret

70



80



90



76

### Swift Chart from GlobFX

*Professional charting for your  
ColdFusion applications*  
by jason clark

76

### Robust E-mail Notification with ColdFusion

*Using CFMAIL in new and  
exciting ways*  
by scott van vliet

7

### From Great Experiences to Great Business

*The key feature in  
MX is... you*  
by norm meyrowitz

8

### Another New MX Tool: *MX Developer's Journal*

*New technologies mean new  
things to learn, and MXDJ is here  
to help you*  
by jeremy geelan

36

xile  
*Cartoon*  
by louis f. cuffari

98

vanguard  
*Platinum RE*  
by dave allbritton

# How Flexible is YOUR Shopping Cart?

macromedia  
ASSOCIATE PARTNER



© 2003, ComCity® Corp.

## Our E-Commerce Solution can adapt to ANY Product and ANY Customer

### Sure, you could build your own shopping cart from scratch, but should you...

ComCity® Corporation has been building shopping carts and hosting E-commerce solutions since 1995. Our experience extends from small shops to multi-server, multi-geographic Enterprise E-commerce.

- **ISP Friendly™** script-based checkout system without complicated dlls, available for Windows (ASP) as well as UNIX (PHP) hosting providers – host with 99.9% of all ISP's without special server installation support.
- **Dual product mode** allows designers to create a cart quickly in *static* mode or use *dynamic* mode for script-based flexibility.
- **Tight Dreamweaver MX integration** allows developers to use Database Server Behaviors to modify Catalog design with ease and flexibility.

### A Shopping Cart *Foundation* for Dreamweaver E-commerce Developers

- Intuit® Quickbooks Integration
- Review orders with any browser
- Customer Control Panel
- Marketing Tools
- Sales Summaries
- Market/Auction based pricing
- Online self-paced training

Mention Ad: PUP1003-10 and receive 10% off the price of SalesCart PRO.

Cannot be combined with any other offer.  
Expires 11/31/03

# SalesCart™

The first **Design-Tool E-commerce** software solution in the world™



For more information, visit [www.salescart.com](http://www.salescart.com) or call 1-888-826-6248 (Toll Free)

**Group Publisher** Jeremy Geelan  
**Art Director** Louis F. Cuffari

**EDITORIAL BOARD**  
**Dreamweaver Editor**  
Dave McFarland  
**Flash Editor**  
Jesse Warden  
**Fireworks Editor**  
Kleanthis Economou  
**FreeHand Editor**  
Louis F. Cuffari  
**ColdFusion Editor**  
Robert Diamond  
**Director Editor**  
Niqui Merret

**INTERNATIONAL ADVISORY BOARD**  
Jens Christian Brynildsen **Norway**,  
David Hurrows **UK**, Joshua Davis **USA**,  
Jon Gay **USA**, Craig Goodman **USA**,  
Phillip Kerman **USA**, Danny Mavromatis **USA**,  
Colin Mook **Canada**, Jesse Nieminen **USA**,  
Gary Rosenzweig **USA**, John Tidwell **USA**

**EDITORIAL**  
**Executive Editors**  
Gail Schultz, 201 802-3043  
gail@sys-con.com  
Jamie Matusow, 201 802-3042  
jamie@sys-con.com

**Editors**  
Nancy Valentine, 201 802-3044  
nancy@sys-con.com  
Jean Cassidy, 201 802-3041  
jean@sys-con.com  
Jennifer Van Winckel, 201 802-3052  
jennifer@sys-con.com

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

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

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

**Worldwide Newsstand Distribution**  
Curtis Circulation Company, New Milford, NJ

**List Rental Information**  
Kevin Collopy: 845 731-2684,  
[kevin.collopy@edithroman.com](mailto:kevin.collopy@edithroman.com),  
Frank Cipolla: 845 731-3832,  
[frank.cipolla@epostdirect.com](mailto:frank.cipolla@epostdirect.com)

**Promotional Reprints**  
Carrie Gebert, 201 802-3026  
[carrieg@sys-con.com](mailto:carrieg@sys-con.com)

**Copyright © 2003**  
by SYS-CON Publications, Inc. All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy or any information storage and retrieval system, without written permission.

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

SYS-CON Media and SYS-CON Publications, Inc., reserve the right to revise, republish, and authorize its readers to use the articles submitted for publication. MX and MX-based marks are trademarks or registered trademarks of Macromedia, in the United States and other countries. SYS-CON Publications, Inc., is independent of Macromedia. All brand and product names used on these pages are trade names, service marks or trademarks of their respective companies.

**MX**  
developer's journal

# From Great Experiences to Great Business

*The key feature in MX 2004 is...you*  
by norm meyrowitz

Macromedia and its developer community have a unique bond based on a mutual fascination with what the Internet could be, a desire to create great experiences, and a need to keep pushing the limits. The active dialogue that takes place between Macromedia and its customers pushes both sides forward.

Together, we are building a world where every digital interaction – whether in the living room, the office, the beach, or the car – is a smart, simple, efficient, and engaging experience. We believe it will happen. It's who we are and why we are here. Macromedia, along with you, our customers, will help make the future better, one interface at a time. Together, with our software and your expertise, we are going to convert great experiences into great business.

Macromedia collaborates deeply with our customers, because you live and breathe our tools and are in the best position to challenge us to make them better. We respond with the features you ask for and try to anticipate trends and add features that will inspire you to take your experience to new heights. You build things that amaze us, and challenge us to put even more features in future versions of the products.

**MX Developer's Journal** is an important venue in which we can converse, and a testament to both the popularity of the products in Macromedia Studio MX 2004 and your expertise.

This first issue of *MX Developer's Journal* launches at the same time that you will begin to experiment with our new MX 2004 product line, providing a perfect place to learn the tips and tricks of the new MX 2004 products together.

The MX 2004 products are evolutionary in their approach, as you will see when you dig deeper into each one. Dreamweaver MX 2004 offers enhanced Cascading Style Sheets (CSS) support; cross-browser validation; built-in graphics

editing; and updated support for ASP.NET, PHP, and ColdFusion server technologies. Flash MX 2004 has new timeline effects, easy import of Adobe Illustrator and PDF content, and CSS support. Fireworks MX 2004 runs faster and delivers a broad array of new design tools and effects. Flash MX Professional 2004 enables advanced content, applications, and video experiences to be created using a forms-based metaphor as an alternative to the traditional timeline development.

We've also expanded the MX universe beyond products with MX Elements and Halo. MX Elements are a series of interface building blocks with a new generation of interactive design patterns embedded – making it faster to build interfaces. These elements bring together a range of technologies including components, templates, CSS, and behaviors. MX Elements, by default, have a distinctive new look and feel we call Halo – we think it will make the digital world a brighter, friendlier place.

Each MX 2004 product provides a rich palette from which the best online experiences will be drawn. Each product is built to help you get more done, to automate routine tasks, and to free your time to create. We think we've made major strides with these releases and look forward to hearing from you about what you'd like to see us take on next time. We're already on the road to planning for the next version – let us know where we should meet up next time.

With this amazing set of tools, the exhaustive information on [macromedia.com](http://macromedia.com), the vibrant community sites, and great new resources such as this magazine, our customers have a broad palette of information from which to create great digital experiences that fuse our community and shared knowledge with their creativity and passion.

We can't wait to see what you're going to do with these new products. ☺

*As president, Macromedia products, Norm Meyrowitz oversees development and marketing for all Macromedia product divisions. Meyrowitz is a recognized authority on the evolution of Web development software and media technology for the Internet. Through his experience at Macromedia, Meyrowitz has overseen the teams creating a vast array of multimedia and Web development products, including Director, Shockwave, Dreamweaver, and Macromedia Flash. [nmeyrowitz@macromedia.com](mailto:nmeyrowitz@macromedia.com)*

# Another New MX Tool: MX Developer's Journal

*New technologies mean new things to learn, and MXDJ is here to help you*  
by jeremy geelan

For many of us who comment regularly on the swirling, whirling Internet technologies space, "Web publishing" has never been a particularly satisfactory term.

If the metaphor of "publishing" was the correct one for the business of making things public on the Web, then how about the rest of the lexicon of publishing, such as "author," "editor," "jacket designer," and so on? It was almost as if half the elements of the publishing ecosystem were missing.

So it was a welcome relief back in 1998 when Kevin Lynch, now Macromedia's chief software architect (then, vice president and general manager for Internet and multimedia authoring tools), championed vector design, and established himself as a key figure in Macromedia's contribution of the open .swf vector format to the Web. It heralded a new focus on one of the concepts that had been missing up till then, namely "Web authoring."

Some people take to newly breaking technologies the way surfers take to newly breaking waves: HTML, XML, XHTML, dHTML, and Web services were to Lynch as water to a fish, and before long he was swimming among such acronyms. First he defined and led the development of Dreamweaver. Then, as president of products, he took on wider responsibility for helping to make a "family" of Macromedia's various award-winning software and Web authoring solutions.

It was Lynch's conviction that creating practical, powerful, and enjoyable solutions would – by transforming the "Web authoring" experience – also transform the end-user Web experience (the "reader," to continue the metaphor).

In a move that foreshadowed the tools and technologies that have now become "MX" tools and technologies, Lynch broke down Macromedia's strategy for better Internet communications into three parts: emotion, context, and interaction. Macromedia pledged to considerably improve each part over what was acceptable at the time. "We're raising the level of human interaction across the Internet," he declared.

Bold words. But to Web developers, as to everyone else, actions speak louder than words. Macromedia's mission – now being continued by Norm Meyrowitz, who succeeded Lynch as president of products – has manifested

itself as action  
after action.  
New versions of the company's Flash animation tools, Dreamweaver Web



**SYS-CON MEDIA**  
**President & CEO**  
Fuat Kircaali, 201 802-3001  
fuat@sys-con.com  
**Vice President, Business Development**  
Grisha Davida, 201 802-3004  
grisha@sys-con.com  
**Group Publisher**  
Jeremy Geelan, 201 802-3040  
jeremy@sys-con.com  
**Technical Director**  
Alan Williamson, 201 802-3040  
alan@sys-con.com

**ADVERTISING**  
**Senior Vice President, Sales & Marketing**  
Carmen Gonzalez, 201 802-3021  
carmen@sys-con.com  
**Vice President, Sales & Marketing**  
Miles Silverman, 201 802-3029  
miles@sys-con.com  
**Advertising Sales Director**  
Robyn Forma, 201 802-3022  
robyn@sys-con.com  
**Director, Sales & Marketing**  
Megan Mussa, 201 802-3023  
megan@sys-con.com  
**Advertising Sales Manager**  
Alisa Catalano, 201 802-3024  
alisa@sys-con.com  
**Associate Sales Managers**  
Carrie Gebert, 201 802-3026  
carrieg@sys-con.com  
Kristin Kuhnle, 201 802-3025  
kristin@sys-con.com

**PRODUCTION**  
**Production Consultant**  
Jim Morgan, 201 802-3033  
jim@sys-con.com  
**Lead Designer**  
Louis F. Cuffari, 201 802-3035  
louis@sys-con.com  
**Art Director**  
Alex Botero, 201 802-3031  
alex@sys-con.com  
**Associate Art Director**  
Richard Silverberg, 201 802-3036  
richards@sys-con.com  
**Assistant Art Director**  
Tami Beatty, 201 802-3038  
tami@sys-con.com

**SYS-CON.COM**  
**Vice President, Information Systems**  
Robert Diamond, 201 802-3051  
robert@sys-con.com  
**Web Designers**  
Stephen Kilmurray, 201 802-3053  
stephen@sys-con.com  
Christopher Croce, 201 802-3054  
chris@sys-con.com  
**Online Editor**  
Lin Goetz, 201 802-3045  
lin@sys-con.com

**ACCOUNTING**  
**Accounts Receivable**  
Kerri Von Achen, 201 802-3062  
kerri@sys-con.com  
**Financial Analyst**  
Joan LaRose, 201 802-3081  
joan@sys-con.com  
**Accounts Payable**  
Betty White, 201 802-3002  
betty@sys-con.com

**EVENTS**  
**President, SYS-CON Events**  
Grisha Davida, 201 802-3004  
grisha@sys-con.com  
**Conference Manager**  
Michael Lynch, 201 802-3055  
mike@sys-con.com  
**National Sales Manager**  
Sean Raman, 201-802-3069  
raman@sys-con.com

**CUSTOMER RELATIONS**  
**Circulation Service Coordinators**  
Niki Panagopoulos, 201 802-3012  
niki@sys-con.com  
Shelia Dickerson, 201 802-3082  
shelia@sys-con.com  
Edna Earle Russell, 201 802-3081  
edna@sys-con.com  
**JDJ Store Manager**  
Rachel McGouran, 201 802-3000  
rachel@sys-con.com

**SYS-CON**  
**MEDIA**





HostMySite.com

**Built for ColdFusion Pros**

**by ColdFusion Pros**

plans from

**\$8.95** / mo.

and **FREE**  
Domain Name

- 24 / 7 / 365 Phone Support
- 99.9+% Uptime
- Macromedia Alliance Partner
- "Full Control" Panel
- CFMX 6.1 or CF 5.0
- SQL Server 2000 or 7.0
- Custom Tags Welcome

Visit [www.HostMySite.com/mxdj](http://www.HostMySite.com/mxdj) for:

**3 Months Free**

and **FREE Setup on Any Hosting Plan**



**"When it comes to ColdFusion hosting,  
HostMySite.com rules them all!"**

James Kennedy  
[mbateam.com](http://mbateam.com)

call  
today!

**877•248•HOST**  
(4678)



# MXDJ Section Editors

## Dreamweaver

Dave McFarland

Author of *Dreamweaver MX: The Missing Manual*, Dave can be relied upon to bring *Dreamweaver MX* to life for MXDJ readers with clarity, authority, and good humor.



## Flash

Jesse Warden

A multimedia engineer and Flash developer, Jesse maintains a Flash blog at [www.jessewarden.com](http://www.jessewarden.com) and says, referring to the MX product range, that "Things are changing, opportunity is on the frontier, a paradigm shift is occurring through Web design, Web applications, et al."



## Fireworks

Kleanthis Economou

A Web developer/software engineer since 1995, now specializing in .NET Framework solutions, Kleanthis is a contributing author of various Fireworks publications and is the technical editor of the Fireworks MX Bible. As an extension developer, he contributed two extensions to the latest release of Fireworks.



## FreeHand

Louis F. Cuffari

Cofounder and art director of *Insomnia Creations* ([www.insomniacreations.com](http://www.insomniacreations.com)), Louis has spent most of his life as a studio artist, including mediums from charcoal portraits to oil/acrylic on canvas. In addition to studio art, he has been involved in several motion picture projects in the facility of directing, screenwriting, and art direction. Louis's creative works expand extensively into graphic design, and he has expertise in both Web and print media. He is deputy art director for SYS-CON Media and the designer of MX Developer's Journal.



## ColdFusion

Robert Diamond

Vice president of information systems for SYS-CON Media and editor-in-chief of *ColdFusion Developer's Journal*, Robert was named one of the "Top thirty magazine industry executives under the age of 30" in *Folio* magazine's November 2000 issue. He holds a BS degree in information management and technology from the School of Information Studies at Syracuse University. [www.robertdiamond.com](http://www.robertdiamond.com)



## Director

Niqui Merret

After many years as an instructor and developer in South Africa, Niqui now works at London-based rich Internet solutions firm, Prismix. She feels that she is a Director addict in a Flash-mad world.



design package, and Fireworks graphics tools have been released, all overhauled to reflect a new emphasis on Web applications. Flash MX Professional 2004 was developed and brought to market, aimed at top-of-the-line Flash developers or people used to using forms-based programming metaphors like Visual Basic or Delphi.

Under Meyrowitz's guidance, Macromedia has made it easier for Flash to work with servers through Web services, and enabled ColdFusion to expose server code. But most important, the focus has broadened from Web-only to "digital interactions" in general. Flash on the handset, for example, is now a growth area for MX developers.

"Making the future better, one interface at a time" may seem a rather lofty goal but that doesn't mean it isn't achievable. In a world where technology is being increasingly embedded into PDAs, phones, and appliances of all sorts, the Macromedia credo is that new "experience hybrids" are a massive growth area for developers and designers. The Macromedia vision is for MX to be the platform that ensures that a good experience is possible on any device. Nothing more, but nothing less.

Wherever applications allow anyone to present information or impart knowledge, developers using the MX family of software can help with the creation and sharing of that touch point as a digital experience; nothing less than the ability to participate, that's what MX tools seek to help developers create.

The organizing principle of *MX Developer's Journal* is participation, too. It is edited "by MX developers for MX developers," and each section editor is a highly devoted IT professional in his or her own right, active at conferences and productive as a book author, Web site owner, or Weblogger. Thus our Flash editor, Jesse Warden, runs the energetic *Jester XL* blog; our Dreamweaver editor, Dave McFarland, authored *Dreamweaver MX: The Missing Manual* (O'Reilly); our ColdFusion Editor, Rob Diamond, runs the high-traffic Web site [BroadwayWorld.com](http://BroadwayWorld.com); and our Fireworks editor, Kleanthis Economou, runs [ProjectFireworks.com](http://ProjectFireworks.com) with equal vigor.

McFarland is highly representative of the top-of-the-line quality we're seeking

in *MXDJ*. Even though he's the Dreamweaver section editor, when he reviewed the first draft of his own article for this month's Dreamweaver section, he found he needed to focus more on just one topic, better organize some parts, and extend a few more once he'd added his example Flash file. So, he removed the entire first section. What remains is a great article about taking a complex, artistic design a designer has made for you – a completed "comp" (composition) – and implementing it into Flash using advanced ActionScript. It's a hybrid approach of timeline and OOP that RAD (Rapid Application Development) people will love.

So whether you have picked up this premier issue of *MXDJ* hoping that it will show you, through a real example, how to take an insanely cool and complex design, and implement it without sacrificing your OOP code, or whether you have picked it up because you are a newbie wondering whether MX Studio is for you, there will be much for you to engage in. You are quite literally holding the future in your hands, the future that the visionary Meyrowitz wants to "make better, one interface at a time."

In this and future issues, the international *MXDJ* team will bring you mouth-watering technical articles that drill down and deal with the code behind this vision. First-rate technical material, fresh insights about rich Internet applications, ingenious new extensions, wide-ranging industry interviews – these are the staple ingredients of the *MX Developer's Journal* banquet.

Best of all, we serve it to you on an all-you-can-eat basis, which is why even this inaugural issue is a hundred pages long. ☺

Bon appetit!  
Jeremy Geelan

*Jeremy Geelan is group publisher of SYS-CON Media, and is responsible for the development of new titles and technology portals for the firm. He regularly represents SYS-CON at conferences and trade shows, speaking to technology audiences both in North America and overseas. [jeremy@sys-con.com](mailto:jeremy@sys-con.com)*

# Complete source code and asset management in Dreamweaver MX—now possible with Surround SCM.

Dreamweaver users know a beautiful Web-based product is only skin deep. Underneath, it's a tangle of hundreds or thousands of ever changing source files. Without a good development process and strong tools, bad things happen. Surround SCM can help.

Surround SCM lets you...

*Track multiple versions of your source files and easily compare and merge source code changes.*

*Check out files for exclusive use or work in private workspaces when collaborating on a team.*

*Automatically notify team members of changes to source files—push changes through your organization.*

*View complete audit trails of which files changed, why, and by whom.*

*Associate source code changes with feature requests, defects or change requests (requires additional purchase of TestTrack Pro).*

*Remotely access your source code repository from Dreamweaver MX.*

Surround SCM adds flexible source code and digital asset control, powerful version control, and secure remote file access to Dreamweaver MX. Whether you are a team of one or one hundred, Surround SCM makes it easier to manage your source files, letting you focus on creating beautiful Web-based products.

## Features:

Complete source code and digital asset control with private workspaces, automatic merging, role-based security and more.

IDE integration with Dreamweaver MX, JBuilder, Visual Studio, and other leading Web development tools.

Fast and secure remote access to your source files—work from anywhere.

Advanced branching and email notifications put you in complete control of your process.

External application triggers let you integrate Surround SCM into your Web site and product development processes.

Support for comprehensive issue management with TestTrack Pro—link changes to change requests, bug reports, feature requests and more.

Scalable and reliable cross-platform, client/server solution supports Windows, Linux, Solaris, and Mac OS X.

Achieve major improvements in Web and e-business development performance through better tool integration and process automation. Gain complete control over your source code and change process with Surround SCM. Manage defects, development issues, and change requests with award-winning TestTrack Pro. Completely automate product testing with QA Wizard. Streamline your development process with Seapine tools and help your team deliver quality software products on time, every time.

Learn more about  
Surround SCM at  
[www.seapine.com](http://www.seapine.com)  
or call 1-888-683-6456



d

f

fw

fb

cf

cb

# A First Look at Dreamweaver MX 2004

*Meeting the standards that  
Web developers look for*  
by dave mcfarland

The Web always changes, and the tools we developers use must also. Dreamweaver MX 2004, the next generation of the popular Web development program, is on its way, featuring improved support for cutting-edge CSS, time-saving enhancements, and performance improvements. In this first look, *MX Developer's Journal* presents an overview of the changes.

## CSS Rises to the Top

Web development is a moving target; once you've mastered one technology, a new standard comes along to take its place. Macromedia's dedication to providing tools that allow Web developers to ride that "bleeding edge" is evident in the significant enhancements to Dreamweaver's Cascading Style Sheet support. From basic text formatting to complex CSS-based page layouts, the program is infused with CSS.

In this release, Dreamweaver throws off deprecated formatting conventions like the `<font>` tag in favor of a cleaner separation of content from presentation. The Property Inspector, for example, now creates CSS styles when a user applies font-face, size, and color properties to text. Basic page properties, like background color and page margins, are all CSS based, so developers can use the program with less fear that Dreamweaver will insert deprecated attributes.

The creation and editing of CSS styles are also streamlined. You can edit a style using the new CSS Properties panel, whose basic editing grid makes changing or adding properties quick and easy. And when properties from multiple styles converge on a selected tag, the Relevant CSS panel is a big help (see Image I). It

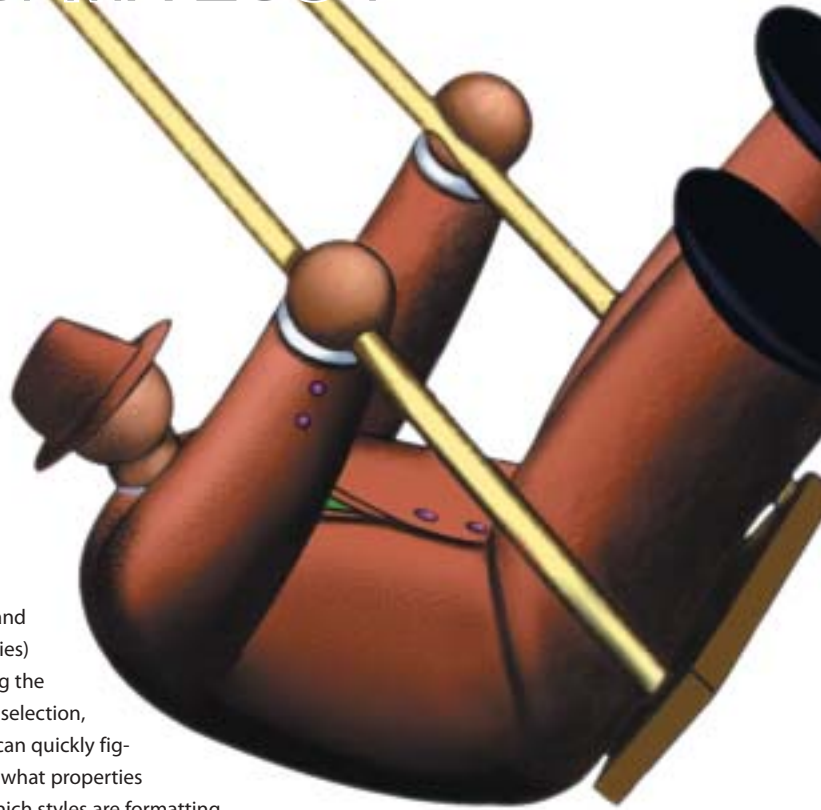
lists all styles (and properties) affecting the current selection, so you can quickly figure out what properties from which styles are formatting the selection.

For example, in Image I the Relevant CSS panel indicates that a `<p>` tag is currently selected; it's nested within three divs, which are themselves within the `<body>` of the page. Highlighting a rule (the `<body>` in this case) displays that rule's properties – properties in blue apply to the selection, while properties crossed out in red do not apply or are overridden by more specific styles. In this example then, the paragraph inherits "Arial" from the `<body>` as well as line height; however, the color (`#333333`) is overridden by another, more specific rule.

CSS support in Dreamweaver MX 2004 isn't limited to text or basic formatting, either. The program now does an excellent job of displaying CSS-based, table-free layouts (see Image II). In fact, the overall rendering of CSS within the program's design view is greatly improved. Working with the basic-building block of CSS-layout – a `<div>` – is also simplified; you can now select divs quick-

ly in design view, and the new Insert Div object provides fine-grained control over placing divs on a page. To jump-start your CSS layouts, the program ships with several sample designs (like the one pictured in Image II).

To make sure your "bleeding edge" design doesn't end up a bloody mess in some poor person's browser, Dreamweaver MX 2004 adds a new cross-browser validation tool. Check your pages against a customizable list of Web browsers to see whether your code will work (see Image III.) For example, if you're not sure how well Navigator 4 supports the CSS "padding" property, Dreamweaver's Browser check can tell you. The feature is very customizable: it can check for errors when the page opens, you can change the list of browsers to check against, and you can even set it to ignore errors you know are there but aren't harmful – such as the fact that the "marginwidth" attribute isn't supported in IE.



## Helping You Work Faster

Other changes to the program are aimed at helping Web developers work faster. Get out of the gate quickly with the new start screen. It lets you open recent pages, create new files, and use supplied sample files to jump-start your design work. In addition, the program can remember files that were open the last time you quit the program, and reopen them when the program launches again.

You no longer have to click through the Insert bar's multiple tabs just to insert the code you

save you hours of work a week.) And if your boss insists that a Web page look just like the Word Doc he or she sent you, the "paste formatted" command preserves complex page formatting (at the expense of some bizarre and unwieldy CSS).

MX 2004 also features improvements to its FTP functionality – it's faster and more reliable (but still won't work in the background). And, if you don't need Dreamweaver's powerful site management features (all you need is one simple change to a page on your server, for example) "site-less" file editing lets you

duced in Dreamweaver MX remain largely unchanged. If you've longed for more server behaviors to further expand the program's capacity to rapidly build database applications, you'll have to wait until the next version (or search through the many third-party extensions).

PHP developers will be happy to see that the User Authentication behaviors available in the other server models have been added for PHP/MySQL, as has the Master/Detail page set. (Unfortunately, still missing are the Go To Detail Page and Go To Related Page behaviors.) For ASP.NET, Dreamweaver introduces

“With every new version of MX the synergy between the company's programs increases”



want; the new "Favorites" tab on the Insert bar holds all of your frequently used Dreamweaver objects (Insert Image and Insert Table, for example) in one easily customized tab. And snippets-addicts can rest their mouse-pads – the program's customizable keyboard shortcuts now support snippets!

Nothing can grind your workday to a halt like receiving content from a client, boss, or co-worker. Oversized, unoptimized JPEGs and highly formatted Word documents often mean hours of extra work. Fortunately, Dreamweaver MX 2004 can help ease this pain.

For quick image modifications, the Property Inspector sports some simple image editing tools, so you can optimize, crop, sharpen, and adjust the brightness and contrast of images – all within Dreamweaver. For more complex work, the program still integrates seamlessly with Fireworks.

Working with Word and Excel documents has also gotten easier. Dreamweaver finally retains basic formatting such as paragraph breaks, bold, italics, and headings when pasting from Word or Excel (this change alone could

skip Dreamweaver's "Local Site" metaphor and work directly on files off an FTP or RDS server.

## Today's Technologies

Modern CSS support isn't the only technological leap forward for the new Dreamweaver. For those in need of privacy, the program now supports Secure FTP for completely encrypted file transfer. In addition, if you communicate with an international audience, you'll find full Unicode support. Now you can build your pages using any font supported by your operating system – even double-byte character sets.

Dreamweaver began life as an HTML editor. It's since morphed into a CSS editor and proven itself to be a capable text editor. This latest version also boasts improvements in XML editing. Dreamweaver's code view supports XML namespaces so you can use the built in code-hints feature while authoring XML, and even validate your XML against an imported XML schema.

On the server side, Dreamweaver MX 2004 doesn't have as much to brag about. The basic server models intro-

duced in Dreamweaver MX remain largely unchanged. If you've longed for more server behaviors to further expand the program's capacity to rapidly build database applications, you'll have to wait until the next version (or search through the many third-party extensions).

PHP developers will be happy to see that the User Authentication behaviors available in the other server models have been added for PHP/MySQL, as has the Master/Detail page set. (Unfortunately, still missing are the Go To Detail Page and Go To Related Page behaviors.) For ASP.NET, Dreamweaver introduces

ASP.NET form control objects. You can build forms for using .NET Web controls; all properties of the form control can be modified in design view using the Property Inspector. Even if Dreamweaver can't do everything for you automatically, at least it can help you find the answers. Dreamweaver's reference panel, which provides helpful references on key Web technologies from industry-leading authorities, has some new additions. You'll find SQL, ASP.NET, and PHP references from O'Reilly, as well as two ColdFusion references from Macromedia. Now the answers are at your mouse-tip.

## The MX World

Dreamweaver doesn't stand alone in the Macromedia Pantheon. With every new version of MX the synergy between the company's programs increases. While the full story remains to be written, Dreamweaver MX 2004 provides some interesting glimpses into what the future might hold. Two fledgling features – MX HTML Elements and Flash Elements – while minor parts of Dreamweaver MX 2004, hold some mystery and a lot of promise.

According to the Macromedia Press releases, "MX Elements are a series of pre-built components available in both Dreamweaver and Flash with a new generation of interactive design patterns embedded, making it fast for designers and developers to build the most effective UI for the task." The only example in this version of

Dreamweaver is "Halo," a slick, chrome-inspired, CSS-only design (see Image II). It's cool, but hardly earth-shattering; what makes it most interesting is the fact that it's worthy of making the Dreamweaver press release. Whether this

is part of some bigger plan for application integration (the Royale

Initiative – [www.macromedia.com/special/royale](http://www.macromedia.com/special/royale) – perhaps?) only time will tell.

In addition, Web developers (with little or no Flash knowledge) may find the new Flash elements helpful. A Flash element is basically a Flash movie whose properties can be customized within Dreamweaver. The only example that ships with Dreamweaver MX 2004 is the Image Viewer (see Image IV). It's basically a Flash-based slideshow projector. After inserting it into a page, you can use the Tag inspector to see various properties, including the name of the slideshow, the URL of images to appear in the show, and caption text for each slide.

While you might not need a slideshow, the idea behind the feature – customizable Flash movies – does hold promise for more interesting tools in the future.

No overview can cover it all. There are thousands of changes to the program: new features, interface enhancements, bug fixes, and performance improvements. But one thing is clear: in Dreamweaver MX 2004, Macromedia shows that they're committed to keeping pace with the rapidly changing standards and technologies that Web developers use and rely on. ☺

*Dave McFarland is the Dreamweaver editor of MX Developer's Journal and author of Dreamweaver MX: The Missing Manual. [davemcfarland@sys-con.com](mailto:davemcfarland@sys-con.com)*

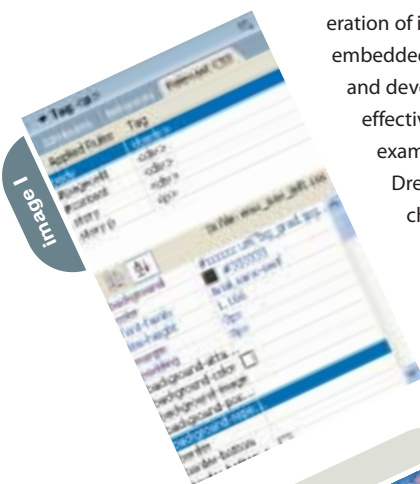


Image I

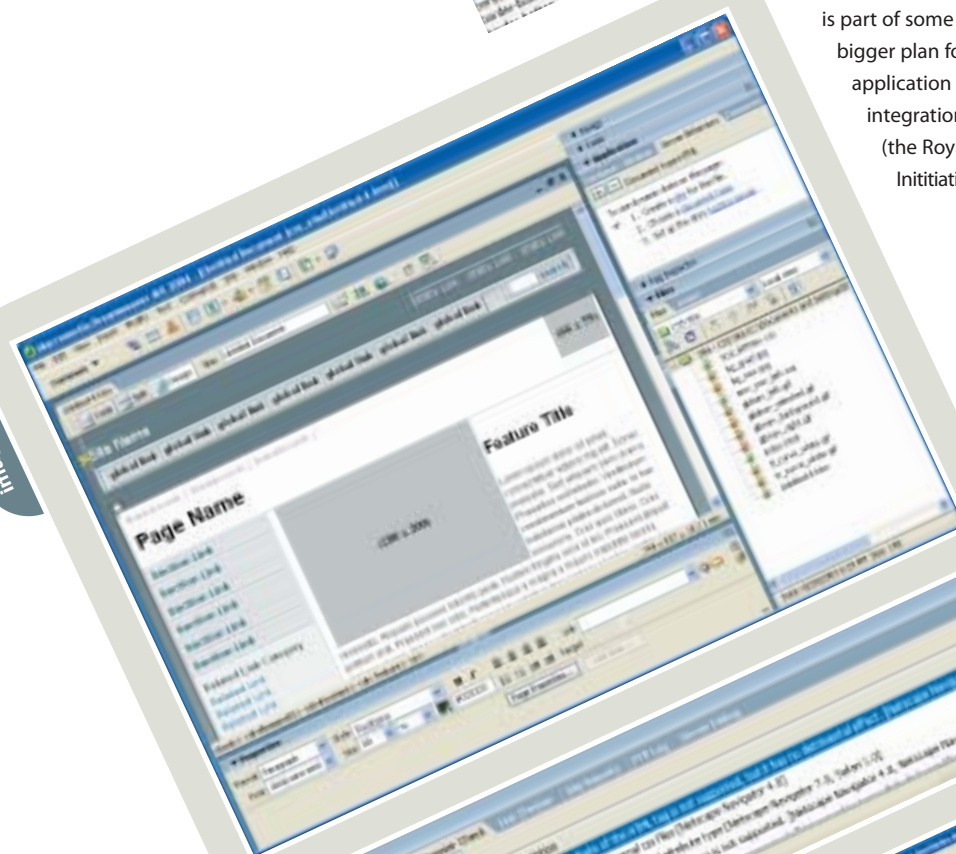


Image II

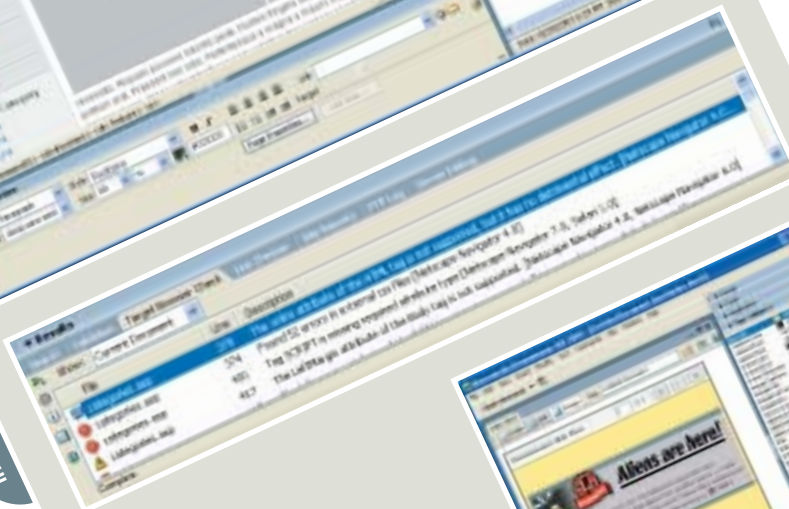
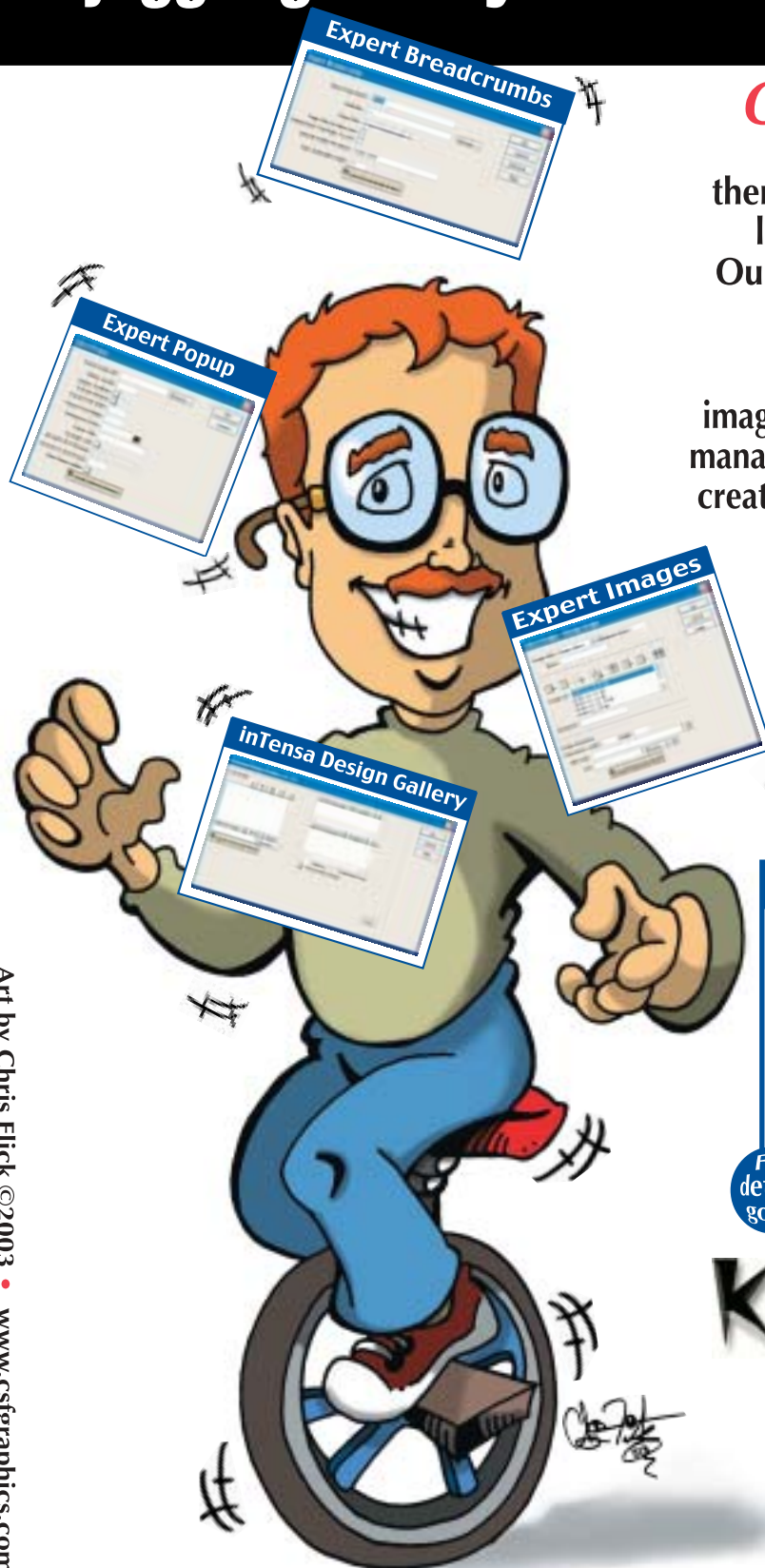


Image III



Image IV

# Are you a web designer that's having trouble juggling all of your web responsibilities?



**Or...** maybe you just want to spend more time designing web sites then programming them. If this sounds like you, then **kaosweaver** can help! Our extensions help web professionals do more work... quicker!

Have you ever wanted to create an image gallery? Maybe find a better way to manage your images? Or always wanted to create breadcrumbs but didn't know how?

## NOW YOU CAN!

Visit [WWW.KAOSWEAVER.COM](http://WWW.KAOSWEAVER.COM)

To find out how **YOU** can easily extend **YOUR** web site projects... **TODAY!**

AN EXCLUSIVE MX DEVELOPER OFFER

**EXPERT PACK!** FOR A LIMITED TIME ONLY!

Expert Breadcrumbs • Expert Images • Expert Popup

3 Incredible extensions for only... **\$65!** OVER 10% OFF!

For details go to [www.kaosweaver.com/mxoffers/](http://www.kaosweaver.com/mxoffers/)

# Kaosweaver

*Weave Fast - Dream More*

12468 Benson Street  
Overland Park, KS 66213  
[kaosweaver@kaosweaver.com](mailto:kaosweaver@kaosweaver.com)  
[www.kaosweaver.com](http://www.kaosweaver.com)

# cascadeinstyle



So you think you're ready to hop on the CSS styling bandwagon? Many people are finding the decreased page load time and quick site-wide changes to be attractive draws, and they're making the leap. The enhancements in Dreamweaver MX 2004 now make this an even more viable option. No, you can't stay in design view the whole time (who'd want to?) but you can preview something close to what you'll actually see in a browser. In this tutorial I'll cover different ways to style simple CSS buttons with no graphics. We'll look at creating the navigation both vertically and horizontally. **by stephanie sullivan**







**Tip:** If you have other graphics on your page with a light source coming from any other direction besides top left, I would strongly urge you to match your buttons to that light source.

Visualize in your mind where the light source is coming from on the other graphics, then match the light source on your buttons by making the two light borders on your buttons on the sides where the light would hit. Consistency in design is very important.

## Using a List

There are a variety of different approaches to creating CSS buttons. What I've settled on, for accessibility reasons, is using an unordered list for the navigation. A list of links is an organized way to present the information to any type of assistive technology. It also degrades quite nicely in Netscape 4 but you will have to decide how much time you put into that ancient browser. The styling can be totally hidden from NN4 in favor of a plain list of links if necessary.

Create an HTML (or XHTML) document. If you want to work on these buttons in an HTML page, you can play with

sional space. You don't have the ability to "cheat" and use the bevel effect in Fireworks to fake the three-dimensional look. You have to rely on your ability to create the illusion of 3D through properly defined code. Remember – the same rules apply in art as in Web pages (or even dressing yourself). Light brings things forward. Dark makes them recede. Thus, in our first example you could create a simple button with four borders of the same color and it will look flat. Or you can make it appear beveled by using lighter colors for the left and top borders and darker colors for the right and bottom borders. This will make it appear that the light source is coming into

“light brings things forward;  
dark makes them recede”

the completed files I created for this project (the files have headers and text in a sample HTML document since I can't stand to just put buttons on a blank page). They're available for download at: [www.communitymx.com/mxdevissue2](http://www.communitymx.com/mxdevissue2).

The basic HTML code used for the buttons is shown in Code I. This code shows the unordered list placed into a div with the ID of menu. The placement of the buttons is controlled through the menu div. All selectors used to define these buttons will begin with the menu ID (#menu) and then continue with the element inside the div that you are defining. By taking advantage of the cascade order like this, you are instructing the browser to apply any selector beginning with #menu only to the elements inside the menu div. This keeps all your other lists "safe" from being rendered with the menu styles. A list in your main content area will not receive the styling you're applying to the buttons. Each selector will be defined like this:

```
#menu elementname {
  property: value;
}
```

## Simple Beveled Button Style

When you're creating a button with CSS, you're working within a two-dimen-

the page from the top-left.

Image I is the look we're creating for the first set of buttons. These buttons will appear to be depressed, or pressed down, on rollover or on tab through. Remember that not everyone will use a mouse to navigate your pages. For accessibility reasons you can add a simple selector (a:focus) that will give the button the same look when it's tabbed over as it has when it's hovered over with a mouse (in the browsers that support it). This gives people a visual cue showing where they are on your page.

These buttons begin with the unordered list code given above. Each link has the null "javascript:;" applied so that it will appear in its link style. This is where Dreamweaver MX 2004's new CSS abilities really shine. Previously I had to preview in a browser (F12) to view the effects my style rules were actually having. Image II illustrates the way MX renders this button style (on the left) and the new rendering by 2004 (on the right).

## Placing the List on the Page

Now the fun begins. You can do most of your work using the Dreamweaver CSS Styles panel. Either create an external CSS document or simply place these styles into the head of your HTML document and move them later. You will need to go into the CSS and hand code if you

will be supporting Netscape 4. If you haven't used the CSS Styles panel before, read through my tutorial, "Using the CSS Styles Panel" at CommunityMX to learn the basics ([www.communitymx.com/abstract.cfm?cid=A1FFDC2ED5D7F133](http://www.communitymx.com/abstract.cfm?cid=A1FFDC2ED5D7F133)).

Since the positioning of the #menu div controls the overall placement of the buttons on the page, you can adjust it to suit your own pages. I find it more attractive if these buttons don't start flush with the left side of the browser chrome, so I set the absolutely positioned div to left: 20px. The top isn't given a setting, thus it comes after the element that it follows (the default is "auto"). You'll come up with a slightly different button that looks nice on your pages later. At that point, you can adjust the width, padding, and left div placement to better fit your page and style. Here's what you'll place on this page:

```
#menu {
  position: absolute;
  width: 130px;
  left: 20px;
  padding-top: 15px;
}
```

### Styling the List Elements

The principles behind styling the unordered list on all vertical button examples begin in the same way. In fact, the redefinition of the <ul> element is identical each time. You will set the font family and size, the margin and padding to zero, and the list-style-type to none. All redefinitions of the <li> element begin the same way as well. The margin and padding are given a value of zero. Code II is the beginning styling of each example button

Due to the cascade, you won't need to set the font in any of the other button selectors. This code will effectively zero out the normal list spacing and remove the bullet so you can have a true button look. You may choose to use font sizing that is more scalable using ems or percents. Since I have no way of knowing what other styles are on your page, I'm setting this at 12px to keep it consistent.

### Styling the Link Element

The styling for the basic button structure is all in the <a> element. The code we just added is what all our vertical buttons will have in common. The code below is where the differences come in. For this first button, we'll

define a background color and two different colored borders. We'll also set the padding to keep the text away from the edges of the button. Since the #menu div itself will have a set width (applied in a moment), we'll set the width of this button to 100% and give it a "display: block". These last two values will make the entire button width clickable, not just the link text itself. This is important if you want it to act like a real graphical button. Add the following code to your CSS:

```
#menu a {
  margin: 0;
  padding: 3px 3px 3px 10px;
  width: 100%;
  display: block;
  color: #000000;
}
```

Now we'll add the color but we'll hide it from that old gal Netscape 4. She's simply too geriatric for me to worry about. She'll receive the basic placement and the rest of the styling in Code III will be hidden. It will render a nice, organized list of accessible links. I've used the Caio Hack on Code III to hide it from NN4. You'll have to hand code the hack if you're going to use it. (For an explanation of the Caio Hack, avail yourself of the article by John Gallant and Holly Bergevin at [www.communitymx.com/abstract.cfm?cid=A3E04D26B481EAB6](http://www.communitymx.com/abstract.cfm?cid=A3E04D26B481EAB6).)

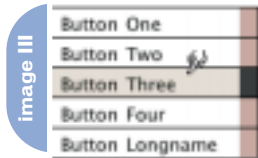
Notice that the borders are defined in two sets of colors to give the button depth. The border colors reverse on the hover style and the background color gets deeper or greyed down, as if it were pressed in. The *a:focus* is set to be the same as the *a:hover* for tabbing accessibility.

Preview the page in your target browsers to be sure you've created a button that is appealing in all of them. Once you're happy with the results, move on to the next button.

### Styling with Creative Borders

In this button style, we're not going to worry about making the button three dimensional. We'll use the borders to give it a nice decorative style. When completed it will look like Image III.

Since the HTML portion of this project is identical each time, you'll reuse it. Simply make a few simple changes to the CSS document you just completed to cre-



ate a new button style (you'll likely want to save a copy of it to go back to).

Let's look only at the portions that will change. Begin with the same unordered list for the HTML portion. Use the identical redefinition of the <ul> element. The <li> styling remains the same with the margin and padding zeroed out. Add a one pixel bottom border to the <li> selector to create the line that separates each button.

```
#menu li {
  border-bottom: 1px solid #000;
  margin: 0;
  padding: 0;
}
```

Style the link to create the button look. As you did previously, hide the styling from Netscape 4. To keep the text from sitting right on the left border, you'll give the links padding. You'll also set the right border to 12px for a decorative border. You'll designate display: block, but this time, instead of setting the width to 100%, we'll set it to 118 pixels. The math behind this is simple. The #menu div width will shortly be set to 140px. From that, you'll subtract the left padding and right border (10 and 12 pixels, respectively). This brings you to the 118px width. If you choose a different width for the link, you'll have a space at the end of the bottom border of the <li> that will be white, and/or extra white space to the right of the decorative border. Experiment with

*Stephanie Sullivan is a Web developer, partner at CommunityMX ([www.communitymx.com](http://www.communitymx.com)), owner of VioletSky Design ([www.violetsky.net](http://www.violetsky.net)), and contributing author of Dreamweaver MX 2004 Magic. [design@violetsky.net](mailto:design@violetsky.net)*



**Tip:** Due to IE5.x PC's broken box model and confusion with padding, the bordered button style will display differently there. This can be hacked if you need to support this browser. Read Holly Bergevin's excellent article at [www.communitymx.com/abstract.cfm?cid=EO989953B6F20B41](http://www.communitymx.com/abstract.cfm?cid=EO989953B6F20B41)

this since it can have interesting effects that we don't have room to explore here. The above steps will create Code IV.

Previously in Dreamweaver MX, you saw what is displayed on the left side of Image IV. In Dreamweaver MX 2004, you'll see a button list almost identical to your browser's (right side of Image IV).

Finally, let's give the button its hover and focus styling:

absolute positioning from the `#menu` div to keep it in the flow of the document, right under the top banner of the page. Set the top padding and zero out the rest, like this:

```
#menu {
    margin: 0;
    padding: 3px 0 0;
}
```

## “this is where dreamweaver mX 2004's new CSS abilities really shine”

```
#menu a:hover, #menu a:focus {
    background: #DBF8D8;
    border-right: 12px solid #333;
}
```

Make the final changes to the menu div for placement. Since you added the line under each button in the `<li>` styling, we'll add a top and right border to the menu div so that the buttons look completely boxed in. Adjust the margin-top to align the buttons on your page.

```
#menu {
    position: absolute;
    width: 140px;
    border-top: 1px solid #000;
    border-right: 1px solid #000;
    margin-top: 18px;
}
```

Preview the style in your browser. Tweak it as desired. You're done!

### Going Horizontal

Since lists are vertical by nature, styling them as buttons is easy. But if you want horizontal navigation, you've got to use images, right? Wrong. With CSS, lists can be horizontal as well. It's just a simple little property value called "inline" that makes the switch. Let's look at the last vertical button set you created. With a few changes in the CSS, it can be horizontal (remember to save a copy of the CSS before you make these changes) (see Image V).

The first change is to remove the

We'll also use the `<ul>` element to make a fun little stripe across the top of the page behind the buttons. Keep all the code from the `#menu ul` selector and add these three declarations to that rule:

```
border: 1px solid #000;
text-align: center;
background-color: #9B9ED7;
```

Since the `#menu a` selector is the one that forms the buttons, very little change is required. Simply remove the `display: block` and the `width: 118px`. (You will be placing the `display: inline` into the `#menu li` selector and this would override it.) If you're using the Caio Hack for Netscape 4, you'll change the borders of the `#menu a` just slightly inside the hack. Replace those padding and border declarations to contain the following:

```
padding: 3px 5px 1px 10px;
border-right: 5px solid #759C74;
```

Make sure you also change the size of the `#menu a:hover`, `#menu a:focus` selector's right border size to be 5px as well. Otherwise, when you roll over the button the border will get larger and everything will jump.

Finally, the changes to the `#menu li` itself will take your list into the horizontal realm. This is the only selector where all the declarations will change. Remember that on the vertical button list, we placed a bottom border on the `<li>` to differentiate the buttons. On the horizontal button list, we'll move that to the right side instead.

I've also placed a right-side margin to put a little space between the buttons; finally, the magical `display:inline` is applied.

```
#menu li {
  margin: 0 5px 0 0;
  display: inline;
  border-right: 1px solid #4D664C;
}
```

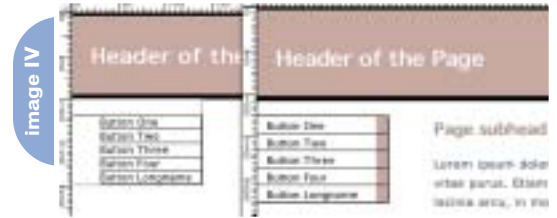
Image VI shows the true power of Dreamweaver MX 2004. Notice the top portion of the image. This is what you will see in design view if you're still using Dreamweaver MX. If you're using MX, you'll have to do a constant F12 (preview) if you want to have any idea of how your buttons will actually render. The lower portion of the image shows the same page in Dreamweaver MX 2004. If you compare it to the earlier image of the completed project, you'll see that it is nearly identical. Obviously, you will always have to preview in many browsers to check your code for bugs and browser differences, but having a nice rendering in Dreamweaver itself is an obvious advantage.

That concludes the changes to your CSS code. There's one final little tweak that must

be made in order to squash a white-space bug that exists in Internet Explorer 6 PC. Go into the HTML document and make your list code "ugly". Strange, eh? It's a simple process really and there are two ways to get the same results. Essentially, there can be no space between the closing `</li>` and the opening of the next `<li>`. Thus, you can either place your cursor in front of the opening `<li>` and hit backspace until it's snuggled up to the `</li>` in front of it, or you can move the closing bracket (yes, that's the proper name for the "greater than or lesser than" signs we all use to enclose our code) of the `</li>` above down to the right in front of the `><li>` below. I tend to like the first method and just run them all together. Following is an example of the second method.

```
<ul
><li><a href="javascript:;">Button
One</a></li
>< li><a href="javascript:;">Button
Two</a></li
><li> etc...
```

You've just learned three methods of styling lists as buttons with CSS. Have fun



and play with the colors and sizes. Remove the styling from the `#menu ul` to make it invisible. Try putting the beveled edges on the buttons or use another styling method. You'll have a great time making something unique to match your style and your Web sites. ☺

code I

```
<div id="menu">
  <ul>
    <li><a href="javascript:;">Button One</a></li>
    <li><a href="javascript:;">Button Two</a></li>
    <li><a href="javascript:;">Button Three</a></li>
    <li><a href="javascript:;">Button Four</a></li>
    <li><a href="javascript:;">Button Five</a></li>
  </ul>
</div>
```

code II

```
#menu ul {
  font-family: "Lucida Grande", Arial, sans-serif;
  font-size: 12px;
  margin: 0;
  padding: 0;
  list-style-type: none;
}

#menu li {
  margin: 0;
  padding: 0;
}
```

code III

```
/**/a{
#menu a {
  background-color: #99CC99;
  border-top: 2px solid #D4E8D1;
  border-left: 2px solid #D4E8D1;
  border-bottom: 2px solid #6B8F6B;
  border-right: 2px solid #6B8F6B;
  text-decoration: none;
}
```

code III

```
}
#menu a:hover, #menu a:focus {
  border-top: 2px solid #6B8F6B;
  border-left: 2px solid #6B8F6B;
  border-bottom: 2px solid #D4E8D1;
  border-right: 2px solid #D4E8D1;
  background: #86AE8C;
}
/* NN4 hack */
```

code IV

```
#menu a {
  display: block;
  margin: 0;
  text-decoration: none;
  color: #000;
  background: #FFF;
  width: 118px; /*this is the width MINUS the left
padding and the right border - otherwise, you'll have a
space before your right border*/
}

/**/a{
#menu a {
  padding: 3px 0 1px 10px; /*this keeps the links from
sitting too far to the left*/
  border-right: 12px solid #9C9; /*this gives you the
right decorative border - make sure to match the size of
the hover border */
}
/* NN4 hack */
```

d

f

fw

fb

f

db

# Can You Be Seen... or Heard?

*Make sure your Web site can be found by everyone*

by bruce lawson

So you need to make your Web site accessible. It's not the purpose of this article to persuade you to do it; let's just assume that you've decided to do it.

Your first port of call should be the Web Accessibility Initiative (WAI)'s Web site at the W3C. They maintain a list of the legislative requirements of countries with specific legislation ([www.w3.org/WAI/Policy](http://www.w3.org/WAI/Policy)). If your country isn't there, you should follow the WAI's general guidelines, found at [www.w3.org/TR/WCAG10-TECHS](http://www.w3.org/TR/WCAG10-TECHS).

Simply put, the guidelines say

- A user should be able to navigate

through the site using a keyboard if they can't use a mouse.

- Any information that is not in text that can be copied and pasted (eg, text in images / text in Flash movies) should have alternative text marked up in the HTML.
- Instructions should avoid requiring sight or accurate color vision (e.g., don't say "click on the red button/picture of a cat/ circle").
- Information that requires listening (e.g., audio recording of conversation) should be available as a written transcript or through subtitles.
- Don't induce epileptic seizures or nausea through flashing or flickering screen art.

This article looks at how to use Dreamweaver to ensure that WAI guidelines are adhered to.

## Defining Accessibility - How Do Disabled People Use the Web?

A year or so ago, if you mentioned accessibility, it was very common to get a look of blank incomprehension back – how could blind or disabled people possibly hope to use the Web?

For many disabled people (or people with repetitive strain injuries, a broken writing hand, etc.), the biggest obstacle to using the Web is the difficulty in using a mouse. One of the aims of accessibility is to allow any Web site to be operated using the keyboard, tabbing for navigation, and using the enter key to push a button. Once this is achieved – and it's very easy to do – then the site is accessible. It could even be controlled by a mouth-stick for a paraplegic user, given appropriate hardware modifications.

For the blind, or visually impaired, the greatest impediment to Web use is, of

course, the inability to see a screen. For blind users whose hearing is unimpaired, there is a technology called a screen reader that works on PCs. This is a speech synthesiser that "reads" the screen through headphones or computer speakers. There are many vendors of these – the main ones are JAWS and Windows Eye, both of which integrate with MSAA (Microsoft Active Accessibility), which is on all Windows machines from Win95 onwards. Any text on the screen is automatically available to MSAA. The challenge is for non-textual parts of the Web page (graphics, photographs, text-in-images) to be available to the screen reader, too. To get the feel of a screen reader, you can download one from [www.readplease.com/rpdownload.php](http://www.readplease.com/rpdownload.php)

Other assistive technologies exist, too – for the blind and deaf, a Braille pad can move pins, driven by special software, to make Braille characters for the user to feel. However, what you as a Web developer need to know is that in order for any of these assistive devices to work, the HTML must be coded to accessibility standards.

## Making a New Site, or Updating an Old One?

If you're creating a brand new site, I suggest (no, actually, I implore) you create your site according to Web standards. This isn't the place to either justify the use of standards, or go into a discussion of how to use them. But while using CSS and XHTML or HTML 4.01 doesn't guarantee accessibility, it makes it considerably easier to make it so.

Tables used for layout can often confuse a screen reader, especially if they're deeply nested. The spacer gifs you'll need to pad the whole thing out will also cause the screen reader to work overtime unless you manually add alt text to them.

image I



image II



image III



# Dreamweaver Extension of the Month

by dave mcfarland

Using CSS for layout means that the screen reader will ignore all that sexy presentational stuff and concentrate on the structure and the content of the site, which is in your (X)HTML.

If you are using CSS and doing some fancy layout, or using alternate style sheets that allow the user to change where the content is positioned on the page, you might not be worried about the order of your DIVs in the HTML. However, a screen-reader is blithely unconcerned with visual fripperies; it will read the content in the order that it appears in the markup. It's structural markup; it needs to be logical. 'Nuff said.

## Setting DW Preferences

The first date I ever had with Cameron Diaz, I spent hours squeezing myself into an Armani suit, getting a \$200 haircut, a manicure, and slicing myself with expensive perfumes, yet forgot to clean my teeth. The moral of the story is, before setting out on the tricky stuff, get the basic preparation work done. (Cameron didn't notice, in case you were wondering, as I'd also forgotten to put on any trousers.) The Dreamweaver equivalent to the pre-date oral hygiene regime – and trousers – is setting Dreamweaver Preferences to do some of the work for you.

Spark up Dreamweaver (MX or MX 2004) and select Edit > Preferences and then the accessibility category in the Preferences dialog box. Under "Show Attributes when Inserting", check all five checkboxes – Form Objects, Frames, Media, Images, Tables (see Image I).

You have just sternly commanded Dreamweaver to remind you, via a dialog box, that you need to add certain extra information. It can't add the information itself, but one of the trickiest things about moving towards designing for accessibility is remembering to add all the extra bits and pieces you need to put in.

## Accessible Images

Now you've updated your preferences, add an image to your site. You'll notice that a new dialog box pops up asking you for *long description* and *alternate text*, which will be added as attributes to the img tag that Dreamweaver will create. The long description is a URL of a page that will describe the contents

of the picture (imagine it's some really complex graph showing profits against the price of fish and the value of the yen). It is entirely optional (and not often used).

The alternate text is the most important, and is mandatory. In a visual browser, you see alt text when you mouse over an image; it pops up like a little yellow 'tool tip'. While screen readers don't care about pictures, they love alt text and proudly read it out. You have 250 characters to describe your image. Here, for example, is some alt text in a blog entry on my homesite ([www.bruceclawson.co.uk](http://www.bruceclawson.co.uk)) specifically about the pictures (see Image II).

While a blind person obviously wouldn't see the image, the point of the blog entry is the picture and they'd understand the context by hearing the phrase "scan of 12-week fetus". Otherwise they won't understand what I mean by "meet the new Lawson".

Image III is an example of bad use of alt text on a Web site header made of four separate images:

The alt text here bears no resemblance to the picture. This could just confuse a non-sighted visitor; the purpose of the site is adequately summed up in the text superimposed on the rightmost picture. This text could have been used as alt text for the logo – a much more logical place for it. Finally, it is not necessary to put alt text explaining shiny happy stock photography models at all.

Which brings us to the question, what do you do with "empty" images, like the decorative people in the last example, or in spacer gifs in old nested table layouts? The answer is simple: use empty alt text, which the screen reader ignores.

Unfortunately, adding blank alt text isn't intuitive in Dreamweaver. If you just hit enter on the alt text/ long description dialog box, Dreamweaver rather stupidly adds no alt text at all (making the markup invalid, as well as ignoring your earlier command to help you make an accessible site). Grrr! After reclining back and receiving a stress-reducing massage from Ms. Diaz, you need to go into the property inspector and select the drop down menu next to alt and select blank (see Image IV).

You need alt text on every image, whether it's an empty image, a hotspot

## Pure Upload

**Extension Developer:** George Petrov  
**Developer Web Site:** [www.dmxzone.com](http://www.dmxzone.com)  
**Price:** \$59

**d**reamweaver makes some aspects of server-side programming a breeze. Pages for adding and updating database records can be produced in seconds. But Dreamweaver can't do it all. Fortunately, thanks to the program's open API, skilled programmers are constantly adding features that Dreamweaver's engineers may never have imagined.

This month we'll look at three extensions from DMXZone – Pure PHP Upload, Pure ASP Upload, and Pure ASP.NET Upload. They're essentially the same extension written for three different server-models, and they let you quickly and easily add file upload capabilities to any page on your site.

A simple dialog box provides all the settings you need to control where uploaded files are stored, what types of files to allow, and the maximum size of uploaded files. In addition, the extension works with Dreamweaver's built-in insert record and update record server behaviors, so you can combine file uploading with regular database functions. In fact, the extension can store the file name, or the full file path into a database field, and even create unique file names when naming conflicts occur.

Although this extension can upload any type of file, many of its features are intended for graphic files. For example, advanced settings let you constrain the minimum and maximum heights and widths for uploaded graphics, as well as letting you store widths and heights into database fields.

To add a bit of polish to the upload process and to provide useful feedback to users, you can also include an upload status window – a pop-up window that indicates the progress of the upload and closes when the upload is complete.

If you need to add file uploading ability to your site quickly and painlessly, this extension can do the trick.



in an image map, a photo of a kitten, a chart showing your company's 750% increase in revenue, or that cute animated gif of a crocodile eating your ex-sweetheart that you put on your blog in a bad mood.

### What Do Links Sound Like?

A very Zen question, you might think. But go to any page on the Web and press the Tab key a few times. You'll see the focus move through all the form fields and all the links. This is a handy way of navigating the site if you're visually impaired; the screen reader reads out the links as you're tabbing through. But imagine your frustration if you were tabbing through links on this potentially fantastic site:

To create world peace and harmony <a href="foo.html">click here!</a>

To learn the secret of everlasting happiness <a href="goo.html">press this!</a>

To ensure that Celine Dion never sings another note <a href="poo.html">go to this page!</a>

The screen reader will just read out "link: click here .. link: press this .. link: go to this page!", and you'd be so bored you'd surf off somewhere else, missing out on all the fun.

The moral? Ensure that the text in

links adequately reflects what's in the link. It isn't just an accessibility must-do; "click here!" is soooo 1998. Everyone knows links are for clicking.

### HTML Navigation

Here's a really simple rule to help accessibility to your site for people with motor skills problems, the elderly, or kids. Don't have tiny, tiny buttons! It can be very tricky to point your mouse onto the active area.

There's a couple of neat HTML accessibility attributes for navigation in forms called `tabindex` and `accesskey`. The tab index controls in what order the user tabs through the fields on the screen. If you're not happy with the default order, you can define your own. (With multi-nested table layouts, it can get very complicated, very quickly.)

You can add the `tabindex` when making a new form in the "input Tag Accessibility Attributes" dialog box that appears when you're adding new fields. However, you probably won't realize until later that your tab index isn't logical; to amend, right click on the field in the design view of Dreamweaver and select 'Edit tag'. In the Tag Editor, select Style Sheet/Accessibility in the left-hand menu. In the right-hand panel at the bottom of the list of attributes you will see Tab Index; you can add this attribute there (see Image V).

You'll see in the dialog box that you can also enter the AccessKey here, which allows jumping to a form element or activating something by pressing alt+ a nominated key – much like activating menu items in Microsoft Word. Take care not to duplicate stan-

dard IE keyboard shortcuts. You'll need to let the user know that there are Access keys; use inline CSS or a class that highlights the chosen letter (by underlining, for example, as in MS Word)

Enter `<span style="text-decoration: underline;"> p</span>hone number`

would appear on the screen as *Enter phone number*, and the user could jump to this form field by pressing Alt+p.

### JavaScript Navigation

Some browsers, and all screen read-

# "This is a the screen

ers, don't know JavaScript from a wombat. Not usually a problem if you go modern and use CSS for rollovers and are careful with your navigation. The cardinal sin of JavaScript navigation is the use of a drop-down menu that can't be controlled from a keyboard. If you must have a JavaScript drop-down menu for navigation, use a 'go' button for the user to confirm.

For accessible pop-up windows, there's a useful tutorial and solution (which costs \$1.99) at [www.dmxzone.com/go?5483](http://www.dmxzone.com/go?5483).

### Tables

Tables are evil, right? Well, no. Although tables for layout are a redundant technique in these CSS days, tables are still absolutely the only way to go for displaying tabular data – that is, any data that would be appropriate to display in an Excel work sheet.

Once again, the Dreamweaver preferences you set help you; after inserting your table and entering number of columns, padding, etc., in the normal dialog, an additional dialog requiring accessibility features leaps up like *Charles' Angels* to the rescue (see Image VI).

The Caption will be inserted, by default, above the table.

But what's that question about headers? And what's that "scope" in the markup? It tells the browser that the headers are for columns, not for rows. Why does it need to know?

One of the problems that screen-reader users have with tables is that you can't scan back up to read the headers with a screen reader as you can with your eyes. Consider the bus timetable in Image VII.

If you're using a visual browser, you'd

image IV



image V



image VI





# handy way of navigating the site if you're visually impaired; reader reads out the links as you're tabbing through"

just look at the name of your bus stop and scan down the column. But, if you're listening to this with a screen reader, by the time you've heard 08.45, you'll have forgotten whether that's the time at Bruce Avenue or DMX zone.

You'll need to add <th> (table header) tags with a scope attribute to tell the screen reader whether the header is for a column or a row. To make a screen reader tell the user the name of the bus stop before each scheduled time, the mark up is

```
<table>
  <tr> (the first row is the row of
  headers)
    <th scope="col">Bus
  station</th>
    <th scope="col">Town Hall</th>
    etc
  </tr>
  <tr> (all the elements in the
  first row)
    <td>07.10</td>
    etc
  </tr>
```

```
<tr> (all the second row)
</tr>
</table>
```

You can now amend the skeleton code that Dreamweaver has made for you to make accessible data tables, and style it for beauty and grace using CSS.

## Accessible Language

This means two things; the first is the use of language that is simple enough to be understood by people with cognitive

*Bruce Lawson is the content commissioner for www.DMXzone.com, one of the largest independent Dreamweaver communities out there. Every weekday, they publish a tutorial on an aspect of Web development of interest to Dreamweaver users, as well as carry extensions, news, views, and all things Dreamweaver related. bruce@DMXzone.com*

You don't have to be Leonardo to draw a "perfect" circle.

You can use a compass.

You can manually code your dynamic websites, or have a tool do it for you. Use InterAKT products to unleash your creativity when developing complex PHP applications.

**Interakt**  
<http://www.interakt.ro/tools/>

**Professional Web Tools**  
 for Dreamweaver MX PHP programming

Bus Station	Town Hall	Diaz Lane	Cameron Blvd.	Bruce Ave.	DMX zone
07.10	07.20	07.30	07.40	07.45	08.00
07.40	07.50	08.00	08.10	08.15	08.30
08.10	08.20	08.30	08.40	08.45	09.00

disabilities. As this term describes everything from Alzheimer's Disease to Parkinson's to Down Syndrome to dyslexia, I leave it to you to decide whether your Web page on Quantum Physics needs to be rewritten for such users.

But, if a screen reader expects to read English and you've got a French phrase in your content, it would probably get in a horrible tangle. Consider this content:

*Cameron breathed, "Oh, mon amour", as Bruce poured the champagne, and then she raised her glass, exclaiming "Prost!".*

*"I must go and save the planet", replied Bruce, swinging on the chandelier out the window. "Ciao!" he waved as he skied down the mountain towards Blofeld's lair.*

In order to make the screen reader pronounce the words as accurately as possible, and not as if they were English, you need to mark them up like so:

```
<p> Cameron breathed, "Oh, <span lang="fr">mon amour</span>", as Bruce poured the champagne, and then she raised her glass, exclaiming "span lang="de">Prost</span>"</p>
. . . .
```

The language codes are here:  
[www.w3.org/WAI/ER/IG/ert/iso639.htm](http://www.w3.org/WAI/ER/IG/ert/iso639.htm)

## Validating for Accessibility

There's a built-in validator for accessibility in Dreamweaver. Some of the warnings it gives you just don't matter; it nearly always tells you not to use color to identify things, for example, whether you are doing so or not. It just isn't that clever. Use it to hunt down mark-up bugs, but nothing replaces the dreaded "t" word ...

## Testing

No amount of push-button validation can replace testing. Here's my usual testing regime:

1. Turn off images in the browser. Does every red "ain't got no image" cross have alt text showing?
2. Unplug your mouse. Feed it some cheese. Edam is good. While it's tucking in, use the keyboard to navigate around your site. Can you?
3. Turn off JavaScript in the browser. Can you still use every form and every link – especially if you're using popup windows?
4. Okay, assuming you don't have the large sums of money to invest in a screen reader, you can see what a screen reader would read out by simulating the text-only browser known as Lynx. The Lynx viewer can be found at [www.delorie.com/Web/lynxview.html](http://www.delorie.com/Web/lynxview.html) and shows you what your site would sound like with no images, JavaScript, or anything but text. It's a sobering experience.

## Accessible - but Usable?

So, your site is now technically accessible – but is it usable? A big problem can be links. A page with a lot of links before the main content is terribly frustrating for a screen-reader user as they have to listen to all the links before they get to the content they're looking for.

Consider my local council's Web site. I selected a link to see how to pay business rates (real estate taxes) and got the page shown in Image VIII.

The content I want is right there in the middle of the screen – but I have to

listen to all the links first. The solution to this is a "skip links" link right at the top of the page. It can even be white on white (or red on red, or whatever your color scheme is) so that sighted viewers don't see it, but the screen reader will pick it up. It's just a link that goes straight to an anchor tag at the top of the content, thereby bypassing all those pesky links...

## Some Useful Resources

Well, it's been nearly 3,000 words and we've only had time for an overview of the main aspects of accessibility; we haven't looked at Flash, SMIL, or even all of the possibilities in HTML. Don't be put off – we've covered 80% of all the techniques that you need to know. Here's some more resources to look over while I get into my tuxedo for tonight's movie premiere with you-know-who. Now... where are my trousers?


- There's a free DMXzone extension that adds good-looking CSS navigation lists to your page – the List-o-Rama! at [www.dmxzone.com/go?5618](http://www.dmxzone.com/go?5618) – as well as a good many \$2 tutorials on accessibility and standards at [www.dmxzone.com/index.asp?Typeld=28&Catld=691](http://www.dmxzone.com/index.asp?Typeld=28&Catld=691)
- [www.accessify.com](http://www.accessify.com) is another great resource for all things accessibility related. Ian Lloyd, who runs the site, has an exceptionally useful collection of favelets that allow you to check pages at [www.accessify.com/tools-and-wizards/accessibility-checking-favelets.asp](http://www.accessify.com/tools-and-wizards/accessibility-checking-favelets.asp), as well as many other resources at [www.accessify.com/tools-and-wizards/default.asp#6](http://www.accessify.com/tools-and-wizards/default.asp#6).
- The industry-standard book on the subject, *Constructing Accessible Web Sites* (Thatcher et al; Apress) is available on Amazon at [www.amazon.com/exec/obidos/tg/detail/-/1590591488](http://www.amazon.com/exec/obidos/tg/detail/-/1590591488).
- For advice and discussion of all things Web standards-related, go to [www.webstandards.org](http://www.webstandards.org). 

image VIII



This is for you.



This is for you. We put in the features you told us were must-haves, plus a few we just thought you'd like.

## Flash MX 2004 Flash MX Professional 2004

- > Improved runtime performance
- > Professional video support
- > Forms-based development

“It’s a must-have upgrade for current users ...”  
-CNET.com, August 2003



## Dreamweaver® MX 2004

- > True WYSIWYG CSS
- > In-line cross-browser validation
- > Native graphics editing

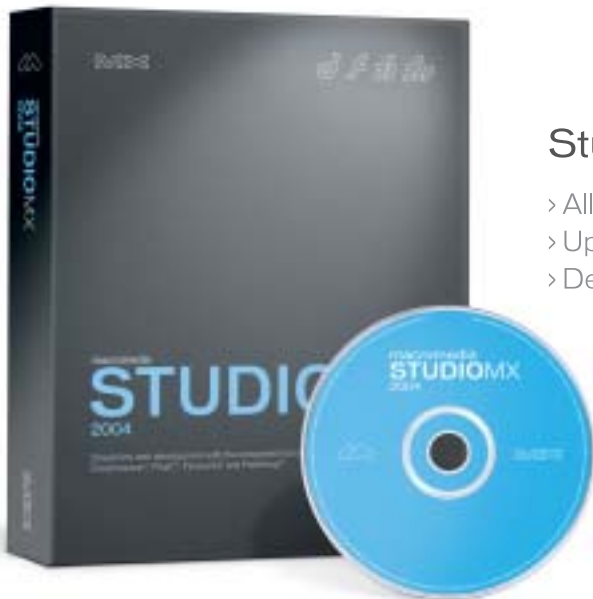
“... a big step forward.”  
-Forrester Research

## Fireworks® MX 2004

- › Faster performance
- › New drawing tools
- › New live effects and photo-editing tools

“... when it comes to web graphics creation, Macromedia’s Fireworks is still the undisputed king.”

-CNET.com, August 2003



## Studio MX 2004

- › All new versions of Dreamweaver, Flash, and Fireworks
- › Updated user interface
- › Deeper cross-product integration



The new versions of Dreamweaver, Flash, and Fireworks in Studio MX 2004 have been awarded the CNET Editors' Choice Award, August 2003.

# Macromedia MX 2004.

Upgrade, try, or buy at [www.mx2004.com](http://www.mx2004.com)

Run with it.





# Pushing Data Out to the Flash Client with XMLSockets

*An essential for games, presentations, and RIAs*

by matthew david

If you've upgraded to Flash MX 2004 Professional or you purchased the FireFly Data Connection Kit for Flash MX, then you've probably read a lot about how easy it is to have Flash pull data into a Flash Movie. The only problem

Other examples can include real-time data transfer for information such as stock numbers, sales figures, and supply chain management. These areas require a tool that can push data out to a client.

tems, chat rooms, and low-latency applications in which the user opens up a connection with the server and requires that the server and the client pass data back and forth. The standard passes full and complete XML documents. The impact enables applications to "chat" over network lines.

You can dig through the specifications of HTML and you will not find support for XMLSockets. But XMLSocket connection has been a standard feature within Flash since version 5, with only one security feature to worry about (see sidebar).

Through ActionScript you can connect to an XMLSocket server, send and receive XML-formatted data, and close connections to the server.

## XMLSocket Server

For XMLSocket information to be passed back and forth between a client and server you need the client (which you know is Flash) and the server.

Although XMLSocket technology is relatively new, there are a number of notable XMLSocket servers on the market.

table 1

Method	Description
Connect	Opening a connection to a XMLSocket server is done through identifying the host server and port number. For instance you can write mySocket. . connect("12.101.101.72", "1024"); where the IP address is listed first in the parenthesis and the port is listed second.
Close	As the name implies, this method closes the connection to the XMLSocket server identified by the new XMLSocket ID. To close the mySocket XMLSocket connection you can write mySocket.close();
Send	You will want to send data to the XMLSocket server. This is done with the "send" method.

is that this is not how applications work in the real world.

In the real world there has to be a constant conversation between the client and the server application. I am going to use e-mail as an example. If you have a traditional POP3 e-mail client (one you get through your ISP or Web hosting company) you have to actively connect to it and check if you have e-mail. You can do this by either hitting the "send and receive" button or setting your e-mail recheck setting to an arbitrary time. A more efficient model would have your e-mail client only check and download e-mail when it's available. Enterprise e-mail systems, such as Microsoft Exchange, leverage both the e-mail client on your desktop and the e-mail server. When e-mail arrives at the server a message is sent to the e-mail client that a new message has been delivered. The server, in this case, pushed data out to the client.

## Using XMLSocket Connections

As part of the evolution of XML, a subset of XML technology called XMLSockets has emerged. XMLSockets are great for solutions such as e-mail sys-

code 1

```
connect = function (ip, port) {
    // Opens Connection to XMLSocket Server
    xsock = new XMLSocket();
    xsock.onConnect = sockConnect;
    xsock.onXML = processXML;
    xsock.connect("12.101.101.72", "1024");
    xsockStatus = "[Connecting]";
};

sockConnect = function (bool) {
    // Validates whether a session has been started with Server
    if (bool) {
        xsockStatus = "[Starting Session]";
        resetGame();
    } else {
        xsockStatus = "[refused]";
    }
};
```

- **UNITY 2.0:** ([www.moock.org/unity](http://www.moock.org/unity)). Colin Moock, esteemed author of the ActionScript Definitive Guide series, has created an easy-to-use and powerful Java-based XMLSocket server. What I like is that you can download a free developer's version of the server before paying up for the full version. You can also buy a sample application that has over 10,000 lines of ActionScript 2.0 code.
- **Shovemedia:** ([www.shovemedia.com](http://www.shovemedia.com)). Shovemedia is another Java-based socket server. The Pong game in this article uses this server for multiple players to connect to.
- **Swocket:** (<http://swocket.sourceforge.net>). Swocket is an open source socket server built in Python.

You can find an even more comprehensive list of XMLSocket servers on the XML/XMLSocket server list over at the Flash Wiki (<http://chattyfig.figleaf.com/flashcoderswiki/index.php?XML%2FXMLSocket%20Servers>).

## XMLSocket Class in Flash

Not a lot has changed since Flash 5 with the use and functionality of the XMLSocket class. As with other objects in Flash, before you can begin using the object you must set a variable that declares the new object. The following will declare the variable mySocket as a new XMLSocket class:

```
mySocket = new XMLSocket();
```

Once you have declared you are using an XMLSocket you can connect, send, and close the XMLSocket class. Table I shows methods and their descriptions.

There are also event handlers you can use to push and pull data to and from an XMLSocket server. These include:

- onConnect
- onClose
- onData
- onXML

If you have worked with the extensive XML features in Flash then you will see

that the XMLSocket server is more limited in features. However, there is a lot you can do with those features.

## Creating a Pong Game with an XMLSocket Server

One of the tried-and-true examples out there using XMLSocket servers is Pong. Pong is a simple game where two or more people connect in a game room and play against each other.

You can download the complete Flash source code to run this game from [www.sys-con.com/mx/sourcec.cfm](http://www.sys-con.com/mx/sourcec.cfm). You will need an XMLSocket server to run the program. The source code is Shovemedia's Pong but you can use any XMLSocket server, such as Unity.

I won't go through the mechanics of the game in detail – there are about 500 lines of code. What I want to direct your attention to is the connection to the XMLSocket server and how it is used to create XML data to pass the players' moves back and forth to each other over the Internet.

Download the files for the Pong Game. You will want to open pongGame.as. Go to line 287. You will see the code shown in Code 1.

Essentially, you have two functions. The first connects to a server through an open port. The second function prints to the Flash Player whether a connection has been successfully made.

## Summary

Pushing data to and from a Flash movie is essential for games, presentations, and Rich Internet Applications. XMLSocket connection gives you an alternative to competing solutions such as XUpdate, Flash Remoting, and traditional GET/PUT HTTP protocols. ☞

*Matthew David is the author of Flash Books (Flash MX Magic, Building Great Flash MX Games, Flash 3D Bible and Flash MX Communication Server Bible), has contributed to over a dozen additional books, and writes 300 articles a year for more than 20 international magazines. His area of speciality is next generation Web solutions.*  
[mdavid@matthewdavid.ws](mailto:mdavid@matthewdavid.ws)

There are two levels of security you need to adhere to when working with XMLSockets. The first level is with the XMLSocket server. When a connection to an XMLSocket server is opened it will remain opened until the connection is killed or closed. Preferably you want to programmatically close the connection instead of ripping the code and "killing" the connection. Latent connections can remain open utilizing precious RAM.

The security concern is that an XMLSocket server does keep an open connection with the application. If you hack in to a system and connect to the XMLSocket server then you will only have to authenticate once to connect to the system. In more traditional Web application solutions, each time you make a request of the server you authenticate, open a connection to the database, collect your information, and then close the connection. XMLSocket servers keep the lines of communication open and stream the data to the Flash client.

While this may not be very secure, you can make it secure by moving your Port connection to the XMLSocket server to above 1024.

The other level of security you need to keep in mind is the new heightened security model surrounding the Flash 7 Player. By default, you will not be able to perform cross-domain authentication. The only way in which you can do this is to have an XML formatted Policy file. A Policy file is a simple affair. It is an XML document that describes which domains can access the content of your site. A simple policy file could include the following:

```
<?xml version="1.0"?>
<!-- demonstration policy file -->
<cross-domain-policy>
  <allow-access-from
domain="110.216.0.40" /> <!-- allows same
IP addresses to connect to the data -->
</cross-domain-policy>
```

You can add a policy file that simply says <allow-access-from>, which will have the effect of opening all content to Flash developers for the site the policy file is stored on.



# The Flash MX 2004 Professional Data Architecture

*Crafting rich Internet applications*  
by owen van dijk

As you might have noticed, the current incarnation of Macromedia Flash, Macromedia Flash MX 2004, has evolved from a simple Web animation package to a full-fledged IDE that allows both developers and designers to work with graphics, animation, video, and sound and spice it up with external sources like databases and Web services.

The more adventurous among us can hook it up to servers like Macromedia Flash Communication Server or the XMLSocket server, Unity 2, adding streaming video and sound, multiuser environments, and Web cam support to their applications. For beginners, the possibilities seem endless and the learning curve is steep. In this article I will guide you through the new data architecture introduced in Macromedia Flash MX 2004 and explain the concepts and ideas that allow you to work with external data. Soon you will be crafting "Rich Internet Applications" too!

## A Little History

If we look at the current technologies and frameworks for working with data we can identify the following key technologies:

- The Microsoft world has a technology called ADO (.NET) that gives developers working with ASP, Visual Basic, and C# access to data sources.
- Java has JDO and JDBC for working with external data.
- The open source world (PHP, Perl, Python) define their own frameworks, each targeted toward specific features and constructs of the language.

Until now, Flash didn't have a standard framework to access external data, thus developers coming from different backgrounds were emulating their preferred framework when working with Flash. For a developer coming from an ASP.NET background, there was no Data Reader, so the only way to have the same workflow was to write your own version

of the Data Reader. The advantages of creating your own framework might be obvious:

- You can really customize your framework for your specific need.
- You don't have to learn new technologies or new APIs; you simply write your own.

But look at the drawbacks:

- It's very time consuming.
- There's a steep learning curve for new developers.
- You're reinventing the wheel.

Flash MX 2004 Professional introduces a standardized and easy way to connect to external data sources, manage data, and make the data visual by binding it to User-Interface controls. This new architecture consists of three layers:

1. **Data Connectivity layer:** This is the Data Access Layer (DAL) with new components providing a consistent interface for sending and receiving

table I

### What works:

A movie is allowed access to data from the same domain	www.clockwork.nl	www.clockwork.nl
A movie is allowed access to data from the same subdomain	data.clockwork.nl	data.clockwork.nl
A movie is allowed access to data from the same ip address	195.64.40.15	195.64.40.15

### What doesn't work

A movie is not allowed to access data from another domain	www.clockwork.nl	www.google.com
A movie is not allowed to access data from another subdomain.	data.clockwork.nl	www.clockwork.nl
A movie is not allowed to access data from another domain if it is served from an IP address, even if the IP address points to the same domain	95.64.40.15	data.clockwork.nl

table II

### Policy files @ work:

www.clockwork.nl/crossdomain.xml	www.data.clockwork.nl
data.clockwork.nl/crossdomain.xml	www.clockwork.nl
195.64.40.15/crossdomain.xml	data.clockwork.nl
www.clockwork.nl:8080/crossdomain.xml	data.clockwork.nl:808



data to and from Web services, databases, and XML files.

2. **Data Management layer:** A repository layer, containing a DataSet and DataHolder component. These components give you an interface to perform basic data operations such as editing, sorting, and updating.

3. **Data Binding layer:** You can have the most advanced data floating around in your application, but if you do not visualize it, for example showing it in a TextArea component, it isn't of much use. The Data Binding layer, which is represented in the IDE through the "Bindings" tab, contains objects such as Formatters and Encoders (see Image I). Here you share the data between components and visualize them by binding the data to "data-aware" components such as the new Macromedia User Interface Components (see Image II).

## Security

Macromedia has tightened the security model in the new Flash Player. This impacts us when we want to work with external data. The rule is pretty simple: you can only access external data if it's loaded from the exact same domain. Table I shows you some examples.

However, there is a simple and elegant solution to this – a feature called "Policy Files," simple XML-based files that have to be uploaded to the server. Table II shows you some examples.

The format of the policy file is very straightforward:

- It contains a single node: "<cross-domain-policy>".
- It has one or more sub nodes named "<allow-access-from>".
- These nodes have one special attribute named "domain" that has the following values:
  - Wildcards ( \* )
  - Exact Domain
  - Exact IP address

Here are two examples:

1. Access all areas

```
<?xml version="1.0"?>
<cross-domain-policy>
  <allow-access-from domain="*" />
</cross-domain-policy>
```

This is the simplest form of a policy

file, stating that anyone is allowed to access data that resides on your server.

2. Specific access for some domains

```
<?xml version="1.0"?>
<cross-domain-policy>
  <allow-access-from
    domain="www.clockwork.nl" />
  <allow-access-from
    domain="*.cnn.com" />
  <allow-access-from
    domain="66.151.149.10" />
</cross-domain-policy>
```

Here I specified explicit access to only a few domains.

The following actions are affected by the new security restrictions:

- XML.load
- XML.sendAndLoad
- LoadVars.load
- LoadVars.sendAndLoad
- LoadVariables
- LoadVariablesNum
- XMLSocket.connect
- Flash Remoting
- Import of a shared library

## Web Services

Software is often seen as an application, something you buy in a store, shrink-wrapped in a box or custom made by a software development company. It is usually distributed on a medium such as a CD-ROM, or in the case of a Web application accessed through a browser. A Web service is a piece of software that can be used through the Internet. You do not own the software, but instead use the service to solve a specific problem. It's much like going to the hairdresser for a haircut or cleaning service to get your laundry done.

Web services are built on standard Internet protocols:

- HTTP GET/HTTP POST
  - Both use HTTP as their underlying protocol.
  - Both have been used widely since the early days of Web programming.
  - They encode values into name/value pairs.
- SOAP on HTTP (Simple Object Access Protocol)
  - Uses HTTP as underlying protocol.
  - XML-formatted messages.

-We can use SOAP on other protocols.

Let's have a look at how you could use Web services in an older version of Flash.

- You had the ability to use HTTP GET and POST through the getURL command since version 4.
- You could access SOAP Web services by creating an XML interface through a scripting language like PHP, ASP, etc. (the Flash Player didn't support the required SOAP Action header).
- You could use AMF, a proprietary protocol by Macromedia Flash Remoting that was fast because of its binary format.

All of these methods require in-depth knowledge of the Flash environment, and sometimes an awk-

image I



image II



image III



image IV





ward workflow for working with external data. In the following section I show you some examples of how to access XML files and Web services with no scripting.

## Examples

Now let's put our knowledge into practice. I will show you two simple examples that will illustrate the concepts and workflow of the new data architecture. (Source code for these examples can be found at [www.sys-con.com/mx/sourcec.cfm](http://www.sys-con.com/mx/sourcec.cfm))

### Hello World: Binding Data to UI Components

Of course we'll start with a simple Hello World program. The common workflow follows:

1. First we have an XML file named "index.xml" that has the following structure:

```
<?xml version="1.0" encoding="UTF-8"
  ?>
- <root>
      <title>Hello World!</title>
</root>
```

2. Open a new document in Flash and drag the following components to the stage:
  - XMLConnector
  - Button
  - TextInput

3. Position them on the stage so it looks like Image III. Select the XMLConnector and open up the Component Inspector (if it is not visible, select it [see Image IV]).

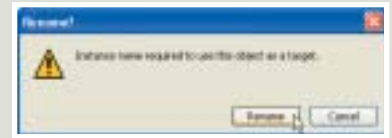
4. The XMLConnector contains several properties; use the settings shown in Image V.

5. The key when using an XMLConnector is defining a schema of your XML file. Flash can help you with that, by analyzing an imported XML file (see Image VI).

6. Select the "index.xml" file and Flash automatically shows you a schematic overview of the structure of the XML file (see Image VII).

7. Now in the Bindings panel we can add a binding by clicking on the + symbol.
8. Select the title node and click OK.
9. We want to send the data from the XMLConnector to UI control, so we select "out".

Owen van Dijk is a software engineer at Clockwork BV in Amsterdam, the Netherlands. His daily work consists of creating software, Web sites, and rich Internet applications for clients such as Bacardi, Coca-Cola, Citroen, and Terres des Homes.  
owen@sys-con.com



10. Now you can bind the "title" node to the TextInput field.
11. Notice that Flash gives a warning when you don't give your component an instance name (see Image VIII).
12. The only thing that needs to be done is triggering the XMLConnector. Select the Button and open the Behaviors panel (see Image IX).
13. Choose Trigger Data Source and select your XMLConnector.
14. Test the movie... and you have made your first application, which accesses an external XML file and copies the value of the title node into the TextInput field with no scripting at all! (See Image X).

### Custom Formatter and Web Services: Formatting Output from a SOAP Response

In our first example, we learned how to work with the XMLConnector and binding data from our Connector to a User-Interface control. Let's take it a step further! I will show you how to use a Web service in Flash MX 2004 Professional and how to format the response returned by the Web service.

1. We're going to use the "Global Weather" Web service by WebserviceX.NET ([www.webserviceX.net](http://www.webserviceX.net)).
2. Open a new Flash Document and drag a WebService Connector to the stage.
3. Open up the Component Inspector and use the following value for the WSDLURL property: <http://www.webserviceX.net/globalweather.asmx?WSDL>. Flash will automatically find the opera-



Image X

The result should look something like Image XIII.

tions you can use. Select "GetWeather" from the dropdown shown in Image XI.

4. If you open the Schema tab, you will see a visual representation of the Web service, which parameters it requires, and what is returned (see Image XII).

5. It's time to add the UI controls to our application – drag the following components on stage and lay them out like this

- a. TextInput named CityName. Set the text value to "Amsterdam".
- b. TextInput named CountryName. Set the text value to "Netherlands".
- c. Textarea named Results.
- d. Button. Set the label value to "Get Weather".

6. Bind the various properties of the Web service to our controls:

- e. The parameter CityName should be bound to the TextInput named CityName.
- f. The parameter CountryName should be bound to the TextInput component named CountryName.
- g. The results should be bound to our Result TextArea.

7. Finally, trigger the WebService-Connector with a click of the button, using the behavior "Trigger Data Source".

8. If we test our movie we get the result shown in Image XIV.

9. Hey! Look what happened...We get an XML-formatted string back. That's not what you want, is it? Fortunately, there is a solution.

10. If you go back to the Bindings panel and select the Result binding, you should see a Formatter option.

11. From the numerous options, you can choose a Custom Formatter to for-



Image XI

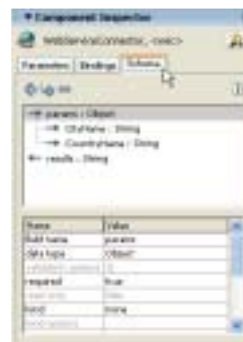


Image XII

## Macromedia Authorized Training



**Future Media Concepts**  
Training a New Generation of Digital Artists™

FMC OFFERS THE COMPLETE  
MX CURRICULUM:

- Coldfusion
- Flash
- Dreamweaver
- Director
- Fireworks
- Freehand



**macromedia  
AUTHORIZED  
TRAINING PARTNER**

Nationwide On-site Training  
877.362.8724

www.FMCtraining.com

New York • Boston • Philadelphia • Washington DC • Miami • Orlando



mat the return value from the Web service. A Custom Formatter accepts an Actionscript class.

- This class extends from `mx.data.binding.CustomFormatter` and needs to implement two methods:
  - format
  - unformat
- If you go back to step 9, note that the Web service returned an XML-formatted document that when saved as a file has the structure shown in Code I.
- Code II formats this XML document and returns the values of the nodes in a formatted string.
- If you save this file as "Weather

Formatter.as" in the same directory, all you need to do is use it in the Custom Formatter (see Image XV).

- Now if you test the movie again, you will see a nice formatted result instead of an XML document (see Image XVI).

### Conclusion

Macromedia Flash MX 2004 Professional adds an easy and simple workflow for developers to access external data resources and perform basic data operations on it. For beginners it makes it easy to create simple RIAs with little scripting, while the heavy ActionScripters among us can really customize input/output with the Formatters and Encoders. ☺☺

code I

```
<?xml version="1.0" encoding="utf-8"?>
<CurrentWeather>
  <Location>Amsterdam Airport Schiphol,
  Netherlands (EHAM) 52-18N 004-46E -
  2M</Location>
  <Time>Oct 02, 2003 - 02:25 PM EDT /
  2003.10.02 1825 UTC</Time>
  <Wind> Variable at 2 MPH (2 KT):0</Wind>
  <Visibility> 3 mile(s):0</Visibility>
  <SkyConditions> partly
  cloudy</SkyConditions>
  <Temperature> 57 F (14 C)</Temperature>
  <DewPoint> 51 F (11 C)</DewPoint>
  <RelativeHumidity> 82%</RelativeHumidity>
  <Pressure> 29.85 in. Hg (1011
  hPa)</Pressure>
  <Status>Success</Status>
</CurrentWeather>
```

code II

```
class WeatherFormatter extends mx.data.binding.CustomFormatter {
function format ( rawValue : String ) :
String {
var retVal : String = "";
// convert to XML object
```

```
var myXML : XML = new XML(
rawValue );
myXML.ignoreWhite = true;

// parse second node to XML tree
myXML.parseXML( myXML.childNodes[1] );

// loop through childNodes
for ( var i : Number = 0; i <
myXML.firstChild.childNodes.length; i++ ) {
retVal += myXML.firstChild.childNodes[ i
].firstChild + newline;
}

return retVal;
}

function unformat ( formattedValue : String
) : String {
var retVal : String = "";
return retVal;
}
}
```

image XIII



image XIV



image XV

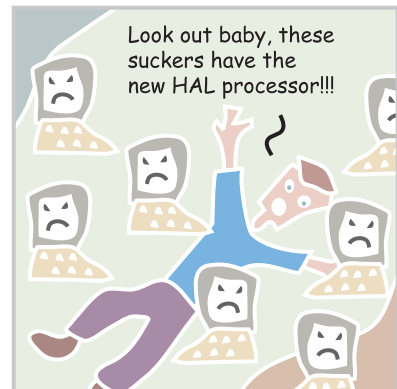
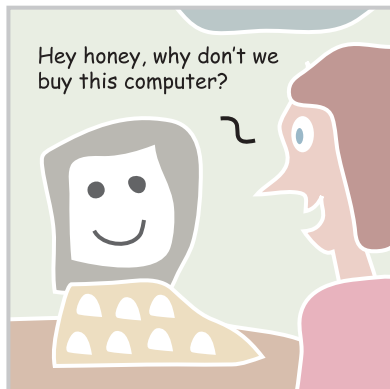


image XVI



# xile

written & illustrated by louis f. cuffari ①



# Think about it ...



## It's everybody's PDF™

Finally, a software company that offers affordable yet flexible PDF solutions to meet every customer's needs. Using activePDF™ to automate the PDF creation process eliminates the need for end-user intervention so your employees can concentrate on what they do best.

Licensed per server, activePDF solutions include a COM interface for easy integration with ColdFusion. Dynamically populate PDF forms with information from a database, convert ColdFusion web pages to PDF on the fly, dynamically print reports to PDF using CF and much more. Users can also merge, stitch, stamp, secure and linearize PDF, all at a fraction of the cost of comparable solutions. Download your free trial version today!



[www.activePDF.com](http://www.activePDF.com)



by jesse r. warden

# From Design to **Reality**

**This article** will cover techniques in dealing with application state and **complex** animations. Additionally, I've included a **few quick** primers on how to implement some of the designs a **Flash** developer can be expected to put into Flash as well as a **sample** application file.



There are many articles dealing with application state, from Visual Basic to PHP. My goal is not to rehash those topics, but to show you some tips and tricks I've learned in merging state and animation in Flash, since one of the main features that draws people to utilizing the Flash Player in a browser is its ability to keep application state.

The term "application state" is very broad, especially when applied to Flash development. Through the use of a timeline, you can lay out your application's states. A programmatic example is keeping the user's name in a variable that doesn't go away when the user navigates from page to page, as would happen on a Web page since the Internet is stateless. A less attractive example is playing "variable toss," where the variable is thrown on the end of the URL variable, with developers praying they never break the chain... or making the assumption that cookies are working on the end user's machine and attempting to use session variables, which is typically done in a PHP or ASP/.NET environment.

This article addresses the ability for components to have state utilizing complex animations. The challenge is changing state and having the interface follow suit, as well as transitioning between states. Because Flash allows timelines within timelines, you can easily build complex animations and nest them in your application state timeline without affecting it adversely in timing. More simply, a designer gives you a complex animation that you're supposed to make use of in your application and/or one of your components. I'll first cover ways you can go about implementing the design, and describe the pros and cons of each technique. Next, I'll describe methodologies in constructing your application as a whole. Finally, I'll describe the code used to keep track of state and manipulate the animations within your component.

If you're already familiar with implementing designs in Flash, and you have your own Flash application construction methodology, then you can skip to Part 3. A discussion on what can be done in the management and meetings up front to get to this phase can be seen at my Web site ([www.jessewarden.com/archives/000309.html](http://www.jessewarden.com/archives/000309.html)).



## Part 1 Design Implementation Primers

What's the best way to take a designer's comp, a compositor's video, an audio-ophile's mix, or an illustrator's layout and merge it with your code?

### Image Editing Quick Primer

First, it helps to know your options. In Flash 5 and up, you can import PNGs, a bitmap format that is lossless (no quality loss in saving to this format) and supports up to 32 bit (8 bit alpha channel). A Photoshop comp can be exported straight from Photoshop to PNG, or, if the effect layers are flattened, brought into Fireworks and converted there. For complex-shaped interfaces and interface elements, this is the most common way to implement them inside Flash so you don't get the white or black background filling in the gaps. Since bitmaps are drawn to the screen in squares, designers use an alpha channel (unknowingly) to mask certain areas. Wherever possible, use a JPEG compressed outside of Flash. Square bitmaps are faster to render than their irregular-shaped bitmap counterparts (they don't negatively affect animation speed as compared to irregular bitmaps). Additionally, depending on the depth of your target monitor, sometimes the bitmap alpha channels show up. Yes, an invisible part of the image can be seen, I know. On 24 bit/16 bit monitors, this can sometimes happen with PNGs. The way to fix it is to put the PNG into a graphic and use the graphic instead of the PNG. Then, duplicate the image on a new layer and guide one of the layers (so you have a copy after you modify the copy; you never know when you'll need to go back to the original, and guided assets do not affect the final SWF file size [exclusion of sounds]). Select the image that is not on the guided layer, and then choose Modify → Break Apart. Then select the Lasso tool. You'll notice three new tools on the bottom of the tool

palette. Choose the Magic Wand Properties, change Threshold to 0, Smoothing to Pixels. This will usually ensure that you'll be able to select all of the alpha channel areas and delete them. Now you don't have to worry about the bitmap falling apart because it's contained in the graphic, and you can animate the graphic. However, the alpha is gone, but you still have your irregular shape intact. The one gotcha is this "auto-smoothes" your image. This option (Flash's custom smoothing of images) can usually be turned off by right-clicking the asset in the Flash Library, selecting properties, and unchecking the checkbox. Now you're stuck with it.

### Quick Video Primer

You can import the video directly into Flash MX and Flash MX 2004 and Pro. However, it is highly recommended that you do two things to ensure that your video will look professional:

- Obtain an uncompressed version of your video. If you compress video twice, it'll look worse and a lot of times come out bigger in file size than it was. "Garbage in, garbage out."
- Purchase Sorenson Squeeze. This product is the professional way to compress your Flash video; you get good control and good results, better than you can get with Flash alone. Then you can import the FLV, or make a video object in Flash and import the FLV into it by right-clicking on it in the Library and then selecting the import button.

### Quick Audio Primer

As with images, it's best if you compress your audio outside of Flash. However, this really is best only if done with MP3 compression. For shorter sounds, it's okay to use Flash. Flash has ADPCM for short sounds, its own Speech



codec for voice-overs (MX+), and MP3 internally, or you can import your own MP3, which won't be recompressed, or just use it RAW. Believe it or not, you're better off mixing down a bunch of sounds rather than compressing each one individually. Sometimes you can gain very slight compression increases if you separate the sounds out, as some sounds compress better than others, but the loss in processing power from uncompressing and playing all of those sounds together, as well as using most if not all of the eight sound channels Flash provides, is not worth the marginal file size savings, not to mention a bigger timeline.

### Vector Art Primer

Although you can import some Illustrator files, my suggestion is to save yourself some heartache by transferring it to a Freehand file or making a bitmap equivalent in Photoshop. If this is not an option, write down your color values – sometimes things do not match up exactly. The same goes for PDF and EPS.

image I

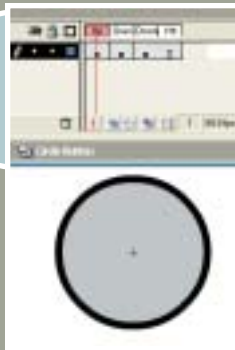


image II

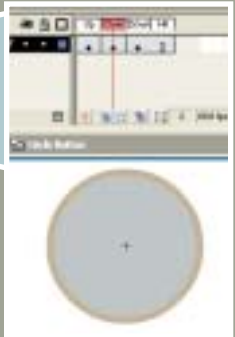


image III



code I

```
function getNextFrameLabel ( newState ) {
// new state is where you want to go
switch ( newState ) {
case "shutdown":
// current state is where you (the animation)
// are at currently
switch ( currentState ) {
case "opened":
return "close to shutdown";
case "closed":
return "shutdown";
case "shutdown":
return null;
}
case "opened":
// current state is where you (the animation)
// are at currently
switch ( currentState ) {
case "opened":
return null;
case "closed":
return "closed to open";
case "shutdown":
return "startup";
}
}
}

function getState(){
return this.state;
}
```

code II

```
function setState(newState){
// get your next frame label to go to
var f1 = getNextFrameLabel(newState);
// show the animation
this.gotoAndPlay(newState);
// create a small engine to see if your at your end state
yet
// so you can populate the components that will appear
// presumably when the transition animation is done
var process_mc = this.createEmptyMovieClip("process_mc",
999);
// your constant names match the framelabels' name
process_mc.targetFrame = this[f1];
process_mc.f1 = f1;
process_mc.onEnterFrame = function(){
if(this._parent._currentframe == this.targetFrame){
// all callback letting you know you can set the data now
this._parent.onReachedTarget(f1);
// thanks, preciate your help movie clip, now die!!!
this.removeMovieClip();
}
};
}

function onReachedTarget (frameLabelThatIsDone){
trace("Done playing: " + frameLabelThatI// function
that populates your componnets
this.setData();
}
```



## Part 2 Methodologies

### Method 1: The RAD Prototype

The quickest way to implement an interface is to simply import it using the methods previously mentioned and lay it out on the stage. It helps to sometimes import a screen-capped version and put it on a guide layer to help in your positioning. Then, you place your code on the objects. This will be very familiar to VB form developers.

Using the new Flash Pro features – the screens, the behaviors panel – and using databinding, you can knock out most interfaces that would take weeks, in days; day's in hours. The process is the same: lay out your interface on a series of forms that are reminiscent of PowerPoint (supporting nested forms), and then click on one of the form elements, click the behaviors panel, and apply your code.

#### Pros

This method is very fast. So fast, in fact, that this is where the multimedia author achieved fame in the Director days, as the process was very similar. The benefit is the media support in Flash; in addition, the flexibility and cross-product editing are superior in speed to other programming environments, save Director (which supports even more media types than Flash does). If something needs to be done in a hurry, this is the way to go. Behaviors created by programmers for the Behaviors panel can still allow very complex interactions and data handling to take place.

#### Cons

OOP? What's that? Although Flash's new data binding and components help you connect to external data sources of all types to keep your application dynamic, the structure of the application is pretty much stuck that way. From the placement of objects to the code placed on the objects, all are there to stay. If you wish to modify code, you must go to the object as well as edit an object's proper-

ties, and it isn't always apparent what has code on it and what doesn't. For wizard-type applications, slide shows, or prototypes, this doesn't usually pose a problem, but for application design, it's not very flexible nor centralized from a coding standpoint.

### Method 2: The Module Approach

Another way that holds much sway in the Web-developed Flash sites arena, as well as CD-ROM and application disciplines, is the use of loaded SWF files, otherwise known as loading movies. Each SWF forms a component, section, or module of the entire project. Flash movies have the ability to load each other either into themselves or one on top of the other, forming levels. As with colored acetate (transparent paper that allows you to write text and images upon it, which you then place on an overhead projector that shows the images on the screen or wall behind you), users have no idea as they are just seeing a bunch of elements on the screen like they usually would. However, this is a very powerful option that has a great many benefits (and pitfalls). I usually recommend this for CD-ROM-type projects, large Web sites, older Flash projects that use Flash video (whether a bitmapped generated sequence or FLV file imported into a blank FLA file, exported alone into a SWF), or projects that have a lot of media such as sounds, video, and large designs. Sites such as 2advanced ([www.2advanced.com](http://www.2advanced.com)) pioneered this methodology into the limelight. It has a certain affinity for design types because they are used to compositing concepts, more so than programmers.

#### Pros

The first thing you'll notice is that your Flash files export a lot faster in a bunch of smaller files than in one big FLA file. This is especially valuable when

you're trying to test some small coding functionality, but you were originally forced to wait for Flash to compress your sounds and video again, which could take many minutes. Another nice thing is that your application uses only the RAM it needs when the content is actually loaded. Although there are still other minutiae loaded in, performance is not compromised.

You can unload the levels you're no longer using, further helping performance. Since Flash still "renders" objects off-screen, this is a great technique for keeping performance in tip-top shape.

From an interface perspective, this allows you to separate out different pieces of the interface to be present only when needed, as well as the functionality of that interface and specific states. You can have many different SWF files with the same menu, but different data, or different colors, or different menu shapes, and so on. However, loading in that SWF will only change the level or current SWF being displayed, so it keeps things pretty flexible. You can also divide functionality in your application into movies instead of deeply nested components. You can still use components, mind you, as well as Branden Hall's Outlet component ([www.waxpraxis.org](http://www.waxpraxis.org)) to allow your SWFs to act as independent components. One level can hold your sound, one can hold the Web service code, three can hold your interface and all its pieces and states, and one can act as the manager for all of the levels.

Videos, kept in their own respective SWF files, are easily loaded in and out. This helps in previewing just the videos as well as updating their compression levels without recompiling the entire application; compressing video takes a lot of time.

Web site wise, users have to download only Web site content they actually view. This also helps in showing dynamic pages without refreshing the page (or as some call it, causing a "postback").

## Cons

Although levels are very flexible, each SWF file streams. You have to keep this in mind when expecting data from other SWFs. Trying to access a variable in SWF that you just loaded into a movie clip or `_level` will fail miserably. Even worse, it may work once in a blue moon because the file was cached and you got lucky, and then you're left wondering if something is wrong with your code, further deviating you from the true problem.

Web site wise, you need to provide a preloader for each SWF you load, or the user will be left wondering why the application is slow the first time around. However, the biggest pitfall is mismanagement. You need to micro-manage the heck out of your `_level` structure as well as your file structure. Even adding just one folder to the file path mix can get confusing when levels start loading each other. Constantly updating your structure on paper helps as does keeping track of who accesses who, when, etc. However, the real problem comes at crunch time. When you want to drastically change the application or site flow, especially when animations already time the loading of certain levels, it gets really difficult not only to modify, but also to keep track of those last-minute changes.

Although it's improved a little in 2004 (e.g., showing how a level can stream in when loaded) you can debug only one level at a time using Flash's debugger. The Flash Player is single threaded, but it's sometimes confusing to keep track of which movie does what and which movie you're currently debugging, causing you to play back the sequence of events in your mind. Overall, debugging is sometimes made more difficult.

Another con is not using shared libraries and having the same global asset duplicated in multiple movies like the `FScrollBar`. Although a background graphic can be in its own movie and remain in the back, a scrollbar is not so flexible. There may be multiple movies that need that, and this generally forces you to use a shared library and keep hard track of what uses that asset. Otherwise, you have 10K duplicated unnecessarily in all the SWFs that use the same scrollbar. Shared Fonts, however, have to be in `_level0` to work this way. Another odd thing to note is that, contrary to the per-

formance increase, `_level0` has the maximum frame rate. If you're seriously trying to optimize your animations, nothing will be as fast as `_level0`.

Loading additional movies causes additional overhead, sometimes slowing your animations down. However, if they couldn't run as well, even without using levels (as the effect is marginal), then you better rethink your animation and not your assembly methodology. Finally, not being able to preview your animations as a whole unless you compile multiple files can sometimes slow things down in the long run. Having to go through points C, D, and E just to reach point B gets frustrating after awhile. Even though the user needs to, you sure as heck don't during development.

For more pros and cons, please see [www.jessewarden.com/archives/000189.html](http://www.jessewarden.com/archives/000189.html) as well as the `chattyfig` archives; many people have offered their opinions on the methodology.

## Method 3: Application SWF

The simplest way to implement an interface in your application is to have all of your components/movie clips and the interface parts in the same FLA file. You implement/skin them in, and just separate the two in your library.

### Pros

Everything is consolidated; you only need to load the SWF up initially; and after that everything is already there for you (excluding external data calls). Using a combination of `attachMovie` and `removeMovieclip/unloadMovie`, you can easily make it dynamic enough because everything you need is right there. Finding a component or design asset is easy, and updating that asset is a pretty quick process. Compile one SWF, and you're done.

Depending on how you design it, you can even use Method 2 to load it in as a self-contained application. Users don't have to load any additional data, they just suffer once, and all of your assets can be loaded prior to frame 1. Not having to worry about preloaders or preloading assets in is a big help.

### Cons

Try this technique on the Web and you'll get burned. First, the export before

first frame option that most components come with in MX and 2004 forces that data to be downloaded first, before anything on the first frame is shown. This causes a lot of newbie Flash programmers to wonder why their users are seeing a blank white screen for seconds on end, and their preloaders don't work. Second, depending on the scope of your project, compiling as well as initialization can take a long time. First, compiling all of that code as well as compressing images, audio, and video all at one time, every time, can drag out your total compile time. If you're just making a quick code edit and wish to test it, or even a compression modification on one asset, all others must be recompressed into a new SWF – not good. Additionally, using this FLA in a source-safe type of environment (CVS) makes it really difficult because doing one minor change doesn't necessarily mean you have to check it in again, does it? For those who don't wish to have binary files in a source control system more than is necessary, this way certainly can rack up the file size, quick.

I recommend this method only for intranet/kiosk/local hard drive delivery projects, and then only for projects that don't use a lot of big media (video, sound, high-res images).

## Method 4: Shared Libraries

These poor guys have gotten a lot of flak from both the designer and programmer camps – designer because they are confusing to implement and programmer because they didn't always work under every condition (like multi-tier). Later versions of the Flash 6 Player have remedied this, and I haven't found any problems utilizing these in Flash 7. The concept of shared libraries is that you have an asset (symbol, i.e., graphic, button, movie clip, font, etc.) in one SWF file. You export that to be available at runtime using the `linkageID` like you normally would, but you also check the export for runtime sharing and identify what the SWF file name will be (usually the same name as the file it's currently in, as Flash by default exports out `cow fla` to be `cow swf`). You can then either drag this symbol to another FLA file to be shared, or create a new symbol in the other FLA file, and then import it in. This allows multiple SWFs to use the

d

f

fu

fb

fd

db



bols to a SWF... because it's already done. At author time, you still have a local copy in your FLA, though, so your FLA may still be just as big; this allows you to break the connection if the symbol needs to be unique to that SWF for some reason. Updating one SWF of components to update the look and feel or functionality of your entire application is a cool concept... and powerful. You can also use the single SWF approach in your development, but just include the asset SWFs, which will be downloaded automatically.

#### Cons

Shared libraries are very difficult to get comfortable with. Once you do,

then utilize them in frame 3 and beyond. However, for older components, or the ones that come with Flash, they use many subcomponents (subclasses), and unchecking all of these can be time-consuming. This concept of application design also spits in the face of the single frame Flash movie that programmers aspired to in 4, 5, and MX. (If you said "3 too!", you're a freak.) Although you can still emulate that "one-time download" effect, the user is still downloading multiple SWFs instead of one, and these do cache, leading to problems in testing or updates. Ever try to explain to your clients how to clear their cache, let alone what a cache is? Russian Roulette is more fun.

"The **goal** from the early days of Flash was to **get** your application into one frame"

same symbol, but the user only has to download it once. Therefore, if your scrollbar is 10K and you use it with three other SWFs, they do not incur the 10K file size. If you update your scrollbar component, those who use the symbol are updated as well without your having to recompile the other SWFs. In later versions of the Flash 6 Player, you can also do multi-tier Shared Libraries (shared symbols within shared symbols).

#### Pros

The ability to create once and have multiple SWF files use the same symbol helps in your updates, as well as saving in file size. Instead of three SWFs using a 10K scrollbar, you now subtract that file size from them, and only incur the 10K download once to the user. Like the level approach, your compiling speed is drastically sped up in that you don't compile shared sym-

you get into three steps: compile the source FLAs for a change, add your components to frame 2 (or some frame not 1 in MX and before they are used code wise), and uncheck "Export Before First Frame". The multiple FLAs are another source of management...and potential debug errors since you may have forgotten to compile the source SWF, broken your linkage by accidentally editing the Shared Symbol, or had to switch to another file to edit a design change. You also have to initialize the components somehow before you use them via attachMovie or with other components. Since they're not exported before the first frame, they will not be in the final SWF file, even with a linkage ID name, unless they are used somewhere in your movie. Therefore, manually placing them on frame 2 is good. This allows your preloader to work in that all of the data is not downloaded in frame 1. You can



For the greatest hits  
of the 70's, 80's and 90's  
call your web host's  
tech support.

For answers call us at **1-866-EDGEWEB**  
3 3 4 3 9 3 2

When calling your web host for support you want answers, not an annoying song stuck in your head from spending all day on hold. At EdgeWebHosting.net, we'll answer your call in two rings or less. There's no annoying on-hold music, no recorded messages or confusing menu merry-go-rounds. And when you call, one of our qualified experts will have the answers you're looking for. Not that you'll need to call us often since our self-healing servers virtually eliminate the potential for problems and automatically resolve most CF, IIS and ASP problems in 60 seconds or less with no human interaction. Plus, our multi-user support system allows you to track support requests for each of your engineers individually, lookup server availability, receive a copy of all errors on your site in real time, and even monitor intrusion attempts on your site in real time. **For a new kind of easy listening, talk to EdgeWebHosting.net**

**By the Numbers:**

- 2 Rings or less, live support
- 100% Guarantee
- 99.998% Uptime
- 150 MBPS Fiber Connectivity
- 24 x 7 Emergency support
- 24 Hour free backup



**EDGE**  
**WEB HOSTING**

*What are you WAITING for?*

[www.edgewebhosting.net](http://www.edgewebhosting.net)

Shared Hosting ¥ Managed Dedicated Servers ¥ Semi-Private Servers  
ColdFusion ¥ SQL Server ¥ .NET ¥ Self-Healing Servers ¥ Value Priced

Win  
a year  
of free  
hosting\*



\*On Shared Hosting or the equivalent value  
See <http://edgewebhosting.net/cfdj> for details



## Part 3 Animations and Application State in Flash

### Application State Applied in Flash

Before we discuss utilizing one of the methodologies, let's talk about state. Application state can mean a few different things. It can refer to a design being "ready" for user input, or to a bunch of buttons grayed out because they application is "disabled," waiting for the user to choose an option on a modal dialogue, for simply "finished" or "game over man!". The nice thing about Flash's timeline and use of frames is laying this out in a timeline allows you to easily see ahead of time, create, and modify these "states." Nesting your use of graphics and movie clips allows you to have many levels of sophisticated state. Whether you actually transition between those states abruptly or gracefully really depends on skill, your time and money requirements for the project, and your style. At any rate, the simple button example in Images I and II shows how Flash handles button states when you make a button symbol.

*Note:* At the end of this article is information on how to get the complex mp3 radio application made using the methodologies discussed below with a complexly designed interface.

And that's just for buttons. Image III is a simple timeline of a more complicated button. This is how components are sometimes set up. You'll have your component code on the top frame (which can

sometimes be just an include file to your class, while in Pro, it's an external ActionScript class auto-linked to the symbol). There's also the frame labels layer that indicates the "states" of your component, which Flash can show by programmatically "playing" or "going to" these frame labels, therefore updating what is shown on those respective frames. Because this is author time, you can view and tweak your states beforehand. This helps a lot if you're required to implement a very complex animation for an "over state" and want to see how it looks beforehand (which helps show the designer how you're actually using her or his design).

Scrubbing the play head back and forth (clicking on the red square, holding it, and moving it left and right along the timeline) will allow you to see your states.

### Event-Based Programming vs Timeline-Based Programming

It goes without saying that the goal from the early days of Flash was to get your application into one frame. Since Flash 5, unless you were seriously pressed for time, putting anything but a "stop();" on the timeline was considered evil and hard coding. You wouldn't be flamed as people knew it would come back to bite you. Harsh justice, I know.

Viewing Image III, you would think that if one section needed to go to the

next, you could just put a "gotoAnd Play('section');" on the frame in question, and at that point the animation would go there. Sure, that works wonders. But does your code know? Having timelines control code died in Flash 5, so picking the habit back up doesn't fly now...well, not in the long run, anyway. Having your code control the timeline from one centralized location is best since it's no longer tied to it. Initsclip and his brother, endinitclip, prevent your code from running more than once, and in Pro, your code is exported on a certain frame on the main timeline and run, and you're done. Therefore, stops are just about the only thing acceptable because the timeline shouldn't control itself, but rather the code.

Informing your code of when a timeline has reached a certain point presents a problem if you can't put code on that spot, then the old adage "How do it know?" (Kenneth Payne, VB Programmer and Jack of All Trades, JRW's first mentor). Some designers and programmers new to Flash will use frames as a sequence of events. Frame 1 starts it, frame 10 does some stuff, and when ready, play frame 20, etc. This, however, has a few issues in dealing with preloading, code portability, reuse, and code centralization. What if the designer adjusts the length of the animation? What if the design itself changes and you must remove layers and animations entirely? Having code moving around the timeline, unless it's a stop, isn't safe.



"It's by **no means** easy, but the ability to sell this application, as well as its **positive** impact on the user, is profound"

"Great, Jesse, but how can we tell when we've reached that point?"

Here's my method. First, I use Method 4 in implementing the design. I decide which elements of the design can be broken into subsequent components, and then build them. Each component plays whatever animation or functionality is needed. There are common problems you run into when doing this, and I'll go over those after I describe my method.

## The Process Clip Revisited

I first have constants – variables with capital names for which I make an agreement to never change their value. They're named after the frame label (plus a "\_DONE"), and their value is equal when that frame label's animation is done. That way, if the timeline gets moved around, I just change the constants variables' values.

Second, I determine which state happens when. Sometimes, you get lucky and only have to code a forced state change, like a Flash button. It goes wherever you are. Sometimes, though, in assembly-type interfaces, the designer has provided a transition from point B to point C, from point C to A, etc. All the permutations I'm assuming are done for me (although I end up doing them because I have little contact with my designers on how the project is put together sometimes, or the way they created it behooves me to re-create it myself. Production Artist, anyone?). With all of these scenarios set up, I code a function that gives me a frame label to go to based on where I am at currently, passing in a frame label that says where I'd like to go. So, if I want to shut down, and my app is currently open, it'll need to play the "close menu" animation. It's generally one big arse nested switch statement (or if/else if you are so inclined). You pass in the frame label your currently at, and it determines where to go (see Code 1).

This allows you to get to any point from any point. This requires you and the designer to hash out all possible scenarios of each component/module, and then for the designer to create the animations for those transitions/scenarios. The programmer may have to modify some of the interface elements; the ones that need code associated with them by giving a movie clip an instance name on a certain frame.

Finally, we get to the engine to instantiate the change in state, as well as know-

ing when state has been truly reached. This can be coded in two ways: lazy evaluation and strict evaluation. Since I'm a programmer, I'll show the lazy way. Strict is merely stating that you cannot change state while transitioning to another state (animating the component/movie clip/showing the animation), while lazy just plays what it needs to, knowing the stops will take care of stopping the animation at its proper time. Designers and sales people obviously love strict because it appears smart, and strict almost always involves transitions. What they don't realize is if you're trying to change a combo box to "open" so you can populate data that just came back from a Remoting call, but you're using strict and your code responds with a "no, sorry, not yet," then the combo box doesn't get its data. You're not allowed to code the combo box to get the data when it's ready; that's cheating. The way around it is using the constants in step one: "He's there, send the data to those who need it." You can either hard code these data broadcasts in your getFrameLabel function, or make a generic function to call whenever state is finished (a button is finally on the stage, ready to have its label set via a Remoting call's record set result and its click handler pointed to wherever). Either way, you use an onEnterFrame function on a movie clip. The enterFrame event fires every frame iteration, ensuring that every time the play head moves, you get an event. Even though most programmers like intervals, the timing is not so exact that you'll get it fired on every frame, while onEnterFrame will. Code 2 shows the functions that go in your component.

Now, before you go asking how to animate components, you simply:

- Put your components in a movie clip
- Give it an instance name
- Place the movie clip on the timeline where it is to be animated
- Give it an instance name
- Animate it

Then, your set data can be something like:

```
this.movingCheckBox_mc.my_ch.set  
Value(true);
```

Now, notice the inefficiency here: either you code one function to populate all of

your components on each state change with data, even if they don't exist, or hard code the data setting to the frame label. I recommend the latter, even though it's more work and you're duplicating function calls, the data that needs to be set will get set when you need it to. This is a bigger deal if the call is asynchronous, like XML or Remoting.

To pull it all together, each component uses the above process and is put in its own FLA file, with its component name as the FLA (in Pro, the component class as its class name.as). You then link the symbol for runtime sharing.

Pull them all together (drag all their symbols into the main movie's library), put your preloader on frame 1, your shared symbols on frame 2, and your app on frame 3.

## Conclusion

Ultimately, the process of implementing complex interfaces and animations in Flash goes smoothly if the details on how it works (transitions from state to state, or not at all) are ironed out, up front, with face-time collaboration. Having the designer and Flash programmer work out how it can be assembled, confirming all states for each component, and having the designer create and animate those states will lead to a smooth process. It's by no means easy, but the ability to sell this application as compared to other, non-Rich Internet Applications, as well as its positive impact on the user, is profound. Besides, true Flash programmers revel in this because they know what they are creating is high-end and cool – the designer can see her or his creation work, and the user ends up reaping the benefits of a labor of love.

For an example using the code here, please see this article's accompanying Flash file, Neos Radio, at [www.sys-con.com/mx/sourcec.cfm](http://www.sys-con.com/mx/sourcec.cfm).

Also, for additional discussion on application state applied to Flash, please see [www.darronschall.com/weblog/archives/000014.cfm](http://www.darronschall.com/weblog/archives/000014.cfm) as well as [www.stalker.com/CommuniGatePro/WebApp.html](http://www.stalker.com/CommuniGatePro/WebApp.html) under "Stateless and Session-based Processing".

Design provided by David Allbritton ([www.daveallbritton.com](http://www.daveallbritton.com)), hacked apart by me. ☺

*Jesse R. Warden, MXDJ's Flash editor, is a senior Flash developer at Surgical Information Systems, an operating room software company, where he currently uses Flash MX, Flash Remoting, .NET, and Oracle to create next-generation rich Internet applications for the OR. He contributed four chapters to the Flash Communication Server MX Bible and has written articles for various publications, including one for Macromedia for a DRK. [www.jessewarden.com](http://www.jessewarden.com) [jessewarden@sys-con.com](mailto:jessewarden@sys-con.com)*



# Customizing Components in Flash MX 2004

*Styles and themes: powerful new tools*

by peter elst

**b**uilding on the success of components in the last release of Flash MX, Macromedia has outdone itself in rebuilding this complete architecture for ActionScript version 2.0.

With this latest release we've been introduced to some new and powerful ways of customizing components. Two buzzwords here are "Styles" and "Themes" – the first is largely used for defining font, color, and spacing properties, while Themes deal with the actual skinning of component assets. I'll discuss these two features in detail, as well as touch on some more advanced topics such as inheritance and code-based skinning.

Of course, these new goodies come at a price. Established Flash designers and developers will need to invest time getting themselves acquainted with the new feature set in Flash MX 2004, and those new to the software will have a lot of ground to cover. This article is a good start, as it will get you more in touch with the component customization options available and how to use them in this latest release of Macromedia Flash.

## Styling Components

Let's take a closer look at one of these new ways of customizing components in Flash MX 2004 (see Image I). Styles is a feature that you'll probably be using most often when tweaking components to suit your project. In Flash MX, you used either the `globalStyleFormat` or `setStyleFormat` component methods for doing this.

You can compare Styles to Cascading Style Sheets, though there are subtle (and not so subtle) differences in implementation. If you look at the new CSS support for `TextField` objects in Flash MX 2004 you'll see that they have a lot in common with the Styles feature used on compo-

nents. One of the main differences between the two is that components can't parse the W3C standard style properties and use their ActionScript equivalent; more about this later in the article.

The basic principle of using Styles is very easy indeed; you only have to use two methods, "getStyle" and "setStyle", to get started. You guessed it, those methods do exactly as their names indicate: set and get style properties. Not very exciting yet, is it? Let's get to the good part... these styles can be layered and applied to various scopes, have inheriting values, and even support custom style declarations (see Image II). It's definitely getting more interesting now.

These various scopes on which Styles can be applied are the following: the component instance, a component class, and the `_global` scope. Let's first look at how to set style properties on a single component instance.

## Applying Styles to a Component Instance

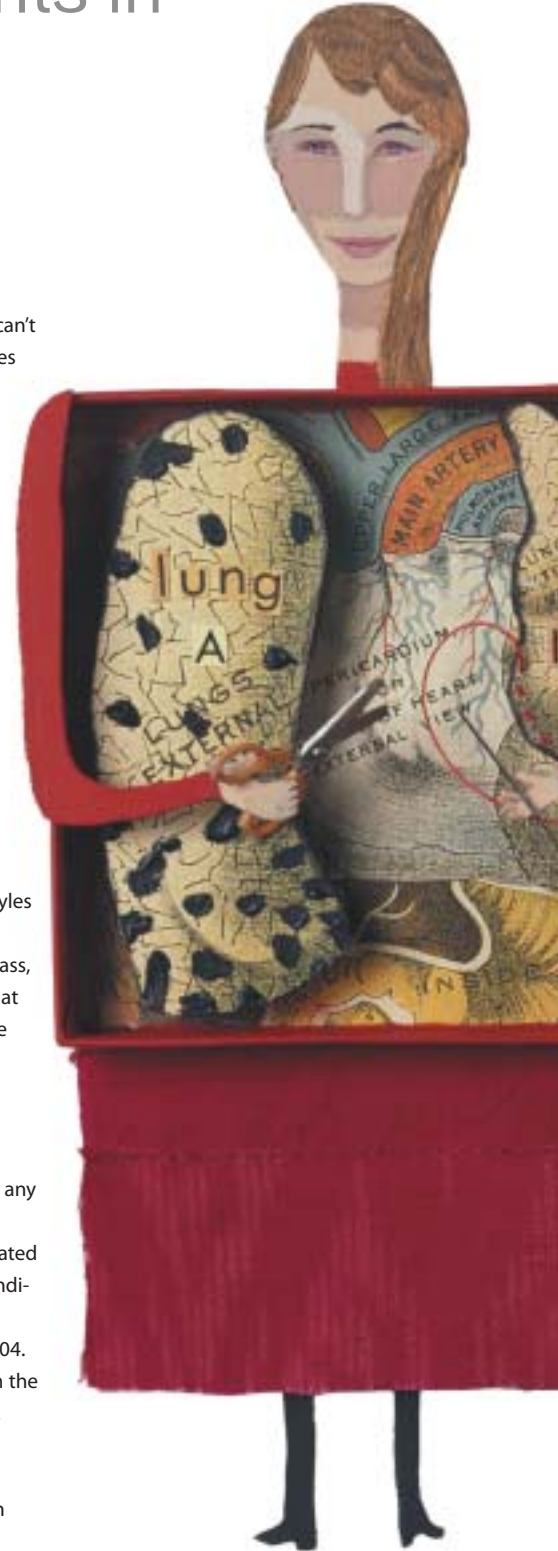
A component instance is basically any single component that you have on stage. These components are instantiated from their component class and are individual copies of the same object.

- Open up a new FLA in Flash MX 2004.
- Drag a `ScrollPane` component from the "Components panel" to your stage.
- Give the `ScrollPane` component instance name "myScrollPane".
- Write the following ActionScript on frame 1 of your main timeline:

```
myScrollPane.setStyle("backgroundColor", "0x003366");
```

If you test this movie you'll see that the `ScrollPane` component now has a

dark blue background color. When you want to retrieve a value for a particular style property on the component, simply use the `getStyle` method and specify the style property as the first argument.





```
myScrollPane.getStyle("backgroundColor"); // traces out 0x003366
```

For a full list of all supported style properties, consult the Flash MX 2004 documentation (Help > Using Components > Customizing Components > Using Styles to Customize ... > Supported Styles). Be sure to look at the documentation for each individual component (Help > Using Components > Components Dictionary) since there might be some component-specific styles listed there.

Of course, when you've got multiple components on stage you won't want to style them all individually. This is where the other scopes come in handy. The first one we'll look at is the component class.



```
_global.styles.ComboBox.setStyle("backgroundColor", "0x003366");
```

If you test this movie you'll see that all ComboBox components on stage have now got a dark blue background color applied. This approach is very useful if you want to use different styles for different components. However, if you want all components on stage to subscribe to your settings, doing it this way could cause you to get finger cramps from defining the styles for every single component class you're using.

### Applying Styles to the \_global Scope

Setting Styles on the \_global scope makes sure that every component has access to the Style properties, and no separate style declarations on the component instance or class are necessary (see Image III).

that the TextArea component does not subscribe to this \_global background Color style property. Why is this happening? Well, the TextArea component, among others, already has some style properties predefined on its component class style scope, which takes precedence over the \_global style scope.

Applying styles to component classes is all very well and good, but it might not always be a good solution. What if you want to give a ComboBox and a TextArea component a particular style and a ScrollPane and List component another one? This is where custom style declarations come into play.

### Creating Custom Style Declarations

Custom styles allow you to create an object that contains various style property settings that you can then have any component subscribe to.

“Flash MX 2004 has given us a host of new features for customizing components”



### Applying Styles to a Component Class

The component class is what you might consider the framework of a component. All instances of a component you drag on stage are instances of their component class. Knowing this, applying styles to the component class will make all instances of that class subscribe to the same settings.

Flash MX 2004 stores these class style definitions in the \_global.styles scope. This scope contains an object for each component, the object name of which refers to the component class name.

- Open up a new FLA in Flash MX 2004.
- Drag a couple of ComboBox components from the “Components panel” to your stage.
- Write the following ActionScript on frame 1 of your main timeline:

- Open up a new FLA in Flash MX 2004.
- Drag a couple of ComboBox and ScrollPane components from the “Components panel” to your stage.
- Write the following ActionScript on frame 1 of your main timeline:

```
_global.style.setStyle("backgroundColor", "0x003366");
```

If you test this movie now you'll see that all components have a nice dark blue background color. Notice that we used \_global.style here, not the \_global.styles scope we discussed earlier for the component class style declarations. That all looks very nice, but it's unfortunately not always a one-stop approach to styling all your components. Let's look at a problem with using the \_global scope to style components:

- Keep using the same FLA with the components.
- Drag a TextArea component on stage. Now if we test this movie, you'll see

- Open up a new FLA in Flash MX 2004.
- Drag ComboBox, TextArea, ScrollPane, and List components onstage.
- Give the ComboBox component instance name “myComboBox”.
- Give the TextArea component instance name “myTextArea”.
- Give the ScrollPane component instance name “myScrollPane”.
- Give the List component instance name “myList”.
- Write the ActionScript in Code I on the first frame of your main timeline.

If you test this movie you'll see that the ComboBox and TextArea component have a dark blue background color, while the ScrollPane and List component have a red background color.

Now how did we do that? It's not that difficult; first we create custom style objects in the \_global.styles scope by instantiating them from the built-in CSSStyleDeclaration class. Having done that we now have two namespaces for



applying our styles to: “myStyle1” and “myStyle2”. The next thing we do is apply a different value for the backgroundColor style property to each of the custom style declarations. If you look at the last four lines you’ll see that we are setting a style property named “styleName” on each of our component instances. This “styleName” property points components to the name of the style declaration in the `_global.styles` scope we want it to subscribe to.

This example, which sets only one style property, doesn’t look very productive, but as you’ll likely use far more style settings in real-world projects this approach can very productive as opposed to applying all those styles manually to component instances.

### Styles and Inheritance

Just as is the case with CSS, Styles applied to components can be set to inherit their value. This means that a child object that doesn’t have a particular style property assigned “inherits” the value of its parent object. A good example of an inheriting style property is “color” versus the “backgroundColor” property, which does not do so.

This style inheritance for components is a very useful feature when working with nested components. Even compo-

nents nested at runtime automatically get these inherited style properties applied.

### Dealing with Differently Scoped Styles on a Component

You might be wondering what happens if you have Styles applied to multiple scopes on a component. Flash deals with this in a very slick way. There is an order in which Flash goes looking for style settings, which also brings the concept of inheriting styles into play.

1. component instance (`myComponentInstance`)
2. component styleName property (`_global.styles.styleName`)
3. component default class style declaration (`_global.styles.ComboBox`)

The three steps above are the same for any style property that Flash applies to a component. First it looks to see if Styles are assigned to the component instance; if so, it disregards all other scopes. If no Styles were defined on the component instance it looks to see if the “styleName” property was set to point to a custom style declaration. If not, it moves on to the default component class style declaration in the `_global.styles` scope.

So far, so good. Now we come to the inheriting styles part. If a particular style property is set to inherit, Flash first moves on to the parent object of the component and goes looking through those three scopes in that object. Finally, when Flash has finished looking through all parent objects without finding the style to be defined it moves on to step 4, the `_global` scope.

Style properties that aren’t set to inherit automatically move on to step 4 if no style definitions were found in any of the other three locations.

4. global style scope (`_global.style`)

We’ve covered a large portion of the Styles API; if you want more information on subjects such as defining color names and setting inheritance for style properties, see the documentation on the `StyleManager` class (Help > Using Components > Components Dictionary > `StyleManager` class).

### Skimming Components

Moving on to skinning components, you may have noticed that – unlike in Flash MX – component assets are no longer stored in the Library panel. You used to be able to look for the Library folders that belonged to a particular component and start editing the movieclips in there to suit your project. Those days are now over; Flash MX 2004 has introduced compiled clips and the SWC file format.

These compiled clips do a couple of things: they hide the component code from prying eyes, allow for easy distribution of components, and automate things like creating live previews for components. All well and good, but how do we get to the component assets? For customizing component skins Macromedia has introduced the so-called “Themes,” which are basically just simple FLA documents that act as containers for all component assets.

### Applying a Theme to Your Project

Flash MX 2004 ships with two themes: “haloTheme.fla” and “sampleTheme.fla” (you can find those in the First Run > ComponentFLA folder of your installation directory). If you open up “sampleTheme.fla”, for example, you’ll notice that there is nothing at all on stage.

Now let’s open the Library panel and see what’s really going on in there. You’ll find a whole bunch of nested folders and if you browse down through Flash UI Components 2 > Themes > MMDefault you’ll find the folders that contain the assets for each of the UI components (Accordion Assets, Alert Assets, etc.).

Things are getting a bit more familiar now; you can now simply edit those movieclips and make all the changes you want. When you’re done with modifying the component assets, simply drag that folder from the Themes FLA to the stage of the FLA you want to apply the skins to. You can then safely delete those assets from the stage of your FLA (they are still contained the Library panel) and if you test your movie you’ll see the new skins applied for the component you customized.

Components can also be skinned purely by using ActionScript. A full

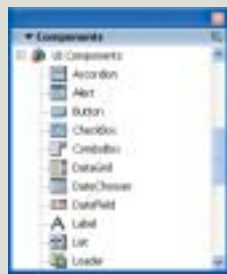


image I



image II



image III

explanation of this would go beyond the scope of this article, but essentially every component contains properties that point to the linkage identifier for the skins it uses. The Accordion, for example, has four of these properties (falseUpSkin, falseDownSkin, falseOverSkin, trueUpSkin) that each point to the linkage identifier of a skin that represents a particular state of the header bars.

If, for example, we want to change the skin for the rollover state of the Accordion header bars we could create a movieclip containing our skin, give it the instance name "rolloverAccordion", for example, and use the following code:

```
myAccordion.falseOverSkin =
    "rolloverAccordion";
```

This code assumes that our Accordion component had the instance name "myAccordion". If you test the movie and move your mouse over one of the header bars you'll see your custom skin applied. For more info on skinning the Accordion component, you might want to consult the Flash documentation (Help > Using


Components > Components Dictionary > Accordion component)

### Conclusion

As you've probably noticed, Flash MX 2004 has given us a host of new features for customizing components. This article has only touched upon the most common uses but should have given you enough information to go out and start customizing components yourself.

If you want to learn more about it, be sure to read through the Flash MX 2004 documentation, which does a pretty good job of introducing all these topics, and be sure to do some experimenting yourself, as that is definitely

the easiest way of getting acquainted with it.

If you want to see customization at work on a real-world application, take a look at "aggregator fla", available at [www.sys-con.com/mx/sourcec.cfm](http://www.sys-con.com/mx/sourcec.cfm). 

*Peter Elst is a freelance developer and founder of MindStudio, which mainly does Flash application development and multimedia projects in general. As a coauthor of Flash MX Components Most Wanted, contributor to various community resources, and Team Macromedia volunteer for Flash, he happily spends most of his time absorbed in ActionScript code and replying to e-mailed questions. [info@peterelst.com](mailto:info@peterelst.com)*

code 1

```
_global.styles.myStyle1 = new mx.styles.CSSStyleDeclaration();
_global.styles.myStyle2 = new mx.styles.CSSStyleDeclaration();

_global.styles.myStyle1.setStyle("backgroundColor", "0x003366");
_global.styles.myStyle2.setStyle("backgroundColor", "0xCC0000");

myComboBox.setStyle("styleName", "myStyle1");
myTextArea.setStyle("styleName", "myStyle1");
myScrollPane.setStyle("styleName", "myStyle2");
myList.setStyle("styleName", "myStyle2");
```

## Content can be unruly. Make it behave with **CommonSpot™**



### CommonSpot™ Content Server.

Put content management in the hands of content owners.

To tame your unruly web site, whip your content into shape with CommonSpot. With a rich, out-of-the-box feature set, you'll be up and running in weeks – not months. Built in ColdFusion, it's easy to integrate and customize CommonSpot to meet your requirements. CommonSpot's intuitive, browser-based interface makes authoring a snap, while its template and data-driven architecture and granular permissions allow for flexible yet powerful control.

Find out why CommonSpot was voted Best Web Content Management Tool in the CFDJ Reader's Choice Awards. Visit [www.paperthin.com](http://www.paperthin.com), or call today to schedule an online demonstration.

With CommonSpot, you'll have content tamed in no time.

- 100% Browser-Based Web Publishing Framework
- Template-Driven Web Pages
- Multi-Level Approval Workflow
- Content Scheduling & Personalization
- Rich Custom Metadata Support
- 508 Accessibility Compliance
- Seamless ColdFusion Integration
- More than 50 powerful, out-of-the box features

**Paper Thin**

**800.940.3087**  
**[www.paperthin.com](http://www.paperthin.com)**



# Flash Remoting vs Flash Player with SOAP

*Flash Remoting offers stability, performance*

by joe orbman

With the recent release of Flash MX 2004 Professional Edition and the SOAP support built into Flash Player some are questioning whether Flash Remoting has become obsolete. Indeed, it appears that Macromedia did not do anything to enhance the Flash Remoting feature, but instead made a significant stride to beef up support for SOAP Web services. The noticeable presence of the WebServiceConnector component, as well as the API for programmatically invoking Web services, makes a lot of developers wonder whether Flash Remoting is overshadowed by SOAP now. Anyone familiar with both approaches will admit that the core difference between them would be performance, though products such as FlashORB add a variety of features that significantly enhance the Flash Remoting approach. This article analyzes the results of a set of benchmark tests designed to demonstrate the performance differences between SOAP and Flash Remoting.

## Test Setup

The tests were designed to demonstrate system performance for transferring simple type, complex type, array of simple types, and array of complex types. Additionally, a set of tests was included in which a combination of the above listed types was transmitted in the same test run. Each test was executed first against the SOAP service and then against a Flash Remoting service. While a test was running, the benchmark client measured the performance of the client/server communication and reported results like “total message throughput” and “number of messages per second”. The same class was exposed as a Web service as well as a Flash Remoting service. FlashORB was selected as the Flash Remoting hosting

server and Apache Axis was selected for the SOAP tests.

## Server-Side Service Class

The server-side service class includes a set of methods corresponding to each type of test:

- **echoString:** This method accepts a string object argument and returns the same string object.
- **echoInt:** This method accepts a primitive integer argument and returns the same value.
- **echoPerson:** This method accepts a data structure representing a person. Each data structure has a set of fields. The method then returns the same object received as an argument.
- **echoPersonArray:** This method accepts an array of Person objects and returns the same array.
- **echoArray:** This method accepts an array of string objects and returns the same array.

The source code for the class exposed as a SOAP Web service and Flash Remoting service is provided in Code I.

## Benchmark Client Application

The benchmark client application consists of five sections, each corresponding to one of the methods in the remote service. Each section has one or more fields to configure the argument(s) for the appropriate method. For example, the “echo Int” section has a text field where the number for the argument can be entered. The “echo Person Array” and “echo String Array” sections use the values from the “echo Person” and “echo String” sections for the actual data, but they also contain a field where the array size can be entered. Each section has a checkbox, which if selected, specifies that



the method should be executed in the test run. This way you can easily select the methods that should be invoked in a test. At the bottom of each section, there is a test summary area. The “Messages Total” fields provide information about the number of messages sent and received during a test. The “Messages/Second” fields automatically calculate the throughput for the running test. At the very bottom of the client UI, there are two buttons responsible for starting a test. The “Test Duration” field can be used to set the duration of the test.

Only one test can be executed at a time – once a test is started, no other test can be executed. The recommended approach is to run one type of a test first (SOAP or Flash Remoting) and once it completes, execute the other type of the test for the same test duration and method selection.

## Client Design

The client application contains an invisible movie clip (invoker) responsible for running a test. In the first frame of the invoker movie clip, 2 proxies are created: one for the SOAP Web service and the other for the Flash Remoting service (see Code II).

As soon as one of the buttons kicks off a test, button click handler delegates the execution to the second frame of the invoker movie clip, which checks the methods that should participate in the test. For each selected method an operation is added to an array. (See Code III. Code III and IV are available at [www.syscon.com/mx/sourcecec.cfm](http://www.syscon.com/mx/sourcecec.cfm).)

Once the servicesToCall array is set up, the Invoker movie clip enters a tight loop (frames 3 and 4). The code continuously invokes methods on the remote service (SOAP or Flash Remoting), until the time in the test duration elapses (see code IV). Frame 4 of the Invoker movie

clip sends the control back to frame 3 to repeat the entire sequence until the test is complete.

## Tests Results

A number of tests were executed to identify the performance of each approach. The tests (and the client application) are designed to demonstrate raw performance of transferring a homogeneous data type (primitive, complex, or array) or a combination of data types. Each message invoked on a remote service includes both transferring an argument to the service (serialization) and receiving the same object as a response (deserialization). A description of each executed test with the produced results can be found in Table 1. Each test was executed with the client and server on the same computer to avoid network latency. However, this could have impacted the results since both client and server compete for the same CPU and memory.

## Benchmark Observations

The Flash 7 SOAP implementation has

a significant memory leak. The duration of each of the executed tests was set to 60 seconds. During the execution of a test, the memory allocated by the browser (IE 6.0) increased between 400% (echoInt test) and 1700% (echoArray test).

Tests passing just primitive arguments or strings had comparable performance for SOAP and Flash Remoting approaches. This can be explained by the rather simple format of the on-the-wire SOAP messages for these data types. As the complexity (and size) of the SOAP messages increases, the performance significantly drops. For example, the throughput of passing an array of 20 elements (each containing a Person object) is 7.22 times higher with Flash Remoting than with SOAP (65 messages/second vs 9).

The SOAP approach experienced a close-to-failure condition with a very large array of strings (2000 elements). Throughout the test, the browser process became very sluggish and nonresponsive, resulting in a throughput of less than 2 messages per second. The Flash

Remoting approach carried out the same test significantly better, with an average throughput of 20 messages/second.

## Test Conclusion

Macromedia has done an outstanding job of making its SOAP integration as simple as possible. Our experience using the Web services APIs was very positive. However, there are several drawbacks in the current Flash Player SOAP implementation. The most noticeable drawback is the memory leak, which makes the approach impractical. The Flash Remoting approach is by far more stable and has better performance. The SOAP approach results in larger payloads and greater CPU processing, resulting from the XML Schema validation and XML parsing. The Flash Remoting approach benefits from a binary representation of request/response as well as the invocation-batching feature. For any application that requires moderate to high remote invocation volume and/or that involves complex data types and large arrays, Flash Remoting is a better fit. ☺☺

*With more than 15 years of software engineering practice and 8 years of distributed computing experience, Joe plays a key role in the day-to-day operations of Midnight Coders. With responsibilities ranging from chief architect to product evangelist, Joe is responsible for product architecture, strategic positioning, and developers' communication. orbman@flashorb.com*

code I

```
// CLASS benchmarks.flashorb.TestService
package benchmarks.flashorb;

public class TestService
{
    public String echoString( String string )
    {
        return string;
    }

    public int echoInt( int integer )
    {
        return integer;
    }

    public Person echoPerson( Person person )
    {
        return person;
    }

    public Person[] echoPersonArray( Person[] person )
    {
        return person;
    }

    public String[] echoArray( String[] array )
    {
        return array;
    }
}

// CLASS benchmarks.flashorb.Person
package benchmarks.flashorb;

public class Person
{
    public String name;
    public int age;
    public String streetAddress;
```

```
public String city;
public String state;
public String zipCode;
}

#include "NetServices.as"
import mx.services.WebService;
// create Web service proxy
var service:WebService = new WebService( "http://localhost:8080/axis/services/TestService?wsdl" );
// create flash remoting proxy
var gwConnection = NetServices.createGatewayConnection( "http://localhost:8081/flashorb" );
var frService = gwConnection.getService( "benchmarks.flashorb.TestService", this );
stop();
function echoInt_Result( result )
{
    _global.echoIntFRCalls++;
}

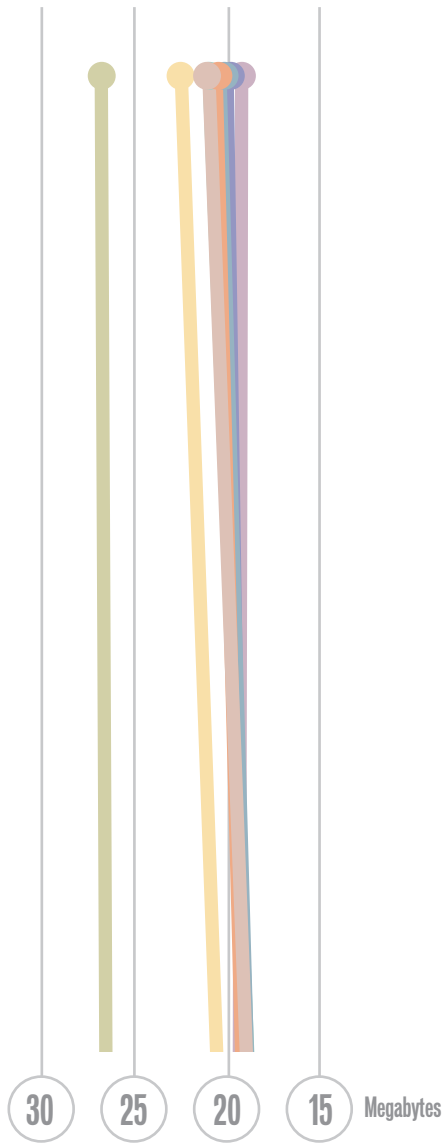
function echoString_Result( result )
{
    _global.echoStringFRCalls++;
}

function echoPerson_Result( result )
{
    _global.echoPersonFRCalls++;
}

function echoPersonArray_Result( result )
{
    _global.echoPersonArrayFRCalls++;
}

function echoArray_Result( result )
{
    _global.echoStringArrayFRCalls++;
```

code II



**echoInt**

In this test a single integer is passed as an argument and is returned by the service. With both SOAP and Flash Remoting the size of the on-the-wire message is relatively small, though when compared to each other, the SOAP message is proportionally larger. The results of the test are shown below:

	<b>SOAP</b>	<b>Flash Remoting</b>
Messages Sent	3727	3895
Messages Received	3727	3895
Messages/second	62	65
Memory allocation	<b>Before: 26.5mb</b> <b>After: 110mb</b>	<b>Before: 26.1mb</b> <b>After: 26.7mb</b>

**echoString**

In this test a string object is passed into a remote method as an argument and is returned by the service to the client. For smaller strings, the size of the SOAP message is significantly larger, but as the size of the string argument increases, the difference in size between SOAP and Flash Remoting on-the-wire messages becomes insignificant.

String size - 20 bytes

	<b>SOAP</b>	<b>Flash Remoting</b>
Messages Sent	3701	3896
Messages Received	3701	3896
Messages/second	62	65
Memory allocation	<b>Before: 19.3mb</b> <b>After: 106mb</b>	<b>Before: 19.3mb</b> <b>After: 20mb</b>

**echoPerson**

In this test a complex type (Person object) is passed as an argument into a remote service method and is returned from the method as a return object. The Person class is a rather 'simple' complex type. It has 6 string fields, however, the results of this tests should be rather indicative of the performance with even more complex data types.

	<b>SOAP</b>	<b>Flash Remoting</b>
Messages Sent	3474	3893
Messages Received	985	3893
Messages/second	16	65
Memory allocation	<b>Before: 19.3mb</b> <b>After: 194mb</b>	<b>Before: 19.1mb</b> <b>After: 21mb</b>

**echoArray**

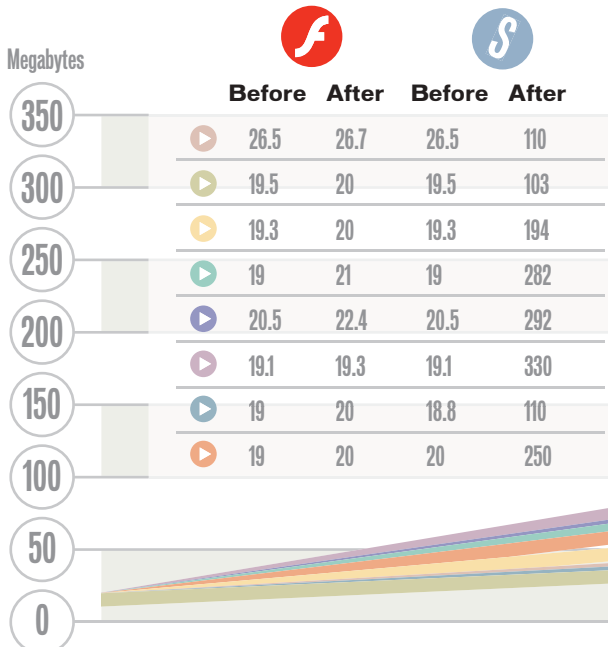
In this test an array of string objects is passed to the remote service and is returned by the remote service. The array size and the size of the string in each cell of the array are configurable.

Array size - 20 elements

	<b>SOAP</b>	<b>Flash Remoting</b>
Messages Sent	3010	3894
Messages Received	1138	3894
Messages/second	19	65
Memory allocation	<b>Before: 20.5mb</b> <b>After: 292mb</b>	<b>Before: 19.1mb</b> <b>After: 22mb</b>

Array size - 2000 elements

	<b>SOAP</b>	<b>Flash Remoting</b>
Messages Sent	87	1215
Messages Received	87	1215
Messages/second	<2	20
Memory allocation	<b>Before: 19.1mb</b> <b>After: 330mb</b>	<b>Before: 21.8mb</b> <b>After: 22mb</b>



test method    test description

**echoPersonArray**



In this test an array of 20 elements, each containing an instance of Person class, is sent as an argument to a remote service. The service returns the argument back to the client as a return type.

Array size - 20 elements

	SOAP	Flash Remoting
Messages Sent	535	3871
Messages Received	535	3871
Messages/second	9	65
Memory allocation	<b>Before:</b> 19mb <b>After:</b> 282mb	<b>Before:</b> 19mb <b>After:</b> 21mb

**echoInt  
echoString  
echoPerson**



In this test 3 operations were repeatedly invoked in the same frame for the entire duration of the test. This test is meant to demonstrate performance when both primitive types and a complex type are passed to a remote service.

	SOAP	Flash Remoting
Messages Sent	1298	3885
Messages Received	1295	3885
Messages/second	22	65
Memory allocation	<b>Before:</b> 18.8mb <b>After:</b> 110mb	<b>Before:</b> 19mb <b>After:</b> 20mb

**echoPerson  
echoArray  
echoPersonArray**

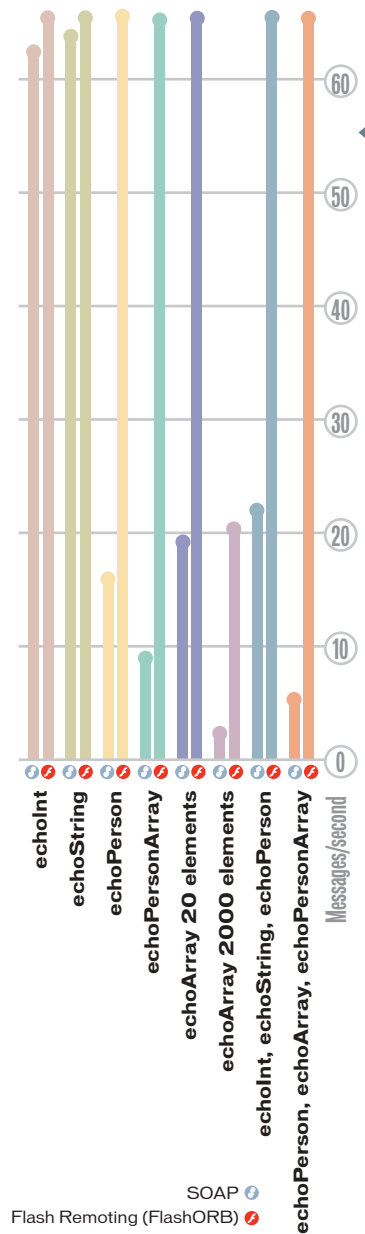


In this test 3 operations were repeatedly invoked in the same frame for the entire duration of the test. This test is meant to demonstrate performance when both primitive types and a complex type are passed to a remote service.

	SOAP	Flash Remoting
Messages Sent	302	3876
Messages Received	302	3876
Messages/second	5	65
Memory allocation	<b>Before:</b> 20mb <b>After:</b> 250mb	<b>Before:</b> 19mb <b>After:</b> 20mb

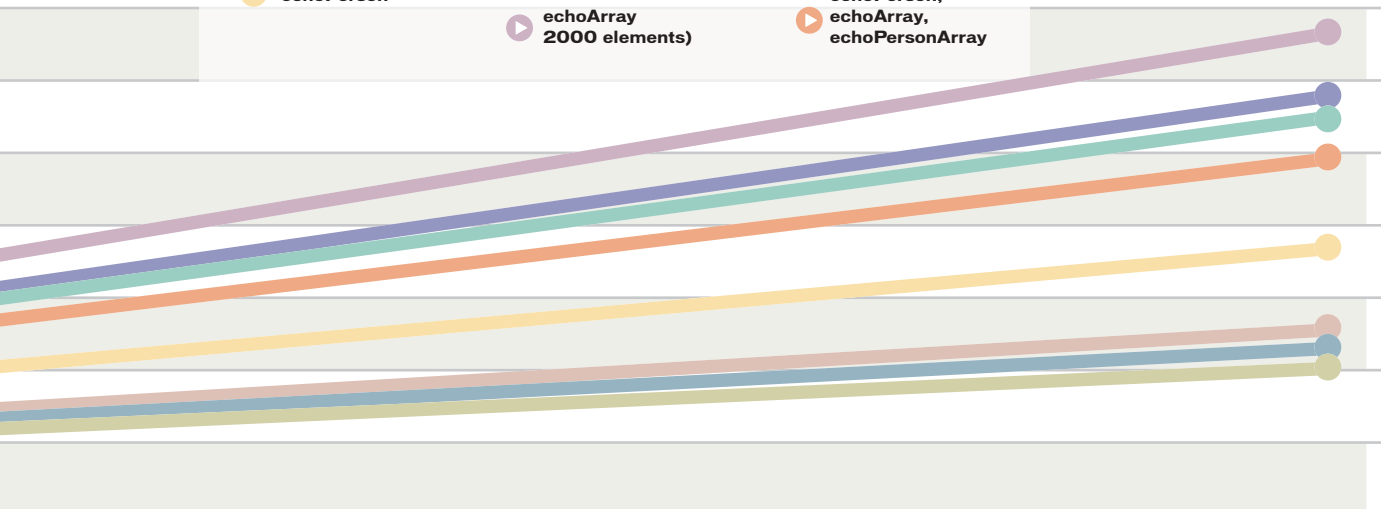
**key**

- echoInt
- echoString
- echoPerson
- echoPersonArray
- echoArray (20 elements)
- echoArray (2000 elements)
- echoInt, echoString, echoPerson
- echoPerson, echoArray, echoPersonArray



soap vs. flash remoting message throughput comparison

**Invoked Methods**



flash plugin memory allocation (before and after soap tests)

# The Executable Internet

*ASP.NET and Macromedia's Flash Remoting combine to offer a compelling rich-client interface*

by ryan moore

As the Internet evolves, the demand for a Web interface that rivals the functionality of desktop applications has become evident. The solution is the "executable Internet," a rich-client technology boasting a client-side browser plug-in capable of making the user's experience of a Web page much more interactive and powerful. The combination of ASP.NET and Macromedia's Flash Remoting is one of the most compelling rich-client interfaces available to overcome today's development limitations.

## The Limitations

In order to fully leverage the immense power of the .NET Framework on the Web, an interactive, responsive, and effective desktop-like user interface is required. Limited by the restrictions of HTML and alternative technologies such as DHTML, developers are forced to conform to browser standards when building their user interface and their application's functionality. These limitations cause Web applications to be much less interactive and powerful than their desktop counterparts.

## Enter the Rich Client

Rich Internet applications offer many of the same possibilities as desktop applications, with the communication power of a Web browser. Rich-client applications have the ability to immediately react to a user's input and then display, process, or validate data based on that input while the user is either on- or offline.

By executing client-side scripts, rich-client applications also make use of the processing power available on the client computer rather than relying only on the Web-hosting server. This feature allows a much more efficient use of bandwidth and processing power than strictly server-side processing.

Rich Internet applications also make the client/server communications taking place in an application nearly invisible. This is accomplished by using an asynchronous, event-driven callback model instead of the traditional Web model. This asynchronous model can also decrease the amount of Web traffic needed to communicate between client and server, and increase the interactivity of the application by allowing the client to retain control of a Web Form while a call is being made to a remote server object.

## About Flash

Macromedia's Flash Player technology is the most widely distributed rich Internet application on the market today. Macromedia Flash Player is currently installed on more than 400 million client devices, including platforms such as Windows, Macintosh, Linux, Sun Solaris, Microsoft TV, and Pocket PC. It is estimated that a version of Flash Player is available to over 98% of Web users. Flash is supported on essentially all version 4.0+ browsers, eliminating the client-interface compatibility problems encountered with other HTML-based technologies such as DHTML and Cascading Style Sheets. Because Flash is supported in both browsers and devices, Flash applications can be deployed consistently across Internet-connected platforms. Flash Player also has a vast array of multimedia capabilities, including support for motion graphics, video, audio, two-way communications, and complex forms.

Traditionally, Flash's capabilities have been utilized by developers for the familiar "skip intro" animations. With the release of Flash MX, the capabilities of Flash have changed. Flash is now an interactive medium capable of processing complex data-driven business logic as well as rich user interfaces.

ActionScript, Flash's scripting language, is a powerful object-oriented scripting language based on the ECMA-262 standard, which is also followed by JavaScript and Microsoft's JScript, making programming in Flash an easy transition for .NET developers.

## Flash Remoting .NET

The component essential to the success of rich-client interfaces is the ability to quickly access server-side data. Macromedia's Flash Remoting for .NET provides an interface for communicating between Flash Player and .NET application servers.

Flash Remoting exposes .NET technologies such as Web services, ASP.NET pages, and .NET assemblies as remote services to Flash, allowing them to be called as if they were local ActionScript objects. Flash Remoting MX is used in .NET applications as a custom server control in ASP.NET pages or as a namespace in .NET assemblies, code-behind class files, and Web services. This gives the .NET developer the flexibility to build server-side logic in a variety of formats, all of which are accessible to the client.

Flash Remoting provides transparent conversions between Flash data types and the server-side .NET data types. These conversions take much of the work out of the hands of both the client and server-side developers, allowing focus to reside on the business logic and client interactivity instead of the object communication.

Flash Remoting for .NET communicates between the Flash client and the .NET server using a message format called AMF (Action Message Format), which is delivered over HTTP and modeled on SOAP. AMF is a binary message format, the likes of which have been found to reduce network traffic up to





50% compared to SOAP-formatted communication. Because it is delivered over HTTP, AMF is also securable via HTTPS and is firewall safe.

The Flash Remoting .NET environment consists of two layers: (1) the net-services layer, residing in the client Flash Player (available on all Flash Players version 6.40+); and (2) the remoting gateway, residing on the .NET Web server. The net-services layer is composed of a Flash include file containing all of the ActionScript classes necessary to send and receive communications on the Flash side. The remoting gateway consists of a .NET DLL that acts as controller on the .NET runtime that, among other things, handles the conversion of data types between ActionScript and the .NET Common Language Runtime. When this controller receives a request, the request passes through a series of filters that han-

dle serializing, logging, and security before arriving at a service adapter that handles the appropriate invocation type.

pub\wwwroot\flashremoting (in a typical IIS install).

Next, create a directory anywhere on your system for the application files. This directory will need to be enabled for Web sharing. In my example, I have shared the folder as netJournalFlash. Create a "bin" directory with write permissions within this directory to function as the local assembly cache. Now we copy a couple of files from the flashremoting directory to the new application directory. Copy the flashremoting/bin directory and the flashgateway.dll file, which is the server-side remoting gateway, to the bin directory of the new application. Also copy the Web.config file to the directory root.

The Web.config file contains one of the essential server-side requirements for Flash Remoting, a reference to the Flash Remoting assembly:

```
<httpModules>
  <add name="Gateway
    Controller"
    type="FlashGateway.
    Controller.Gateway
    Controller,
    flashgateway" />
</httpModules>
```

If the server receives a Web request containing AMF, it forwards this request to the Flash remoting assembly.

### .NET Programming

The ASP.NET code in this example consists of two pages, productList.aspx and productData.aspx. To access data from an ASP.NET file and pass data to and from Flash files, a Flash Remoting custom server control must be used within the page. First, register the Flash gateway:

```
<%@ Register TagPrefix="MM"
Namespace="FlashGateway"
Assembly="flashgateway" %>
```

The Flash control is added to the page with the following statement:

```
<MM:Flash id='myFlash' runat=server />
```

The Flash Remoting custom server controls contain three properties used to access variables passed to and from Flash:

- Flash.Params
- Flash.Result
- Flash.DataSource

The Flash.Params property is an array

of parameters passed from Flash to the .NET application. The Flash.Result property is used to return data to Flash after the server-side processing has occurred. The Flash.DataSource property is used to bind .NET DataSets to Flash Remoting controls.

Let's take a look at the example files. When productList.aspx is invoked, a connection is made to a local Access database and a DataSet is retrieved consisting of the product ID and name for each product in the database. This DataTable is then bound to the Flash control using the control's DataSource property and DataBind() method.

```
myFlash.DataSource =
myDataSet.Tables[0];
myFlash.DataBind();
```

ProductData.aspx is very similar to productList.aspx, except that it requires a parameter to be passed to it from the Flash client and returns the detailed listing for that single product. In this file, we first check to make sure that a parameter

“Macromedia Flash Player  is currently installed on more than 400 million client devices”



has been passed through the Flash Control, then make the connection to the datasource. The SQL query is then built, using the passed parameter to determine which product to select further data for:

```
string sqlQuery="SELECT descrip
tion, location, price FROM
productData WHERE pid=" +
myFlash.Params[0].ToString();
```

And the resulting DataTable is bound to the Flash control as shown in productList.aspx, one of the source files.

### Time for Some Action(Script)

Now that we've constructed our server-side .NET code, it's time to tackle the

```
getService(defaultService,
this);
```

The gateway.aspx file is a blank ASP.NET file used only when developing in the Flash Authoring Environment. In production, the setDefaultGatewayURL is removed, and the gateway is supplied through a parameter in the HTML that embeds the SWF file in the Web page.

Once this connection is made, the remote .NET service methods may be accessed as if they were local Flash ActionScript resources. The service function we will use will reside inside the netJournalFlash application (or whatever you named your app), so we set our default service to netJournalFlash. To make a call to an ASP.NET page containing a remoting object we would like to invoke, a call to the service function is

with the line:

```
myCombo.setChangeHandler("loadImage
Data");
```

The comboBox has been set to execute the loadImageData function when an item has been selected.

In the function loadImageData, we call the productData.aspx page, passing the value of the product we would like to retrieve the data for:

```
flashService.productData(myCombo.ge
tValue());
```

When the response is received by the productData.aspx page, it is automatically handled by the productData\_result function. In this function, we set the price and description text fields to their respective values, as well as load the image associat-

“It is estimated that a version of  Flash Player is available to over 98% of Web users”

front-end Flash. Flash files are constructed on a timeline consisting of a number of layers. In our file, the top layer (as seen in the “Timeline” window) is titled “functions”. When the first frame of this layer is selected, the code for this frame appears in the “Actions” panel. This frame is where all of our ActionScript code will reside. More information about programming in Flash can be found at [www.macromedia.com/support/flash](http://www.macromedia.com/support/flash).

As mentioned earlier, the netservices layer is the client portion of the Flash Remoting model. The netservices layer is initiated in ActionScript with the following call:

```
NetServices.setDefaultGatewayUrl("h
ttp://localhost/netjournal/gate
way.aspx");
gatewayConnection =
NetServices.createGatewayConnection();
defaultService = "netJournalFlash";
flashService
=gatewayConnection.
```

made, with the name of the ASP.NET page being the name of the method being called:

```
flashService.productList();
```

This function calls the productList.aspx page and waits for a response. When a response is received, Flash automatically forwards this response to a function with the name of the call followed by “\_Result”, in this case:

```
function
productList_Result(result)
```

When the result is successfully received, this data is then bound to the Flash comboBox with the instance name “myCombo” using the Flash DataGlue ActionScript object, also included with Flash Remoting:


```
DataGlue.bindFormatStrings(myCombo,
result, "#title#", "#pid#");
```

ed with this product with the line:

```
theImage = "images/"+result.
getItemAt(0).location;
loadMovie(theImage, imgHolder);
```

Line one creates a string variable named “theImage” and sets it to the images directory, followed by the first result’s location column. Line two then loads this image into the imgHolder movieClip on the stage, and our Flash is complete!

### Conclusion

As the demand for a more and more interactive Web user experience increases, the need for rich-client interfaces has increased exponentially. As you have seen in the example, the combination of .NET objects, Macromedia Flash Remoting, and Macromedia’s Flash Player create a powerful rich-client interface capable of producing desktop-like applications in a Web browser interface. 

Ryan Moore is the lead software architect and a principal of Balance Studios Inc. of Green Bay, WI. Ryan, a .NET programmer and Certified Macromedia Flash Developer, splits his time between Web and desktop application development. [interactive@balancestudios.com](mailto:interactive@balancestudios.com)

A new tool for MX professional developers and designers...



### ADVERTISE

Contact: Robyn Forma  
robyn@sys-con.com  
(201) 802-3022  
for details on rates  
and programs

### SUBSCRIBE

[www.sys-con.com/  
mx/subscription.cfm](http://www.sys-con.com/mx/subscription.cfm)  
1 (888) 303-5282



**MX**  
developer's journal



# W

*hen I do my Fireworks MX seminars, I am always surprised at how many experienced Web designers underutilize Fireworks' capabilities. As an example, recently one person said they spent hours designing and writing all the code for a navigation bar with some dynamic behavior. We re-created this person's project, with additional functionality, in Fireworks MX. And we did it all in about 15 minutes.*

turbocharge your

collbar



by charles e. brown





In this tutorial, we're going to examine some aspects of creating a navigation bar in Fireworks MX. We will create three buttons for our small navigation bar, and then we will instruct Fireworks to write the JavaScript code necessary to create rollover effects for the buttons. In the end, we will have Fireworks generate a separate page for each button with all the required behaviors.

### Create the Navigation Bar's Background

In most instances, my normal workflow would be to define the site using Dreamweaver MX. Then, through

*“Nothing gives a button as much pizzazz as a shadow”*

Dreamweaver, I access Fireworks MX to create the graphical interfaces necessary for my design (navigation bars, images, etc.). Once all that is completed, I return to Dreamweaver to add any coding or, should it be necessary, server-side behaviors such as connections to a database. I also use Dreamweaver to serve as the final assembly point for all my various assets (Flash or Shockwave movies, sound files, etc.).

For purposes of this tutorial, we will restrict ourselves to working in Fireworks MX. However, I'm sure you'll have no problems applying the concepts to larger projects. I'm going to assume you have some familiarity with Fireworks MX.

Let's begin by creating a new Fireworks MX document. Make it 500 x 500 pixels with a resolution of 72 dpi. Also, use a white background.

Using the vector Rectangle tool, draw a rectangle along the top of the canvas. You can use the Property inspector to ensure that the rectangle is 500 pixels wide and 50 pixels high.

Set the background color to #AFAFAF.

To make it a bit more interesting than a simple gray rectangle, select Sand from the Texture field in the Property inspector. A texture amount of 40% will create a subtle effect.

It is always a good idea to name each component, or object, of your work. We do this for two reasons:

- It makes it easier to locate and edit the component.
- If you use it in a JavaScript environment, the names are needed for proper identification.

Naming conventions require that we use lowercase with mid-word capitalization and no spaces (underscores are used in lieu of spaces).

You can use either the Layers panel or the Property inspector to do the naming. Name the rectangle navBackground.

### Creating the Button Symbols

There are two approaches you can take to creating the buttons. You could draw the buttons on the rectangle. However, using this technique, you lose some of the functionality of Fireworks MX. If you use this technique, you will need to draw each button separately, define the behaviors for each button separately, and write the text for each button separately. In other words, you will be giving yourself a lot of extra work.

As you will see, a far better technique is to create the button, with all of its properties, once as a symbol. Then, once the symbol is created, we can use as many copies (or instances) of it as we need. Further, if you need to change any of the properties, you need only to change them in one place and all the instances will reflect those changes.

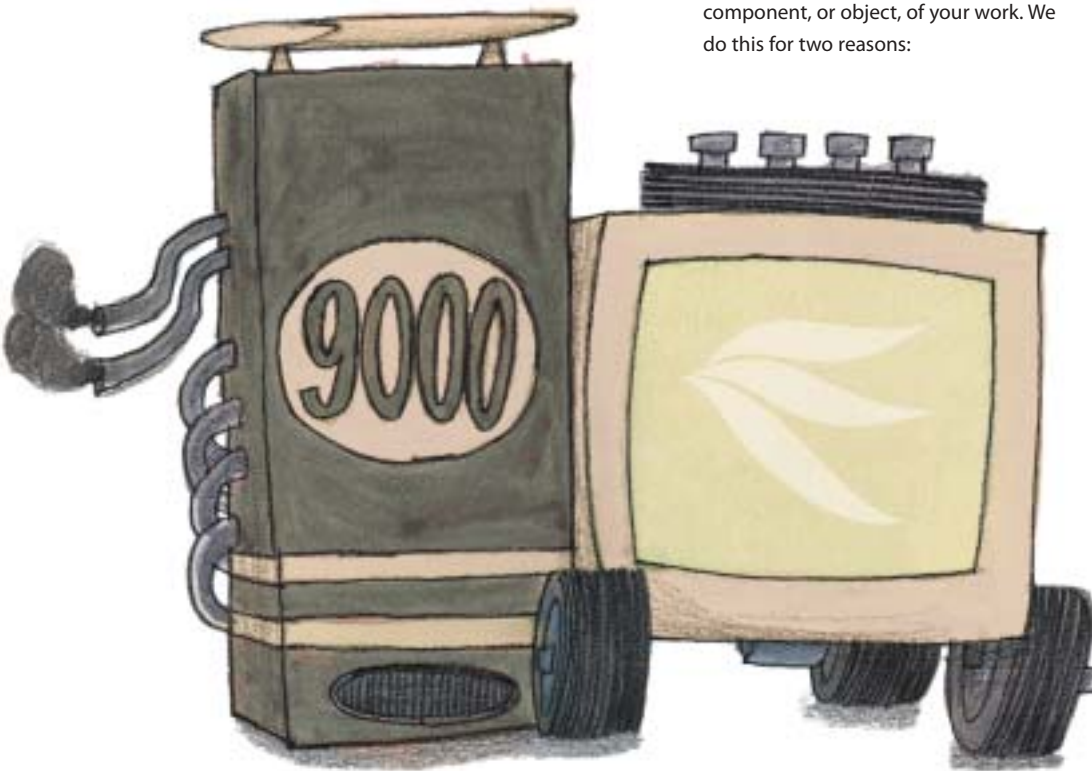
Let's begin by selecting Edit -> Insert -> New Button. Draw a rectangle that is 150 pixels wide and 45 pixels high. Leave the Sand texture set as before but set the color to #0000FF.

To make your button more interesting, use the Distort tool and convert your rectangle into a parallelogram (see Image I).

Nothing gives a button as much pizzazz as a shadow. On the Property inspector, click on the + for effects. Select Drop Shadow and set the Distance to 15, Opacity to 65%, Softness to 4, and Angle to 315. Make the Color black.

Using the Text tool, put the text Button 1 inside the button (of course, in reality you could use whatever text you need). Set the Font to Arial, the Size to 20, and the Color to black.

If your text looks off-center, we can



easily fix that. Either use Select -> Select All or Shift + Click on the text and the Rectangle. Once you have done that, select Modify -> Align -> Center Vertical then Modify -> Align -> Center Horizontally.

Your finished button for this stage should look like Image II.

It is important that your navigation bar is capable of giving your site's visitors a lot of visual cues. We are going to look at several of them.

The first one will let your visitor know which button is selected. We do this with the Over stage. Click on the Over tab at the top of your button symbol canvas. Don't be surprised that the new canvas is empty. Just select the Copy Up Graphic. That brings the graphic over from the previous stage.

We are going to give the button the effect of lighting up when the visitor rolls his or her mouse pointer over it. Using the Pointer tool, select the parallelogram and change the color to #FFFF00. That's all there is to it.

We now want to give the visitor a cue that the button has been depressed. Let's go to the Down tab and, once again, Copy Up Graphic. Select the parallelogram and set the color to #FF0000.

You'll notice that there is an option box at the top of the canvas called Include Nav Bar Down State. A lot of new designers don't understand this process. Let's assume our Web site will have three Web pages. The image of the navigation bar, with the buttons, will need to be included on each page. However, using this feature Fireworks MX will show the Down state properties of the button associated with that page. That way, your visitor now has a visual cue as to what page he or she is on. The rest of the buttons will be in their normal Up state. We will see this in action when we generate the Web pages at the end of this tutorial.

Make sure this feature is checked.

Even though the button stays in its Down state when on the page associated with it, many designers still like to give it the rollover effect for the sake of consistency.

Select the Over While Down tab and Copy Down the Graphic.

Select the parallelogram and change the color to the rollover we used before:

#FFFF00. Again, leave the Include Nav Bar Down State checked.

There is one other tab called Active Area. This gives us the ability to change the size of the slice. Basically, it will allow you to change the amount of space around the button where the button will recognize the mouse pointer. Most of the time, there is no reason to change this quality.

Select Done.

## Assembling the Navigation Bar and Buttons

Of course, the button is placed nowhere near the navigation bar. This is easily fixed by using the Pointer tool and dragging it to the left side of the navigation bar (in this tutorial, we will be putting three buttons on our bar: left, center, and right).

This next step is very important. Since this button is associated with the home page, name it index using the Property inspector. In the Property inspector's Link field, type index.htm. You will notice that we gave the object the same name as the link (minus the htm extension). This point is critical. We are going to have Fireworks MX generate a separate Web page for each button on the navigation bar. In order for it to work properly, the names should match.

One last thing: you'll notice the width now includes the slice area. Using the Property inspector, change the width back to 150. You may need to reposition the button slightly so that it is sitting on the left side of the bar. I am purposely letting the button be about 25 pixels higher than the navigation bar background. This accentuates the shadow effect.

That takes care of the first button. Do we need to go through this whole process for each button? Here is where the power of Fireworks MX really gets fired up.

If it isn't already, open the Library panel by selecting it from the Panels area, pressing the F11 key, or using the Window -> Library menu selection. You should see the button symbol located in it. Just drag the button to the navigation bar and place it about the center of the bar.

There is a slight problem here. We now have two buttons that say Button 1. Notice that there is a Text field in the

Property inspector. For our purposes, type Button 2 here and press enter. The text on your button should have changed. Thanks to Fireworks, we can create one button and change the text as needed. This will give all of your buttons a consistent look and behavior.

image I



image II



image III

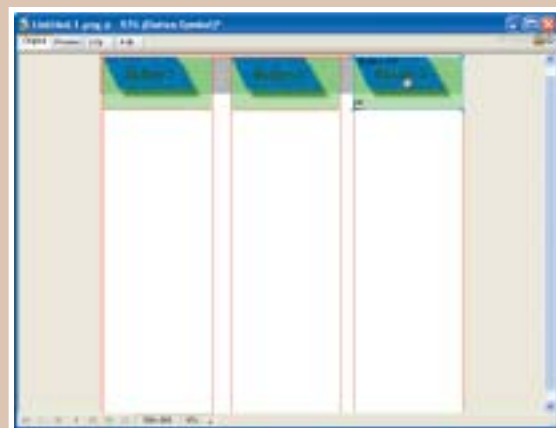


image IV





Call this button page2 and the Link will be page2.htm. As before, change the width to 150 and reposition as necessary so that it is in the center of the bar.

Finally, drag one more instance of the button to the navigation bar, change the text to Button 3, name it page3, and link it to page3.htm. Change the width to 150 and position it at the right side of the bar. Your canvas should now look like Image III.

This may be a good time to save your work to the folder of your choice with a name of your choice. You can preview your work by using the Quick Export button, located on the upper right side of your canvas, or by pressing the F12 key. You should see the rollover and Down state behaviors in action. Of course, we have not generated the Web pages yet, so the links won't work.

*“Fireworks just took a job that could have taken hours and turned it into minutes”*



## Exporting

In reality, we would put a lot more content and design work on our page. But for this tutorial the only content will be the navigation bar.

Select File -> Export. Simply exporting the document will not do the trick. You must also select Options and then the Document Specific tab.

Once you're in that tab, you see an option for Multiple Nav Bar HTML Pages. Make sure that's checked. This will force Fireworks to create a separate HTML page, with a navigation bar, based on the name of each button. In our case, three pages will be produced: index.htm, page2.htm, and page3.htm.

While we're on this page, I also find it convenient to change the naming convention. By default, Fireworks names the images by their position in the table that will be created automatically. In my opinion, this can be confusing for a beginner. Under the Slices section, with the three dropdown lists, drop menu one should say doc name, the second should say underscore, and the third should be changed to slice.

One last thing before we do our Export. If you're going to export to a specific HTML editor, you can click on the General tab and select the HTML style. Most of the time, I use Dreamweaver HTML. This setting is not overly critical. It just puts it in a format style that makes it easier for the editor to handle.

Select OK to come out of the Options window. We'll call our export nav\_bar and export HTML and Images. We'll export an HTML file and Export Slices. Make sure that you have the option Include Areas without Slices checked.

Remember, when you have a rollover image such as we created here, all that is happening is that JavaScript is swapping one image for another, so each state of each button must have a separate image. That being the case, each button will create four images. In addition, additional images will be created to help spacing and such.

You may find it a bit confusing to put your HTML pages and images in the same folder, especially in larger projects. Go ahead and create a new folder, if you don't have one, called images. Then select the Put Images in a

Subfolder option and click the Browse... button, double click on the images folder, and select the Select Images button located in the lower part of the screen.

When you return to the Export screen, you should see images/ next to the Browse... button. Select the Save button. Fireworks will generate three HTML pages, each with all the required HTML and JavaScript code necessary to create the navigation bar and the rollover effects. In addition, it generates all the necessary images in the images subfolder.

Using Windows Explorer, or navigating from your Web browser, open the index.htm file you just created. You will see button 1 in its Down state and the other two buttons in their Up state. If you click on Button 2, you will be taken to page2.htm with that button in its Down state.

Just to show you what was down, view the source code, using your Web browser, for each page (usually this feature is located under the View menu item). When you do this, you'll see a fair amount of code which, without Fireworks, would have taken hours to create and debug.

In the images subfolder, we can easily see the names of the pages and the slices they are associated with. I always tell my seminar students that it is a good learning experience to open each of the four images created for each button so you can see what behavior is associated with each name.

At this point, you can open the files in Dreamweaver MX to do whatever additional doctoring you might want to do (for instance, adding titles or dynamic technology).

## Summary

Fireworks just took a job that could have taken hours and turned it into minutes.

As an interesting follow-up exercise, look at some of your favorite Web sites and try to re-create their navigation bars in Fireworks. Pretty soon you'll be able to turn out eye-catching work in minutes. ☺

*Charles E. Brown is the author of Fireworks MX From Zero to Hero and Beginning Dreamweaver MX. He also contributed to The Macromedia Studio MX Bible. [charles@charlesbrown.net](mailto:charles@charlesbrown.net)*



# International Conference & Expo

# Edge 2004 EAST

Development Technologies Exchange

**February 24-26, 2004**

**Hynes Convention Center, Boston, MA**

**ARCHITECTING JAVA, .NET, WEB SERVICES,  
OPEN SOURCE, AND XML**

**Addressing:**

- ▶ Application Integration
- ▶ Desktop Java
- ▶ Mobility
- ▶ ASP.NET
- ▶ VS.NET
- ▶ JavaServer Faces
- ▶ SOA
- ▶ Interoperability
- ▶ Java Gaming

Register By  
November 21, 2003  
**SAVE** Up To  
**\$400**



For more information visit  
**www.sys-con.com**  
or call  
**201 802-3069**

Over 200 participating companies will display and demonstrate over 500 developer products and solutions.

Over 3,000 Systems Integrators, System Architects, Developers, and Project Managers will attend the conference expo.

Over 100 of the latest sessions on training, certifications, seminars, case-studies, and panel discussions promise to deliver real world benefits, the industry pulse and proven strategies.

**Full Day Tutorials Targeting**

- Java • .NET • XML
- Web Services • Open Source

Conference program available online!  
[www.sys-con.com/edgeeast2004](http://www.sys-con.com/edgeeast2004)

**Contact information: 201 802-3069 • [events@sys-con.com](mailto:events@sys-con.com)**





# The Fireworks Fuzzy Edge Quick Knockout Mask

*Keep your edges in line*  
by linda rathgeber

**m**asking fuzzy or irregular edges is typically a slow, painstaking procedure of hand selection with the Polygon Lasso tool, or if you prefer a vector mask, the Pen tool. Plugins that automate the process (e.g., Extensis Mask Pro, Corel Knock Out) are expensive for one-trick ponies. And shame on Extensis and Corel – they haven't been made available for Fireworks. This article will teach you to quickly mask objects with irregular edges using the new Fireworks MX 2004 Replace Color tool, a simple Alpha masking technique, and some sneaky edge-refining tricks.

I'm going to use a gorgeous flower image (see Image I), photographed by Daniel Reyes of Sydney, Australia, to demonstrate the technique. (If you'd like to download a copy and follow along, go to [www.playingwithfire.com/flower.zip](http://www.playingwithfire.com/flower.zip).)

one image into another or make part of an image invisible, you use an alpha mask.

## Preparing the Alpha (?) Mask

Since the mask we'll create is going to work like a cookie cutter, it could be called a clipping mask. But since our mask will fade part of the image, it will work as an Alpha mask. Instead of concerning ourselves with these technical details, let's just do it.

### Step 1

- Select the image by clicking on it and choose Edit \_ Clone.
- Open the Layers panel and select the lower, flower-image thumbnail.
- Hide the selected image by clicking on the hide object icon (the eyeball).

There is good contrast between the

### Step 2

Select the Replace Color tool. Set the properties for the Replace Color tool in the Property inspector, as follows:

Size: 70  
Shape: Round  
Tolerance: 50  
Strength: 200

To set the Change property:

- Click in the Change fill color box in the Property inspector to open a swatches pop-up.
- Use the built-in color picker of the swatches pop-up to select the green color adjacent to the upper left flower petals.

To select the color you want to change the green to:

- Click in the To Fill color box in the

“Masking fuzzy or irregular edges is typically a slow, painstaking procedure”

Unzip the image file, and open it in Fireworks MX 2004.

## A Word About Masks

In Fireworks you can apply two kinds of masks: clipping masks and Alpha masks. Clipping masks work like cookie cutters, clipping the underlying objects to the shape of the mask. If you've ever pasted one object inside another in Fireworks, you've used a clipping mask.

Alpha masks affect the visibility of underlying objects. They are made with grayscale gradients that tell Fireworks how much to blend the underlying pixels with the pixels of the background it sits on. If you want to fade

flower and the background of the image in Daniel's photo. That means it can do most of the work of preparing the Alpha mask by adjusting the contrast.

- Click on the image to select it.
- Choose Filters \_ Adjust Color \_ Brightness/Contrast. The Brightness/Contrast dialog box will open (see Image II).
- Move the Brightness slider all the way to the left, to the -100 position.
- Move the Contrast slider all the way to the right, to the 100 position. This reduces the number of colors in the image to four or five, and makes the next step, replacing colors, go very quickly (see Image III).

Property inspector.

- Use the built-in color picker to select the black swatch in the swatches popup (see Image IV)
- Click and drag the Color Replacement brush over the image background to replace all of the green pixels with black

Let's select a new color to change.

- Click in the Change fill color box to open the swatches pop-up.
- Use the built-in color picker of the swatches pop-up to select the aqua color near the top right of the image. Leave the remaining settings as they are.



- Sweep quickly over the image with the Replace Color brush to change all the aqua pixels to black.

We need to remove the white and yellow pixels in the background next. It's necessary to exercise some care when replacing these pixels because there are white and yellow pixels at the edges of the flower that we need to preserve. Enlarging the image will make this a snap.

- Click in the Set Magnification box at the bottom of the document window. Choose 300% from the pop-up menu (see Image V). Alternatively, select the Zoom tool from the bottom of the Tools panel and click a couple of times in the area you want to work on first.
- Make the Change color either white or yellow.
- Reduce the size of the Replace Color brush to 18 or less.
- Replace the white or yellow pixels in the background with black, being careful not to go over the outside edges of the flower petals.

When you've finished replacing the color you selected, switch the Change color and carefully remove the unwanted pixels of the other color.

There are some scattered blue and pink pixels left in the background. Use the Replace Color tool to change them to black, or paint over them with the Paint Brush tool.

### Step 3

In this step we'll prepare the flower portion of the image for our Alpha mask.

- Click in the Change fill color box on the property inspector to open the swatches pop-up.
- Use the swatches pop-up color picker to select a white area inside the flower.
- Click in the To color box in the Property inspector and use the swatches pop-up color picker to select the red color of the flower.
- Sweep over the flower with the Replace color tool to change the remaining white pixels to red.
- Replace the white Change color with the yellow from the image.

- Sweep over the flower with the Replace Color tool to replace all the yellow pixels with red.
- Replace any pink pixels with red.
- Replace the black pixels inside the flower with red.

The entire image has quickly been reduced to black and red. Next, we want to change it to black and white.

- Set the Change color to the red from the image.
- Replace the To color with white.
- Set the Tolerance to 100.
- Brush over the red pixels in the image with the Replace color brush to change them to white (see Image VI).

Our Alpha mask is nearly finished. The last step before masking the image is feathering the edges of the mask.

### Step 4

If we use the Alpha mask as it is, the edges of the flower petals might look jagged. To soften them, we'll select the black background and pour a feathered fill into the selections.

- Click on the Magic Wand tool.
- Set the Tolerance for the Magic Wand tool to 33 in the Property inspector.
- Set the Edge property for the Magic Wand to Feather, and the amount of feather to 1 (see Image VII).
- Press Shift and select the three black sections of background around the flower.

We'll fill the selections with the Paint bucket tool.

- Click in the Fill Color box on the Tools panel.
- Select a black swatch to set the fill color to black.

*Note:* If you see a color in the Stroke Color box in the Tools panel, click on it and change the color property to None.

- Click on the Paint Bucket tool to select it.
- Click three or four times in each of the areas you selected with the Magic Wand.
- Choose Select \_ Deselect to turn off the marquee (marching ants).



Image I

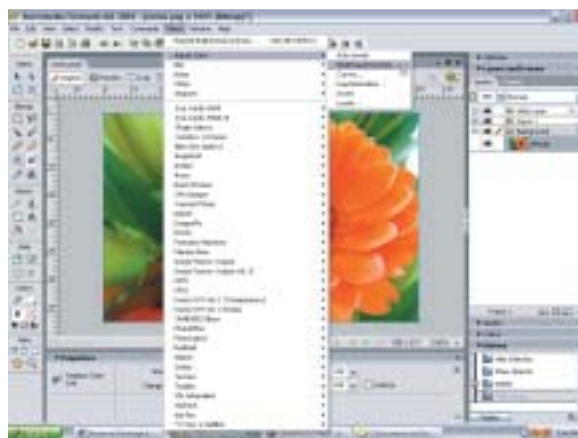


Image II

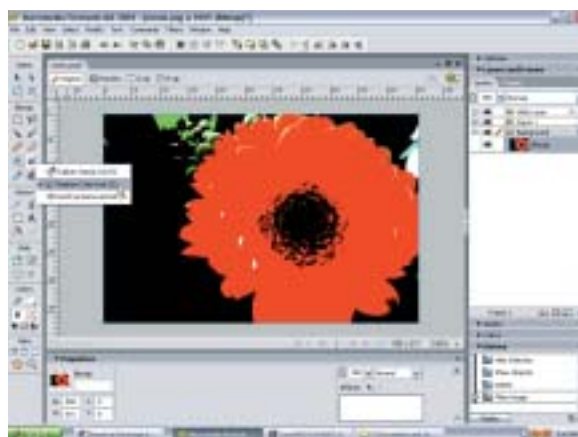


Image III

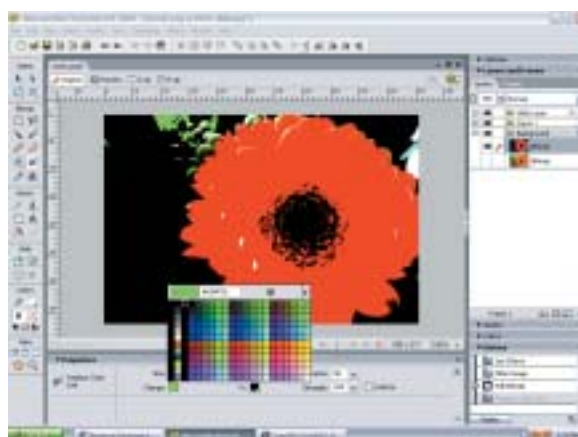


Image IV



image V

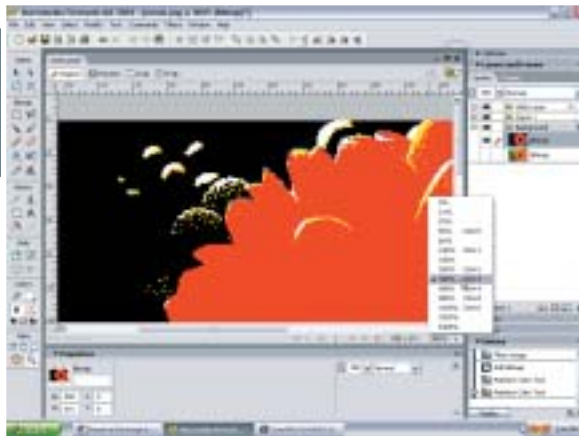


image VI

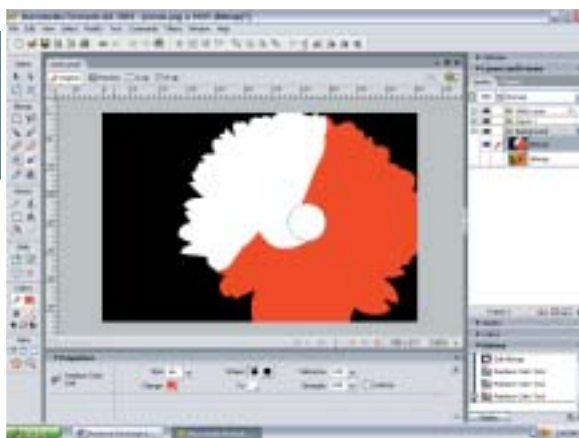


image VII

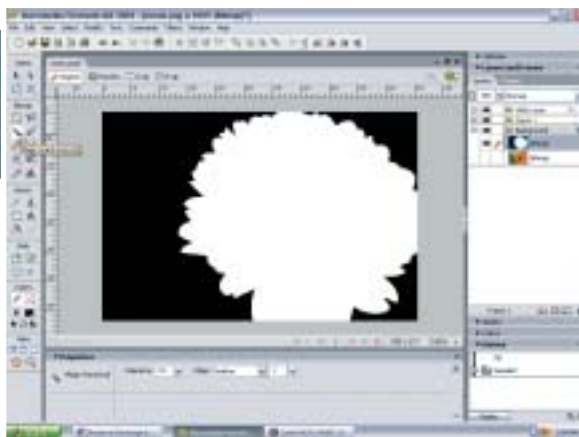
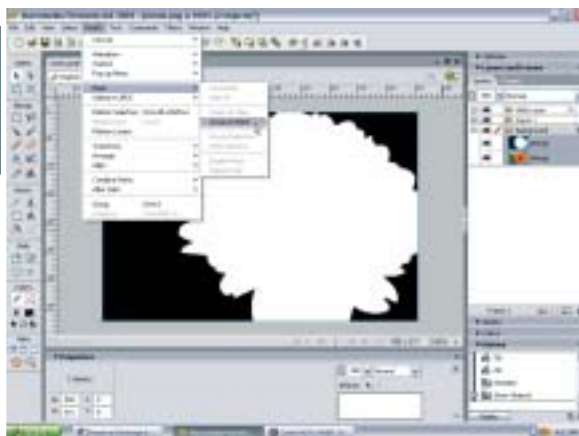


image VIII



The edges of the flower shape are now anti-aliased to the black background and our Alpha mask is complete.

### Applying the Alpha Mask

Before applying the mask we need to unhide the original image. In the Layers panel, make the hidden image visible again by clicking on its thumbnail. Now for the fun part!

- Press Shift and select both the Alpha mask thumbnail and the original image thumbnail.
- Choose **Modify \_ Mask \_ Group** as mask (see Image VII).

Shazam!

The black background disappears, instantly converting the flower to an image object floating on a transparent canvas (see Image IX). That is just way too cool. If you'd like to perform this trick again, choose **Modify \_ Ungroup**. Shift + select the two bitmaps again and choose **Modify \_ Mask \_ Group** as mask.

The edges of the knocked-out flower may still look a bit rough. In the next section I'll show you how to smooth them by applying some Fireworks Live Effects.

### Refining the Edges

The edges shouldn't need much touching up, particularly since you would reduce the dimensions of an image this size before using it on a Web page. If you feel the edges need smoothing all the way around the flower, then with the masked image still selected

- Click on the Add Effects button [+] on the Property inspector.
- Choose **Shadow & Glow \_ Glow**.
- Click in the color box of the Glow dialog and select a medium color from the edges of the flower petals (see Image X). If you're not satisfied with the color you can change it later.
- Click on the down arrow next to the Glow width text box and use the slider to change the width to 1.
- Set the Glow softness to 1 by means of its slider, or by simply typing a new number into the text box.

If just one side of the image edge needs smoothing, choose the Drop



# The **Leading Magazine** for Enterprise and IT Management



# LinuxWorld Magazine

There is no escaping the penetration of Linux into the corporate world. Traditional models are being turned on their head as the open-for-everyone Linux bandwagon rolls forward.

Linux is an operating system that is traditionally held in the highest esteem by the hardcore or geek developers of the world. With its roots firmly seeded in the open-source model, Linux is very much born from the "if it's broke, then fix it yourself" attitude.

Major corporations including IBM, Oracle, Sun, and Dell have all committed significant resources and money to ensure their strategy for the future involves Linux. Linux has arrived at the boardroom.

Yet until now, no title has existed that explicitly addresses this new hunger for information from the corporate arena. *LinuxWorld Magazine* is aimed squarely at providing this group with the knowledge and background necessary to make decisions to utilize the Linux operating system.

Look for all the strategic information required to better inform the community on how powerful an alternative Linux can be. *LinuxWorld Magazine* does not feature low-level code snippets but focuses instead on the higher logistical level, providing advice on hardware, to software, through to the recruiting of trained personnel required to successfully deploy a Linux-based solution. Each month presents a different focus, allowing a detailed analysis of all the components that make up the greater Linux landscape.

*Regular features include:*

*Advice on Linux Infrastructure*

*Detailed Software Reviews*

*Migration Advice*

*Hardware Advice*

*CEO Guest Editorials*

*Recruiting/Certification Advice*

*Latest News That Matters*

*Case Studies*

# SAVE 30% OFF!

REGULAR ANNUAL COVER PRICE \$71.76

# YOU PAY ONLY \$49<sup>99</sup>

12 ISSUES/YR

\*OFFER SUBJECT TO CHANGE WITHOUT NOTICE

# SUBSCRIBE TODAY!

WWW.SYS-CON.COM

OR CALL

1-888-303-5282

**FOR ADVERTISING INFORMATION:**

**CALL 201 802.3020 OR**

**VISIT WWW.SYS-CON.COM**



image IX



image X



image XI



image XII



Shadow Live Effect.

- Click on the Add Effects button [+] on the Property inspector.
- Choose Shadow & Glow \_ Drop Shadow.
- Click in the color box of the Drop Shadow dialog and select a medium color from the edges of the flower petals.
- Adjust the Shadow width to 1.
- Set the Shadow softness to 1.
- Click on the down arrow next to the angle text box and use the built-in dial to rotate the shadow to the correct position.

Other live effects you can use to adjust the edges of your extracted image are Inner Glow and Inner Shadow.

### Finishing Up

To check the quality of our work, let's add a background. Select the Vector Rectangle tool and drag out a white rectangle the same size as the canvas. With the rectangle still selected:

- Click on the down arrow of the Fill type selector and choose Gradient.
- Choose the Waves gradient from the gradient pop-up palette (see Image XI).
- Click in the Fill Color box to open the Gradient editor.
- Click on the black chip and change its color to medium gray or blue.

To send the gradient-filled rectangle behind the image, choose Modify \_ Arrange \_ Send to Back (see Image XII).

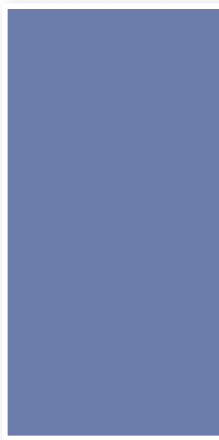
That's it! Congratulations on a job well done. I hope you'll make much use of the techniques you learned while following this exercise. ∞

*Linda Rathgeber is a writer, Web developer, and Team Macromedia member who coaches newcomers using Macromedia Dreamweaver and Fireworks. Her graphic work has been featured by the independent film company King Pictures, in book ads for author Bill Stott, in the Fireworks 4 and Fireworks MX sample files, and on the companion CD-ROMs of Joseph Lowery's Fireworks Bibles. Visit Lynde at [www.playingwithfire.com](http://www.playingwithfire.com) lightly@sc.rr.com*

# #1 Circulation in the World!

*JDJ is the highest circulation  
i-Technology magazine in the world!*

162,019\*



**Java  
Developer's  
Journal**

120,031\*



**Dr. Dobb's**

75,561\*



**MSDN**

\*JUNE 2003 BPA AUDIT STATEMENTS \*All circulation numbers are publishers' most current own data, six-month average circulation through June 2003.



Carmen Gonzalez  
Senior VP Marketing



Miles Silverman  
VP Marketing



# vectorvortex

# M

**FreeHand, Fireworks, and Flash**  
a combination that will save you  
time and money

by tom green

acromedia Studio MX includes a number of relatively unheralded gems. Perhaps it's because Macromedia hasn't hyped these features enough, or maybe users spend so much time focusing on a particular tool that they have simply overlooked or ignored how the other pieces of Studio MX actually work with each other. A great example of this is the stepchild of Studio MX: FreeHand MX.









For some odd reason FreeHand has fallen into the role of “always a bridesmaid, never the bride.” This started even before Macromedia existed. When PostScript drawing tools hit the graphics industry in 1988, the two heavyweights that fought it out for supremacy were FreeHand from Aldus and Illustrator from Adobe. Those of us who were there had ringside seats to an amazing slugfest between these two products as they attempted to dominate the market by “out-featuring” each other. When Adobe bought Aldus, the war ended and FreeHand, surprisingly, was sold to Macromedia. Illustrator rose to prominence and FreeHand essentially faded from the radar of the graphics community.

With Macromedia’s embracing of the Web, FreeHand became even more overlooked because graphic artists and Web developers still thought of it as a print tool. I must confess, I was one of them. It wasn’t until FlashForward 2000 when Todd Purgason of Juxt Interactive did a

developers a productivity tool that saves time and money.

The common factor among the three apps is vectors, and I’m going to show you how they integrate.

The first major area of integration is quite common among Flash MX developers. It is symbols. All three applications use them and the symbols produced in FreeHand MX and Fireworks MX 2004 move freely between the applications.

To demonstrate this point I’m going to use a logo. Inevitably, clients hand us a vector version of the logo and we either add it to a series of Web pages as a static branding element or we distort it or animate it in either Flash MX 2004 or Fireworks MX 2004. An issue we confront is that the logo is filled with Pantone colors. Moving a Pantone color from the CMYK color space to the RGB color space is a rather inconsistent process. The fidelity of the “converted” color as compared to the original color is like throwing grenades. The best you can hope to do is to come close. FreeHand, with its print roots, allows you to work with these colors and the conversion, at least from my perspective, is a bit more accurate than that produced by Fireworks MX.

In this illustration using the college’s logo, I have been supplied with an Illustrator .eps image filled with Pantone colors (see Image I). In order to prepare it for use in Fireworks MX 2004 or Flash MX 2004, I usually do two things:

1. Create a Symbol Library
2. Convert the color from the CMYK color space to the RGB space

Creating the symbol library is not at all difficult. In the case of the logo I determined that I needed only two symbols. The first will be the graphical section of the logo and the second will be the word “HUMBER.” I first select the logo and ungroup it – Modify, Ungroup. Ctrl-Shift-G (PC) or Command-Shift-G (Mac). In this way I have the various elements prepared for conversion to FreeHand MX symbols. If the elements to be converted to symbols are already ungrouped, I’ll marquee each using the Selection tool and group the selection – Modify, Group. Ctrl-G (PC) or Command-G (Mac).

Converting a selection to a symbol is simple in FreeHand MX. Select the object

to be converted and press the F8 key (see Image II). If you open the Library (F11) or click the Library tab on the Assets panel, you will see that the selected object has been converted to a symbol named “Graphic-01.” If you double-click the name in the Library you can change the name to something more meaningful such as “LogoGraphic.”

Unlike Fireworks MX 2004 or Flash MX 2004, selecting a symbol in FreeHand MX does not give you a visual clue – the little arrow in the bottom left corner of the selection – that the object is a symbol. Instead, the item will be selected in the library. Also, in certain instances, the FreeHand MX library will contain a Brushes folder. Feel free to delete it.

Once an object is converted to a symbol, all copiers of that object on the page are called “instances.” The neat thing about these is if you edit a symbol, all instances of that symbol on the page reflect the change. To edit a symbol, you double-click the image of the symbol in the preview area of the library. This opens the Symbol Editor. Any changes made will be reflected in the instances.

### Converting the Color

Though the Pantone images will be converted to RGB in Fireworks MX 2004 and Flash MX 2004, you can always take the attitude, “Hey, I’m here anyway.”

The conversion is done by opening the Swatch panel and double-clicking one of the listed Pantone colors. This will open the FreeHand MX Mixer pane. Click on the RGB button and the RGB values will be shown. Click once on the “Add to Swatches” button beside the color swatch to open the “Add to Swatches” panel. I rarely use the color breaks to name a color. In this instance I would name the color “Humber Gold.” Finally, the color is destined for Web use, so selecting either “Spot” or “Process” is immaterial. When you click the Add button, the dialog box disappears. When you click the Swatch tab, the new color appears in the color swatches (see Image III).

In order to avoid confusion you might wish to remove the Pantone colors from the swatches. What you can’t do is simply remove the color. FreeHand will not let you do this because, according to FreeHand, the color is in use and it will kick out a dialog box notifying you of this fact.



Image I



Image II

session on the FreeHand/Flash connection that FreeHand crossed my radar again. Two years later, when I was writing my book *Building Web Sites with Macromedia Studio MX*, I discovered that FreeHand is one serious Web application that, when teamed up with Flash MX 2004 and Fireworks MX 2004, gives Web

All instances of the Pantone color will have to be replaced with the new RGB version. This is not as complicated as it sounds.

Open the library, select a symbol, and double-click its preview to open the Symbol Editor. Change to the subselect tool and click on the area to be changed. Open the swatches and drag the color swatch into the selection. When finished, close the Symbol Editor (see Image IV).

Once the color in the symbols is in place, select the Pantone in the swatches, open the Options panel, and select "Remove" from the dropdown list. Save the image.

If the client is positively fanatical about accurate color, Pantone Corporation does have Hexadecimal versions of their colors. If this were the case with the college I would be replacing the 117CVC (Gold) with the approved #C49E0D and the 282 CVC with #00214D. In this instance I would do the color replacement in the Symbol Editor of Fireworks MX 2004. The reason is that the application was developed precisely for the purpose, among other things, of optimizing Web graphics.

### Moving FreeHand MX 2004 Symbols into Fireworks MX 2004

There are three ways of getting the artwork from FreeHand MX into Fireworks MX 2004. The first is to simply select the objects in FreeHand, copy them, open a new Fireworks MX 2004 document, and paste. The second method is to open a blank Fireworks MX 2004 document and simply drag the selected objects from the FreeHand MX page onto the Fireworks MX 2004 canvas. In both cases the objects will appear as individual layers in Fireworks MX 2004. The problem is that the symbols don't move. If you open the Fireworks MX 2004 library you will see that only the first symbol in the FreeHand MX library appears in the Fireworks MX 2004 library. In the case of the Humber logo this is not exactly a deal killer. You can select the Fireworks MX 2004 object missing from the library and convert it to a Fireworks symbol. You could also group the object on the FreeHand page and convert it to a single symbol (see Image V).

If the object is composed of multiple symbols, your best plan of attack is to

import the FreeHand MX file into Fireworks MX 2004. When the Vector Files Options dialog box appears, click OK. The cursor will change into a sideways "L." Click the cursor and the artwork drops onto the Fireworks canvas and the symbols move into the library.

If the original FreeHand drawing is composed of layers, they too will move into Fireworks MX 2004. All you have to do in this case is to select "Remember Layers" in the File Conversion area of Fireworks' Vector Files Options dialog box.

Once there you can apply filters and so on without affecting the original drawing. In the example shown in Image VI, I applied the EyeCandy 4000 Glass filter to the image. If you are thinking of applying the Fireworks effects that were added to FreeHand MX in FreeHand...don't. When the image is imported into Fireworks MX 2004, the FreeHand image becomes a dog's breakfast of masks and layers in Fireworks MX 2004.

### Moving FreeHand MX Symbols into Flash MX 2004

Moving the symbols from FreeHand MX into Flash MX 2004 uses techniques

such as "Drag and Drop," "Copy and Paste," and "Import." When it comes to Flash, however, there are some real issues to keep in mind.

If you "Drag and Drop" or "Copy and Paste" from FreeHand MX into Flash MX 2004 the objects will appear on the Flash stage and only the first FreeHand MX symbol will appear in the Flash MX library. You won't get the various symbols



Image III



Image IV





//  
...the only limit  
to the possibilities is  
the limit they...

...place on  
the creativity"

that compose the object (see Image VII).

In this case, the old adage "Let the software do the work" comes into play. Using the previous two techniques would result in some extra work. Instead open Flash MX 2004 and select File, Import, Import to Library. When the Import FreeHand dialog box opens, click OK. When you open the Flash MX Library you will have both symbols from the FreeHand MX library along with the composite artwork from FreeHand.

With the symbols in the library you can "play" with the logo by applying any number of the new timeline effects to the logo, distortions, or animations you may need. No matter what you do, the integrity of the original FreeHand MX logo is maintained because you are working with instances of the symbols in the Flash MX library.

If you don't need the artwork in Flash MX 2004 simply select "Export Symbols"

from the FreeHand MX library's Options menu. This will create a FreeHand document composed only of the symbols you chose, and these symbols can be imported directly into the Flash MX 2004 library or the Fireworks MX 2004 library (see Image VIII). In this way valuable vector drawings are available to everyone without compromising the integrity of the original, which can be stored in a secure location.

### 'Comping' Sites in FreeHand MX

When I tell developers they can create their wire frames or page "comps" in FreeHand MX, slice them up in Fireworks MX 2004, and then have them on the Web in short order thanks to Dreamweaver MX 2004, I am usually greeted with a blank look at worst or indifference at best. I attribute this attitude to FreeHand MX's

print roots. This is all doable, and the best part of it is that multiple pages in FreeHand are contained in a single FreeHand MX file. This makes file management an absolute breeze and injects efficiency into your workflow. Instead of a dozen separate comps that would need to be imported into Fireworks MX 2004, they can be imported from one FreeHand MX document.

In this example I am going to create some very simple comps that use text, images, and a simple navigation bar to the side for the main pages of a site that talks about a city in Bermuda. The point is to demonstrate a technique, not to win an award for art direction or Web design.

The first step is to open a new document in FreeHand MX. Once the document is open, select the Document tab in the Properties panel. In the Page Size dropdown select "Web" to give yourself a standard 550 by 400 Web page. If you need a different size, select "Custom" from the menu and, when prompted, enter the page dimensions. Build the page.

The layout calls for four more pages. This can be accomplished in two ways. The first is to select "Add Pages" from the Document panel's Panel Options. This will open the "Add Pages" dialog box. I entered "4" in the "Number of New Pages" input box. The interesting aspect of this dialog box is that you can change the size of the page if you choose. Having made your decision, click OK. The preview area of the Document Inspector will show you the thumbnails of the added

*Teacher, author, lecturer, chief cook and bottle washer. Instructor at Humber College's School of Media Studies in Toronto, Tom is also the author of Building Web Sites with Macromedia Studio MX and Building Dynamic Web Sites with Macromedia Studio MX 2004. Both are published by New Riders. tgreen17@cogeco.ca*



Image V



Image VI

pages. The second method is to click the Add Page icon at the bottom of the Document Window – it looks like a page with a “+” sign. The downside to this method is that you add one page at a time.

To move to the page, select the page number – found at the bottom of Document Window – from the pop-up list. When the page appears you can construct the page with graphics and text.

image VII



When finished you can select “Fit All” from the Page View pop-up at the bottom of the Document Window to get an overall view of the pages. As a Web developer you will most likely want to rearrange them to reflect a site map layout. To do this, select the Page tool on the toolbar and drag the pages into the order you wish. I love this feature because it’s like sticky notes with content on them. Once you’re finished, save the FreeHand MX document and quit FreeHand MX. After you save this file do not move it from its Save location.

Before we move into Fireworks MX 2004 there are two other major options available to you. You can select the Action tool, which looks like a blue dot with an arrow, click on a navigation element of the page, and drag to the page the element points to. That blue line shows the path. You can then open the Navigation panel and actually add Flash ActionScript to the document to allow the navigation. This will require saving the page as a .swf, but it is a quick way of creating a click-through for the client or other members of the team.

The other option is to turn the site map into a deliverable for the client. Remember, FreeHand is a vector application and as such turns out some rather crisp “proofs”

through laser printers or color printers that can be presented to the client.

### Importing Multiple-Page FreeHand MX Documents into FreeHand MX 2004

Of the features of the integration between FreeHand and Fireworks MX 2004 that are “slick,” this, to me, tops the list. The ability to bring only the pages I wish to manipulate, optimize, or other-

image VIII



wise prepare for the Web is a huge time-saver.

Launch Fireworks MX 2004 and select File, Open. Don’t click on the Open button on the Start page. Doing this will open the first page of FreeHand MX file and will not allow you to choose the page to work on. When the Open dialog box appears, navigate to the folder containing the FreeHand MX file and select it. The reason it was important not to move the FreeHand MX file to another location is that moving the file after the save will break the links to the images and Fireworks will prompt you to locate them.

Selecting the FreeHand MX file will open the Vector File Import Options dialog box. The area that is of prime importance to you is the “page” pop-down menu in the File Conversion area. When you open this menu, each of the pages in the FreeHand MX document will be listed. Select the page number you wish to open and click OK. The page will open in Fireworks MX 2004 (see Image IX).

From here you can optimize the images on the page and apply slices and links to the page using the Web Layer. You can also add Alt tags to the page or apply any of the Effects in the Fireworks MX 2004 Effects menu to the objects on the page.

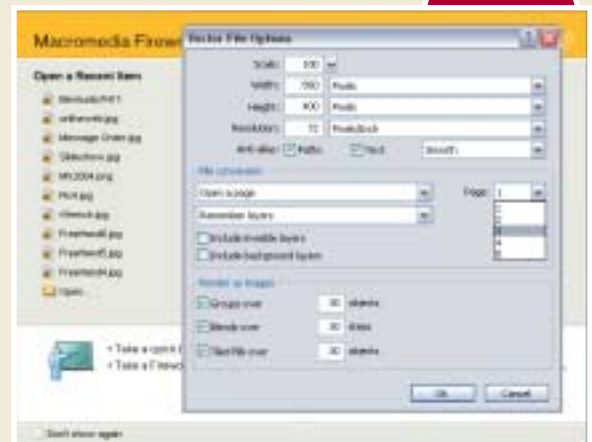
Though I have demonstrated two general techniques there are other uses for the

FreeHand MX to Fireworks MX 2004 workflow. For example, the Type management tools in FreeHand MX are far more powerful and precise than those in Fireworks MX 2004. In this instance you could, in FreeHand MX, create a series of headlines or subheads that can be placed into Fireworks MX. In this case you will have to decide whether to convert the text to outlines in FreeHand MX or permit Fireworks MX 2004 to handle the chore. You could

create a vector-based animation using FreeHand MX’s release to layers command and turn that animation into a .gif animation using the Frames feature of Fireworks MX. If I am doing animations in Flash MX, I will usually create all of my major headlines and subheads in FreeHand MX and bring them into Flash MX 2004. You could use FreeHand MX’s vector distortion filters to create some very interesting masks for use in Fireworks MX 2004. As I tell my students, the only limit to the possibilities is the limit they place on the creativity.

In the case of the FreeHand MX, Fireworks MX 2004, and Flash MX 2004 workflow, that limit is more a bias than anything else. ∞

image IX





# Swift Chart from GlobFX

*Professional charting for your ColdFusion applications*  
by Jason Clark

In its simplest form, text-based data provides a basic view of its structure and meaning. Charting enhances that view and puts the data in an entirely different light. ColdFusion has offered charting capabilities since version 5.0, but the functionality has always lacked some finesse – something that is present in other commercial charting solutions.

I discovered Swift Chart, a charting product, while working on a project for a client. This client required a charting solution that would produce professional-looking charts within their content management solution. Unfortunately for the client, ColdFusion's built-in charting didn't produce the impact that the client desired. After some fairly extensive investigating, I found GlobFX Technologies ([www.globfx.com](http://www.globfx.com)), the company behind Swift Chart.

GlobFX goes about their charting solution in a different sort of way. Their solution includes two products: Swift Chart and Swift Chart Generator.

Swift Chart is a robust client-side chart generation tool. This complete tool allows for data input to styling the chart, and then exporting them into several formats, such as SWF, PNG, BMP, and JPG to name a few. Swift Chart Generator is a server-side product that generates charts dynamically. It supports ASP, PHP, and ColdFusion, and runs on Windows, Linux, and Solaris. However, it only supports ColdFusion on the Windows platform, due to the use of COM. (COM is the means for ColdFusion to talk to the server.)

Both the client-side and the server-side products can generate a number of charts, including pie, bar, line, column, and XY scatter charts. GlobFX has just released a new version of the two products, which will additionally generate

combination, area, bubble, and stock charts.

## Making a Chart Style

Every copy of the Swift Chart Generator product includes the client-side tool, since the client-side tool also doubles as a "style sheet" generator for the server-side solution. This approach allows you to enter data into the client tool and see what your chart will look like as you change the various chart style

properties. When you have finished designing a chart style sheet, it can be saved and used when you generate your charts dynamically. If you're not a designer at heart, the client-side tool has several built-in styles to get you started.

There are many options in the client tool to assist in creating a chart style. We'll walk through the main tabs of the client tool. The Data tab is where you enter your labels and values. This data can either be entered manually or



Image 1



Image 2

imported from a spreadsheet (we'll discuss dynamic charting later in the article). After you have established the data, you're ready for the next step – styling the chart (see Image I).

As you can see, there are various built-in styles to choose from, as well as the ability to load previously created styles. To construct a new style, select a plot style and then begin customizing the various options to make your own unique look. Being able to change the visual impact of your charts live is a much more intuitive approach than changing source code. After selecting your plot style (i.e., bar, line, pie, etc.), you can then begin customizing the individual properties of the chart (see Image II).

The Plots tab has various options available, from changing the fonts and colors of the values to selecting the colors for the columns and borders. Aside from just choosing solid colors, you have the option of custom gradient fills, which adds some flair to the chart. The rest of the tabs permit customization to the axes, legend, title, background, and the general layout of the chart.

The interface to the client tool is quite easy to work with; it's very similar to a Paint type application. Almost every conceivable part of the chart can be customized from this tool: values, axis, legends, colors, and a neat feature animation. The charts already look fantastic, but the animation supplements the impact of the charts by animating the bars or pie slices as they are drawn. For an example of this, view the online example I created at [www.fusetalk.com/cfdj/swiffdemo.html](http://www.fusetalk.com/cfdj/swiffdemo.html).

### Server-Side Generation

After you've played the designer role and created chart styles, you can then either save your charts right from the client tool, or save the chart styles and plug them into the server-side generator. The server-side generator installs a COM object. This object provides the means for the communication between ColdFusion and the server. This permits you to create charts dynamically using the server-side generator.

This is a very simple task. There are two choices for talking to the COM object: JScript and VBScript. Each language has an array of methods and prop-

erties that you have at your disposal (see Image III).

If the application in which you are deploying the Swiff Chart Generator gets a fair amount of traffic, you can use the built-in caching functionality to alleviate the load. Swiff Chart Generator allows you to turn caching on and off, and to define where the chart cache will be kept from a server module (see Image IV). You can also change the caching parameters from the JScript or VBScript interface to the COM Object (see Code I).

GlobFX supplies quite a number of sample code snippets to get you started in generating charts. I've written a CFC to illustrate just how extensible, yet simple, this tool is to use.

As illustrated in this code snippet, talking to the COM object is very easy, and the properties that I've used in this example are only a snapshot of the properties available. To generate my first chart, I called the CFC with the code shown below, and produced a very professional-looking chart.

```
<cfscript>
    ChartObj =
    CreateObject("component", "swiffchart");

    ChartObj.GenerateChart("a|b|c|d|e", "1|
2|3|4|5", "2|4|6|8|10", "First
Series", "Second
Series", 560, 420, true, "My First
Chart", "the subtitle", "c:\inetpub\www-
root\SanFrancisco.scs");
</cfscript>
```

Pricing for Swiff Chart starts at \$149 USD for the client tool and \$799 USD for the Generator tool, which includes a copy of the client tool (see Image V). GlobFX ([www.globfx.com](http://www.globfx.com)) also offers evaluation versions of both products, which are downloadable from their Web site. If professional charts are a requirement in your next project, this product should be on your list. It's easy to use and flexible, and the great-looking charts make it a strong contender in aftermarket charting solutions. ∞

*Jason Clark, cofounder and CEO of Fusetalk Inc., has more than 10 years of experience in programming, specifically, and the IT sector, in general.*  
[jclark@fusetalk.com](mailto:jclark@fusetalk.com)

image III



image IV

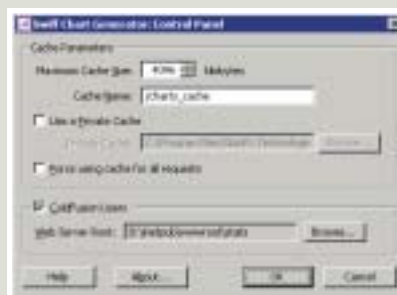


image V



“ColdFusion has offered charting capabilities since version

5.0



code 1

```
<cfcomponent>
  <cffunction name="GenerateChart"
  access="public" output="yes"
  returntype="string">
    <!-- BEGIN ARG DECLARATION -
    -->
    <cfargument
    name="CARG_LABELS" type="string"
    required="true">
      <cfargument name="CARG_VAL-
      UES" type="string" required="true">
        <cfargument name="CARG_VAL-
        UES2" type="string"
        required="true">
          <cfargument
          name="CARG_SERIES1CAPTION"
          type="string" required="true">
            <cfargument
            name="CARG_SERIES2CAPTION"
            type="string" required="true">
              <cfargument name="CARG_WIDTH"
              type="numeric" required="true">
                <cfargument
                name="CARG_HEIGHT" type="numeric"
                required="true">
                  <cfargument name="CARG_ANI-
                  MATE" type="boolean"
                  required="true">
                    <cfargument name="CARG_CHART-
                    NAME" type="string"
                    required="true">
                      <cfargument name="CARG_CHART-
                      SUBTITLE" type="string"
                      required="true">
                        <cfargument name="CARG_CHARTSTYLE" type="string"
                        required="true">

    <!-- LOCAL VARIABLES --->
    <cfscript>
      var seperator = "|";
      var ignoremultseperators = true;
      var chartoutput = "";
    </cfscript>

    <!-- INSTANTIATE CHART OBJECT --->
    <cfobject type="com" name="chart"
    class="SwiffChartObject.ChartObj.1" action="create">

    <cfscript>
      ignoremultseperators= true;
      chart.SetSeparators(seperator, ignoremultsepara-
      tors);

      // SET THE CHART LABELS

      chart.SetCategoriesFromString(arguments.CARG_LABELS);

      // ADD ONE SERIES AND SET THE CHART VALUES
      chart.AddSeries();
      chart.SetSeriesValuesFromString(0,
      arguments.CARG_VALUES);
      chart.SetSeriesCaption(0,
      arguments.CARG_SERIES1CAPTION);

      // IF ANOTHER SERIES IS PASSED ADD IT TO THE CHART
```

```
      if(Len(arguments.CARG_VALUES2) gt 0){
        chart.AddSeries();
        chart.SetSeriesValuesFromString(1,
        arguments.CARG_VALUES2);
        chart.SetSeriesCaption(1,
        arguments.CARG_SERIES2CAPTION);
      }

      // SET THE CHART TITLE
      chart.SetTitle( arguments.CARG_CHARTNAME );

      // IF A SUBTITLE IS PASSED
      if(Len(arguments.CARG_CHARTSUBTITLE)){
        chart.SetSubtitle(arguments.CARG_CHARTSUBTITLE);
      }

      // APPLY THE CHART STYLE
      chart.LoadStyle( arguments.CARG_CHARTSTYLE );

      // VARIOUS PARAMETERS
      chart.SetWidth( arguments.CARG_WIDTH );
      chart.SetHeight( arguments.CARG_HEIGHT );
      chart.SetLooping( false );
      chart.CompressSWF( true);
      chart.AnimateChart( arguments.CARG_ANIMATE );

      // GET THE HTML FOR THE CHART AND OUTPUT IT
      chartoutput = chart.GetHTMLTag();
      writeoutput(chartoutput);
    </cfscript>
  </cffunction>
</cfcomponent>
```

Advertiser	URL	Phone	Page
ActivePDF	www.activePDF.com	949-582-9002	37
American Media Training	www.americanmediatraining.com	310-258-8068	3
CFDynamics	www.cfdynamics.com	801-394-1254	81
ComCity	www.salecart.com	888-826-6248	6
Edge Web Hosting	www.edgewebhosting.net	866-EDGEWEB	45
Edith Roman	www.edithroman.com	800-223-2194	97
Future Media Concepts	www.fmctraining.com	877-362-8724	35
HostMySite.com	www.hostmysite.com	877-248-4678	9
Interakt	www.interakt.ro/tools	4021-312-5312	25
Java Developer's Journal	www.sys-con.com	201-802-3021	71
Kaosweaver	www.kaosweaver.com		15
LinuxWorld Magazine	www.sys-con.com	888-303-5282	69
Macromedia	www.mx2004.com		27-29
Macromedia	www.macromedia.com/go/cfmxad	415-553-7186	Cover IV
MX Developer's Journal	www.sys-con.com/mx/subscription.cfm	888-303-5282	59
NetQuest	www.nqcontent.com	357 22 813560	Cover III
New Atlanta	www.newatlanta.com	678-256-3011	87
Oddcast Media Technologies	www.oddcast.com	212-375-6290	Cover II
PaperThin	www.paperthin.com	800-940-3087	47
Seapine Software	www.seapine.com	888-683-6456	11
SYS-CON Events	www.sys-con.com	201-802-3069	65
SYS-CON Media	www.sys-con.com/2001/syb.cfm	888-303-5282	91



# WHERE COLDFUSION EXPERTS HOST



*"CFDynamics is everything I look for in an ISP. With more features and fewer restrictions than similar provider plans, consistent top-notch customer service and incredible attention to detail, CFDynamics immediately comes to mind whenever CF developers ask me to recommend a host that delivers great service at a decent price."*

Simon Horwith

Macromedia Certified Instructor  
Certified Advanced ColdFusion  
MX Developer  
Certified Flash MX Developer

At CFDynamics, we put the warmth back into client relationships. Our technical support staff consists of personable and intelligent CF system administrators who really enjoy solving problems. We don't employ machines to do our telephone work; and we never mix sales with service.

At CFDynamics, ColdFusion web hosting is our complete focus. You'll speak with a real system administrator every time you call. You'll also enjoy real benefits, not just promotional tidbits. With accounts starting at \$16.95/month, our clients enjoy free SQL Server Access, free account setup, unlimited email accounts and a 30-day money-back guarantee. Real service. Real satisfaction. Forget about cold computer voices and confusing phone loops - let CFDynamics show you the warmth in ColdFusion hosting.

**CFDynamics**  
A Division of Resonance, Inc.

Are you going to MAX in November 2003?

Take the:

**MAX**  
**Challenge!**  
2003

[www.cfdynamics.com/maxchallenge](http://www.cfdynamics.com/maxchallenge)

866 - CFDYNAMICS

[WWW.CFDYNAMICS.COM](http://WWW.CFDYNAMICS.COM)

# coldfusion in the t r e n c h e s

## CFMX Is RX for Midwest Health Care Organization

by ben forta



### The Organization

The Midwest is home to a world-class health-care provider, research facility, and medical school. For close to a century, this organization has been pushing the limits of modern medicine, while making significant contributions to the medical community.

The organization has treated more than 6,000,000 patients since its founding. Today it is a multibillion dollar operation with facilities in several states, employing more than 45,000 physicians, nurses, and health-care workers, and just as committed as ever to excellence in health care and medical research.

### The Web Sites

The organization actually has two very different public-facing sites with very different goals.

The public .com site is a consumer-information site serving over 12,000,000 page views each month. It provides extensive articles and information on diseases and medical conditions, prescription and over-the-counter drugs, healthy living, preventative medicine, and more. Under the banner proudly declaring "Reliable information for a healthier life," a vast amount of content is made available at no cost to readers. Visitors can sign up for regular e-mail newsletters and even register for personalized content relating to information they provide.

The organization also maintains a dedicated site specifically for patient care on a .org domain. This multilingual site provides

physicians and patients with vital personalized content, and also lists job postings, research, medical services, international services, and other related information. There are also numerous internal sites and applications (see Image 1).

### Using ColdFusion

By now it should be blatantly apparent that the public-facing Web sites are highly content-centric. In fact, it would not be a stretch to consider them to be massive content management and publishing systems. And at the heart of the content management systems sits ColdFusion.

The organization has been using ColdFusion since the early Allaire days. And as ColdFusion evolved, their developers have been quick to leverage the features and capabilities it offered. The sites now use a combination of ColdFusion 4.5, ColdFusion 5, and ColdFusion MX running on Windows and Solaris boxes (using IIS and Apache). As is the case in all large organizations, data is stored in a variety of databases, including Oracle, Sybase, Microsoft SQL Server, and DB2.

### Configured for Scalability

The production .com site is split across two physical data centers linked by fiber backbones. The network architecture and configuration has been optimized for high-performance content management. Each location is configured as follows:

- Arrowpoint (now Cisco) switches



ColdFusion helps one of the world's premier health-care providers to deliver quality and timely content



- SonicWall SSL accelerators to improve the performance of secure data transmission
- Five clustered ColdFusion servers running on Windows 2000
- Three clustered dedicated media servers running on Linux (so as to reduce load on the application servers)
- Two clustered Verity K2 servers running on ColdFusion MX on Windows 2000
- Two Sybase servers running on Solaris, configured for failover

Image I



- Internal newsletter application that can be customized by department
- Committee Minutes system that allows for the archiving and retrieval of minutes
- Employee directory services system
- IT issues tracking system
- Job posting board

Similar configurations are employed on the development and build sites so as to eliminate deployment problems caused by differing environments (see Image II).

The use of dedicated servers is important. Early performance problems taught the application developers that static media content (images, graphics, any media saved to disk) can be served more efficiently from dedicated boxes. This way the media is served faster (there is no application server running to slow things down), and the ColdFusion servers can run faster too (as resources are not being used for responding to simple HTTP requests). Similarly, the dedicated Verity K2 servers allow for lightning-fast indexing and searches with no other processing getting in the way. This is a well-thought-out and highly scalable architecture, one that also allows for future growth and expansion.

And this is just one of the public-facing .com sites. ColdFusion is used for all sorts of applications (both public facing as well as internal):

- Content creation and management
- Content workflow and approval
- Content publishing and data searching
- Human resources self-serve applications (like timecard and benefits management)
- Change management tracking, which is crucial for compliance with recently mandated HIPAA (Health Insurance Portability and Accountability Act of 1996) regulations
- Single Sign On system (built on Web services), which allows corporate clients to seamlessly integrate their custom-hosted corporate sites into their own corporate portal without the need for additional login steps

ColdFusion was originally brought in to replace an existing Interwoven solution, and to bring hosted applications in house. As per an internal technical specialist, ColdFusion is used because of its

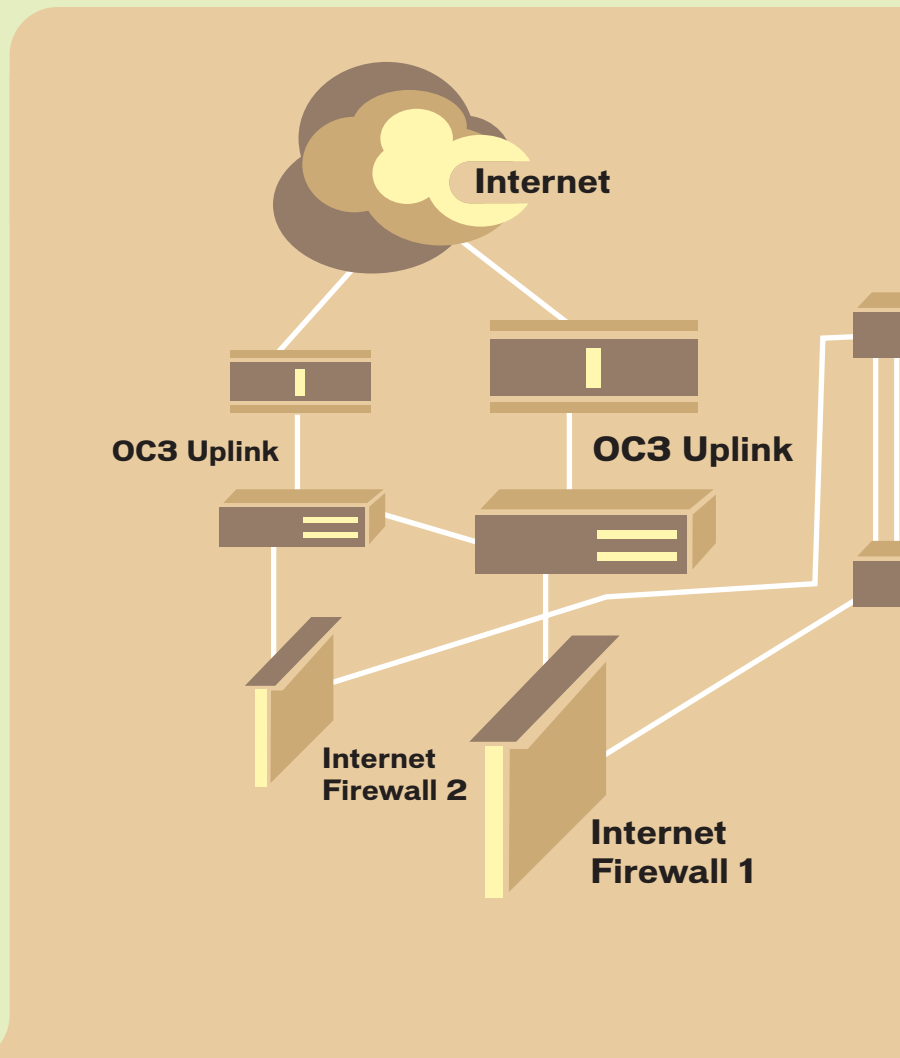
“rapid development, ease of use, and minimal learning curve.”

This is not to say that ColdFusion is the answer to all problems – on the contrary. Whereas ColdFusion empowers users with “rapid prototyping and quick results,” some developers have failed to understand where ColdFusion should be used and where it should not. Some early attempts at building entire complex systems in ColdFusion failed to scale. As the technical specialist puts it, “it is so easy to develop things that just shouldn’t be developed in CF.” But with that important lesson learned, ColdFusion is hard at work doing what it does best.

*Note:* The subject of when and when not to use ColdFusion is frequently covered in my *CFDJ* column <BF> on <CF>.

### Developers and Development

The sites (and their supporting applications) are created and maintained by



# The content-management systems empower nonprogrammers to deliver compelling content, scalable content, and even entire custom sites for corporate clients

hundreds of developers, with seven dedicated full time to maintaining the public-facing .com site. Developers have a diverse set of skills, from ColdFusion to Java to .NET to JavaScript and CSS to database administration and network management. They use Macromedia Dreamweaver, ColdFusion Studio, HomeSite, and Macromedia Flash, and some content creators use the recently released Macromedia Contribute.

Having access to so many different but complementary skill sets is invaluable, and has allowed the various teams to build well-designed and highly structured applications, using the various

technologies for what they do best.

ColdFusion is used for most of the presentation layer. Back-end processing is implemented in a mixture of .NET and Java (especially when integrating with mainframes and legacy systems) and some ColdFusion as well. Being able to develop back ends on any appropriate platform and front ends in ColdFusion makes for an ideal and very compelling combination of power and productivity.

On the .com site, for example, a number of the health-management programs utilize both ColdFusion and Java.

ColdFusion handles the display and personalization components, and JavaBeans are used for the number crunching and data manipulation. In addition to being more scalable, this split saves time as development can be split across team members leveraging their respective skills.

image 11

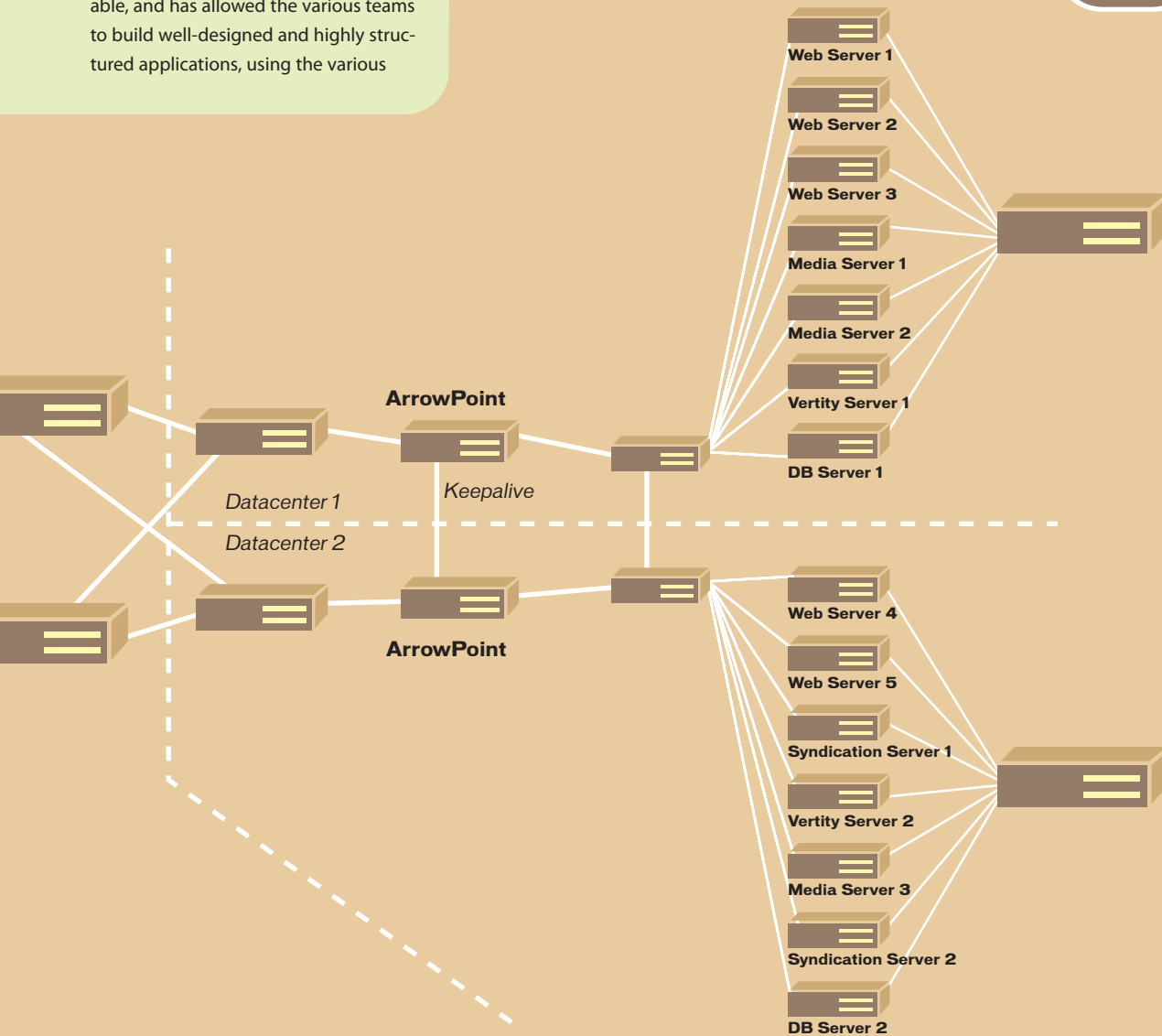




image III

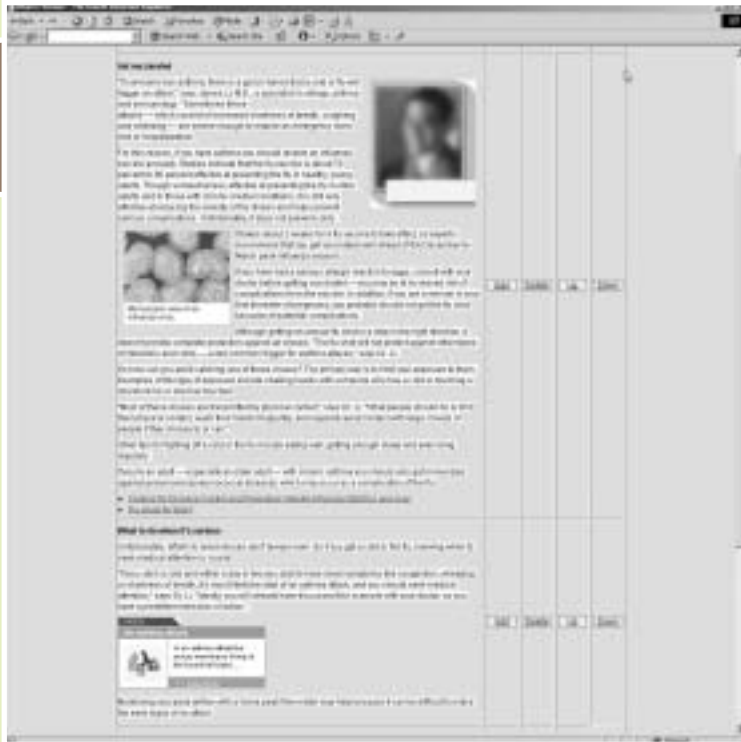


image IV



image V



Recent development projects have employed MVC (model-view-controller) design, and ColdFusion is used at the V and C layers. Several applications employ the community-driven Fusebox methodology and framework ([www.fusebox.org](http://www.fusebox.org)) to facilitate code reuse, simplified maintenance, and troubleshooting. And the development team is now working with Mach-II (a new Web application framework focused on easing software development and maintenance, [www.mach-ii.com](http://www.mach-ii.com).)

Scaling up and monitoring a variety of systems is always a concern. Using a Mach-II based application, JMX interfaces can be accessed for an immediate snapshot of system health. The framework also allows for cloning of new control interfaces to different systems as they come online.

### Content Management

The core content management systems are powered by two ColdFusion-powered solutions. CommonSpot (by PaperThin, Inc.) is used on a research site because it works well out of the box, and is also flexible, extensible, and very lightweight. The .com site is built on top of a highly modified Macromedia Spectra application. Yet another Spectra modification is in the works, this time to migrate from WDDX ([www.openwddx.org](http://www.openwddx.org)) data stores to XML using XPath for searches.

These content-management systems empower nonprogrammers to deliver compelling content, locale-based content, and even entire custom sites for corporate clients. Special features (like various health-related calculators) can be embedded in sites quickly and easily. Project personnel can create and edit articles consisting of embedded images, popups, slideshows, and other multimedia via simple Web interfaces (see Image III). Using this simple interface they can move sec-

tions around, delete them, and create new ones. Other screens (like the one shown in Image IV) allow for reordering the content in articles, inserting and removing content, and making quick edits.

All site navigation elements are XML driven. Obviously, requiring staff to manipulate XML would just not work, so data-entry screens (like the one shown in Image V) allow project personnel to interactively build these elements while previewing the end result. The preview uses a Macromedia Flash movie that is dynamically updated using the generated XML.

To improve performance, not all content is served dynamically. A custom ColdFusion application dynamically renders static HTML pages when content changes. This dramatically improves the performance of highly content-centric applications where content tends to be created more often than updated. (Incidentally, similar techniques are employed on parts of Macromedia's own site, [www.macromedia.com](http://www.macromedia.com).)

### What's Next

With an eye on current trends and future technology, the organization is already leveraging SOAP-based Web services. ColdFusion MX powers data syndication feeds to CNN, Walgreens, and others. IBM WebSphere Portal is now being deployed, and ColdFusion MX will be an important part of that initiative too.

### Summary

This health-care pioneer was an early adopter of ColdFusion, which has grown to play a significant role within the organization's Internet and intranet presence. With a diverse mixture of developers and skill sets, and with highly content-centric applications, ColdFusion is a perfect fit for the fast-growing and ever-changing needs of one of the world's foremost health organizations. ☺

*Ben Forta is Macromedia's senior product evangelist and the author of numerous books, including ColdFusion MX Web Application Construction Kit and its sequel, Advanced ColdFusion MX Application Development, and is the series editor for the new "Reality ColdFusion" series. For more information visit [www.forta.com](http://www.forta.com). [ben@forta.com](mailto:ben@forta.com)*

# BlueDragon

# is CFML

# BlueDragon



**BlueDragon runs CFML natively within**  
**A Standalone Server**  
**Any J2EE Application Server**  
**The .NET Application Framework**



With over 10,000 corporate and individual customers in over 65 countries around the world, New Atlanta has provided reliable server-side products since 1997.

# Robust E-mail Notification with ColdFusion

*Using CFMAIL in new and exciting ways*

by scott van vliet

There are a number of features included in ColdFusion MX 6.1 that solidify its position as the best choice for Web application development – from both the developer and client perspectives. These features include enhanced performance and scalability, as well as enhancements to key functionality, including the CFMAIL tag.

## The Challenge

Last year, I was given an RFP to implement a Digital Asset Management System (DAMS) for a leading Japanese automotive manufacturer based in Gardena, California. The general system requirements asked for a unified system that would provide users

across North America with access to digital assets for use in advertising and branding efforts. My first thoughts were, “Okay, I’ll go and find vendors who provide this service or find a packaged application that addresses the requirements outlined in the RFP.” Piece of cake, right? Not really.

After a week of research, I came across three companies that offered such a service. But the prices for these services were outrageous! Not only was there an initial setup fee, there was a per-seat user license. When calculating the cost for the thousands of dealers, vendors, and regional and corporate users that would be accessing this site, we knew that this was not an option.

We were coming down to the wire on the deadline for the proposal and we had yet to select our proposed solution. Then the project manager asked, “How long would it take to develop a custom digital asset management system for the client?” I was taken aback! In the past, I have always been pushed to find a pre-built or customizable solution. “Depends on a few things – most importantly the platform,” I responded.

The project manager mentioned that the client was predominantly a Java shop; however they were in the process of implementing .NET into their infrastructure.

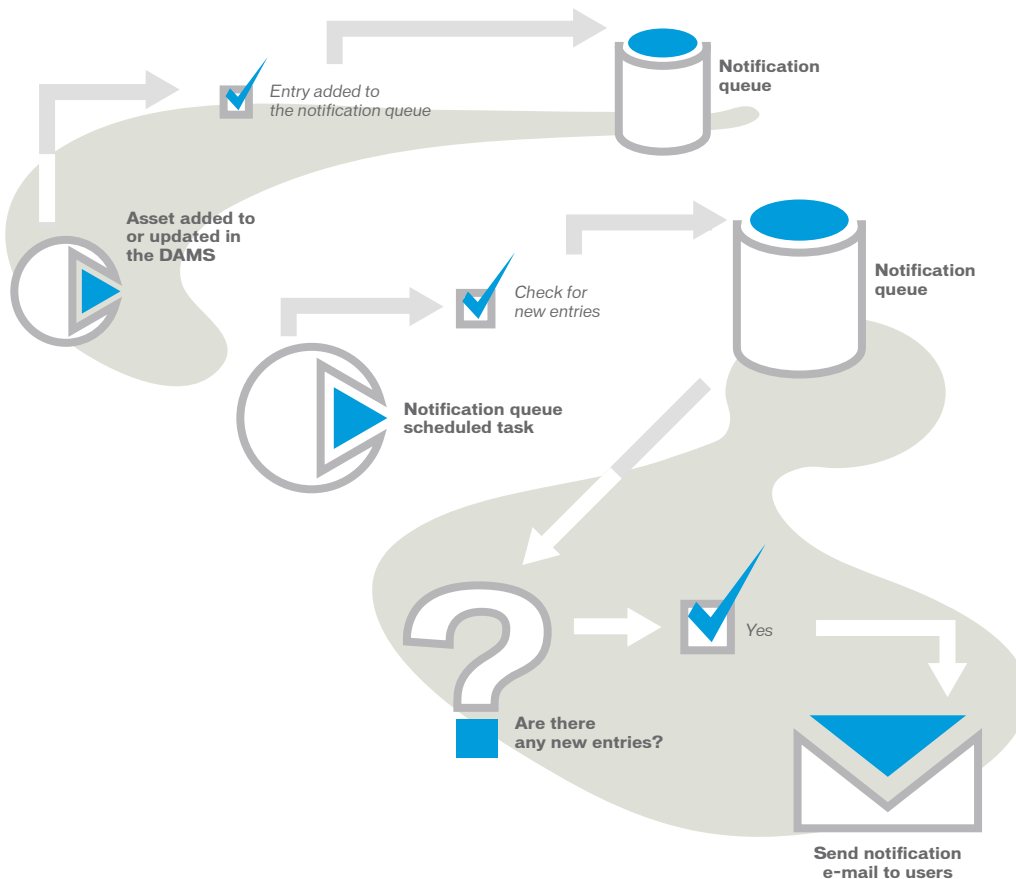
“Well, if we choose Java, it will take anywhere from 12–18 months. With .NET, it will take approximately 8–10 months,” I responded. This was not going to work, since the RFP asked for a six-month timeline on the implementation of the proposed solution. Well, we can deliver a custom enterprise solution to them in less than six months,” I informed the project manager. “Really?” he asked. “Yes...” I replied, “with ColdFusion.”

## The Obvious Choice

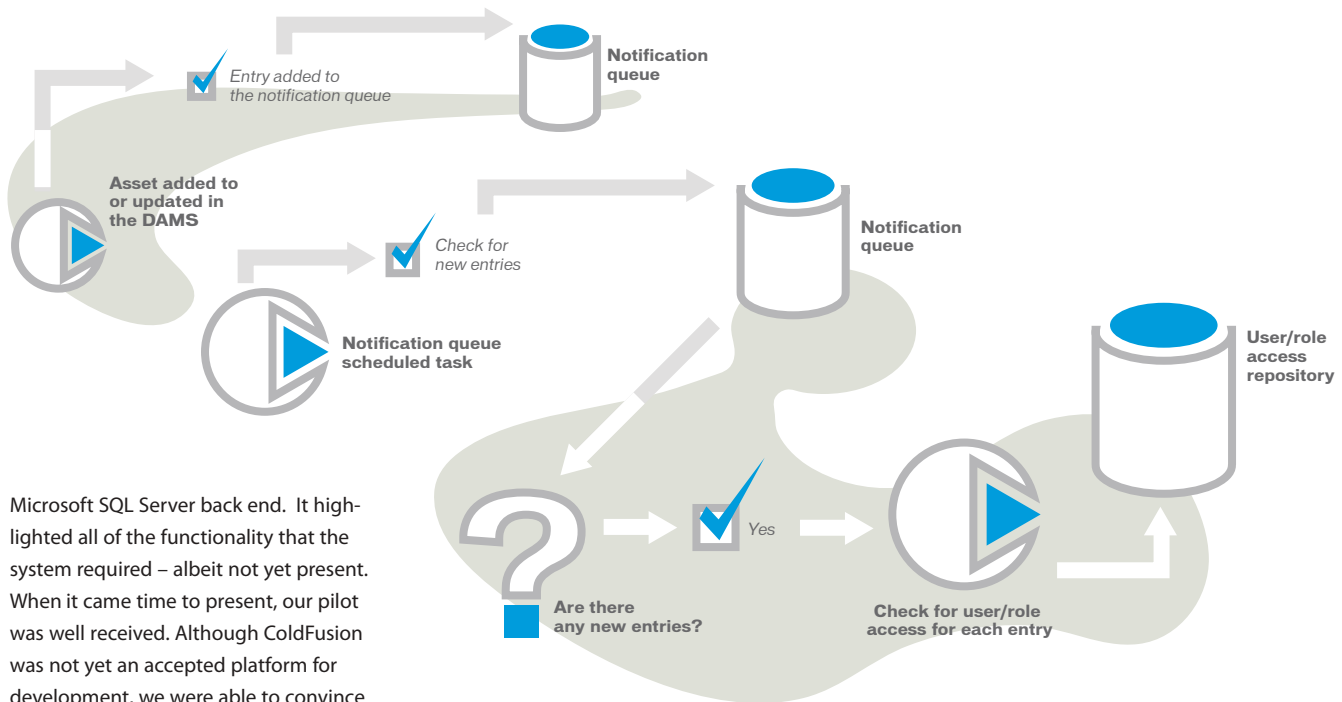
After some deliberation, I was able to convince the project manager to pitch a custom solution utilizing ColdFusion. However, I was to deliver a demonstration of the proposed system when the proposal was due – in two weeks. No sweat!

Ready to have some fun, I returned to the RFP and began to access the functionality required of the proposed solution. These features included file and file metadata management, category management, workflow management, e-mail notification, custom reporting, and user/role-based security management. Without a pause for breath, I began working on the pilot to demo for the proposal delivery.

Within a week, I had a partially functional demo – built in ColdFusion with a







Microsoft SQL Server back end. It highlighted all of the functionality that the system required – albeit not yet present. When it came time to present, our pilot was well received. Although ColdFusion was not yet an accepted platform for development, we were able to convince the client otherwise and managed to win the project. Within a few days, we began development of the custom DAMS.

Now, the client has a robust, scalable DAMS that was built with their needs in mind, and has the flexibility to be modified with ease. Add another notch to ColdFusion's belt!

### Enhancements to the CFMAIL Tag

One of the features that this system offers is e-mail notification. This feature is as trivial as it is mission critical. There are several products on the market that provide e-mail list functionality. However, most of these products do not offer the appropriate functionality to integrate seamlessly with the applications that we ColdFusion developers create and have to work with.

Fortunately we have been blessed with the simplicity of the CFMAIL tag, which has solved our e-mail needs. Now, with ColdFusion MX 6.1, we have even greater power to deliver secure e-mail applications with SMTP authentication.

Code I shows an example of the CFMAIL tag with SMTP authentication.

Notice the two new attributes to the CFMAIL tag: username and password. These attributes represent the Username and Password that are required by some mail servers to authenticate before a client can send e-mail messages.

This new feature of the CFMAIL tag has opened up a new door for our DAMS. Until

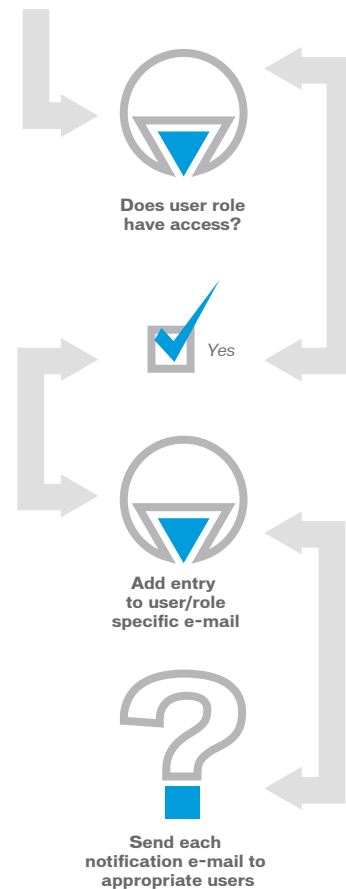
ColdFusion MX 6.1, we were relying on our own mail servers to deliver daily and weekly notifications to the thousands of users of this system. Not only was this a load that we did not want to carry, we had already subscribed to a service that provided SMTP access for other applications that we supported. However, these SMTP servers were open only to authenticated users.

Now that we utilize SMTP authentication, we have been able to streamline our application, and our network, by isolating all e-mail-related functionality to our subscription service.

### Behind the Scenes

One of the requirements for the e-mail notification function that the DAMS required was to send daily update e-mails to users of the system with all of the additions and updates made to the system that day. Thus, when an asset is added to or updated in the DAMS, an entry is made within a Notification Queue. This queue is scanned nightly for new entries by a ColdFusion Scheduled Task. If new entries are found, they are purged into a list of new entries. This list of new entries is distributed to the users of the DAMS. Image I outlines this process.

This model would work. However it does not take into consideration this system's security model, which includes user- and role-based access restriction to assets within the DAMS. Therefore, users



should not receive a notification about an asset that they do not have access to.

With this in mind, we had to rethink how the Notification Queue worked. The ColdFusion Scheduled Task would have to check if a user had access to a particular asset before the Notification Queue

Image II



Scott Van Vliet is a senior consultant for Hitachi Consulting. He has worked with ColdFusion for over four years in the automotive, entertainment, and telecommunication industries. svan.vliet@hitachiconsulting.com

entry could be sent to that user. Image II outlines the revised process.

Now that we have rethought the process, how do we achieve this in code? Simple! Let's start by dissecting the ColdFusion Scheduled Task that actually sends these messages.

The first thing we must do is check the Notification Queue to see if there are any new entries. (see Code II)

If the value of EntryCount from the query above is zero, then we exit the task since there are no entries to purge and send. However, if EntryCount is greater than zero, we must continue with the task as there are new entries in the Notification Queue.

```
<cfif selNotificationQueue.EntryCount>
```

Now that we know we must purge the Notification Queue and send an e-mail message to each authorized user, we can begin to build the message.

First, we load in our default header and footer text from text files located on the server whose paths are stored in Request variables in the Application.cfm

(see Code III).

Now, we must obtain a list of the users in the system. We do this by querying the Users table (see Code IV).

With our list of users, we can now iterate over them to create a customized e-mail message based on their permissions. To do this, we query the Notification Queue to find entries that the user has access to. We determine whether or not a specific user has access to an entry with a user-defined SQL function that checks the restrictions table for both user and role access (see Code V).

If the query in Code V returns more than one record, then the user has access to one or more new entries in the Notification Queue. Therefore, we must send them an e-mail message (see Code VI).

In Code VI, we begin the construction of our user-specific e-mail message. After setting the subject and the content header, we loop over each entry from the Notification Queue that the user has access to. Then, we complete the content by adding the footer text.

Now we are ready to send the user-specific e-mail message. Once the message has been sent, we continue looping

through the list of users, sending each user a custom e-mail message based on their access (see Code VII).

Once we have finished looping over all of the users, we must now update the status of the entries that were sent in this task to Sent, stored in the database as 1. We can use the ValueList() function to obtain a list of the Entry IDs (see Code VIII).

That's it! We have now updated the users of the DAMS with relevant information about only those assets that they have access to.

## Conclusion

The enhancements to CFMAIL are just the icing on the cake for ColdFusion MX 6.1. The performance gains I've observed in the applications I've built running this release are astounding, and continue to motivate me to introduce ColdFusion to all of my clients. The Notification Queue dissected in this article is only one of the many robust systems that can be built using CFMAIL. I would encourage you to extend your thoughts beyond e-mail forms and imagine how you can use CFMAIL in new and exciting ways. ☺

code I

```
<cfmail to="user@domain.com"
from="notifications@domain.com"
subject="Asset Updates - 9/4/2003"
server="smtp.domain.com"
username="username"
password="password">
</cfmail>
```

code II

```
<cfquery name="selNotificationQueue"
datasource="#request.datasource#">
SELECT      COUNT(EntryId) As EntryCount
FROM        NotificationQueue
WHERE       EntryWasSent = 0
</cfquery>
```

code III

```
<cffile action="read"
file="#request.notificationHeaderText#"
variable="header">

<cffile action="read"
file="#request.notificationFooterText#"
variable="footer">
```

code IV

```
<cfquery name="selUsers" datasource="#request.data-
source#">
SELECT
        UserId, UserFullName, UserEmail
FROM
        Users
</cfquery>
```

code V

```
<cfloop query="selUsers">
        <cfquery name="selEntriesForUser"
datasource="#request.datasource#">
SELECT      EntryId, EntryText
FROM        NotificationQueue
WHERE       EntryWasSent = 0
AND         dbo.IsUserExcluded(AssetId,
#selUsers.UserId#) = 0
```

```
</cfquery>
```

code VI

```
<cfif selEntriesForUser.RecordCount>
        <cfset subject = "Asset Updates - " &
DateFormat(Now(), "m/d/yyyy")>
        <cfset content = header>
        <cfloop query="selEntriesForUser">
<cfset content = content &
selEntriesForUser.EntryText & Chr(13) & Chr(10)>
        </cfloop>
        <cfset content = content & Chr(13) &
Chr(10) & footer>
```

code VII

```
<cfmail
to="#selEntriesForUser.EntryEmail#"
from="#request.notificationFromEmail#"
subject="#subject#"
server="smtp.domain.com"
username="username"
password="password">
#content#
</cfmail>
```

code VIII

```
</cfif>
</cfloop>

<!-- Now that these Entries have been sent,
mark them as old -->
<cfquery name="updNotificationQueue"
datasource="#request.datasource#">
UPDATE
        NotificationQueue
SET
        EntrySent = 1,
EntrySentDate = getDate()
WHERE
        EntryId IN
(#ValueList(selNotificationQueue.EntryId)#)
</cfquery>
</cfif>
```

# SUBSCRIBE TODAY TO MULTIPLE MAGAZINES

## AND SAVE UP TO \$400 AND RECEIVE UP TO 3 FREE CDs!\*



RECEIVE YOUR DIGITAL EDITION ACCESS CODE INSTANTLY WITH YOUR PAID SUBSCRIPTIONS

### 3-Pack

Pick any 3 of our magazines and save up to **\$275<sup>00</sup>**  
 Pay only \$175 for a 1 year subscription plus a FREE CD

- 2 Year – \$299.00
- Canada/Mexico – \$245.00
- International – \$315.00

### 6-Pack

Pick any 6 of our magazines and save up to **\$350<sup>00</sup>**  
 Pay only \$395 for a 1 year subscription plus 2 FREE CDs

- 2 Year – \$669.00
- Canada/Mexico – \$555.00
- International – \$710.00

### 9-Pack

Pick 9 of our magazines and save up to **\$400<sup>00</sup>**  
 Pay only \$495 for a 1 year subscription plus 3 FREE CDs

- 2 Year – \$839.00
- Canada/Mexico – \$695.00
- International – \$890.00

**CALL TODAY! 888-303-5282**

**Pick a 3-Pack, a 6-Pack or a 9-Pack**

<input type="checkbox"/> 3-Pack	<input type="checkbox"/> 1YR	<input type="checkbox"/> 2YR	<input type="checkbox"/> U.S.	<input type="checkbox"/> Can/Mex	<input type="checkbox"/> Intl.
<input type="checkbox"/> 6-Pack	<input type="checkbox"/> 1YR	<input type="checkbox"/> 2YR	<input type="checkbox"/> U.S.	<input type="checkbox"/> Can/Mex	<input type="checkbox"/> Intl.
<input type="checkbox"/> 9-Pack	<input type="checkbox"/> 1YR	<input type="checkbox"/> 2YR	<input type="checkbox"/> U.S.	<input type="checkbox"/> Can/Mex	<input type="checkbox"/> Intl.

**TO ORDER**

- Choose the Multi-Pack you want to order by checking next to it below.
- Check the number of years you want to order.
- Indicate your location by checking either U.S., Canada/Mexico or International.
- Then choose which magazines you want to include with your Multi-Pack order.

<b>MX Developer's Journal</b>			
U.S. - Two Years (24) Cover: \$143	You Pay: \$49.99 /	Save: \$167 + FREE \$198 CD	
U.S. - One Year (12) Cover: \$72	You Pay: \$29.99 /	Save: \$60	
Can/Mex - Two Years (24) \$168	You Pay: \$79.99 /	Save: \$137 + FREE \$198 CD	
Can/Mex - One Year (12) \$84	You Pay: \$49.99 /	Save: \$40	
Intl - Two Years (24) \$216	You Pay: \$89.99 /	Save: \$127 + FREE \$198 CD	
Intl - One Year (12) \$108	You Pay: \$59.99 /	Save: \$30	
Digital Edition - One Year (12)	You Pay: \$19.99		
<b>Linux World Magazine</b>			
U.S. - Two Years (24) Cover: \$143	You Pay: \$79.99 /	Save: \$63 + FREE \$198 CD	
U.S. - One Year (12) Cover: \$72	You Pay: \$39.99 /	Save: \$32	
Can/Mex - Two Years (24) \$168	You Pay: \$119.99 /	Save: \$48 + FREE \$198 CD	
Can/Mex - One Year (12) \$84	You Pay: \$79.99 /	Save: \$4	
Intl - Two Years (24) \$216	You Pay: \$176 /	Save: \$40 + FREE \$198 CD	
Intl - One Year (12) \$108	You Pay: \$99.99 /	Save: \$8	
<b>Java Developer's Journal</b>			
U.S. - Two Years (24) Cover: \$144	You Pay: \$89 /	Save: \$55 + FREE \$198 CD	
U.S. - One Year (12) Cover: \$72	You Pay: \$49.99 /	Save: \$22	
Can/Mex - Two Years (24) \$168	You Pay: \$119.99 /	Save: \$48 + FREE \$198 CD	
Can/Mex - One Year (12) \$84	You Pay: \$79.99 /	Save: \$4	
Intl - Two Years (24) \$216	You Pay: \$176 /	Save: \$40 + FREE \$198 CD	
Intl - One Year (12) \$108	You Pay: \$99.99 /	Save: \$8	
<b>Web Services Journal</b>			
U.S. - Two Years (24) Cover: \$168	You Pay: \$99.99 /	Save: \$68 + FREE \$198 CD	
U.S. - One Year (12) Cover: \$84	You Pay: \$69.99 /	Save: \$14	
Can/Mex - Two Years (24) \$192	You Pay: \$129 /	Save: \$63 + FREE \$198 CD	
Can/Mex - One Year (12) \$96	You Pay: \$89.99 /	Save: \$6	
Intl - Two Years (24) \$216	You Pay: \$170 /	Save: \$46 + FREE \$198 CD	
Intl - One Year (12) \$108	You Pay: \$99.99 /	Save: \$8	
<b>.NET Developer's Journal</b>			
U.S. - Two Years (24) Cover: \$168	You Pay: \$99.99 /	Save: \$68 + FREE \$198 CD	
U.S. - One Year (12) Cover: \$84	You Pay: \$69.99 /	Save: \$14	
Can/Mex - Two Years (24) \$192	You Pay: \$129 /	Save: \$63 + FREE \$198 CD	
Can/Mex - One Year (12) \$96	You Pay: \$89.99 /	Save: \$6	
Intl - Two Years (24) \$216	You Pay: \$170 /	Save: \$46 + FREE \$198 CD	
Intl - One Year (12) \$108	You Pay: \$99.99 /	Save: \$8	
<b>XML-Journal</b>			
U.S. - Two Years (24) Cover: \$168	You Pay: \$99.99 /	Save: \$68 + FREE \$198 CD	
U.S. - One Year (12) Cover: \$84	You Pay: \$69.99 /	Save: \$14	
Can/Mex - Two Years (24) \$192	You Pay: \$129 /	Save: \$63 + FREE \$198 CD	
Can/Mex - One Year (12) \$96	You Pay: \$89.99 /	Save: \$6	
Intl - Two Years (24) \$216	You Pay: \$170 /	Save: \$46 + FREE \$198 CD	
Intl - One Year (12) \$108	You Pay: \$99.99 /	Save: \$8	

<b>WebLogic Developer's Journal</b>			
U.S. - Two Years (24) Cover: \$360	You Pay: \$169.99 /	Save: \$190 + FREE \$198 CD	
U.S. - One Year (12) Cover: \$180	You Pay: \$149 /	Save: \$31	
Can/Mex - Two Years (24) \$360	You Pay: \$179.99 /	Save: \$180 + FREE \$198 CD	
Can/Mex - One Year (12) \$180	You Pay: \$169 /	Save: \$11	
Intl - Two Years (24) \$360	You Pay: \$189.99 /	Save: \$170 + FREE \$198 CD	
Intl - One Year (12) \$180	You Pay: \$179 /	Save: \$1	
<b>ColdFusion Developer's Journal</b>			
U.S. - Two Years (24) Cover: \$216	You Pay: \$129 /	Save: \$87 + FREE \$198 CD	
U.S. - One Year (12) Cover: \$108	You Pay: \$89.99 /	Save: \$18	
Can/Mex - Two Years (24) \$240	You Pay: \$159.99 /	Save: \$80 + FREE \$198 CD	
Can/Mex - One Year (12) \$120	You Pay: \$99.99 /	Save: \$20	
Intl - Two Years (24) \$264	You Pay: \$189 /	Save: \$75 + FREE \$198 CD	
Intl - One Year (12) \$132	You Pay: \$129.99 /	Save: \$2	
<b>Wireless Business &amp; Technology</b>			
U.S. - Two Years (24) Cover: \$144	You Pay: \$89 /	Save: \$55 + FREE \$198 CD	
U.S. - One Year (12) Cover: \$72	You Pay: \$49.99 /	Save: \$22	
Can/Mex - Two Years (24) \$192	You Pay: \$139 /	Save: \$53 + FREE \$198 CD	
Can/Mex - One Year (12) \$96	You Pay: \$79.99 /	Save: \$16	
Intl - Two Years (24) \$216	You Pay: \$170 /	Save: \$46 + FREE \$198 CD	
Intl - One Year (12) \$108	You Pay: \$99.99 /	Save: \$8	
<b>WebSphere Developer's Journal</b>			
U.S. - Two Years (24) Cover: \$360	You Pay: \$169.99 /	Save: \$190 + FREE \$198 CD	
U.S. - One Year (12) Cover: \$180	You Pay: \$149 /	Save: \$31	
Can/Mex - Two Years (24) \$360	You Pay: \$179.99 /	Save: \$180 + FREE \$198 CD	
Can/Mex - One Year (12) \$180	You Pay: \$169 /	Save: \$11	
Intl - Two Years (24) \$360	You Pay: \$189.99 /	Save: \$170 + FREE \$198 CD	
Intl - One Year (12) \$180	You Pay: \$179 /	Save: \$1	
<b>PowerBuilder Developer's Journal</b>			
U.S. - Two Years (24) Cover: \$360	You Pay: \$169.99 /	Save: \$190 + FREE \$198 CD	
U.S. - One Year (12) Cover: \$180	You Pay: \$149 /	Save: \$31	
Can/Mex - Two Years (24) \$360	You Pay: \$179.99 /	Save: \$180 + FREE \$198 CD	
Can/Mex - One Year (12) \$180	You Pay: \$169 /	Save: \$11	
Intl - Two Years (24) \$360	You Pay: \$189.99 /	Save: \$170 + FREE \$198 CD	
Intl - One Year (12) \$180	You Pay: \$179 /	Save: \$1	

\*WHILE SUPPLIES LAST. OFFER SUBJECT TO CHANGE WITHOUT NOTICE

Subscribe Online Today [www.sys-con.com/2001/sub.cfm](http://www.sys-con.com/2001/sub.cfm)



# lingo

mixing  
flash  
and  
director

# stickman

## Director



is a great application for new media promotion and promotions together can create an audience for with beginning stages of the Stickman process a prerelease for a "stickman" This stickman interface also acts as a countdown to the

by niqui merret



# n can

marketing. Putting these  
the marketing tool  
t

The stickman has a series of actions that it can perform. These have been created as Flash animations that are called by the user. They can be called as individual actions or can be set up as a sequence. The main project is split into a Flash movie that has a stickman with associated animations. This has been nested in a Director movie with the con-

trols. The user can then save out sequences of actions so that they will be available the next time the project is run.

### Director Is the Shell

Director is the container for multiple types of media. It can incorporate formats ranging from bitmaps, sound, digital video, and 3D, to Flash. It allows you

to integrate these formats together to output to a variety of mediums. Although Director has the bulk to output to CD-ROMs, kiosks, and other heavy media, it also has the ability to compress down to a more than acceptable Web size with the shockwave player.

The different media elements that are loaded into Director are merely actors for



Director to control. For all the power that Flash is capable of, Director can take it to the next level, adding further control and functionality.

Director is able to not only control the different types of media that it has stored within, but allows the user far greater control of operating-system functions. A text file could be written to the hard drive, then retrieved by any other presentation. This method of saving and retrieving a text file was useful to stickman because it allowed actions and sequences to be saved to the user's machine.

Through the power of extras, further control can be applied across applications and within Director. Extras enable Flash movies to run in Director. If these extras are not included, Flash will not be displayed.

### Planning

As a general rule, all projects should be well planned with full specifications laid out before work commences. One of the areas that will need to be considered is how to split up the project between

Flash and Director. Some areas of a project would be best suited for Flash and others for Director. This may depend on a number of factors, but will generally be based on what you are comfortable with.

It is preferable, however, to take all the elements for a project and look at the applications with an unbiased opinion before making any decisions. It would be easy with the stickman to create the whole project in Flash, but that would not necessarily accommodate the full project specifications. The ability to write a text file to the hard drive would be lost.

The other option would have been to bring the stickman in as a series of Flash movies and just link from one to another. This approach would have saved the need to communicate with Flash altogether. It would, however, have meant adding in a whole series of cast members for the project, a new one for each action.

Due to differences in the scripting language, there may be cases where ActionScript would be better to use than Lingo. Performing date calculations, for

There are three types of variables in Director: local, global, and property. We will look at both local and global variables in this article. Stating a name and assigning a value to the variable will create a local variable. The variable is available until the end of the handler. This is normally when the keyword *end* is encountered. The value is not stored unnecessarily in the computer's memory and cannot be used anywhere other than in the handler. A global variable must be declared as a global before it is used. It is not dependent on any levels or attached to any objects. This can be used at any point in the project. The value is stored until the project is quit or it is cleared as a variable. For example:

Local variable in Director:  
`StickmanAction = "walk"`

Global variable in Director:  
`Global gStickmanAction  
gStickmanAction = "walk"`

Flash variables, however, are timeline variables by default. This means that they

*"you will find that as you are sending variables between*



example, would be easy and quick in Flash, and the values can then be passed back to Director with ease. Each action for the stickman could carry a date and display information about the last date it was called, how many times during the day it was called, as well as any other useful information.

### Variables

You will find that as you are sending variables between Flash and Director there are some differences in the way in which each program handles them. Director and Flash have different conventions for naming and working with variables. Director will treat a variable as a local variable unless otherwise stated. Flash, on the other hand, will treat variables as timeline variables by default.

are available at any point on the timeline after they have been created. However, they are grouped to objects or levels, so in order to be used, they must be called with a reference to their associated level or object. If you wish to use a local variable in Flash, it must be declared as a variable when it is created. It will then be cleared out of memory as soon as the block of code has executed. Flash also has global variables, which are treated in a similar way to Director globals in that they are declared as global every time they are used. For example:

Local variable in Flash:  
`var StickmanAction;  
StickmanAction = "walk";`

Timeline variable in Flash:

```
StickmanAction = "walk";
```

Global variable in Flash:

```
var _global.StickmanAction = "walk;"
```

Notice that both Director and Flash use the keyword "global" when referring to the global variables. This must be continued throughout, otherwise it will be considered to be local or timeline depending on the situation used.

## From Within Flash

The original way to work with Flash and Director was to call Director commands from within the Flash movie. A Flash button would contain an on(press) event that contained a call to Lingo. The following will tell a Director movie to move the playback head to another frame on the timeline.

```
on(press){
    getURL("lingo: go to frame
25");
}
```

The main problem with this method

A custom handler can be created to act as a holder for commands that could be called from a number of different events. The handler is called by its name from any location in the Director presentation and the scripts associated with it will run. The following is the above link converted to a custom handler and then called from a mouseDown event.

```
On gotoStickman
    Go to "stickman"
```

End

```
On mouseDown
    GotoStickman
```

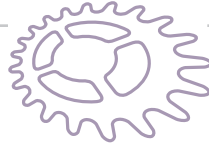
End

This could be fairly limiting as it stands, so to add to the flexibility, one can pass through different parameters. The following is the same handler as above, only the marker name can be sent through when the handler is called.

```
On gotoStickman target
    Go to target
```



# & director



there are some differences in the way in which each program handles them'

is that only expressions can be passed to Director. One of the methods to get around this is to call custom handlers in Director.

## Director Handlers

Handlers contain Lingo commands that run when the associated event calls them. A handler could be associated with a mouseDown event, and when the user clicks down, the relevant script will execute. The following example will send the playback head to a marker called "stickman" when the user clicks on the sprite that the event is associated with.

```
on mouseDown
    go to "stickman"
end
```

```
End
On mouseDown
    GotoStickman "stickman"
End
```

## Flash and Custom Handlers

The same method is used from Flash to call up any Lingo commands. The getURL option basically tells Flash that the commands in the brackets must trigger the object holding the Flash movie. This object is usually a browser window, but in this case the Flash is nested into a Director movie. The Lingo part of the command tells Director that what follows the colon will be code that it will understand as Lingo. The following example calls the Director handler that will move the playback head to the stickman marker in the movie.

```
on(press){
    getURL("lingo:
GotoStickman");
}
```

The custom handler in Director:

```
On gotoStickman
    Go to "stickman"
End
```

The only problem with this is that you cannot send a string parameter through to the Director movie, so the above case would not work if we tried to do the following:

```
on(press){
    getURL("lingo: GotoStickman
```



```
"stickman");  
}
```

```
The custom Director handler:  
On gotoStickman target  
    Go to target  
End
```

## Director Control

With the `getUrl` command from Flash limited only to expressions and calling custom handlers, a serious loss of integration between the two applications occurs. To counter this, there are some commands in Director that will control the Flash movie. We have now moved mostly into Director to tell the Flash movie how to behave. This can basically be done with two different methods: `getVariable()` and `setVariable()`. They are as they sound; one will get a variable from a Flash movie and return the value to Director; the other will set the value of a variable in Flash.

The following example will send the title of the action to the Flash movie to be displayed on the stage. The displaying variable is called `displayAction` and the value that will be displayed is "walk". Sprite 1 is the Flash movie that the variable must be sent to. This must refer to the channel that the sprite is in on the score.

```
on mouseDown  
sprite (1).setVariable(displayAction,  
"walk")  
end
```

The syntax is:

```
sprite  
(whichSprite).setVariable(variableName  
, newValue)
```

The process to get a variable from Flash to be used in Director is the same. The main difference is that it will return a value from Flash whereas the `setVariable` will not return anything.

The following will return the state that stickman is in. Is he acting or stationary?

```
On mouseDown  
sprite (1).getVariable(State, true)  
end
```

The syntax is:

```
sprite
```

```
(whichSprite).getVariable(variableName  
, stringOrReference)
```

The `stringOrReference` option is used to control the variable's value type. If this is set to true, then a string value will be passed; if this is false, then a literal will be passed. This is set to true by default.

## Accessing Flash Functions

There are times when merely getting and setting variables does not offer enough control. The next level is to call Flash functions from Director. These can be either custom functions or Flash functions. This process needs to be broken down into two steps. A reference to the Flash object will need to be created. The functions can then be attached directly to the object.

The following code will need to be attached to the Flash sprite on the stage as a sprite script. The reference will need to be stored in a global variable so as to be accessible at a later stage. The `beginSprite` handler ensures that the code will set up the reference to the Flash sprite as soon as the playback head reaches the Flash sprite in the score. The reference in this example has been created to the main timeline in Flash on level 0.

```
global gStickmanMain  
on beginsprite me  
    gStickmanMain  
    =getVariable(sprite(me.spritenum), "_level0", false)  
end
```

Now that you have created the reference to the main timeline you can call up Flash functions from anywhere in Director. The following example will now tell the Flash movie to play frame 2 of the main timeline.

```
global gStickmanMain  
on mouseUp me  
    gStickmanMain.gotoAndPlay(2)  
end
```

In the case of `stickman`, we need to create a reference to the `stickman` movieclip so that any of our `stickman` functions can be called. The following code is the same as above but creates a

reference directly to the movieclip `stickmanMC`.

```
global gStickmanMC  
on beginsprite me  
    gStickmanMC=getVariable(sprite  
(me.spritenum), "_level0.stickmanMC", false)  
end
```

The following will tell the `stickmanMC` to stop on frame 2:

```
global gStickmanMC  
on mouseUp me  
    gStickmanMC.gotoAndStop(2)  
end
```

To call a custom function in Flash that has been created in the `stickmanMC` is a simple process of calling the function and passing through the correct parameters. The following calls a Flash function called `stickmanAction`, which takes a parameter that is the type of action to be performed.

```
global gStickmanMain  
on mouseUp me  
    gStickmanMain.stickmanAction("walk")  
end
```

The basic Flash function that is called is the following:

```
function stickmanAction(action) {  
    gotoAndStop(action);  
    display_txt.text = action;  
}
```

## Some Tips for Using Flash in Director

One of the main things to consider when using Flash and Director is the frame rate. Director plays Flash back at a slower frame rate and therefore this needs to be increased. This can either be done in Flash or in the properties inspector in Director. The frame rate that I set in `stickman` was 30. Another property to consider is the DTS (direct to stage) option. This forces the Flash movie to be on top of all other content. This is useful if the Flash movie is large and there is not much else on the stage. In any other case it may cause the Flash movie to flicker during playback. ☺

After many years as an instructor and developer in South Africa, Niqui now works at London-based rich Internet solutions firm, Prismix. She feels that she is a Director addict in a Flash-mad world. [niquim@prismix.com](mailto:niquim@prismix.com)



# SYS-CON MEDIA

**304,187 of the World's Foremost IT Professionals**

**DIRECT MAIL, EMAIL OR MULTI-CHANNEL**

**Target CTOs, CIOs and CXO-level IT professionals and developers who subscribe to SYS-CON Media's industry leading publications**

**Java Developer's Journal...**  
The leading publication aimed specifically at corporate and independent java development professionals



**PowerBuilder Developer's Journal...**  
The only PowerBuilder resource for corporate and independent enterprise client/server and web developers



**ColdFusion Developer's Journal...**  
The only publication dedicated to ColdFusion web development

**LinuxWorld Magazine...**  
The premier monthly resource of Linux news for executives with key buying influences



**WebLogic Developer's Journal...**  
The official magazine for BEA WebLogic application server software developers, IT management & users

**Web Services Journal...**  
The only Web Services magazine for CIOs, tech, marketing & product managers, VARs/ISVs, enterprise/app architects & developers



**.NET Developer's Journal...**  
The must read iTechnology publication for Windows developers & CXO management professionals

**XML-Journal...**  
The world's #1 leading XML resource for CEOs, CTOs, technology solution architects, product managers, programmers and developers



**WebSphere Developer's Journal...**  
The premier publication for those who design, build, customize, deploy, or administer IBM's WebSphere suite of Web Services software



**Wireless Business & Technology...**  
The wireless magazine for key corporate & engineering managers, and other executives who purchase communications products/services

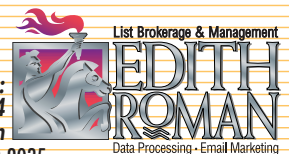
Recommended for a variety of offers including Java, Internet, enterprise computing, e-business applications, training, software, hardware, data back up and storage, business applications, subscriptions, financial services, high ticket gifts and much more.

**NOW AVAILABLE!**  
The **SYS-CON**  
Media Database  
304,187 postal  
addresses



For email information:  
contact Frank at 845-731-3832  
frank.cipolla@epostdirect.com  
epostdirect.com 800-409-4443 fax 845-620-9035

For postal information:  
contact Kevin at 845-731-2684  
kevin.collopy@edithroman.com  
edithroman.com 800-223-2194 fax 845-620-9035



# Platinum RE

freelance designer Dave Allbritton created this interface concept to be translated into a Flash application. "I wanted to take a different approach in making a futuristic interface that would inspire other designers and developers to try new things – something you would want to use just for the sake of how it looks. I like to create designs that have tons of flexibility by incorporating sliding compartments and spaces that are revealed to the user when the interface transforms."

You can view Dave Allbritton's online portfolio at [www.daveallbritton.com](http://www.daveallbritton.com)



# Liberate your ColdFusion Talents

**NQcontent v2**  
WEB CONTENT MANAGEMENT

The only CMS  
with the developer in mind

NQcontent is the ColdFusion based Content Management platform that eliminates repetitive development work and extends your talents to give you room to deliver robust, dynamic, websites that work all the time!



- Best Content Management Tool  
2002 WINNER
- Most Innovative Application  
2002 WINNER
- Best e-Business Software  
2002 WINNER
- Best Web Application  
2002 WINNER



Visit [www.nqcontent.com](http://www.nqcontent.com) for more information



how many people does it take to change the web?

macromedia®  
COLDFUSION®  
MX

See for yourself. Upgrading to ColdFusion MX gives you amazing new features and a powerful Java™ architecture. Incorporate XML and web services with ease. Build flexible and maintainable applications with ColdFusion Components. Integrate with J2EE™ and .NET to deliver rich user interfaces with native connectivity to Macromedia Flash.™ Easily migrate existing applications and save time with the powerful new Macromedia MX development tools. Try ColdFusion MX today and see what you can do now.

Download the free 30-day trial at  
[www.macromedia.com/go/cfmxad](http://www.macromedia.com/go/cfmxad)

