

MX

developer's journal

volume 3 issue 4 2005 www.mxdj.com

THE LEADING MAGAZINE
FOR MACROMEDIA MX
DEVELOPERS & DESIGNERS

Adobe



macromedia®



DREAMWEAVER

Best Practices with CSS



FLASH

ProgressBar Component



FIREWORKS

Masking with Bitmaps



COLDFUSION

10 Issues Identified & Discussed



DIRECTOR

Collision Detection

“ We share a common vision for the future of content and Internet applications across multiple operating systems, devices and media. ”

- Kevin Lynch
Chief Software Architect
Macromedia



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

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

To tour the site—and see the future of video—visit:
macromedia.com/go/video5

ARE YOU READY FOR

MX7

WE ARE

CFDynamics

- Reseller program
- State of the art data center
- Comprehensive on line support
- 24/7 network monitoring

CFDynamics, an innovative leader in ColdFusion hosting, caters to professional developers. We're here to help our clients use this brand new technology. While other developers waited patiently for the new version of ColdFusion, our clientele has already experienced powerful new features such as a full-feature reporting engine and flash forms through free hosting of Blackstone. CFDynamics is now ready for you to experience Coldfusion MX7.

Contact CFDynamics at 1.866.233.9626 and [experience Coldfusion MX7.](#)

*Blackstone is the code name for MX 7. Free BlackStone hosting is no longer available because ColdFusion MX7 is now released.





8 **New Beginnings**
Welcome to the future
by charles e. brown



26 **Skinning the V2
ProgressBar Component**
Getting started with skinning
by colin moock



38 **10 Issues Identified &
Discussed**
*Web services-enabling a
ColdFusion application*
by steven rubenstein

7 **Bigness Sure
Ain't Badness**
by jeremy geelan

16 **Macromedia Hits
the Headlines**
*Blogging during
acquisition*
by mxdj news desk

20 **The MX
Blogosphere**
*Taking a stroll around
the MX World*
by mxdj news desk

50 **vanguard**
Step into My Office
by agencynet

22 **Best Practices with CSS in
Dreamweaver MX**
*Cascading Style Sheets (CSS) in
changing the website design process*
by rachel andrew

46 **How I Rolled My Own Collision
Detection**
A helpful addition to your toolbox
by andrew m. phelps

34 **Uncovering Fireworks Masks:
Masking with Bitmaps**
We look at vector objects
by kim cavanaugh

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.

Want to learn more?

Visit www.seapine.com/webdev to discover how Surround SCM can save you time and headaches. Be sure to download our white paper, **Change Management and Dreamweaver**, and learn why change management is the new must-have tool for web development.

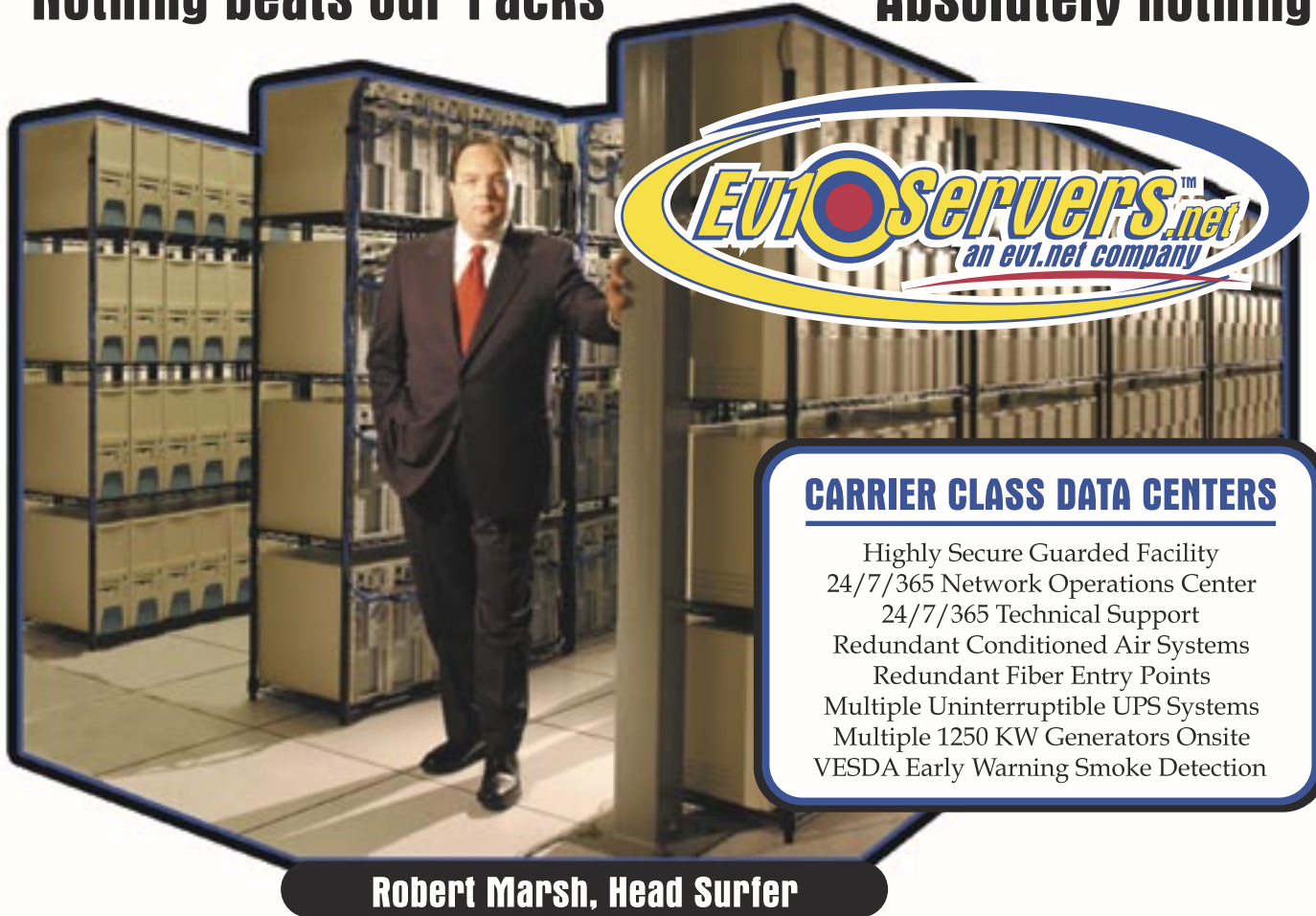


www.seapine.com/webdev
1-888-683-6456



Nothing beats our racks

Absolutely nothing



CARRIER CLASS DATA CENTERS

- Highly Secure Guarded Facility
- 24/7/365 Network Operations Center
- 24/7/365 Technical Support
- Redundant Conditioned Air Systems
- Redundant Fiber Entry Points
- Multiple Uninterruptible UPS Systems
- Multiple 1250 KW Generators Onsite
- VESDA Early Warning Smoke Detection

Robert Marsh, Head Surfer

START YOUR OWN WEB HOSTING BUSINESS TODAY!

from
\$299
Instant Activation!

Dedicated Server

- Dual Xeon 2.4 GHz**
- 2 GB RAM • 2 x 73 GB SCSI HD**
- Remote Console • Remote Reboot**
- 2000 GB Monthly Transfer Included**

Over 20,000 Servers!

1-800-504-SURF | ev1servers.net

**PLESK7
RELOADED**
Preferred Control Panel

IP Compliant. Price subject to change. Quantities Limited.
*Per month. Set-Up fees apply. See web site for complete details.

Group Publisher Jeremy Geelan
Art Director Louis F. Cuffari

EDITORIAL BOARD

Editor-in-Chief

Charles E. Brown charles@sys-con.com

Dreamweaver Editor

Dave McFarland

Flash Editor

Brian Eubanks

Fireworks Editor

Joyce J. Evans

FreeHand Editor

Ron Rockwell

Louis F. Cuffari

Director Editor

Andrew Phelps

Captivate Editor

Tom Green

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 Moock **Canada**, Jesse Nieminen **USA**,

Gary Rosenzweig **USA**, John Tidwell **USA**

EDITORIAL

Editor

Nancy Valentine, 201 802-3044

nancy@sys-con.com

Associate Editor

Seta Papazian, 201 802-3052

seta@sys-con.com

Technical Editors

Jesse Warden • Sarge Sargent

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

Subscriptions

E-mail: subscribe@sys-con.com

U.S. Toll Free: 888 303-5282

International: 201 802-3012

Fax: 201 782-9600

Cover Price U.S. \$5.99

U.S. \$29.99 (12 issues/1 year)

Canada/Mexico: \$49.99/year

International: \$59.99/year

Credit Card, U.S. Banks or Money Orders

Back Issues: \$12/each

Editorial and Advertising Offices

Postmaster: Send all address changes to:

SYS-CON Media

135 Chestnut Ridge Rd.

Montvale, NJ 07645

Worldwide Newsstand Distribution

Curtis Circulation Company, New Milford, NJ

List Rental Information

Kevin Collopy: 845 731-2684,

kevin.collopy@edithroman.com,

Frank Cipolla: 845 731-3832, frank.cipolla@epostdirect.com

Promotional Reprints

Dorothy Gil, 201 802-3024

dorothy@sys-con.com

Copyright © 2005

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

Bigness Sure Ain't Badness...

by jeremy geelan

In the world of architecture, as many of the designers among *MX Developer's Journal's* readership are well aware, there's a wonderfully erudite theory, with a equally wonderfully dumb name: Bigness.

The Dutch architect Rem Koolhaas wrote, just six years before the turn of the millennium, in 1994, "Bigness is a theoretical domain at this fin de siècle. In a landscape of disarray, disassembly, dissociation, disclamation, the attraction of Bigness is its potential to reconstruct the Whole, resurrect the Real, reinvent the collective, reclaim maximum possibility."

Ignoring the impossibly grand Koolhaasian rhetoric of Bigness as "the one architecture that could survive, even exploit, the now-global condition of the tabula rasa," (not for nothing was Koolhaas dubbed a "rhetorical architect" as well as a creator of real physical buildings!) it seems pretty clear that the Theory of Bigness can also throw light on why two companies like Macromedia and Adobe might opt so enthusiastically to coalesce.

In the world of commercial software, bigness allows what Koolhaas called "Manhattanism without Manhattan" – put another way, like a skyscraper-block returned in a single building, a mega-company can facilitate a great variety of programs, and won't be constrained by any grid. Two plus two can, and will, make five.

In this issue, as editor-in-chief Charles E. Brown makes clear in his "New Beginnings" feature, *MXDJ* celebrates the potential, for designers and developers, in the new amalgamation, as and when it is all ratified and approved in due course.

In terms of bigness, a company with \$1.67 billion per annum and one with \$422 million is no out-and-out brontosaurus. IBM's annual revenues, after all, are nearly \$97 billion and Microsoft's are almost \$39 billion. But the particular Adobe-Macromedia combo is definitely a very nifty Tyrannosaurus Rex, with long and slender legs for an animal of its size. Combined with \$2 billion joint revenues, it's a formidable new software dinosaur, and achieving such a size will defi-

nately allows many things to happen that would not have been possible before.

As Charles points out in his article, one of the most energetic areas of development in the Adobe-Macromedia future is certain to be software aimed at the mobile market. So we will doubtless be appointing a Mobile Editor soon to the magazine. And we would love to hear, too, from the kinds of people who in the past enjoyed Adobe Magazine, which – entirely appropriately – moved to the Web in 1999 and flourishes in Australia/New Zealand, South East Asia, and India. The address for article submissions: <http://mxdj.sys-con.com/main/proposal.htm> – or send an e-mail to mxdj@sys-con.com.

Of all the various comments about the developments in April, I like these two the best. First, I liked the way that Adobe CEO Stephen Elop said: "Both Macromedia and Adobe are passionate about creating and enabling great experiences across a wide range of devices and operating systems." Then I liked what Macromedia's Kevin Lynch said: "We will continue to be busier than ever here as we build our next generation tools, servers and client technology, and over time, the combination will enable a complete, cross media, rich client technology platform."

One hears many (frankly) empty utterances before, during, and after the kind of business transaction that is now in train. But both these consummate i-technology professionals seem to have latched on to something almost tangible, indisputable, and that is that the Adobe-Macromedia-in-combination story is more interesting than the story of either Adobe or Macromedia would be solo. Both are forward-looking and both demonstrably recognize that passion pays, that great software will always be in demand – as will great developers.

If you want to continue keeping track of this powerful force for innovation around cutting-edge platforms for delivering content and applications, then stay 100% tuned. I know that we all involved in the magazine will! These are truly interesting times. ☺

Jeremy Geelan is group publisher of SYS-CON Media, and is responsible for the 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



New Beginnings

Business guru Tom Peters once wrote “we need to embrace change; to welcome it; to transplant ourselves periodically.” Through change, we grow and avoid complacency or staleness. With change, we start on the proverbial Spring cleaning and make way for the new and exciting.

This month, most certainly, Macromedia has forced us to change...and for that we should be grateful. With the merger of Macromedia and Adobe we are standing on the edge of the next generation of web design and development...welcome to the future.

While I know I am going to get a ton of nasty e-mail about this, as I already have had after several blogs I published at mxdj.sys-con.com, we have to face one very real reality: by the end of this year, or somewhere thereabouts, Macromedia as we know it will be gone. The new name is not going to be Macromedia-Adobe or some variation of that. The new name is going to be Adobe. Within the next year, all the familiar interfaces (and how we use our tools) will be gone. The products we are all comfortable with will be incorporated into the Adobe family of products.

by **charles e. brown**

Adobe



macromedia®





This Is Not Armageddon

If we listen to the many postings I have seen on a variety of chat boards, Armageddon has arrived. Absolutely nothing is further from the truth. Why is this necessarily bad? Both Macromedia and Adobe are well-established and very reputable names in our industry. Photoshop has been a standard for many years and the PDF format (which is owned by Adobe and created with their package Acrobat) is growing in many unique ways. On the Macromedia side, Flash and Dreamweaver are indisputable industry standards. Put them together and you have potential like you have never seen before. However, it grows from there.

This week, Adobe just released its next generation of products called CS2 (Creative Suite 2). This release introduces a radical new Adobe Creative Suite component: Adobe Bridge. It acts as a hub for productivity, imagery and creativity, providing multi-view file browsing and smooth cross-product integration across Adobe Creative Suite 2 software. It also provides access to Adobe Stock Photos, a new stock photography service that offers one-stop shopping for high-quality, royalty-free images for layout and design. With advanced mobile authoring, for standards such as SVG-t, Adobe Creative

Suite 2 accelerates design and publishing workflows – for print, the Web and the latest generation of handheld devices.

Bridge to the Future?

To quote Shantanu Narayan, president and chief operating officer at Adobe: “Through their work in design, advertising and publishing our customers change the way the world looks everyday. Our goal is to develop a technology platform that the creative community can rely on and build upon. With Creative Suite 2 we’ve delivered the essential tools designers need to efficiently realize and create their best work: online, in print or for the latest handheld device.”

Adobe Bridge is a visual file browser that lets designers easily browse, organize, and process design assets within Adobe Creative Suite 2 software components. Designers can preview multi-page Adobe PDF files, drag and drop from floating Compact Mode windows, process images with built-in Camera Raw 3.0, edit and search for Adobe XMP metadata, and track assets managed in Version Cue CS2. Simplifying the purchase and management of stock imagery, Adobe Stock Photos is a single resource that brings together, at one convenient location, more than 230,000 images from some of the world’s leading stock pho-

tography providers, including: Photodisc by Getty Images; Comstock Images by Jupitermedia; Digital Vision; imageshop royalty free by zefaimages; and amana.

Version Cue CS2, a powerful file-version manager and collaboration tool, helps users visually preview, track and access historical and alternate versions of files in Adobe Creative Suite 2 projects. Workgroup users can easily share their work, and can now initiate a Web-based collaborative review of PDF project files. Using a single dialog box in Adobe Bridge, designers can synchronize color settings for Photoshop CS2, Illustrator CS2, InDesign CS2 and GoLive CS2, avoiding painful after-the-fact color adjustments. New global export presets allow for consistent creation of Adobe PDF documents across components of Adobe Creative Suite to save time and avoid costly errors when projects go to print. Presets are stored in a single location and can be customized for specific workflows.

A Top Priority: The Mobile Market

Have absolutely no doubt: Adobe is looking toward the mobile market. Focusing on that market will be Adobe’s GoLive software.

Once again, quoting Shantanu Narayan: “GoLive is now at the leading edge of cross-media content authoring, eliminating the gap between what can be delivered on the Web and what can be authored for mobile devices. The future of the Web is being driven by millions of consumers who are demanding graphically-rich Internet services on their handsets and GoLive will be at the center of this content revolution.”

At the heart of GoLive will be a defector Object Oriented way of building websites. GoLive CS2 unlocks the full potential of CSS with new and enhanced tools that enable designers to work visually while automatically generating lean, efficient CSS code. Designers can now drag and drop pre-built, extensible CSS block objects onto Web pages to quickly construct classic liquid CSS designs that previously took considerable time to produce. The new Layer and Grid Text Box tools enable designers to interactively place and size CSS-based layers and text boxes, while the enhanced Layout Grid, Layout Text Box, and Layer objects all pro-

“We can now take the incredible power of Macromedia’s outstanding collection of tools and combine them with the innovative tools of Adobe.”

Brad Lawryk: "Before MX Kollection, I 'thought' I was doing a **pretty good job**, and now ...

Everyone tells me I'm doing a **brilliant** job!"

If you own Dreamweaver,

MX Kollection is a must!

List Database Information

- Create dynamic lists
- Order records within a list
- Insert, Edit, Delete field in list
- Automatic navigation bars
- Automatic list filter
- Sort column by clicking on the header
- Automatic row counter
- Delete multiple records directly from list
- Create master/detail lists

Manage Recordsets Visually

- Create, Edit recordsets visually
- Visually add tables to query
- Contextual menu to refresh fields
- Large query management Zoom In/Out
- JOINS between tables
- Define and apply complex conditions
- Automatically generate SQL conditions
- Add GROUP BY for aggregated fields
- Extract information from multiple tables
- Optimized SQL generation
- Smart SQL queries for list filters
- Automated database introspection

Rapid HTML Form Creation

- Generate Insert Record Form
- Generate Update Record Form
- Insert, Edit or Delete form field
- Redirect to page after form submit
- Table and CSS form generation

Form Validation and Error Handling

- Validate form fields
- Rich formats library
- Allow custom validation formats
- Error handling
- Preserve submitted values on error

Upload Files and Images

- File Upload
- Image Upload
- Resize and sharpen image on upload
- Show Dynamic Image
- Download Uploaded File
- Delete file from specified folder
- Show If File Exists on Server

Send E-mails

- Send e-mail after form submit

Online HTML Editor

- Edit HTML content visually
- Use your own CSS styles
- Upload and manage server images
- Format tables and align images

Enhanced HTML Form Controls

- Date Picker
- Dependent Drop-downs
- Editable Drop-down (Combo-box)
- Masked Textfield
- Numeric Textfield
- Dynamic Search
- n...1 Dependent field
- Smart Date

Security & User Authentication

- Login form generator
- Login with User Levels
- Remember me feature
- Encrypted password
- Restrict access to page
- Generate password with fixed length
- Update user password
- Send password by e-mail

DREAMWEAVER PRODUCTIVITY TOOL FOR DYNAMIC WEBSITE DEVELOPMENT

Download a trial version here - <http://interaktonline.com/go/MXKollection/>

Find out how easy is to use our tools - **save 30%** by using the coupon in the right.

Caution: InterAKT extensions are known to cause a high increase in productivity. For those of you who can't deal with the free time please continue to hand-code or refer to our competitors for help.



\$100 coupon - 199920002001



duce CSS by default.

Adobe GoLive CS2 delivers comprehensive mobile authoring tools based on open standards to support mobile layout (CSS, XHTML), multimedia and interactivity (SVG, SVG-t), and video (3GPP, MPEG-4) for mobile devices. The new SVG-t Editor supports both SVG-t 1.1 and 1.2 and enables designers to use SVG-t files created in Adobe Illustrator CS2 and other applications, quickly select SVG elements and their source code and then add a wide range of interactivity.

GoLive also delivers a robust MMS templating and editing environment and flexible device emulation for previewing content for Sony Ericsson and Nokia devices and is easily extensible to support other manufacturer handsets. The Live Rendering engine includes a new Small Screen Rendering option for previewing how content developed for the Web will appear on mobile devices.

Repurposing print content for the Web is easier than ever with GoLive CS2. Designers can now package print assets in InDesign CS2; open, preview, and optimize that content in GoLive CS2; and then export automatically as XHTML Web

pages. This jump-starts Web designs, which can then be finessed and site-managed in GoLive CS2. In addition, Smart Object support is extended to include easy creation of smart favorite icons, used in URL addresses and Favorites menus to brand websites.

Let's now turn to the core of the Adobe family: Photoshop.

"Stunning Images" With CS2 Software

Photoshop CS2 integrates a new set of intuitive tools, including an enhanced Spot Healing Brush, for handling common photographic problems such as blemishes, red-eye, noise, blurring and lens distortion. Smart Objects allow users to scale and transform images and vector illustrations without losing image quality – as well as create linked duplicates of embedded graphics – so that a single edit updates across multiple iterations.

Responding to requests from film, broadcast and video professionals, Photoshop CS2 now allows non-destructive editing and the creation and editing of 32-Bit High Dynamic Range (HDR) images, ideal for 3D rendering and advanced composi-

ing. FireWire Previews can also be viewed on a television monitor through a new direct export feature.

Photoshop CS2 users will test the limits of creativity with new tools like Vanishing Point and Image Warping. Vanishing Point cuts tedious graphic and photo retouching tasks by allowing users to clone, paint and transform image objects while retaining visual perspective. Reinventing workflows such as product packaging development, Image Warping makes it easy to fold, stretch, pull, twist and wrap an image into shape by selecting an on-demand preset or dragging custom control points.

"Photoshop CS2 pushes the envelope with powerful features and simplified workflows that provide photographers and creative professionals the freedom to deliver stunning images," said Bryan Lamkin, senior vice president of Digital Imaging and Digital Video Products at Adobe. "In light of the four million digital SLRs expected to ship this year, more photographers than ever will be making Photoshop CS2 their digital darkroom of choice."

With Photoshop CS2 software, find-

figure 1



CommonSpot™

Efficient Content Management

fast. easy. affordable.



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

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

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

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

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

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

Paper | Thin



figure 2



figure 3



ing and working with digital photos and images has never been easier. The popular File Browser has been upgraded to Adobe Bridge, which functions as a hub for productivity, imagery and creativity, providing multi-view file browsing and smooth cross-product integration across Adobe Creative Suite 2 software. Adobe Bridge also provides access to Adobe Stock Photos, a new stock photography service that offers users one-stop shopping – across five elite stock image providers – to deliver high-quality, royalty-free images for layout and design (see separate press release).

The new Camera Raw 3.0 workflow allows settings for multiple raw files to be simultaneously modified. In addition batch processing of raw files, to JPEG, TIFF, DNG or PSD formats, can now be done in the background without launching the main Photoshop executable. Integrated, non-destructive cropping and straightening controls allow raw files to be easily prepared for final output.

Simplifying Photoshop's rich interface, task-based menu presets make it easier to find the features needed for specific work. Users can create their own custom presets, highlighting favorite items and now have the ability to define event-based scripts and time-saving operations that execute automatically when triggered by actions, such as saving or printing a file. Multiple Layer Controls speeds editing, enabling the simultaneous selection and manipulation of numerous layers within a file.

Adobe Creative Suite 2 delivers a tightly integrated set of professional design tools, enabling superb image-editing, illustration, page layout, Web and mobile authoring, and Adobe PDF workflow support. Offering the full, new versions of the world's leading design tools – Adobe Photoshop CS2, Illustrator CS2, InDesign CS2, GoLive CS2, and Acrobat 7.0 Professional – Adobe Creative Suite 2 gives designers such powerful new features as enhanced support for digital images captured in raw format; a new paradigm in vector graphics creation; significantly more productive page layout tools; and the capability to design, develop, and deploy mobile sites that conform to open industry standards. And because the components of Adobe Creative Suite 2 are so tightly knit and cohesive, creative

“EPCOT founder Walt Disney always had a very firm eye on the future and embraced change and development. If we want to stay in this business, we must do the same”

ideas flow onto the page uncompromised and undiminished, just as designers envision them.

Key Role for Macromedia's Tools

While I will grant that much of Adobe's efforts have been focused on print, this is where Macromedia will come into play: we can now take the incredible power of Macromedia's outstanding collection of tools and combine them with the innovative tools of Adobe. Adobe has not been strong with server technology: Dreamweaver is needed to help harness much of this technology on the server. E-Learning is developing into a huge industry, and Macromedia is poised with Captivate, Authorware, and Breeze. And, of course, it goes without saying that Flash has virtually taken over this industry with most websites incorporating it in one form or another. Neither Adobe, nor for that matter anyone else, comes even close with a competitive product (although, Microsoft has a product it is planning on introducing this fall and which, many believe, may have been part of the motivation behind this merger).

Two of the features I found especially interesting in Photoshop CS2 were the Vanishing Point and Image Warp.

With Vanishing Point (see Figure 1) you can effortlessly edit and transform with tools that automatically adjust to the visual perspective of your images with the revolutionary, plane-based Vanishing Point. Define your perspective planes visually with the perspective grid tool, then paint, clone and drag objects around corners and into the distance. Cutting literally hours off precision design and photo retouching tasks, one use and you'll wonder how you ever lived without it.

Image Warp (see Figures 2 & 3) allows you to expand your creative options with live image warping directly on the canvas. Start with one of the comprehensive set of included presets, or

customize your warp by simply dragging control points on the warp grid. Image Warp makes short work of wrapping graphics around objects, and when coupled with the new Smart Objects features, warps stay fully re-editable.

Getting the idea? This is not the end, but only the beginning.

If I may, let me give you a bit of what some may call a “doom and gloom” prognostication: Just as we get use to these great new tools, they will probably change yet again. All at once, this is what makes our industry so exciting yet, at times, so frustrating.

Conclusion: A Firm Eye on the Future

As I am writing this article, I am overlooking the EPCOT theme park located in the Walt Disney World complex in Orlando, Florida. EPCOT stands for Experimental Prototype City of Tomorrow (although, some people feel it stands for Every Person Comes Out Tired). Disney always had a very firm eye on the future and embraced change and development. If we want to stay in this business, we must do the same.

• • •

Over the next few months, we will be bringing more in-depth analysis of the Adobe products. Let's face it, we will now need to work with them.

Let's embrace this New Beginning. 

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

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

ADVERTISING
Senior Vice President, Sales & Marketing
Carmen Gonzalez, 201 802-3021
carmen@sys-con.com
Vice President, Sales & Marketing
Miles Silverman, 201 802-3029
miles@sys-con.com
Advertising Sales Director
Robyn Forma, 201 802-3022
robyn@sys-con.com
Advertising Sales & Marketing Manager
Dennis Leavey, 201 802-3023
dennis@sys-con.com
Advertising Sales Manager
Megan Mussa, 201 802-3023
megan@sys-con.com
Associate Sales Managers
Dorothy Gil, 201 802-3024
dorothy@sys-con.com
Kim Hughes, 201 802-3025
kim@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
Tami Beatty, 201 802-3038
tami@sys-con.com
Assistant Art Directors
Andrea Boden, 201 802-3034
andrea@sys-con.com
Video Production
Frank Moricco, 201 802-3036
frank@sys-con.com

SYS-CON.COM
Consultant, Information Systems
Robert Diamond, 201 802-3051
robert@sys-con.com
Web Designers
Stephen Kilmurray, 201 802-3053
stephen@sys-con.com
Percy Yip, 201 802-3054
percy@sys-con.com

ACCOUNTING
Financial Analyst
Joan LaRose, 201 802-3081
joan@sys-con.com
Accounts Payable
Betty White, 201 802-3002
betty@sys-con.com
Accounts Receivable
Gail Naples, 201 802-3062
gailn@sys-con.com

EVENTS
President, SYS-CON Events
Grisha Davida, 201 802-3004
grisha@sys-con.com
National Sales Manager
Jim Hanchrow, 201 802-3066
jimh@sys-con.com

CUSTOMER RELATIONS
Circulation Service Coordinators
Edna Earle Russell, 201 802-3081
edna@sys-con.com
Linda Lipton, 201 802-3012
linda@sys-con.com
JDJ Store Manager
Brundila Staropoli, 201 802-3000
bruni@sys-con.com



Macromedia Hits the Headlines

*"Blogging During Acquisition" becomes hottest issue of the month
by mxdj news desk*

It is wholly unprecedented in the (brief) history of blogging: with a \$3.4 billion business deal in play, the Macromedia bloggers nonetheless remained determined to continue their commitment to avoiding a "legal blackout" interfering unduly with the Macromedia commitment to communicate more directly and more often with the developers on whose success as a company depends.

Macromedia's John Dowdell was in the forefront, "erring on the side of caution" by "going dark" on the morning of the announcement, but taking care as a courtesy that his blog didn't merely fall silent all day, which would have possibly been misinterpreted. "Sorry I didn't post today," he eventually wrote, "I've been waiting on getting approval for a post – rare, and in this case, apparently low in the priority stack. I hope that this single post here is okay for apologizing for not posting at all today."

Three days later he followed up in much more detail. "In five years some law school textbook will likely cite this 'blogging during acquisition' situation," he wrote on April 21. "Anyway, we're still feeling our way here," he continued, "and I'd prefer to go slowly, erring on the side of caution. People reading this can speculate about what an Adobe/Macromedia combination would be able to accomplish, but us staffers cannot, at least not without risking massive complications. I'd feel more comfortable if such speculations occurred elsewhere, not on my weblog."

Dowdell pointed out early on that since Weblogs "reach more people, and remain easily searchable, and are documents which can (with comments) actu-

ally have multiple authors," the potential complications they could introduce – if allowed complete free rein – are considerable.

Macromedia's product manager for MX, Mike Chambers, on the other hand, took the role of 'Blogger-in-Chief' and rose to the occasion admirably. In an April 18 blog entitled, succinctly, "Macromedia + Adobe," Chambers weighed in:

"Well, as you probably have probably heard by now, Macromedia and Adobe announced this morning that our companies are combining. If you are like me, then I am sure there are a ton of questions going through your head. We have put up a mini site (<http://www.adobe.com/aboutadobe/invrelations/adobeandmacromedia.html>) with more information, but I wanted to make a quick post to put some more context around what is happening."

"First, Macromedia and Adobe have announced that we are combining into a single company," Chambers continued. "However, and this is a very important point, this has not occurred yet, and will not occur until approved by stockholders and government regulators."

"This is an important point, because during this period, there will be some legal constraints on what I (and other Macromedia employees) can and can't speak about in public (see the end of the post for an example). So, please keep this in mind if you ask a question and are referred to the FAQ or web site."

"We are still working on 8ball, Maelstrom and everything else we have been working on"

Chambers' blog continued: "So, with that in mind, I can't actually go into too many details. Day to day work

at Macromedia will continue on as it has been. We are still working on 8ball, Maelstrom and everything else we have been working on. We are still planning to ship sometime in the second half of this year. We are still committed to Flash as a development platform (probably more than ever now). We are still committed to our server products, such as ColdFusion (7 has been incredibly successful) and Flex."

"How will this affect the Macromedia culture?" Chambers asked rhetorically. "We have worked hard over the past couple of years to get closer to you, communicate more directly and more often. We have tons of employees participating in the community, whether it be through mailing lists, forums, weblogs or contributing code. I think that it is this culture of openness and participation that has made us so successful over the past couple of years. This is not going to change. Indeed, I think that this culture of openness and participation is one of the reasons that Adobe was interested in bringing our companies closer together."

"We are going to work hard to make sure that everything moves forward smoothly, but there are sure to be some bumps along the way. However, I am confident that as long as we continue to be open, and communicate clearly we will all come out of this in a much stronger position. We will be able to compete more effectively, innovate more, and be more responsive than we would have been able to do on our own."

Chambers then, "because this post contains some forward looking statements about the combination of our companies, and because we are now in a regulatory period," included a slew of

legalese longer than the actual post!

But the "net net" was clear: Macromedia was living up to its promise to be as transparent as possible. Three cheers for Mike Chambers!

Quote Unquote: Early Reactions

*Mike Chambers, April 18, 2005
Macromedia MX Product Manager
and 'Blogger-in-Chief'*

"How do I feel about this personally? Well, of course, I was a little surprised at first. But, now that I have had some time to think about it for a while, I am actually quite excited.

I strongly believe in the potential of Flash as a cross-platform solution for deploying rich content and applications (I can't stress the cross-platform part of that enough). Together, the combined company will have the resources on our own to make the platform successful on a larger scale. So, personally, I am excited about this because, after the combination closes, Flash as a platform will be driven by a much, much larger company (combined we would be over \$2 billion a year in revenue), with a lot more resources at its disposal. I think it makes it much more likely that Flash will play a significant role as the next generation application / content platform (one that is ubiquitous and cross platform).

I think it makes things very interesting..."

*Jeremy Allaire, April 18, 2005
Former CTO of Macromedia, Now
Founder and President of Brightcove*

"Macromedia lost the enterprise publishing race to Adobe, and Adobe lost it with the Web publishing community. So the deal combines the best of both worlds. It gives Macromedia a huge sales channel, especially on the enterprise side. This will probably make the channels as strong as say Microsoft has."

*Kevin Lynch, April 18, 2005
Chief Software Architect,
Macromedia*

"This has been a move that's been a long time coming as the relationship between our companies has been growing far beyond our early clashes during the frenetic times of the internet bubble.

"I've gotten to know Bruce Chizen,

Shantanu Narayan and others at Adobe over the past few years (I've known some other folks at Adobe for even a longer time) and we all share a common vision for the future of content and internet applications across multiple operating systems, devices and media. There have been times when this led us to conflict as we worked in similar areas, but today we find ourselves having worked on different parts of this vision and are in a very complementary position with our technology and our customers.

Together we will offer an incredible set of products.

Many creative professionals and web developers already use our products together, and we will be able to provide an even more efficient authoring and development environment to create, manage and deliver information. There are also significant opportunities to help digital video creators not only produce great work but also deliver it seamlessly over the internet.

In the rapidly growing mobile devices area, we are able to provide a very strong set of products for developers, content providers, and operators to create and deliver rich mobile content.

For enterprise developers, we can provide a wider set of development tools and solutions that help connect people and business systems, providing an effective, highly productive user experience. For mainstream business users, we provide a more complete environment for dynamic, engaging collaboration over the internet that enables both on-line and off-line work.

We will continue to be busier than ever here as we build our next generation tools, servers and client technology, and over time, the combination will enable a complete, cross media, rich client technology platform.

I'm looking forward to seeing both familiar and new faces at Adobe, and to working together to build excellent software to help millions of people better communicate and interact, enabling great experiences."

*Mike Chambers, Macromedia
April 18, 2005
Macromedia + Adobe*

Well, as you probably have probably heard by now, Macromedia and Adobe

announced this morning that our companies are combining. If you are like me, then I am sure there are a ton of questions going through your head. We have put up a mini site with more information, but I wanted to make a quick post to put some more context around what is happening.

First, Macromedia and Adobe have announced that we are combining into a single company. However, and this is a very important point, this has not occurred yet, and will not occur until approved by stockholders and government regulators. This is an important point, because during this period, there will be some legal constraints on what I (and other Macromedia employees) can and can't speak about in public (see the end of the post for an example). So, please keep this in mind if you ask a question and are referred to the FAQ or web site.

So, with that in mind, I can't actually go into too many details. Day to day work at Macromedia will continue on as it has been. We are still working on 8ball, Maelstrom and everything else we have been working on. We are still planning to ship sometime in the second half of this year. We are still committed to Flash as a development platform (probably more than ever now). We are still committed to our server products, such as ColdFusion (7 has been incredibly successful) and Flex.

How will this affect the Macromedia culture? We have worked hard over the past couple of years to get closer to you, communicate more directly and more often. We have tons of employees participating in the community, whether it be through mailing lists, forums, weblogs or contributing code. I think that it is this culture of openness and participation that has made us so successful over the past couple of years. This is not going to change. Indeed, I think that this culture of openness and participation is one of the reasons that Adobe was interested in bringing our companies closer together.

We are going to work hard to make sure that everything moves forward smoothly, but there are sure to be some bumps along the way. However, I am confident that as long as we continue to be open, and communicate clearly we will



all come out of this in a much stronger position. We will be able to compete more effectively, innovate more, and be more responsive than we would have been able to do on our own.

Factpoint 1: The April 18 Statement by Adobe

Adobe to Acquire Macromedia

Combined Company to Deliver Industry-Defining Technology Platform for Rich, Interactive Content

SAN JOSE, Calif. - April 18, 2005 - Adobe Systems Incorporated (Nasdaq: ADBE) today announced a definitive agreement to acquire Macromedia (Nasdaq: MACR) in an all-stock transaction valued at approximately \$3.4 billion.

The combination of Adobe and Macromedia will provide customers a more powerful set of solutions for creating, managing and delivering compelling content and experiences across multiple operating systems, devices and media. Together, the two companies will meet a wider set of customer needs and have a significantly greater opportunity to grow into new markets, particularly in the mobile and enterprise segments.

"Customers are calling for integrated software solutions that enable them to create, manage and deliver a wide range of compelling content and applications - from documents and images to audio and video," said Bruce Chizen, chief executive officer of Adobe. "By combining our powerful development, authoring and collaboration software - along with the complementary functionality of PDF and Flash - Adobe has the opportunity to bring this vision to life with an industry-defining technology platform."

Under the terms of the agreement, which has been approved by both boards of directors, Macromedia stockholders will receive, at a fixed exchange ratio, 0.69 shares of Adobe common stock for every share of Macromedia common stock in a tax-free exchange. Based on Adobe's and Macromedia's closing prices on Friday, April 15, 2005, this represents a price of \$41.86 per share of Macromedia common stock. Upon the close of the transaction, Macromedia stockholders will own approximately 18 percent of the combined company on a pro forma basis.

In the combined company, Chizen will continue as chief executive officer and Shantanu Narayen will remain president and chief operating officer. Stephen Elop, president and chief executive officer of Macromedia, will join Adobe as president of worldwide field operations. Murray Demo will remain executive vice president and chief financial officer. Dr. John Warnock and Dr. Charles Geschke will remain as co-chairmen of the Board of Directors of the combined company and Rob Burgess, chairman of the Macromedia Board of Directors, will join the Adobe Board.

"Both Macromedia and Adobe are passionate about creating and enabling great experiences across a wide range of devices and operating systems," said Elop. "Our combined teams will be a powerful force for innovation around cutting-edge platforms for delivering content and applications."

Integration

The two companies are developing integration plans that build on the cultural similarities and the best business and product development practices from each company. The companies will make additional details and information about the acquisition available at <http://www.adobe.com/aboutadobe/invrelations/adobeandmacromedia.html>.

"While we anticipate the integration team will identify opportunities for cost savings by the time the acquisition closes, the primary motivation for the two companies' joining is to continue to expand and grow our business into new markets," said Chizen.

The acquisition, which is expected to close in Fall 2005, is subject to customary closing conditions, including approval by the stockholders of both companies and regulatory approvals. The transaction will be accounted for under purchase accounting rules.

Due to the absence at this time of estimates of the acquisition-related restructuring costs and the allocation of the purchase price between goodwill, in-process R&D, other intangibles and equity-based compensation expenses related to SFAS 123R, Adobe is currently unable to provide GAAP estimates on future earnings.


The transaction is currently expected

to be break-even to slightly accretive to earnings in the first twelve months after closing on a non-GAAP basis. The company's target of break even-to-slightly accretive to earnings on a non-GAAP basis assumes no adverse impact from the loss of deferred revenue in the first twelve months following the close due to purchase accounting.

Factpoint 2: Adobe Systems Incorporated

- *Market:* graphics, video and business document software
- *Headquarters:* San Jose, Calif.
- *Founded:* 1982
- *Chief executive officer:* Bruce Chizen
- *Stock:* adbe (nasdaq)
- *Employees:* approximately 4,000
- *Revenue:* \$1.67 Billion in fiscal 2004
- *Customers:* creative professionals, enterprises and governments, photo and video hobbyists
- *Market presence:*
 - more than a half-billion copies of adobe reader have been distributed worldwide.
 - the world's top 10 pc manufacturers ship their systems with adobe pdf technology pre-installed.
 - more than 700 thousand copies of adobe creative suite shipped in its first year (October 2004).

Factpoint 3: Macromedia, Inc.

- *Market:* web and application development software
- *Headquarters:* San Francisco, Calif.
- *Founded:* 1992
- *Chief executive officer:* Stephen Elop
- *Stock:* macr (nasdaq)
- *Employees:* approximately 1,500
- *Revenue:* \$422 million in cy 2004
- *Customers:* web and application developers, business users, mobile ecosystem
- *Market presence:*
 - Macromedia Flash Player is installed on more than 98% of Internet-connected desktops and more than half a billion users have the Flash client installed.
 - More than 25 million Flash-enabled devices have shipped.
 - Macromedia products are used by 90 percent of Fortune 1000 companies. 

Recent News:
 Microsoft is Gold sponsor
 Joel Spolsky (of Joel on Software)
 to give keynote
 Pre-Conference classes added for
 Monday and Tuesday
 Hotel block price expires 4/15/05

Accessibility / Usability

Raymond Camden
 Michael Smith

coldfusion

Advanced CF

Michael Dinowitz
 Geoff Snowman

Matt Liotta
 Deployment / Platform Simon Horwith

Tim Buntel
 Christian Cantrell
 Charles Arehart
 Jeff Peters

Manager
 Empowered Programming

Steve Drucker
 Ben Forta

MX Integration

Hal Helms
 Dave Watts

Shlomy Gantz
 Boot Camp



June 29 - July 1, 2005

Washington DC Area

7th Annual ColdFusion Conference

3 full days!

New Venue

75% Bigger



www.cfunited.com

Produced by
 TeraTech
 Programming
 www.teratech.com
 301.424.3903
 info@cfunited.com

- TeraTech
- Microsoft
- NewAtlanta
- CFDynamics
- Fog Creek
- Savvy
- Interakt
- UNIVERSAL Mind
- Paper Thin
- MX developer's journal
- COLDFUSION developer's Journal
- macromedia USER GROUPS
- SYNTHESIS
- AlvatoWeb



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

Ben Forta

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

Chuck Hoffman

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

Daniel Gregorio

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

Kathleen Ballard



The MX Blogosphere

This month's edition of our new regular feature – taking a stroll around the MX world
by mxdj news desk

If one of the true signs of a vibrant developer community is an active blogosphere surrounding a technology, then the MX suite of technologies certainly passes that test with flying colors. In case you're not yet actively blogging yourself, MXDJ brings you here a comprehensive selection from some of the best known blogger-commentators (and some of the not so well-known too), by way of whetting your appetite.

Don't forget that you can blog yourself now, too, at the MXDJ site – just follow the link from <http://mxdj.sys-con.com>.

Blog Topic: The Acquisition

Macromedia and Adobe Not Sure If it's a Good Thing?

By The Blogster (<http://williajay.linux-world.com/read/1214759.htm>)

"Being a user of Macromedia products for the last 15 years, I find the Adobe purchase to be a much better fit for Adobe than Macromedia. Macromedia has created such a wonderful web presence; Adobe has tried but hasn't come close. There is also the issue of support. I haven't had an issue with Macromedia's support, they have always been very patient and worked through my problems.

Over on Slashdot, they're talking about this: "According to John Dvorak the reasoning behind Adobe's recent (and to many, surprising) purchase of Macromedia for \$3.4 billion is that Adobe was afraid Microsoft was going to do it first."

Blog Topic: Flash

Class Deserialization: OpenAMF & Flashcom

By Jesse Randall Warden
(<http://www.jessewarden.com/>)

Had a hell of a time debugging some OpenAMF calls yesterday. Turns out, when Flash deserializes your class, it basically takes a vanilla object, puts properties on it and assigns their values, and then points that instance's `__proto__` property to the prototype of the class you registered via `Object.registerClass`. The downside to this is it doesn't run your setter functions on any getter/setters you have set on the class. Naturally, your getters fail immediately because they look to a private equivalent variable which is different, and when you call the setter... it's really a function.

How Flash manages to keep "firstName" the public property and "first-Name" the public getter function in the same namespace is beyond me, but regardless, I've tested in Flashcom last night, and the same thing happens there, too, so it appears to be how Flash deserializes your class.

The way we, "solved it" as my manager says, or "worked around it" as I claim, is emulating, EXACTLY the Java class equivalents. So, you have private properties in the Java model class, like:

```
private String firstName;
```

And same on the Flash side:

```
private var firstName:String;
```

And instead of getter/setter functions in Flash, you just use the get/set function methodology:

```
public function getFirstName():String
{
```

```
return firstName;
}
public function setFirstName(val:
String):Void
{
    firstName = val;
}
```

I really don't like this at all, and personally feel that there should either be an event letting you know when the class is deserialized (Flashcom does this for server-side ActionScript classes via the `onInitialize` event) so you can then run the getter/setters yourself, OR Flash should just intrinsically know there are getter/setters in place, and set the private variables accordingly. This gets sticky though because you're now having the Flash Player run code on your classes. Thus, I vote for the first.

Blog Topic: Flash

ProgressBar "Gotcha"

By Colin Mook from "moockblog"
(<http://moock.org/blog/>)

"I just noticed this undocumented limitation with the v2 ProgressBar component, and I figured it was worth sharing...

When a ProgressBar's mode property is set to "event" (the default) it won't broadcast `complete()` or `progress()` events. If you want to register for events from a ProgressBar make sure its mode is set to "polled" or "manual". If you are using the ProgressBar in "event" mode then you should register for events with the ProgressBar's source rather than with the ProgressBar itself.

Have fun preloading..."

Intermedia.NET...

Now serving the hottest ColdFusion.

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

Intermedia.NET's premier hosting services include:

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

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



INTERMEDIA.NET

Call us at: 1.888.379.7729

e-mail us at: sales@intermedia.NET

Visit us at: www.intermedia.NET

d

f

fw

cf

db

Best Practices with CSS in Dreamweaver MX

Cascading Style Sheets (CSS) is changing the website design process
by rachel andrew

The increasing use of Cascading Style Sheets (CSS) is changing the website design process. To meet the rising needs of CSS-conscious designers, Macromedia Dreamweaver MX includes many new and improved CSS-related features. With these new features, you can plan for future updates, build websites that are more compatible with World Wide Web Consortium (W3C) standards, and comply with new accessibility laws. This article discusses best practices when using CSS and highlights specific CSS features in Dreamweaver MX.

Style sheets are, in general terms, a collection of formatting rules that control the appearance of content in a web page. You can implement CSS on your pages three different ways:

- **Inline:** a one-time style placed in the code

- **Embedded:** a style sheet that controls the elements on one web page
- **External:** one external style sheet that can control the appearance of many web pages

Many websites, in fact, use combinations of these different kinds of CSS styles.

An important consideration in the use of CSS styles is the fact that different browsers (and different versions of the same browser) render CSS styles differently. In addition to variations in web browser compatibility, you should be aware that many other types of browsers exist, such as aural browsers, TV-based browsers, handheld devices such as Palm Pilots, and TTY (teletypewriter).

If you want to learn more about the basics of CSS before proceeding on with

a “best practices” analysis, review some of the materials listed in the More Resources section at the end of this article.

What does CSS “Best Practices” Mean? Most technologies have accepted standards, and CSS styles have their own set of standards. Although non-standard CSS practices (including its lack of use) exist on the Internet, there are many benefits to adhering to the defined standards.

In general, developers should aim to “separate content from presentation” as much as possible on all web pages. Proper separation is recommended in order to:

1. *Increase the longevity of your website*
Using non-standard practices may seem more convenient at the moment, but a future version of a browser may not be so tolerant of shortcuts. Individually changing all of your web pages in the future could be very time-consuming and defeat one of the main purposes of using CSS styles in the first place.
2. *Make your website more accessible to all users and devices*
Government legislation in some areas has required websites to be accessible to users with disabilities. Browsing devices that assist people with disabilities, such as an aural browser, are especially dependent on the best use of CSS styles.
3. *Allow you to update and maintain your website with ease*

Properly implemented, CSS styles allow you to make changes to your website on one page and see those changes take effect globally across your entire website instantaneously.

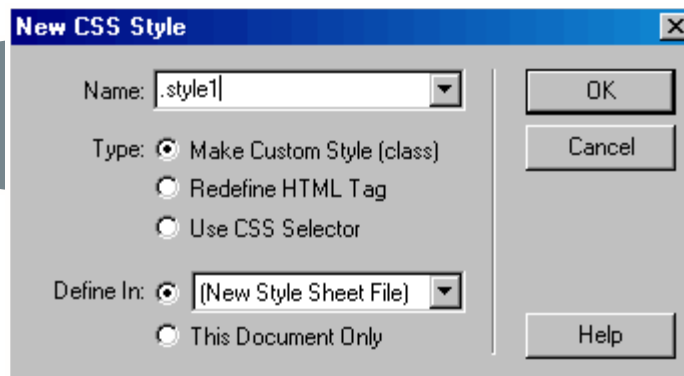


figure 1

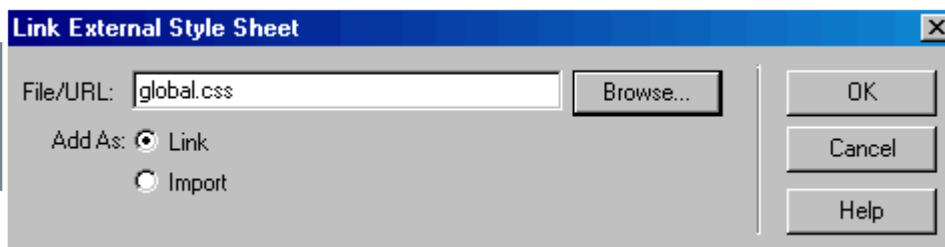


figure 2

Best Practices with CSS in Dreamweaver MX

One of the first decisions you'll have to make involves selecting which kind of CSS style to use. Below is an analysis of the different kinds of CSS styles with best practices in mind.

Inline CSS

Simply put, you should avoid inline CSS styles. Apart from any other considerations, using inline CSS means that you are not taking advantage of the true powers of CSS-and it means that you are not separating content from presentation.

Dreamweaver MX uses inline CSS mark-up primarily for positioning page elements (these positioned items are called "Layers" in the Dreamweaver MX UI), or to enable a specific DHTML effect which requires an inline style to alter an object's properties with JavaScript.

Embedded CSS

Embedded CSS is less than optimal

pages that link to the style sheet.

2. Easily update the look of all pages by changing the values in one file.
3. Make your pages much smaller and quicker to download, because all of the style information has been removed to one file.

Other CSS best practices will be mentioned as specific CSS-related features are outlined below.

Creating a CSS style in Dreamweaver MX

When you create a new style in Dreamweaver MX, you have two choices in the New CSS Style dialog box: New Style Sheet File or This Document Only (see Figure 1). Choosing New Style Sheet File will start the process for creating an external style sheet. This option requires you to name and save the CSS file before actually creating a style. The other option, This Document Only, will embed the style directly into the head of your document.

You can also select an existing style

the external style sheet (see Figure 2).

Linking to the style sheet is the most common approach. Selecting the "link" option will attach your external style sheet to your page by adding the following markup:

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

All CSS-enabled browsers support the linking option, and this method should be followed if you need older browsers (such as Netscape 4.x) to "see" the styles.

If you choose the "import" option, the style sheet will be attached with the following mark-up:

```
<style type="text/css">
<!--
@import url("mystyles.css");
-->
/style>
```

The import method of attaching

“Developers should aim to “separate content from presentation” as much as possible on all web pages”

because it only affects the style of the page on which it is written. During an update process, inconsistency across your site will result if any pages are missed. Additionally, your style information-which exists on each and every page of your site-has to be downloaded every time a user views a page.

External CSS Files

External CSS files should always be your first choice. By creating a CSS file that contains all of the style information for your website, you can:

1. Maintain a consistent look across all

sheet in the New Style dialog box to edit or add more definitions to it.

Should You Link or Import Your External Style Sheet?

After creating your external style sheet, you will need to attach it to each page (or attach it to a template). To do this, click the Attach Style Sheet button on the CSS Panel. The Link External Style Sheet dialog box will appear, allowing you to browse to the name of your style sheet. After you've located the style sheet, you can choose to link or import

a style sheet is not recognised by the older, version 4, browsers. However, it can come in handy. Developers often use the import method in order to include CSS styles designed for new browsers while also linking to an external style sheet to include styles for Netscape 4.x. This technique is explained in more detail below.

One Strategy to Cover Newer and Older Browsers

You can use both the link and import methods to attach two different style

figure 3



“Dreamweaver MX provides you many tools that will assist you in the development of your CSS-based websites”

sheets that, between the two of them, cover multiple browsers—including Netscape 4.x. One style sheet only contains values supported in the older browsers; and one style sheet contains additional or different properties for the newer browsers with more robust CSS support.

To accomplish this, you first attach the style sheet for the older browsers using the link method. Then, select import to attach the style sheet for newer browsers. The resulting code will look similar to this:

```
<link href="oldbrowsers.css"
rel="stylesheet" type="text/css" />
<style type="text/css">
<!--
@import url("newbrowsers.css");
-->
</style>
```

The version 4 browsers will completely ignore the imported style sheet and render the page according to the values in the linked one, leaving you free to fully use the capabilities of CSS for those browsers that support it. This approach eliminates the possibility of causing an unreadable page in older browsers.

The CSS Property Inspector

You can now easily switch into CSS mode within the Dreamweaver MX Property inspector. By default, the Property inspector displays the original font tags in HTML mode. To switch the Property inspector into CSS mode, click the CSS/HTML toggle (the small “A” next to the font pop-up menu). Instead of the font tag list, you will be presented with the list of CSS styles currently available

(see Figure 3).

You can always toggle back into HTML mode to work on a non-CSS based website.

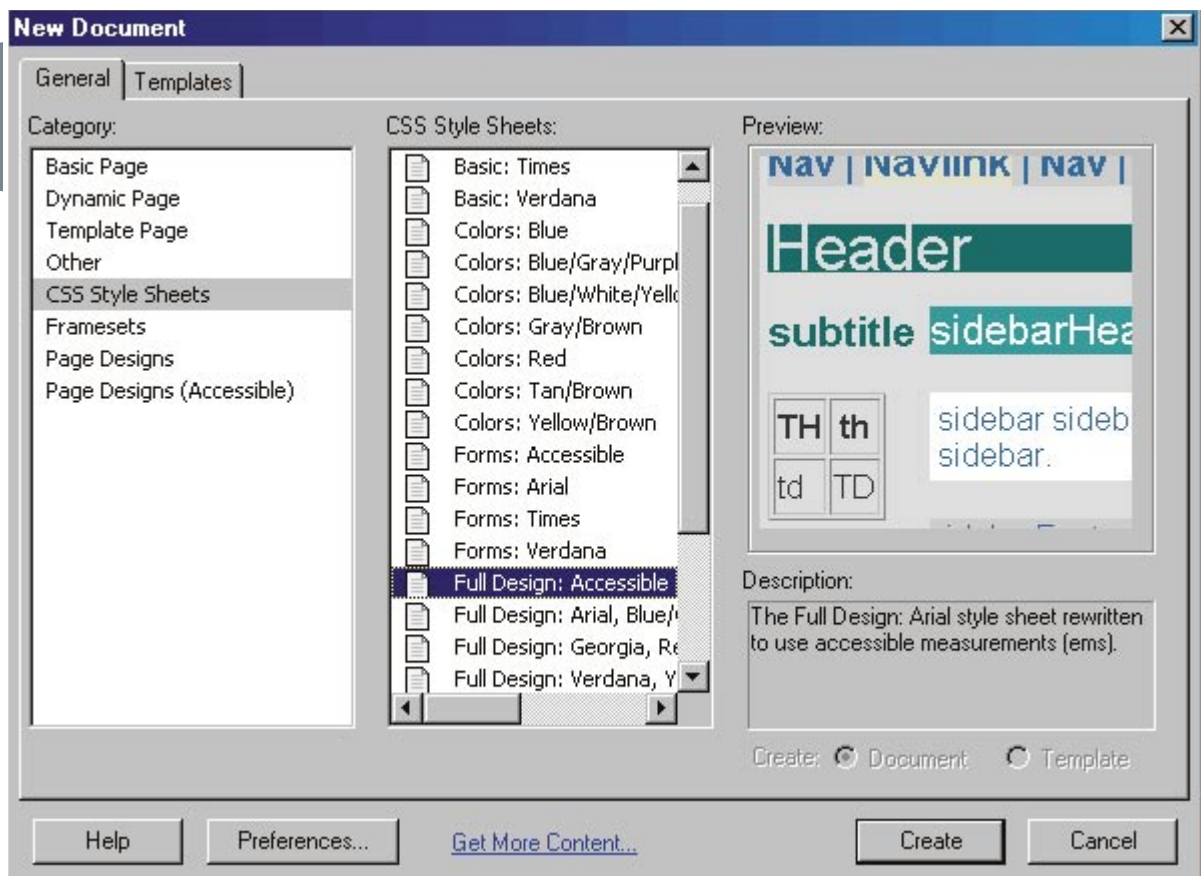
Ready-Made Style Sheets

A particularly exciting new CSS-related feature in Dreamweaver MX is the inclusion of ready-made CSS style sheets. New users to CSS can begin by experimenting with one of these ready-made style sheets.

To get started, choose File > New. In the New Document dialog box (under category) select CSS style sheets. A whole list of ready-made style sheets will appear in the right-hand pane. To implement best practices, you should choose one of the style sheets marked Accessible (see Figure 4).

You can now save this document into

figure 4



your site folder and attach the CSS file to your pages as described above.

Design Time Style Sheets

This convenient Dreamweaver MX feature allows you to apply a style sheet so you can see how your site really looks while you are still in Design view. This style sheet will not appear on your live site. From a best practice viewpoint, this feature is especially useful. If you are using the import and link method described earlier (with two style sheets), attaching a Design Time style sheet will allow you to design your website using one of those two style sheets. When you want to see how the page looks with the other style sheet, you can easily change the style sheet.

Design Time style sheets will also be helpful for developers who are using style sheets that will be housed on the server-side (with, for instance, ASP, PHP, or ColdFusion Application servers) or accessed with JavaScript on the client side. These server-side style sheets are another way to cope with the lack of CSS support in older browsers. However, in previous versions of Dreamweaver, this approach restricted you from seeing the effect of the style sheet applied to your page while you were designing. Design Time style sheets allow you to apply your style sheet so you can work visually within Dreamweaver. Another benefit to using Design Time style sheets is that when you upload your site files, you won't have to remember to go through the site removing unnecessary style sheets.

Validation

Whether you have created your own style sheet or used and edited a ready-made style sheet, validating your CSS styles will help to ensure that you are not accidentally using any non-standard tags or incorrect code. Frequently, problems with CSS will disappear once the style sheet is validated and the errors corrected.

Dreamweaver MX does not have a CSS validator, but you can accomplish this by using the W3C CSS Validation Service on the W3C website. Alternately, you can validate your HTML or XHTML mark-up from within Dreamweaver MX by choosing File > Check Page > Validate

Markup (for HTML) or File > Check Page > Validate as XML for XHTML.

Dreamweaver MX provides you many tools that will assist you in the development of your CSS-based websites. With Dreamweaver MX and a solid understanding of CSS and the importance of writing valid, accessible CSS code, you are well on your way to creating websites that will stand the test of time.

More Resources

For basic information about CSS Styles, see:

The W3C CSS Pages

CSShark: The CSS Know-How Site. Note, in particular, The Basics page.

The W3Schools CSS tutorial

"What is the W3C© and why do we care?"

by Julie Hallstrom, in the DevNet on Macromedia.com


The Dreamweaver MX Help documentation.

To access this material within Dreamweaver MX, choose Help > Using Dreamweaver > Adding Content > Inserting and Formatting Text > About Cascading Style Sheets. Also, try out the Designing with Cascading Style Sheets tutorial in the Tutorials section. Even experienced Dreamweaver users would do well to check out the new CSS capabilities by reading through these excellent help files.

For further, more advanced information on CSS, consult these sites:

- *New York Public Library's Online Style Guide provides a thorough XHTML and CSS Guide.*
- *Peter-Paul Koch's charts browser support for various aspects of CSS: CSS 2 tests.*

Bluerobot.com's The CSS Layout Reservoir (a collection of "borrowable" multi-column layouts).

- *More borrowable code can be found on CSS Layout Techniques: for Fun and Profit on glish.com.* 

Rachel Andrew runs her own web solutions company in the UK, edgeofmyseat.com, the company website also being home to various "web standards"-focused articles and Dreamweaver extensions. Rachel is involved with the Web Standards Project on the Dreamweaver Task Force, and hopes to encourage best practices in the support and use of W3C Standards in Dreamweaver. info@edgeofmyseat.com

macromedia®



Real world. Real training. Real results.

Learn from the best.

All Macromedia Authorized Training Partner locations provide you with instructors thoroughly trained and certified by Macromedia. Only MATPs offer courses that include hands-on tasks and real-world scenarios to give you valuable practical experience.

To find a Macromedia Authorized Training Partner, use our locator at:

www.macromedia.com/go/findaclass

Get the recognition you deserve!

Show your colleagues, employer, or prospective clients that you have what it takes to effectively develop, deploy, and maintain dynamic applications—become a Macromedia Certified Professional.

Become a Macromedia Certified Professional today:

www.macromedia.com/certification



macromedia®
**TRAINING AND
CERTIFICATION**

Copyright © 2005 Macromedia, Inc. and its licensors. All rights reserved. Macromedia and the Macromedia logo are trademarks or registered trademarks of Macromedia, Inc. in the United States and/or other countries. Other marks are the properties of their respective owners.



Skinning the V2 ProgressBar Component

This article explains how to completely change the graphical appearance of (i.e., "to skin") the V2 ProgressBar component. Before we begin, a warning: the techniques described in this article apply to the ProgressBar and to many other V2 components, but not to all. Some components require advanced skinning techniques not discussed in this article. For example, the Button component's skin, perhaps surprisingly, is primarily code-based. Skinning a Button component requires a thorough understanding of the Button skin code. Likewise, the specific steps required to skin components made up of multiple subcomponents, such as the DataGrid or the List, are not covered in this article. This article will help you get started with skinning and will give you the skills to skin many V2 components, but for some components you should expect to do further research (see Further Study at the end of this article for suggested resources).

by colin moock

The Basics: Styles Versus Skins

A component's appearance can be changed in two ways: 1) by setting the component's styles, 2) by skinning the component. Setting a component's styles means using code to change the component's color and font without changing its graphics. For example, the default appearance of a ProgressBar component is shown at Figure 1.

And Figure 2 shows what the ProgressBar component looks like with the "themeColor", "fontFamily" and "fontSize" styles set.

Here's the code used to style the ProgressBar shown in Figure 2:

```
pBar.setStyle("themeColor", 0xFF0000);  
pBar.setStyle("fontFamily", "Times New  
Roman");  
pBar.setStyle("fontSize", 16);
```

In Figure 2, the ProgressBar's color is now red instead of green and its text is set in Times New Roman instead of Verdana, but the basic graphical shapes of the ProgressBar are the same as they were in Figure 1. By setting a component's styles, we can gain some control over the appearance of the component, but when we need to change more than just colors and fonts we must skin the component. Skinning a component means replacing the component's graphics entirely, producing a completely new appearance for the component. For example, if we were creating an adventure game we might want to skin our game's ProgressBar to make it look like Figure 3.

Later we'll examine how the above "adventure game" ProgressBar was made. But before we do, it's worth pointing out that not everything about the appearance of a component can be changed, even with skins. The internal display system of each component determines what aspects of the component's appearance can be changed. For example, the variable-width "load bar" section of the ProgressBar component shown in Figures 1, 2, and 3 is produced by stretching a thin graphic between the left and right side of the component. Because the graphic is stretched, it must be a simple, abstract shape not, say, a repeating image such as the apple

shown in Figure 4.

Filling a region with a repeated graphic like the apple in Figure 4 is known as "tiling." Tiled skins are not natively supported by the V2 ProgressBar. Hence, modifying a ProgressBar to look like the one shown in Figure 4 would require custom code and in-depth knowledge of the ProgressBar's own skinning implementation.

What, Exactly, Is a "Skin"?

In interface-programming terms, the "skin" of a user interface is its graphical appearance, which is typically separated from code so that the interface can be changed visually without affecting the underlying logic.

Specifically, for a Flash V2 component, a "skin" is a collection of movie clips that contain the component's graphics. Each movie clip represents either an entire state of the component (e.g., enabled, disabled, pressed, etc) or a subsection of a particular state or states. For example, a skin movie clip might contain an entire disabled button face or just the arrow icon for a scrollbar. The skin is broken into multiple sections so that it can be assembled and resized at runtime in response to either user input or application logic. Each movie clip in a skin has a specific Linkage Identifier that is used by the component to attach the movie clip to the Flash movie stage. Once on stage, the movie clips are positioned and scaled to produce the graphical appearance of the component.

Figure 5 shows the skin movie clips for the ProgressBar component. The Linkage Identifier for each movie clip is shown in black text.

Among the various skin movie clips shown in Figure 5, those whose IDs begin with "ProgTrack" are the background of the ProgressBar while those whose IDs begin with "ProgBar" are the foreground of the ProgressBar – the part that stretches horizontally to indicate load progress. Both the "track" (background) and the "bar" (foreground) are made up of three separate sections: a left and right edge – the "caps" at the ends of the bar – and a middle section that stretches between the caps. When the ProgressBar is resized, the caps remain the same size and are placed at either end of the bar, while the middle section is scaled horizontally to

fill the space between the caps. Finally, the ProgIndBar is a special graphic that animates between the left and right caps if the duration of the load operation being represented is unknown. The ProgIndBar movie clip is only used when the ProgressBar instance's indeterminate property is set to true.

Changing the Skin of a Single Instance

As we've just seen, a component's skin is made up of graphics stored in movie clips (well...that's generally true, but some component skins are made of movie clips that link to ActionScript 2.0 classes instead of containing graphics). To change a skin, then, we need to tell the component to render itself using our own, customized movie clips instead of its default set. There are two ways to tell the component to use our movie clips instead of its own:

- change which Linkage Identifiers the component uses when attaching its skin movie clips
- set the Linkage Identifiers for our custom movie clips to match the Linkage Identifiers used by the component (in other words, override the component's default skin movie clips)

The first option requires more work, but lets us skin a component on a per-instance basis. The second option is easier to implement, but forces every instance of the component being skinned to use the custom skin. Let's take a look at both options, starting with changing the Linkage Identifiers a component uses when attaching its skin movie clips. The V2 components store the Linkage Identifiers for their skin movie clips in so-called "skin properties". For example, the Linkage Identifiers for the ProgressBar component (as shown in Figure 5) are stored in the following properties: ProgressBar.progTrackLeftName, ProgressBar.progTrackMiddleName, ProgressBar.progTrackRightName, ProgressBar.progBarLeftName, ProgressBar.progBarMiddleName, ProgressBar.progBarRightName, ProgressBar.progBarIndName. We can change which movie clips a component instance uses for its skins by setting its skin properties. Let's see how this works by making a progress bar with a black



square for its track's left cap instead of the default rounded grey left cap. Admittedly, this is a boring skin but it demonstrates the general principle. We'll make a more interesting skin later. Follow these steps:

1. Create a new Flash document (.fla file).
2. On Frame 1, Layer 1 drag a ProgressBar component to the stage.
3. Create a new movie clip symbol named Square.
4. In the Square movie clip, draw a small square shape 4 pixels wide and 4 pixels high. Position the shape with its top left corner at the Square symbol's registration point as shown (greatly magnified) in Figure 6.
5. Select the Square movie clip symbol in the Library
6. On the Library panel's pop-up options menu (top-right), select Linkage...
7. On the Linkage Properties dialog, for Identifier, enter "SquareSymbol".
8. Click OK to accept the Linkage Properties settings.
9. With the ProgressBar component instance selected on stage, add the following code to the Actions panel:

```
onClipEvent (initialize) {
```

```
    this.progTrackLeftName =  
    "SquareSymbol";  
}
```

10. Export the movie using Control > Test Movie. The ProgressBar component should look like the one in Figure 7 (again greatly magnified to show the new skin). Note the black square on the left side of the bar.

The preceding steps describe how to modify the skin of a component instance placed on stage manually in the Flash MX 2004 authoring tool. We changed the skin of the component by setting the progTrackLeftName property within the component's "initialize" event. But if we had created the instance with code at runtime, we would not be able to use onClipEvent() as we did in Step 9. Instead, we would use the createClassObject() method's initObj parameter to set the skin property on the new component instance. The following code does just that; it creates a ProgressBar instance at runtime with createClassObject() and uses the initObj parameter to set the progTrackLeftName property to "SquareSymbol". The result is exactly the

same as that shown in Figure 7.

```
import mx.controls.ProgressBar;  
this.createClassObject(ProgressBar,  
    "pBar", 0, {progTrackLeftName:"Square  
Symbol1"});
```

Where to Find a Component's Skin Properties

As we've just seen, in order to skin a component we can assign new movie clip Linkage Identifiers to its skin properties. But how do we determine which skin properties a component uses? There are two options.

First, we can look in the Flash MX 2004 Help files. The so-called "ellipsis" release of the Flash authoring tool (Version 7.2) contains updated Help files that include the skin properties for all V2 components. For each component, the skin properties are listed in the following location:

- Using Components > Components Dictionary > [component name] component > Customizing the [component name] component > Using skins with the [component name] component
- Using Components > Components Dictionary > ProgressBar component > Customizing the ProgressBar component > Using skins with the ProgressBar component

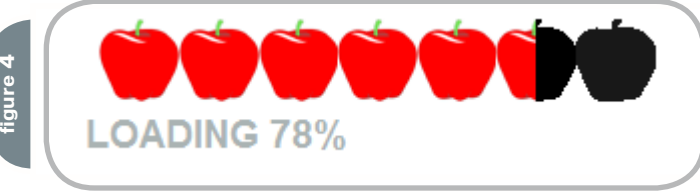
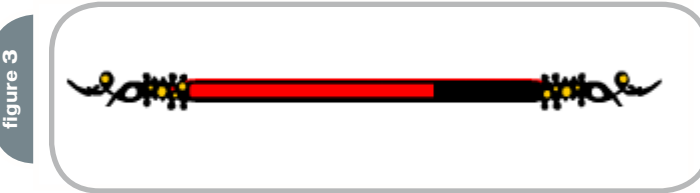
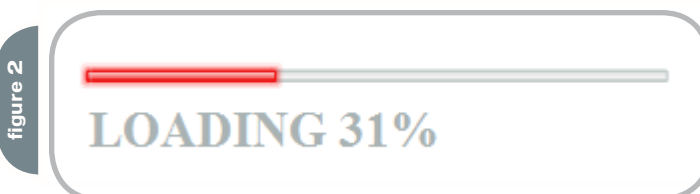
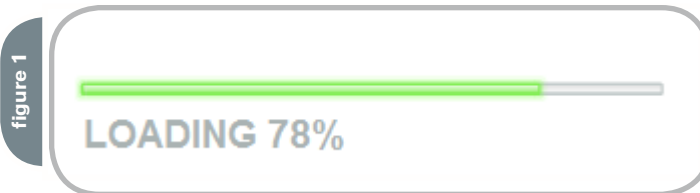
Alternatively, we can find the skin properties for a component by looking at the component's class source code. The component classes are located in the Flash MX 2004 application folder, in the following directory:

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

For example on an English language Windows system, the ProgressBar component's class is located here:

- \Program Files\Macromedia\Flash MX 2004\en\First Run\Classes\mx\controls\ProgressBar.as

Within that file we can find the following skin property definitions:



```

var progTrackLeftName:String =
"ProgTrackLeft";
var progTrackMiddleName:String =
"ProgTrackMiddle";
var progTrackRightName:String =
"ProgTrackRight";
var progBarLeftName:String =
"ProgBarLeft";
var progBarMiddleName:String =
"ProgBarMiddle";
var progBarRightName:String =
"ProgBarRight";
var progIndBarName:String =
"ProgIndBar";

```

However, locating the skin properties in the source code for a component's class can be difficult because the code is often not thoroughly commented and the skin property names do not follow a single naming convention. Some skin property names end with the word "Skin", as in "Button.trueUpSkin"; some skin property names end with the word "Icon", as in "RadioButton.falseOverIcon", while still others follow no convention at all, as in "NumericStepper.upArrowDisabled". Hence, the easiest way to determine the skin properties for a class is through the Flash MX 2004 Version 7.2 updated documentation.

No matter how you find a component's skin properties, once you know them you can skin the component by setting those properties to the Linkage Identifiers of your own custom skin movie clips.

Creating Skins That Can Be Colored with Styles

In our earlier SquareSymbol example, we replaced a ProgressBar instance's left track cap with a black square. In the example, we permanently set the color of the square to black – any attempt to change the color of the left track cap to red using styles would now fail. However, with a single line of code, we can make a skin's color alterable with styles. Using the same document we created earlier, follow these steps to see how it works:

1. Select the ProgressBar instance on stage.
2. In the Properties panel, set the ProgressBar's instance name to "pBar".
3. Edit the Square movie clip symbol.
4. Select the shape inside the Square movie clip.

5. From the Modify menu, choose Convert to Symbol.
6. For the symbol Name, enter "SquareShape". For Behavior, choose "Movie clip".
7. Edit the SquareShape movie clip symbol.
8. Rename Layer 1 to "graphic".
9. Add a new Layer to the SquareShape movie clip and name it "actions".
10. Select Frame 1 of the actions layer.
11. In the Actions panel, enter the following code:

```

mx.skins.ColoredSkinElement.
setColorStyle(this, "themeColor");

```

That code registers the skin movie clip as a styleable skin element. As a result, it will change color when the "themeColor" style is set.

12. Return to the main document timeline.
13. Select Frame 1.
14. In the Actions panel, enter the following code to set the ProgressBar theme color to red:

```
pBar.setStyle("themeColor", 0xFF0000);
```

15. Export the movie using Control > Test Movie. The ProgressBar component should look like the one in Figure 8 (again magnified to show the new skin). Notice that the left track cap is

now red because it is registered to change color when the "themeColor" style is set.

A detailed look at styleable custom skins is outside the scope of this article, but Figure 8 should give you a sense of what's possible. For more information on associating skins with color styles, see the following document in the Flash MX 2004 Version 7.2 updated Help: Using components > Customizing components > About skinning components > Linking skin color to styles.

Changing the Skin of All Instances in a Document

So far our SquareSymbol example has shown how to change the left track cap of a single ProgressBar instance only. To change the left track cap of all ProgressBar instances in a document we have several options. We can:

- set the progTrackLeftName property on the ProgressBar class's prototype
- use the default Linkage Identifier for our custom left track cap movie clip
- set the progTrackLeftName property in a ProgressBar subclass

We'll examine the first two techniques in this article; for information on subclassing component classes see the resources under Further Study.

Here's an example of the first technique,

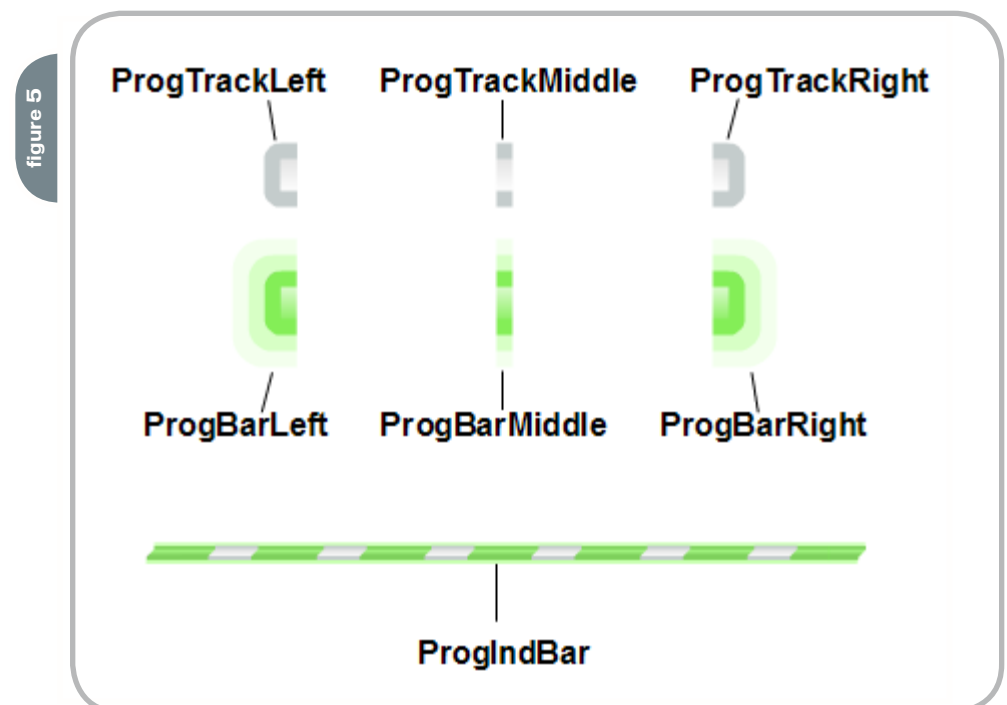
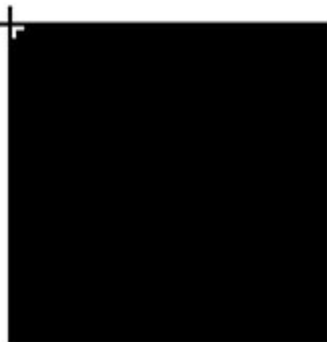




figure 6



setting the progTrackLeftName property on the ProgressBar class's prototype:

```
import mx.controls.ProgressBar;
ProgressBar.prototype.progTrackLeftName = "SquareSymbol";
```

That code tells the ProgressBar class to use the SquareSymbol movie clip when creating any ProgressBar component's left track cap. If the above code appears on frame 1, then all ProgressBar instances placed on frame 2 or later will be affected. That said, it is still possible to customize the left track cap for any single instance. To customize a single instance, we follow the exact steps we studied earlier in the SquareSymbol example--using onClipEvent(initialize) (for author-time instances) or createClassObject()'s initObj parameter (for runtime instances) to set the instance's progTrackLeftName property to the desired movie clip. For example, to apply a triangular cap to a single ProgressBar instance we would use the following code (assuming we had also created a triangle movie clip and given it a Linkage Identifier of "TriangleSymbol"):

```
onClipEvent (initialize) {
    this.progTrackLeftName =
    "TriangleSymbol";
}
```

Now let's consider the second technique for changing the left track cap of all ProgressBar instances in a document -- using the default Linkage Identifier for our custom left track cap movie clip. We'll start by determining the default Linkage Identifier for the skin movie clip we want to customize, and then we'll assign that Linkage Identifier to our own custom skin movie clip, overriding the component's default skin. Let's return to the document we created earlier and follow these steps to see how it works:

1. On the main timeline, select the on-stage ProgressBar instance.
2. Delete all code from the Actions panel.
3. Open the following document in the Flash MX 2004 Version 7.2 updated Help:
 - Using Components > Components Dictionary > ProgressBar component > Customizing the ProgressBar component > Using skins with the ProgressBar component
4. In the skin-property table, under the Description column for the progTrackLeftName property, find the property's "default value." The "default value" is the default Linkage Identifier for the left track cap skin movie clip. In our case, it's "ProgTrackLeft".
5. Change the Linkage Identifier for the Square movie clip symbol to "ProgTrackLeft". Note that case sensitivity matters! Make sure to use a capital P, T, and L when setting the Linkage

Identifier.

6. Export the movie using Control > Text Movie. The ProgressBar component should once again look like the one in Figure 7.

Because we set the Square symbol's Linkage Identifier to match the ProgressBar's default Linkage Identifier for the left track cap skin, the ProgressBar uses our Square instead of its own skin. If a document's Library contains a movie clip exported with the same Linkage Identifier as a component's internal skin movie clip, then the document's movie clip overrides (i.e., is used instead of) the component's internal movie clip.

Overriding a component's skin movie clip is an attractive way to skin components, especially for designers, because it involves no code whatsoever. However, for a complicated component, creating skin movie clips and assigning Linkage Identifiers can be time consuming. Fortunately, we can use so-called "themes" to reduce the labour required to override the skin of every component instance in a document.

Themes

A "theme" is a collection of skin movie clips and code used to change the appearance of the entire set of V2 components. Each theme is contained within its own .fla file. Macromedia provides two themes with Flash MX 2004, each of which offers its own graphical look for the V2 components. The default theme used by the V2 components is the Halo theme, which resides in a file named HaloTheme.fla in the following location:

- **Windows:** \Program Files\Macromedia\Flash MX 2004\{LANGUAGE_CODE}\Configuration\ComponentFLA\HaloTheme.fla
- **Macintosh:** HD/Applications/Macromedia Flash MX 2004/Configuration\ComponentFLA/HaloTheme.fla

The other theme supplied by Macromedia is known unceremoniously as "Sample Theme." It resides in a file named SampleTheme.fla in the following location:

- **Windows:** \Program Files\Macromedia\Flash MX 2004\{LANGUAGE_CODE}\Configuration\ComponentFLA\SampleTheme.fla

figure 7

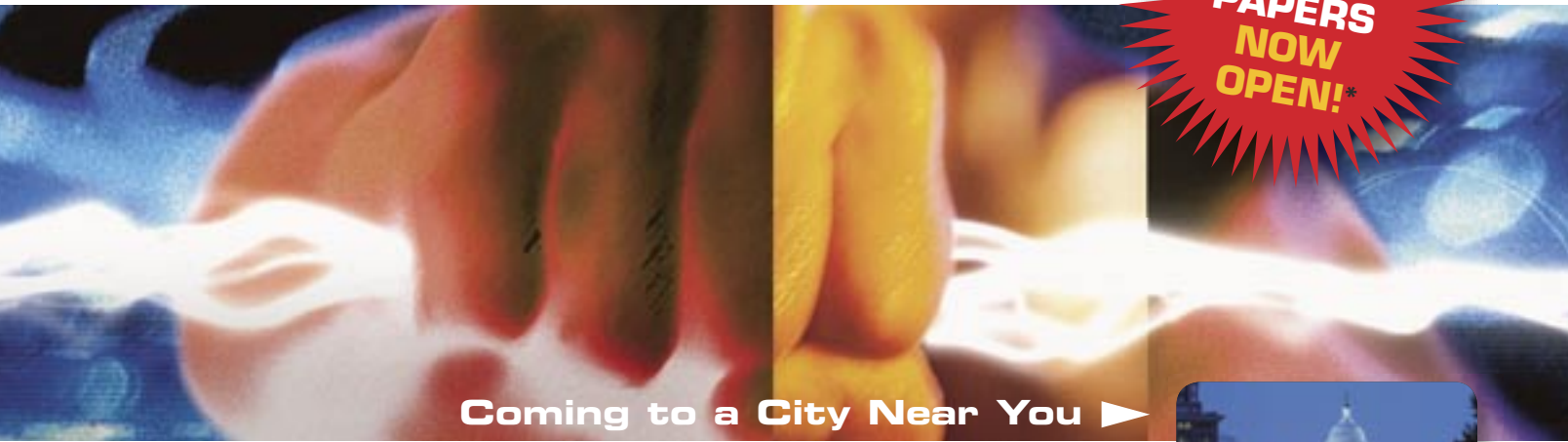


figure 8





[Engage and Explore] the technologies, solutions and applications that are driving today's **Web services** initiatives and strategies...



Coming to a City Near You ▶

Web Services Edge Fall Conference Series

3 Dynamic Conference Programs Targeting Major Industry Markets

20+ seminars within 5 tracks will address the hottest topics & issues:

- ▶ Web Services: The Benefits and Challenges
- ▶ Web Services Security
- ▶ SOA (Service-Oriented Architecture) and ESB (Enterprise Service Bus) Strategies
- ▶ Interoperability, Incremental Integration, & Open Source
- ▶ The Management Process in Developing a Web Services Strategy

Why Attend:

- ▶ Improve the return on your technology investment
- ▶ Develop & sharpen your strategy and identify key action steps
- ▶ Find new ways to reach and impress customers with Web services
- ▶ Maximize the power of your enterprise
- ▶ Protect your business from security threats
- ▶ Assess Web services as a viable option

Program Features:

- ▶ Keynotes
- ▶ Tutorials
- ▶ Panel Discussions



Attention Exhibitors:

- ▶ An Exhibit-Forum will display leading Web services products, services, and solutions

Register Today! www.SYS-CON.com/Edge2005

Sponsored by



*Call for Papers email: grisha@sys-con.com

For Exhibit and Sponsorship Information ▶ **Call 201 802-3066**



The Westin Washington, D.C.
Washington, D.C.

September 7-8, 2005

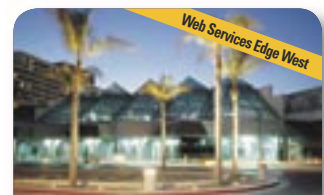
1



Helmsley Hotel
New York, NY

September 19-20, 2005

2



The Westin Santa Clara Convention Center
Santa Clara, CA

October 24-25, 2005

3



- **Macintosh:** HD/Applications/Macromedia Flash MX 2004/Configuration/ComponentFLA/SampleTheme fla

Creating custom themes and using the supplied themes is not our focus here; we're interested in themes only because they supply us with the skin

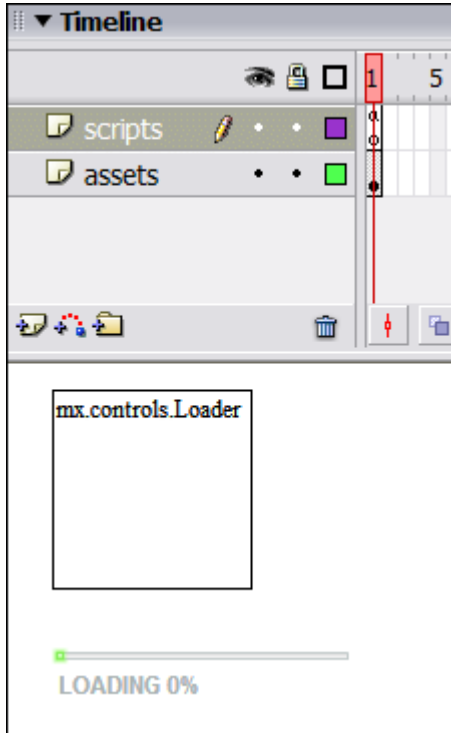


figure 9

movie clips we need to override a component's skin, complete with the correct default Linkage Identifiers!

Here's how it works. Each theme has an entire set of skin movie clips for every available V2 component. To skin a component in a new document, we simply copy the component's skin movie clips from one of the supplied themes into the new document, then alter the graphics as desired. PRESTO! Our new skin will automatically be applied to all component instances in the new document.

Let's use this "copy-from-theme" approach to build the adventure-game ProgressBar example shown earlier in Figure 3. But as we build the example, remind yourself that this technique works only with components that use graphics-based skins. As we learned earlier, components that are nested or use code-based skins require advanced techniques that are not covered here, but can be found in the Flash MX 2004 Version 7.2 help.

The Adventure-Game ProgressBar Example

Earlier we skinned the left track cap of a ProgressBar but we didn't actually use the ProgressBar to show any load progress. This time we'll link our adventure-game-style ProgressBar to a Loader component so that we can see it in action.

Before we start, here's a warning to

heed when using ProgressBar with Loader: when a ProgressBar's mode property is set to "event" (the default) it won't broadcast complete() or progress() events. If you want to register for events from a ProgressBar make sure its mode is set to "polled" or "manual". If you are using the ProgressBar in "event" mode then you should register for events with the ProgressBar's source rather than with the ProgressBar itself.

Bearing that warning in mind, follow these steps to create a ProgressBar linked to a Loader component:

1. Create a new Flash document (.fla file) named skinnedV2ProgressBar.fla.
2. Rename Layer 1 to "assets".
3. Create a new layer and name it "scripts".
4. On Frame 1 of the assets layer drag a ProgressBar component to the stage.
5. In the Properties panel, set the ProgressBar component's instance name to "pBar".
6. On Frame 1 of the assets layer drag a Loader component to the stage.
7. In the Properties panel, set the Loader component's instance name to "loader". Your document should now look like the one shown in Figure 9.
8. Finally, select Frame 1 of the scripts layer and add the following code to the Actions panel:

```
// Make the load bar red.
pBar.setStyle("themeColor", 0xFF0000);
// Tell the ProgressBar to show progress for the Loader.
// (The source property specifies the object for which
// load progress will be displayed.)
pBar.source = loader;

// Load an image into the Loader.
loader.contentPath = "http://marsrovers.jpl.nasa.gov/gallery/press/spirit/"
                    + "20050113a/site_
A89_CY_navcam_360_cyl-A365R1.jpg";
```

Our ProgressBar is now fully functional. Let's get skinning.

First, we need to copy the ProgressBar skin assets from the HaloTheme.fla file to our document's Library. Follow these steps:

1. Select File > Import > Open External Library.
2. Open HaloTheme.fla, which is in the following location:
 - **Windows:** \Program Files\

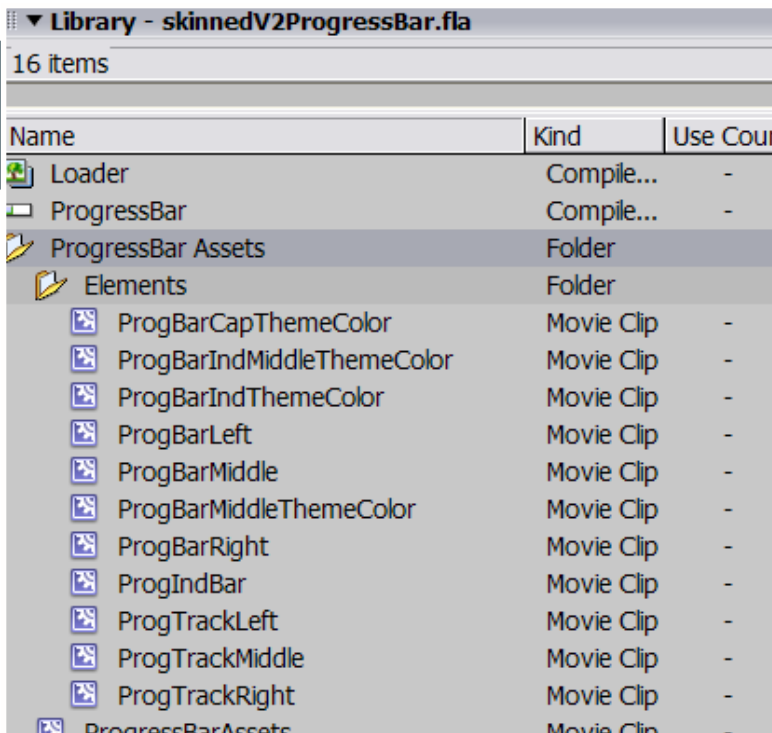


figure 10

Macromedia\Flash MX
2004\{LANGUAGE_CODE}\
Configuration\ComponentFLA\
HaloTheme fla

- **Macintosh:** HD/Applications/
Macromedia Flash MX 2004/
Configuration\ComponentFLA/
HaloTheme fla

3. With both the skinnedV2ProgressBar fla Library and the HaloTheme fla Library open, drag the following folder from HaloTheme fla to skinnedV2ProgressBar fla's Library: Flash UI Components 2 > Themes > MMDefault > ProgressBar Assets

Your skinnedV2ProgressBar fla Library should now look like the one shown in

In the Elements folder (shown in Figure 10), you should recognize the following symbols as the ProgressBar skin movie clips (shown earlier in Figure 5): ProgBarLeft, ProgBarMiddle, ProgBarRight, ProgIndBar, ProgTrackLeft, ProgTrackMiddle, and ProgTrackRight. Those symbols are the movie clips that you would normally redesign when skinning the ProgressBar.

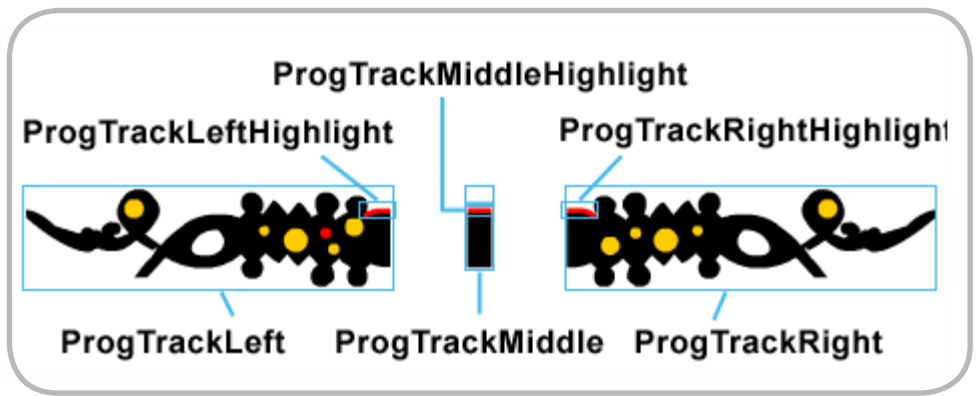
In addition to the skin movie clips from Figure 5, the Elements folder also contains a series of symbols whose names end with "ThemeColor": ProgBarCapThemeColor, ProgBarIndMiddleThemeColor, ProgBarIndThemeColor, and ProgBarMiddleThemeColor. Those movie clips contain the raw shapes for the caps and middles of the ProgressBar's track (background), load bar (foreground), and indeterminate bar (animated bar that indicates an unknown load progress). The raw shapes are contained by the "ThemeColor" movie clips so that they can be associated with a style (exactly like we saw earlier under "Creating skins that can be colored with styles"). Each "ThemeColor" movie clip contains the following single line of code, which causes its color to change when the "themeColor" style is set:

```
mx.skins.ColoredSkinElement.  
setColorStyle(this, "themeColor");
```

Finally, the ProgressBar Assets folder itself contains a single movie clip symbol named ProgressBarAssets. That symbol simply forces the ProgressBar skins to export with the movie.

For our purposes, we're only interested

figure 11



in skinning the track (background) of the ProgressBar, so we'll only need to change the following movie clips: ProgTrackLeft, ProgTrackMiddle, and ProgTrackRight. In addition to changing those movie clips, we'll also create three new movie clips to contain our track's highlight shapes: ProgTrackLeftHighlight, ProgTrackMiddleHighlight, and ProgTrackRightHighlight. We place the highlight shapes in movie clips so that we can associate them with the "themeColor" style (i.e., so that they change color whenever the "themeColor" style is set). The highlight movie clips are nested inside the track movie clips.

Figure 11 shows the completed skin movie clips for our ProgressBar skin.

Notice that the ProgTrackMiddle movie clip has a transparent (i.e., 0% alpha) shape just above the ProgTrackMiddleHighlight movie clip. The transparent shape forces the track's middle section to line up with the left and right caps. Transparent shapes are often used to force proper alignment in graphical component skins.

Now follow these steps to apply the adventure-game style skin to the ProgressBar in skinnedV2ProgressBar fla.

1. Edit the ProgTrackLeft movie clip and delete everything on its stage (you may have to unlock layers to do so).
2. Delete all but one of the layers in ProgTrackLeft's timeline.
3. Rename the remaining layer "graphic".
4. Draw a shape like the one shown for ProgTrackLeft in Figure 11. Make sure to place all graphics below, and to the right of, the symbol's registration point (i.e., the registration point should be at the top-left corner of the graphic).
5. Draw a shape like the one shown for ProgTrackLeftHighlight in Figure 11.
6. Select the ProgTrackLeftHighlight shape.
7. Convert the ProgTrackLeftHighlight

shape to a movie clip by choosing Modify > Convert to Symbol.

8. On the Convert to Symbol dialog, for Name, enter ProgTrackLeftHighlight, then click OK.
9. Edit the ProgTrackLeftHighlight symbol.
10. Rename Layer 1 to "graphic".
11. Create a new layer and name it "actions".
12. Select Frame 1 of the actions layer.
13. Add the following code to the Actions panel:

```
mx.skins.ColoredSkinElement.  
setColorStyle(this, "themeColor");
```

14. Repeat steps 1 to 11 to create the ProgTrackMiddle and ProgTrackRight movie clips, substituting the appropriate graphics and symbol names from Figure 11. Remember to include the transparent shape above ProgTrackMiddleHighlight in ProgTrackMiddle.

And that's it! The adventure-game ProgressBar skin is done. Choose Control > Test Movie to see it if worked. If it didn't work, try comparing it with the downloadable example file provided at the beginning of this article.

Further Study

For more information on skinning and customizing the V2 components, see the following resources:

- Skinning the Flash MX 2004 Components
- Flash MX 2004 Version 7.2 updated Help: Using Components > Customizing Components > About Skinning Components
- Building and Testing Components in Macromedia Flash MX 2004
- Flash MX 2004 Version 7.2 updated Help: Using Components > Creating Components

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



Uncovering Fireworks Masks: Masking with Bitmaps

In part two of this 2-part article on how masks function in Fireworks, we look at vector objects
by kim cavanaugh

In last month's article I led off with the same quotation from Yeats. At the risk of seeming culturally challenged, I'll lead off with it again for the second treatment of how masks function in Fireworks, simply because it is so apt.

A mask in a graphical editor like Fireworks (or Flash, among others) has the same function that Yeats describes – to control how much we can see of the object behind the mask. The interplay between the mask and the mask wearer is what makes our hearts beat. And of course, when we're dealing with art in any form we do want to get the hearts of our viewers to beat just a little harder. The idea is to grab and draw and maintain attention with our graphics, be they in print or online.

Last month you saw how to do that with bitmap masks. The focus for this month's article is on how vector objects can be used in Fireworks to hide and reveal images and objects in our compositions. And just as in the previous article

you'll have the chance to see some practical examples of how vector-based tools in Fireworks can be used for some common creative effects. But first the obligatory review.

Masks and Masking Reviewed

What is a mask? A mask is a graphical object that is placed on top of another image and given instructions on how it should interact with its partner. Masks are always a pair of objects working in concert to allow creative interplays that modify the appearance of the two. Masks are a marriage between two objects: a masking object lies above its partner and, through its properties, hides or reveals the masked object sitting below.

As well as the how of masking, it's also important that you understand the why of masking and the advantages that these techniques afford you.

1. Masking is a non-destructive process that does not alter the masked object. If things go awry, deleting or editing the mask does no damage to the original photograph or art.
2. Masking techniques can help you overcome Fireworks' limited tool set for working with bitmap images. It should come as no surprise that Fireworks is not Photoshop, but many people still insist on grumbling that the tools are dissimilar between the two applications. Fireworks is made to produce screen graphics and Photoshop targets print images. So while they both push pixels around, Fireworks doesn't have the precise bitmap editing tools that Photoshop contains. Does this mean that you can't get creative with Fireworks? Not at all, but you do have

to use different techniques, and many of those involve masking.

3. Masks remain editable in your source Fireworks file unless you take conscious steps to merge your objects together and flatten them. So while Fireworks may not allow you to select pixels with the same precision that Photoshop has, the ability to go back a year from now and open a Fireworks PNG file and make some minor adjustments to a mask has significant advantages over techniques that may throw pixels away. Once those pixels are gone, they're gone for good. Masks help you avoid making changes that you can't recover from, or edit, when future revisions need to be made.

Masks themselves are relatively easy to understand. When one graphical object interacts with another as a mask, new transparency values are created for the masked object based on the color of the masking object. If the masking object is entirely white then nothing will appear to have changed in the masked image. If on the other hand, the masking object is completely black the masked image will be totally transparent. Additionally, areas outside a mask will be hidden as well. Figure 1 illustrates how a mask with varying fill properties affects the masked image below it, creating areas that are opaque, transparent, and semi-transparent based on the size and position of the mask and the colors used in the masking object.

When it comes to Fireworks then, knowing how to create vector objects and using them in different combinations allows you a great deal of creative license. Those who master Fireworks have learned

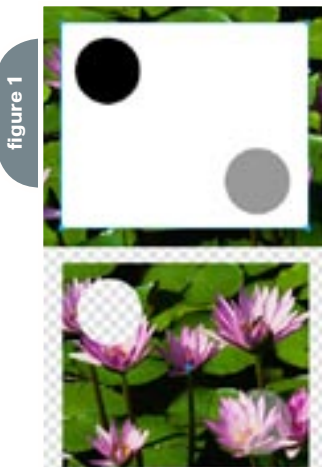


figure 1

how to mask properly, and whether they use bitmap objects or vector shapes, they have a handle on how the interplay between objects can be used to maximum advantage.

Using Masks in Fireworks for Creative Effects

The rest of this article will focus on those fundamental vector masking skills and three common methods for using masks in Fireworks for creative effects.

The use of vector objects can be broken into three categories based on the effects you hope to achieve:

1. Masking with text allows you to paste a bitmap image such as a photograph inside text objects. You can also create text masks with complex stroke-and-fill properties that give you even greater control over the output in your final composition.
2. Vector masks can be used to crop or fade an image to perform tasks such as cropping an irregular shape or creating a collage. This type of mask can be performed either with the automatic tools that Fireworks provides or with your own custom combination of fills and strokes on a simple vector shape.
3. Removing backgrounds is also accomplished with vector masks, most commonly created using the Pen tool to create an irregular shape that covers the part of the image that you want to show, while hiding the rest of the photograph.

You'll take a look at each of these methods in turn as the rest of this article shows how each of these techniques is accomplished.

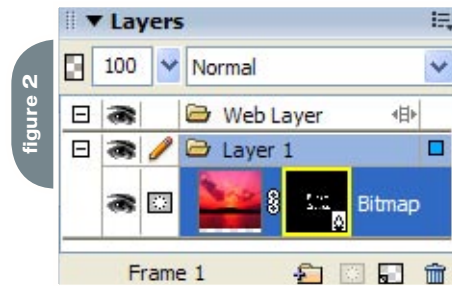
Masking with Text by Pasting Inside

Combining text with images is a great way to create interest in a composition.

Any image can be pasted inside text, which automatically creates a mask.

1. To begin, start a new Fireworks document and choose File > Import to find an appropriate image to work with. In these examples I've selected sunset.jpg from My Documents/My Pictures/Sample Pictures that Windows contains. (Macintosh users can find some great photos to work with at MacintoshHD:Library:User Photos.)

2. Choose File > Import and browse to the image you wish to use. After you choose Open you'll be returned to your canvas where you can click once to import the image. Save your file.
3. Using the Text tool type some text directly on top of your image. A nice heavy font works best here.
4. Select the image and cut it from the canvas by choosing Edit > Cut, or press the Ctrl/Cmd+X.
5. Select the text and choose Edit > Paste Inside. Your photograph will now replace any fill that has been applied to your text.
6. In the Layers panel note the way that your photograph has been combined with the text. Figure 2 illustrates how a text mask applied in this manner will appear in the Layers panel.
7. Click on the thumbnail of your text on the right in the Layers panel to select the text. You will see a yellow border appear around the thumbnail and your text will also have a yellow box surrounding it.
8. With the text selected you can now use the Property Inspector to change some of the text properties. In the example you see in Figure 3 I have added a stroke color and set the stroke to 3 pixels wide. I've also applied two instances of a drop shadow effect to



- the text to get a good heavy shadow.
9. Notice that when using the Paste Inside technique that by default you see the path outline and the fill and stroke, as indicated in the Property Inspector. This differs from the technique you'll use next to apply a mask manually.

Masking with Text Objects

In the first method for applying a text mask we cut and pasted the image into a text object. In this next lesson you'll see how to create a text object and set its properties before applying your mask. This technique uses the ability that masks have to set varying degrees of transparency to create an effect.

1. Start a new Fireworks document with a black canvas color and import your practice photograph. Save your file.
2. Import the image you wish to mask by choosing File > Import and browsing

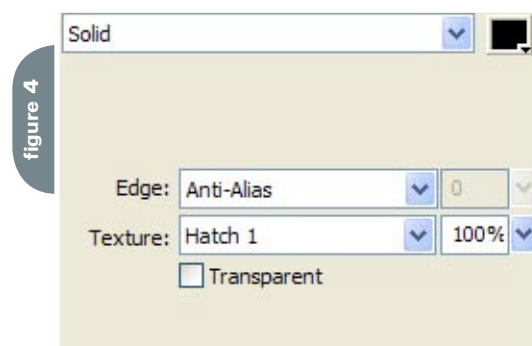
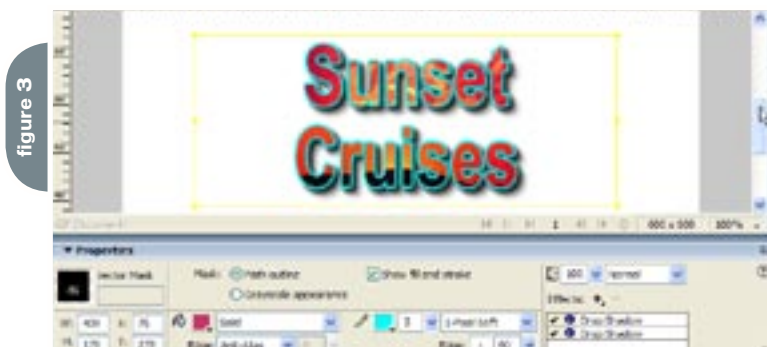




figure 5

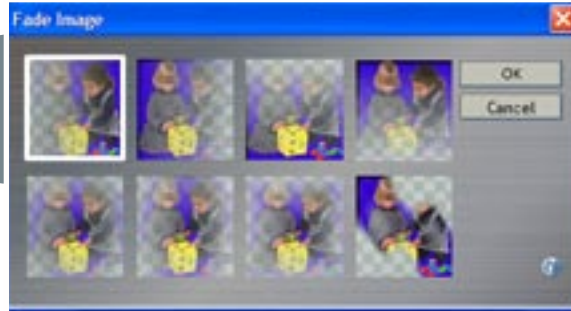


figure 6



figure 7

- to your photograph.
- Use the Text tool to create a text object with a message of your choosing. In my example I used an Arial font set to 96 pixels, bold.
 - Click on the Fill color button in the Property Inspector and select the Fill

- Options button at the bottom of the pop-up window that appears. Set the fill color to black and the Texture value to Hatch 1 with 100% of the texture showing. Figure 4 shows how your Fill options should appear.
- Use the same method for creating a Stroke on your text. Select the Stroke color button and click on the Stroke Options button to see all of the possible selections. For my example I chose a white stroke color with a grain texture applied at 88% and a width of 4 pixels. Refer to Figure 5 for a look at how your text should appear.
- Once you have the text appearance set the way you'd like, Shift+click the text and the photo so they're both selected. Or, if those are the only two objects in your document press Ctrl/Cmd+A. Both the text and the image must be selected before you proceed.
- Choose Modify > Mask > Group as Mask to apply the mask. Depending on the colors you've used the lighter colors in the text will allow the image to be seen while the darker colors will hide the photo.



figure 8

- You can add additional effects to get some interesting results by selecting the text object in the Layers panel and then turning to the Property Inspector to apply your effects. In Figure 5 you see my sample where a red glow effect has been added.

Masking with Vector Shapes

Now that you've seen a couple of examples of how masks can be created with text, let's move on to the next category – masking with vector shapes.

Once again there's an automated method for making masks like this and techniques that allow you greater control over your final output. Let's look at both methods so you have a handle on the basic principles.

Masking with the Fade Image Command

Using the Fade Image command that ships with Fireworks allows you to quickly fade images using 8 presets that ship with Fireworks. The command automatically draws an object with a gradient fill then applies the mask to a selected image to create a fade effect. This method is fast and easy, but since preset fills are used the creative options are a little limited. Let's take a quick look at how it's done.

- Open a file or import one of your existing images to a new Fireworks document. Select the image you wish to fade.
- From the Menu Bar select Commands > Creative > Fade Image.
- Click on one of the fade thumbnails, shown in Figure 6, to set the shape and direction that you want your image to fade. Click OK.
- Your image will now be converted to a masked object with portions of the image changing from transparent to opaque based on the size and position of the masking object that the command creates. You can modify the fade by adjusting the position and width of the gradient inside the masking object by adjusting the control handles that appear on screen.

Cropping with Vector Masks

Now that you've seen the automated method for creating a fade let's look at how the same principles are

figure 9



applied in a method that gives you more creative control. By creating your own object and setting its shape, position, size, and fill properties you have complete freedom to create your mask in countless ways.

1. Once again start by creating a new Fireworks document and importing one of your practice images.
2. In this example you'll create an oval crop with a textured appearance by positioning an oval shape over the top of your image. Using the Ellipse tool, draw an oval shape over the top of your image.
3. With the oval shape still selected, set your fill to the Jeans pattern by clicking on the Fill type drop-down in the Property Inspector.
4. Set the edge of your shape to Feather with a value of 16 pixels. Set the stroke to None. Figure 7 displays the correct Fill and Stroke settings for this example.
5. Once your settings for the oval shape are correct, select both the shape and the image under it by Shift+clicking the two objects, or select them both in the Layers panel.
6. From the Menu Bar choose Modify > Mask > Group as Mask. Your two images will now be combined as you see in Figure 8.

In this example the Jeans texture was selected because it demonstrates how color variations work when a mask is created. By using a texture that has a range of colors from dark to light, the effect of applying the mask is a textured appearance for your image.

Of course there are endless possibilities for using masks in this manner, including using textures and gradient fills to further modify how your composition will appear. You should be able to see clearly from this example that adjusting

the masking object so there are ranges of shades from dark to light results in an entirely new appearance for your image.

Removing Backgrounds with Masks

Perhaps the most common use of masks is for the removal of objects from the background. Unlike Photoshop, where the pixels in the background are physically removed from the image, with masks the background is made transparent by not covering it with a mask. Typically the shape itself is created by using the Pen tool since this allows you the greatest possible control over the position of the anchor points that define your shape and the lines that connect your points.

By necessity this example will be brief, but there are several excellent tutorials online that cover the use of the Pen tool and how masks are used to remove backgrounds. The key to remember is that your shape must cover the area of the image that you want to remain visible, while leaving the background outside the mask.

1. Start once again by creating a new Fireworks document and importing one of your practice images.
2. Select the Pen tool and stitch a basic shape around the part of the image you wish to keep. By clicking and releasing with your mouse you'll create square anchor points that are connected by straight lines, as you see in Figure 9. Using this method allows you to quickly "rough in" the shape of your mask.
3. With your basic shape in place, use the Pen tool to modify your anchor points as necessary. Click directly on top of an existing anchor point, represented by the tiny squares at the border of the shape, and while holding down the mouse button, gently drag away from the point. With a little bit (OK, maybe a lot) of practice you'll be able to modify your points so that the lines in the shape precisely cover your image.
4. Additional modifications to the anchor points and their connecting lines can be achieved by using the Subselection tool (the white arrow in the Tools panel) to modify the shape.
5. Once your shape covers the part of the image you wish to keep set the

Stroke to None and the Fill color to white. If you want to blur the edges of the masked object a bit set the Edge to Feather with a value of 3 or 4 pixels. This is often helpful where you need to hide any minor imperfections in your mask.

6. Select both the masking object and the object to be masked then choose Modify > Mask > Group as Mask. The parts of your image not covered by the masking object will now be hidden from view.
7. For final adjustments to your mask simply select the thumbnail image of the shape in the Layers panel. Your path outline will appear in yellow and you can then use either the Pen tool or Subselection tool to further modify your mask. Figure 10 shows the final output that I achieved in this quick example.

Summary

In this article you've seen the three most common methods for combining vector objects with images to create masks. Masks can be generated with text objects, with basic shapes, or with complex shapes that you draw with the Pen tool. All three techniques are important skills to develop as you continue to develop your creative chops with Fireworks. ☺

Kim Cavanaugh has been teaching and writing about web design software from Macromedia for over 5 years. He has written two books about Dreamweaver and Fireworks, collaborated on books about Dreamweaver, Fireworks, Flash and Contribute, and continues to write extensively about Studio MX tools for CommunityMX.com. In addition to his tutorials at CommunityMX, you can find more of his tutorials at his Beginner's Guide to Dreamweaver and Fireworks website (www.dw-fw-beginners.com) and read about things that interest him at his BrainFrieze blog (www.brainfrieze.net). cavanaugh_l@firm.edu



figure 10

d

f

fw

cf

db

10

ISSUES IDENTIFIED & DISCUSSED

For those not familiar with web services, they are a way for servers to exchange messages with each other using XML standards. Despite the extreme hype, web services do not enable you to do anything that was not previously possible. Since the earliest

versions of ColdFusion, CF developers have used the CFHTTP tag to post an HTTP request to another server and then analyzed the response. Submitting (or receiving) a request via web services simply does the same thing, albeit using XML standards.

by **steven rubenstein**

CFMX and Web Services

With ColdFusion MX, Macromedia made implementing web services very easy. (Although CFMX 7 was recently released, the web services implementation has no major changes or enhancements.) To receive a web services request, simply create a ColdFusion Component (CFC) and set the "access" value to "remote" for those functions which should be accessible as a web service.

One benefit of web services is that they are "self-describing," meaning the web service itself provides the documentation that tells others systems (or people) the inputs (name and field type) and output field type for that web services function. This documentation is Web Services Definition Language, or WSDL. To generate the WSDL file for each "access=remote" function in a ColdFusion Component, simply call that component via your browser followed by "?wsdl". For instance:

```
http://localhost/YourComponent.  
cfc?wsdl
```

The resulting file is a WSDL document, which is just an XML document.

To submit a web service via ColdFusion, you can use the CFINVOKE tag just as if you were submitting a request to a normal CFC function. Other methods of invoking functions will work as well, such as within CFSCRIPT. It doesn't matter whether the web service request is sent to a server running ColdFusion, Java, .NET, PHP, and so on.

How to Web Services-Enable Applications: A Case Study

So implementing web services functions in ColdFusion is simple. But that's very different from web services-enabling your application. This requires more than just creating some CFC functions set to "access=remote." It requires authenticating the user submitting the request, authorizing they have permission for that function, and validating that they own each target of that request.

For instance, if the function is to update a product, you first authenticate the user, i.e., whether they are logged in. If the user isn't logged in or their session has expired, they must first log in. Once you

have determined their identity, you next authorize whether they have permission to update products. This is based on how you implement permissions, which can be via roles, individual permissions, and so on. Finally, you must verify whether the user owns the product he/she is trying to update, i.e., that the product belongs to that user and not another user (not to mention that the product itself exists).

Of course, you also need to validate the updated values just as you would when submitting via a normal browser-based form. And if there are any errors, just as you display error messages to the user, you need to tell the user submitting the web service that their update request was not successful.

What follows is a case study on web services-enabling Averum Billing. It will hopefully provide MXDJ readers with an overview of the various issues we encountered. Some of these problems are specific to ColdFusion while others are generic problems that apply regardless of your development platform.

There are 10 primary issues, which I list below and will discuss:

1. File size limit
2. Session management
3. Redundant CFCs
4. Application scoped CFCs
5. Custom IDs vs. internal IDs
6. All fields are required
7. Returning queries
8. Boolean fields
9. Error messages
10. Debugging

1. File Size Limit

For developers using Fusebox or other methodologies where all pages are called via index.cfm or some other file, it's a natural preference to prefer that all web services requests use the same ColdFusion Component as well. Unfortunately, that just ain't gonna happen. Java has a 64k file size limit, which can quickly be reached with a few functions and their arguments.

You can try moving all logic outside of the component (which is recommended anyway) but you will still hit the limit eventually. Plus you will also find that the CFRETURN tag must be in the CFC itself and not an included file, so you will need a variable to store the returned value.

Bottom line: if you are truly web ser-

vices-enabling your application, you must give up the dream of using the same CFC for all web services functions. You'll need to use multiple CFCs. So you may as well separate the functions into logical groupings – CFC1 and CFC2 are not exactly intuitive.

2. Session Management

If your application requires users to log in, they'll need to do so before accessing any data via web services as well. Despite ColdFusion's documentation, managing sessions in web services is not simply a matter of using CFAPPLICATION or cookies. Normal browser-based sessions based on cookies do not work. A more creative solution is required.

It's possible that including CFID and CFTOKEN in the URL when calling the web service component would work, but we didn't try this. It's not a standard practice in web services for the URL to invoke a web service to vary. That's why you have input variables. Moreover, the default setting for a browser-based ColdFusion session to timeout is 20 minutes. However, our clients preferred a longer timeout period. For Averum Billing, we chose 2 hours, which would have been dangerous if storing those sessions in memory.

As usual, the best way to resolve new issues is to see how other sites have dealt with them. After reviewing the web services implementations of several other popular sites, we choose to use a UUID variable as the session ID, which can generally be assumed to be unique.

After logging in with a username and password (and in our case, a company name as well), Averum Billing's web services login function returns a UUID, which must then be submitted with each subsequent web services request. This authenticates the user so that the username and password do not need to be submitted in each request.

For additional security, Averum Billing tracks the IP address used when logging in. All subsequent web services requests for that UUID must originate from the same IP address. Otherwise the request is rejected because the user isn't logged in.

Like any session, there is a need to store "session" variables. However, because we cannot store normal session variables (between requests), we technically do not have a session and cannot



store variables in the Session scope. Instead, to simulate session variables, we chose to add a `WebServiceSession` table to our database that tracks the necessary session information.

While this limits the number and type of session variables we can store, it was the only scalable solution. We initially implemented sessions using an application-scoped structure where the structure index was the UUID of the web services session. The resulting value would also be a structure to store the necessary session variables. For instance:

```
<CFSET Application.webServiceSession[theUUID].userID = "1">
```

However, this method has an inherent scalability problem. Adding a new session or updating an existing session required updating an application-scoped variable, which in turn requires using `CFLOCK` to ensure only one process is updating the variable at a time. If you expect more than a few customers to be accessing the web services interface at the same time, the constant locking of the application-scoped variable becomes a major bottleneck.

Finally, like any session, it must end at some point either by logging out or timing out. While we chose not to build a "logout" web services function, a web services session can time out due to inactivity, just like a browser session. However, whereas `CFAPPLICATION` does this for you automatically, we had to implement this capability ourselves.

To do this, we created a "session" variable that stores the last date/time the "user" submitted a web services request. This is used to determine whether the web services session should be "timed out" due to inactivity, which is checked with each request. We chose to time out sessions after 2 hours of inactivity. With each web services request, we first validate the UUID to ensure it is an active session and, if so, then compare the current date/time with the last request to determine whether the session has "timed out." To actually delete the timed-out sessions, we set up a scheduled script that deletes all web services "sessions", i.e., the rows in the database where the last request was more than 2 hours ago.

3. Redundant CFCs

While some developers prefer to store all components in a single directory, we chose to store each component in the same directory with its associated functionality. But this would require that customers refer to the web services components in a whole bunch of different directories. Of course, it would be more convenient if all components accessed via web services were in the same directory.

While it initially seems ludicrous to have redundant components, it was actually necessary anyway. The web services components have the same name, but with a "WS" prefix. Most web services functions in Averum Billing have the same function (method) name as their associated non-web services function.

For many reasons, the normal CFC function and the web services version of that function accept a different list of arguments. We will go into more detail later about some of the reasons for this, including custom IDs, searching, updating, and boolean fields. Of course, the most obvious example is the UUID that must be passed in to validate the session.

Another simple reason is that Averum Billing contains fields that are used internally, but which are not specified directly by the client. In general, our client is not even aware the fields exist. These fields exist as arguments in the normal CFC function, but not in the web services CFC function. While Java does allow you to have multiple functions with the same name but with different arguments, ColdFusion does not.

4. Application Scoped CFCs

Once we created our first web services component, a funny thing happened when we tested it. For simplicity, we used `CFINVOKE` to call the component just as we would any other. We assumed there was no difference when calling the function as a component versus calling it as a web service. In other words, we figured "`<CFINVOKE Component=`" worked the same as "`<CFINVOKE Webservice=`".

As you have probably guessed by now, this was not the case. Many of our components access functions in other components. In those instances, we simply used "`CFINVOKE Component=`" to access the other component. When calling a component that is not in the

same directory, the "`Component=`" value must start from the root public directory of your site, which requires setting up a mapping in the ColdFusion Administrator. This is fairly annoying, and we would have preferred that CFMX be more intelligent in this respect rather than implementing this in a normal Java fashion.

This method of calling a function in another component works fine if the component is called via "`<CFINVOKE Component=`" but, for some strange reason, doesn't work when called as a web service. Yes, it's the exact same function that is being called, but CF is evidently a bit more cranky when called as a web service. So we found that the only way to access a function in another component when called via a web service is to store that component in the Application scope.

It's a good idea to store commonly-used components in the Application scope for performance reasons. So we were considering doing this anyway, at least for the more heavily-used components. But the web services requirement did not give us much choice in the matter.

The only downside of storing components in the application scope is that we then had to remember to "update" the component value in the application scope whenever that CFC file is updated. To do this, we added a simple link in our admin interface that can reset these values when necessary.

The components that are called as web services are not stored in the application scope though because they are never called by other components. They are only called directly by clients when accessing a web services function. So storing them in the application scope provides no benefit.

On a random note, you will also find that if you update a web services CFC, ColdFusion will not recognize the changes in the updated CFC until you restart ColdFusion. It is possible it may recognize the changes after a certain amount of time, but we have not been willing to wait around long enough to find out.

We also use several functions that are written in `CFSCRIPT`. Many of these functions are also stored in the application scope since ColdFusion gets cranky about accessing functions via cached components. Before including the ColdFusion file that defines the function, you must first

check whether the function already exists. If it does exist and you try to define a function with the same name (even if it's the same function) ColdFusion returns an error. Doing a check in the file that defines the function is too late – ColdFusion returns an error even if the CFSCRIPT is within a CFIF statement that says to only execute the CFSCRIPT if the function does not exist.

A final note about storing components in the application scope (or any persistent scope actually). Any variables stored using the “Variables” scope in a persistent component are stored persistently as well. While the ColdFusion documentation does warn you about this, it did not occur to us. As you know, you must use the “var” command when creating a temporary variable in a component. The complication is when a web service mirrors the functionality of a non-component based browser feature, where a persistent Variables scope is not a problem. So while we initially thought it was good habit to use the Variables scope instead of leaving a variable un-scoped, it came back to haunt us as we had to de-scope many variables that were accessed via a web service. Plus we had to use “var” in our web services CFCs to create a temporary variable.

5. Custom IDs vs. Internal IDs

Now that we have dealt with the basics of implementing web services, we can deal with the specifics of the actual web services functions and how your customers will use them. Averum Billing has a custom ID field for each item so that our clients can specify their own primary key values rather than using the primary key assigned automatically by Averum Billing.

For instance, when a new product is created, the productID is assigned automatically by the database. This number generally has no meaning to our client since they have their own productID already for that product. By using the custom ID, a field we creatively named productID_custom, they can refer to the product in Averum Billing using the same ID they use in their own system. Whereas the productID is an integer field, the productID_custom is a varchar (text string) field so that clients can enter any value they want.

When specifying the productID via web services, our client can specify either our internal productID or their productID_custom. But the web services

function must know which ID they are specifying. There are several ways this can be achieved:

- a. Having both productID and productID_custom fields as arguments in the web services function, where productID is numeric and productID_custom is string. Then if the productID_custom is not blank and productID is 0, this means they are using the productID_custom value. However, this method is not explicit and the client could accidentally specify values for both fields, in which case it is unclear which field to use. Plus, in some functions, it may be possible to specify multiple productID's in a comma-delimited list, or multiple productID_custom's.
- b. To ensure it is explicit whether to use the productID or productID_custom, instead of having both productID and productID_custom fields, we can instead have a single productID field and then add an argument where the client can specify which version to use. We named this field “useCustomIDFieldList”. The default is to use the productID. To specify that the

productID_custom should be used instead, just enter “productID” for the “useCustomIDFieldList” value. If there are other arguments in the function for which the custom ID may be used, just enter that field name in useCustomIDFieldList, which is a comma-delimited list of fields for which the custom ID should be used. This method works great, except that it still is not explicit whether to use the productID or productID_custom field if there is only one. Plus if the productID argument is a string, that violates the web services concept of a self-defining function.

- c. The final method is to use the useCustomIDFieldList argument and also have both productID and productID_custom arguments. This is the most explicit method and maintains the field type integrity (except where the productID may be a comma-delimited list, in which case it must be of type string instead of numeric).

6. All Fields Are Required

One of the annoying things about web services is that all arguments are

Advertiser	URL	Phone	Page
CFDynamics	www.cfdynamics.com	866-233-9626	3
EV1 Servers	www.ev1servers.net	800-504-SURF	6
HostMySite.com	www.hostmysite.com/mxdj	877-248-4678	5
InterAKT	http://interaktonline.com/go/MXKollection/		11
Intermedia.net	www.intermedia.net	888-379-7729	21
Macromedia	www.macromedia.com/go/video5	415-252-2000	2
Macromedia WPS	www.macromedia.com/go/webupdate	415-252-2000	52
Macromedia Training	www.macromedia.com/certification	415-252-2000	25
MX Developer's Journal	www.sys-con.com/mx/subscription.cfm	888-303-5282	43
PaperThin	www.paperthin.com	800-940-3087	13
Seapine Software	www.seapine.com/webdev	888-683-6456	5
CFUNITED	www.cfunited.com	301-424-3903	19
ColdFusion Developer's Journal	www.sys-con.com/cfdj/subscription.cfm	888-303-5282	45
Web Services Edge 2005	www.sys-con.com/Edge2005	201-802-3066	31



“We should warn you that debugging your web services is more complicated than debugging normal ColdFusion code”

required, whether you need to specify a value for that argument or not. Unlike accessing normal functions in ColdFusion, the Required clause in the CFARGUMENT tag is ignored for web services. All arguments are required. Period.

While this does not seem like a big deal, it is actually quite annoying, especially if the web services function is for searching or updating. When searching, there are dozens of search criteria you might use to filter the results. Because all arguments are required though, you need a way to specify which arguments should be used for searching. In Averum Billing, we have solved this problem with an argument named “searchFieldList”. This is a comma-delimited list of the arguments to use for searching.

Similarly, when updating an item, for instance a product, you may want to update the product name but not the price. Again, you do not want to be forced to update all product fields if you only want to update the name. That would require you to know the current value of the product fields you do not want to update. But we certainly do not want to create a separate web services function for each product field so you can only update that field. Instead, we added an argument named “updateFieldList” which is a comma-delimited list of the fields to update.

A somewhat-related side effect of the required field problem is Averum Billing’s support for custom fields. Averum Billing allows clients to create their own custom fields for users, companies, products, etc. But it is not possible to submit arguments to web services that are not listed in the function. Java may support such “overloading” but ColdFusion does not. And it certainly violates the self-describing spirit of web services standards.

For customers to specify custom fields via web services, we instead have an argument named “customField” where they can enter the custom field values in an XML

string where <customField> is the main tag and the individual custom field tags are the custom fields names as defined when they created the custom field.

7. Returning Queries

Like all functions in CFCs, web services functions specify the field type that is returned. When the return type is a query, you need to consider that queries are treated differently in ColdFusion than in other languages. They are generally treated as a dual-index array whereas ColdFusion essentially treats queries as a structure/array combination. For instance, to access a particular row in a ColdFusion query, you can use:

```
queryName.fieldName[row]
```

But when the query is returned via web services to another language, instead of “fieldName” being the index, the index is a number. But the number must correspond to the field name, and the client requesting the query must know which index number corresponds to the field name. Fortunately, ColdFusion seems to be consistent in how the field names are ordered – they match the order in which the fields are listed in the SELECT clause of the query.

In your documentation, simply list the returned fields in the order in which they are selected. This provides a minor issue if the query returns internal fields that are not listed in the web services documentation. You cannot hide this fact since it is easy enough to determine the array length. So you either have to specify that the query not return these internal fields when the query is requested for web services, or just have some fun and let your client guess what the unlisted field is.

8. Boolean Fields

Boolean fields are generally specified as 0/1 or True/False. They presented

a minor challenge in our web services interface. Internally, our components used numeric field types for Boolean fields that are stored as bit fields in the database. We could have specified a numeric field type for our web services as well, but in keeping with the self-describing nature of web services, we preferred the argument type to accurately reflect the data type.

This presented two issues though. First, since our internal components and code use the numeric value, we created a simple function to convert the Boolean value to a 0 or 1. Second, there are instances where the Boolean value may be null, e.g., a nullable bit field. Well, it seems that null is not a valid Boolean value and the web service will return an error if you submit a blank value for a Boolean field. This is very annoying. In these instances, the data type for the “Boolean” argument must instead be a string, which can be blank.

On a side note, the MySQL database does not seem to support bit fields. So we used a tinyint field instead. The alternative was to use a varchar field of size 1, but we preferred to stick with a numeric field.

On a complete tangent, we also learned the hard way that in ColdFusion MX, if your query returns a bit field, ColdFusion will display the value as a 0 or 1. However, if you use ColdFusion’s ValueList function to return a list of all values in the query for a bit field, for some annoying reason, ColdFusion converts the 0’s to False’s and the 1’s to True’s.

9. Error Messages

When a user submits a form, you validate the field values and, if there are any mistakes, you display the error messages and re-display the form. This is slightly more difficult in web services, which are generally submitted by a server and thus no one is there to read the error messages. More importantly, each web ser-

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





vice function must specify the data type of the return value. If the function returns a numeric value or a query, clearly you cannot return an error message as a text string.

In .NET, I believe it is possible to create an exception where the returned value is not of the type specified. But trying to do this in ColdFusion will cause an exception error. (Unfortunately, ColdFusion also returns an error if you call a web services function without all of the arguments or if the argument value does not match the field type. This is incredibly annoying because there is no way to catch this error via CFTRY.)

There are two issues regarding how to handle error messages:

- 1) how to tell the client submitting the request that there was a problem with their request, but in a way that their server can understand in an automated manner;
- 2) when clients are initially setting up the web services on their end, how to let them access the error messages so they know why the request was unsuccessful.

For the server, our documentation lists the value returned if the request is not successful. For a numeric return type, we return -1. For a string, we return a blank value. For a date field, we return January 1, 1970 at 12:00 AM. For a query, we return a blank query with a single field named "error". (If the user does not have permission, we do not want to return a blank query with the actual field names. Granted they can get this from our documentation, but we still need to differentiate between an error and a blank results set.)

Now that the server knows the request was rejected, the programmer might want to know why. In the "session" variables for each web services login session, Averum Billing stores the last error message generated for that session. We could have stored the last error message generated for each web services function or perhaps the last 10 error messages, but we determined that would be a waste of storage.

There are many types of error messages, including:

- The user's web services session has timed out (or never existed)

- The user does not have permission for the requested function
- The client does not have permission for the requested item, e.g., the product may belong to a different Averum Billing client.
- Form validation issues

10. Debugging

Finally, on the topic of error messages, we should warn you that debugging your web services is more complicated than debugging normal ColdFusion code. The error messages generated by ColdFusion are often useless and misleading. Checking the ColdFusion error log may help, but it is generally the same non-descriptive error message.

There are several ways to debug your web services code. One way is to call the function as a normal component instead of a web service, but that's not always useful if the error is web services-specific.

Another method is to use CFTRY within the code where the CFCATCH clause includes a CFMAIL tag to email the error message to you or a CFFILE tag to write/append the error message to a text file. Of course, our experience seems to suggest that ColdFusion ignores the CFCATCH code unless it is within a file that is not included directly by the CFC itself, i.e., it must be at least 2 files down from the component.

Finally, like other code, you can walk through the code and comment out parts that may be causing errors, and then gradually un-comment the code to find the error. When doing this, just make sure there's still a return value specified for the function.

If you're truly stuck and are totally frustrated, we recommend inserting a bunch of CFFILE tags that append a number or other text to a file, where each successive CFFILE increments the number. This is the quickest way to get feedback about where the error is occurring. Unfortunately, this method does not work well if the error is in the CFC itself since ColdFusion evidently caches any functions with remote access and updates its cache only when ColdFusion is restarted. (This was mentioned in #4 above.)

One interesting fact we learned the hard way is about the function

QuerySetCell. Even if setting the value to an integer, when listing the values in a query using ValueList, ColdFusion returns the value with a single decimal point. The only way to resolve this was to set the value using the ToString function around the actual integer value. ValueList also converts bit fields from 0/1 to True/False.

We also ran into a problem with a query that was looped thru via CFLOOP Query="". The variables within the query were properly-scoped using "queryName.fieldName", but the query was returning junk data. Using ValueList confirmed that the query was correct. Oddly, the solution was to remove the "queryName:" in the CFLOOP.

Finally, for temporary values that will be used in your function, you should create the variable using "<CFSET var >". However, this does not apply to values that will be returned via a CFINVOKE tag when accessing another function. The "var" value will actually take precedence over the returned value, including for queries.

Summary and Disclaimer

This article was intended to help ColdFusion developers web-services enable their CFMX application. It recounts the issues encountered by Averum, Inc. while web services-enabling Averum Billing, Averum's billing system for Application Service Providers (ASPs).

We do not suggest the methods used by Averum are the only way or best way to implement web services throughout your application. However, the primary goal of this article was to make you aware of the various issues we encountered, share our solutions, and stimulate debate among the ColdFusion community about best practices for web services-enabling your application. ☁

Steven Rubenstein is founder and CEO of Averum, which develops billing solutions for ASPs and the first-ever product to reconcile credit card transactions with your bank account. He has been programming with ColdFusion since 1997. Before Averum, Steven founded an e-finance company and Emaze Software, which developed auction software. srubenstein@averum.com

The bible for
<CF_Developers>



ADVERTISE

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

SUBSCRIBE

www.sys-con.com/
cfdj/subscription.cfm
1 (888) 303-5282



COLD FUSION Developer's
Journal





One From the Vault: How I Rolled My Own Collision Detection

A helpful addition to your toolbox
by **andrew m. phelps**

Recently in a Shockwave3D project, I had some problems with the `modelsUnderRay` command. More precisely, `modelsUnderRay` seemed to have some problems of its own. Occasionally it just didn't give the correct result, it would miss a model somehow. In the particular application I was building (a landscape that let the user's drive a car around on the surface), this was dramatic because the when `modelsUnderRay` failed, the car fell through the world. At first I started to develop little stop-gap bits of code that ensured that the built-in functions would work. And then it occurred to me, why not just write my own?

So in my application, I was firing a ray 'down' from the position of the car, and intersecting the land, and then moving the car 'up' so that it would ride on the surface. I was also orienting the car to the normal of the triangle that it was on top of. This is shown in Figure 1.

So as you can see from the diagram, missing the landscape would have rather drastic consequences. Additionally, in doing it myself, I was able to get not only the normal from the original triangle, but the blended normal that weights the three disparate normal at the vertices of the triangle, which allows for a much smoother motion across the surface. The desired "smooth normal" is shown in Figure 2.

So I needed a way to fire a ray at the landscape, see where it hit, and get the resulting normal vector. The code for my collision detection routine is generalized to the code in Listing 1 (which unlike my terrain example uses ALL of the objects in the world, in the same way that `modelsUnderRay` does). Note also that to debug

this I used my 'Visual Debug Lines' that I presented in a previous *MXDJ* column (<http://mxdj.sys-con.com/read/47857.htm>).

The true "workhorse" of this routine is the `ghTriangleRayCollision` routine, that performs the magic of ray/triangle collision detection. This routine is based on an algorithm that I found at the SoftSurfer graphics site [see References, item #1], and that has been published several times in various places. A competing algorithm that could also be used would be the Akenine-Moller algorithm, which is presented in [see References, item #2]. The code for `ghTriangleRayCollision` is presented in Listing 2.

Finally, I wanted a way to average the normals of the triangle I hit, and get the normal of the exact point, or `texel`, within the triangle, weighted from the three triangle vertices. So I modified the code in Listing 1 to return not only the texture coordinates, but the three normals for the triangle that I hit. This is simply a matter of reading them out of the mesh in the same way the texture coordinates are read. Once I had the normals, then I knew the following information: the position of the points of the triangle (`aTriangleHit` in the returned array), the point of collision (`vHitPosition` in the returned array), and the normals at the vertices of the triangle (`aTriangleNormals`). With these three pieces of information I used a technique called Barycentric Coordinates to average the normals across the triangle. The Barycentric Coordinates are computed and returned as a vector of weights in the `ghGetBarycentricCoords3D` function in Code Listing 3.

The final application of the weights to the original vectors occurs in the `ghGet-`

`TexelValues` handler, that takes an array of values (the normals) and a vector that represents the Barycentric weights of the point. The code for `GetTexelValues` is shown in Listing 4.

Using these methods, I rolled my own collision detection that was far more accurate than the `modelsUnderRay` command that exists in the 3D Xtra. It will detect backfacing collisions, will not return a collision when the ray intersects a model's bounding sphere but not the model, and is generally more precise around the edges of individual faces. Additionally, by adding some averaging methods, I was able to more smoothly interpolate across the surface of the triangle, rather than have sharp transitions between the normals of two adjoining triangles. This made for a much smoother terrain-following algorithm, without the overhead of constantly interpolating between the last and current face. Hopefully you too will find these functions a helpful addition to your toolbox in developing your own 3D worlds.

References:

- Sunday, Dan. "Intersection of Rays, Segments, Planes and Triangles in 3D." The SoftSurfer Graphics Algorithms Collection. Available Online: http://www.softsurfer.com/Archive/algorithm_0105/algorithm_0105.htm
- Akenine-Moller, Thomas, and Eric Haines. Real-Time Rendering. 2nd Edition. A. K. Peters. Natick, Massachusetts, 2002. ☁

Andrew M. Phelps is in the Information Technology Department at the Rochester Institute of Technology in Rochester, NY (<http://andysgi.rit.edu/>). amp5315@rit.edu

figure 1

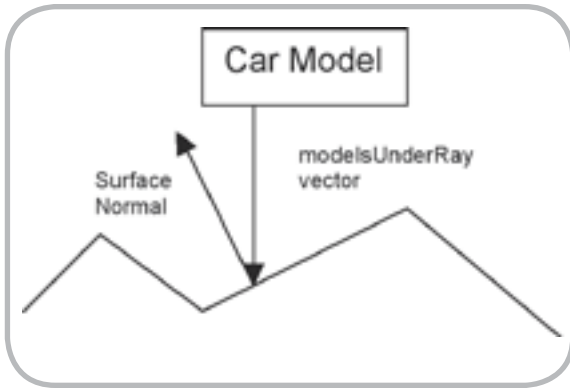
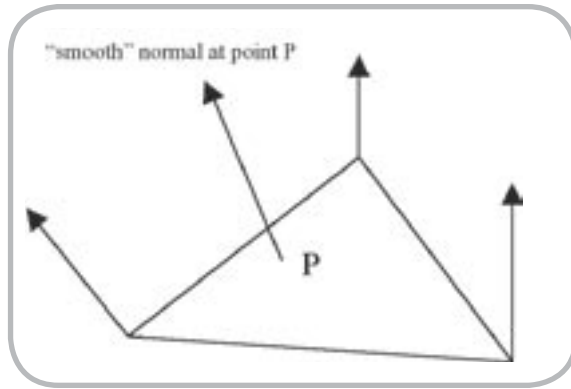


figure 2



listing 1

```

-----
--TraceRayOrSegment3D
-----
--Purpose: Traces a ray through the 3D world and returns
--either no collision, or a collision with an object.
--Generally used for pt to light or pt to reflected point
--calculations. Returns an array: the first element
--is an integer: 0=no collision, 1= collision, the second
--element of which is a vector describing the point hit,
--the third is an array of the triangle hit, the fourth is
--an array of texture coordinates at the hit location, and
--the final value is a reference to the model that was hit.
--
--a_3dWorld -- a 3D world in which to trace
--a_mObstacles --a list of models that are valid obstacles
--a_aOwnTri --the triangle the ray is cast from (ignored)
--a_vStartPos --vector starting position of the ray/segment
--a_vDir --vector direction of the ray/segment
--a_vEndPos --end point of the segment (use int 0 for ray)
--a_bQuickCheck --whether to use ModelsUnderRay for a quick
--look at whether or not we hit a model (ignored backfacing
--collision and uses bounding sphere -- non accurate !!!!)
--a_bReturnFirst --returns the first hit found if true,
--otherwise returns the hit guaranteed closest to the
--a_vStartPos vector in world space (a longer search)
--a_bVisualDebug --whether or not to use visual debug lines

--ONLY REQUIRED FOR DEBUG
--a_iX - X count of lumel in map
--a_iY - Y count of lumel in map
--a_ilc - which light pass this is
--a_shShader -- debug shader reference
--a_rgbColor1
--a_rgbColor2

--viscious HACKERY
--a_bReturnAll --return all hits along the vector

--returns:
--[TRUE/FALSE,vHitPosition,aTriangleHit,aTexCoords, mModel]
-----
on ghTraceRayOrSegment3D a_3dWorld, a_mObstacles, \
a_aOwnTri, a_vStartPos, a_vDir, a_vEndPos, a_bQuickCheck, \
a_bReturnFirst, a_bVisualDebug, a_iX, a_iY, a_ilc, \
a_shShader, a_mModel, a_rgbColor1, a_rgbColor2, \
a_bReturnAll

--use models under ray for a quick look at collision
if a_bQuickCheck then
  p1Options = [#maxNumberOfModels: 1, #levelOfDetail: \
    #simple ]

  aHitModels = a_3dWorld.modelsUnderRay(a_vStartPos, \
    a_vDir, p1Options)

  if aHitModels.count = 0 then return [false]
end if

--initialize flags
bHitAFace = false
vHitPosition = vector(0,0,0)
aHitList = []

--calculate ray end if non-segment
if a_vEndPos = 0 then a_vEndPos = a_vStartPos + \
  (a_vDir * _global.g_fRayLengthCap)

repeat with iObj = 1 to a_mObstacles.count
  repeat with iFace = 1 to a_mObstacles[iObj]\
    ].meshDeform.face.count

    --get information about this face
    aIndicies = a_mObstacles[iObj]\
      ].meshDeform.mesh[1\
        ].face[iFace]

    --get the verticies of this face
    vA = a_mObstacles[iObj]\
      ].meshDeform.mesh[1\
        ].vertexList[aIndicies[1]]

    vB = a_mObstacles[iObj]\
      ].meshDeform.mesh[1\
        ].vertexList[aIndicies[2]]

    vC = a_mObstacles[iObj]\
      ].meshDeform.mesh[1\
        ].vertexList[aIndicies[3]]

    --convert this triangle to world space
    vP1 = a_mObstacles[iObj].getWorldTransform() * vA
    vP2 = a_mObstacles[iObj].getWorldTransform() * vB
    vP3 = a_mObstacles[iObj].getWorldTransform() * vC

    aTri = [vP1, vP2, vP3]
    if aTri <> a_aOwnTri then --dont collide with own face

      --check for collision using custom maths
      aTriCol = ghTriangleRayCollision([a_vStartPos, \
        a_vEndPos], aTri)

      if (aTriCol[1] = 1) then --hit something
        --set flags and store this triangle
        bHitAFace = true
        vHitPosition = aTriCol[2]
        aTriangleHit = aTri

        --get texture infos from this triangle

```



```

aUV1 = a_mObstacles[iObj\
].meshDeform.mesh[1\
].textureLayer[1\
].textureCoordinateList[\
aIndicies[1]]

aUV2 = a_mObstacles[iObj\
].meshDeform.mesh[1\
].textureLayer[1\
].textureCoordinateList[\
aIndicies[2]]

aUV3 = a_mObstacles[iObj\
].meshDeform.mesh[1\
].textureLayer[1\
].textureCoordinateList[\
aIndicies[3]]

aTexCoords = [aUV1, aUV2, aUV3]

mModel = a_mObstacles[iObj]

--debug marker
if _global.g_bDebugMarkRayTriangleCollisions then
  put "HIT A TRIANGLE on model " & mModel.name
end if

if a_bReturnFirst = true then
  --we are about to return, so debug this path
  --if debugging is on
  if a_bVisualDebug then
    sName = "connector " & a_iX & a_iY & \
      a_iLC & iFace & a_mModel.name

    ghDebugRay(a_3dWorld, sName, a_rgbColor1, \
      a_rgbColor2, a_vStartPos, a_vEndPos, \
      a_shShader)

  end if

  --we hit something not ourselves, so return
  return [true, vHitPosition, aTriangleHit, \
    aTexCoords, mModel]
else

  --store this hit in the list of all hits
  --along this ray for later determination
  --of the closest hit
  aHitList.add([true, vHitPosition, \
    aTriangleHit, aTexCoords, mModel])

end if

else
  --self face intersection or blank pass
end if

end if

end repeat
end repeat

--debug this useless pass debugging is on
if a_bVisualDebug then
  sName = "connector " & a_iX & a_iY & a_iLC & iFace \
    & a_mModel.name

  ghDebugRay(a_3dWorld, sName, a_rgbColor1, a_rgbColor2, \
    a_vStartPos, a_vEndPos, a_shShader)

end if

```

```

--if we are storing all hits, calculate the closest one
--and return it

if a_bReturnFirst = false and a_bReturnAll = false then
  if aHitList.count > 1 then

    aClosest = aHitList[1]
    fDist = (a_vStartPos - aHitList[1][2]).magnitude

    repeat with iC = 2 to aHitList.count

      fNewDist = (a_vStartPos - \
        aHitList[iC][2]).magnitude

      if fNewDist < fDist then
        fDist = fNewDist
        aClosest = aHitList[iC]
      end if

    end repeat

    return aClosest

  else if aHitList.count = 1 then
    return aHitList[1]
  end if

end if

if a_bReturnAll = true then
  if aHitList.count >= 1 then

    --order the hitlist from closest
    --to farthest. total pain.

    aOrderedList = []

    repeat while aHitList.count > 0

      aClosest = aHitList[1]
      fDist = (a_vStartPos - aHitList[1][2]).magnitude
      iPos = 1

      repeat with iC = 2 to aHitList.count

        fNewDist = (a_vStartPos - \
          aHitList[iC][2]).magnitude

        if fNewDist < fDist then
          fDist = fNewDist
          iPos = iC
          aClosest = aHitList[iC]
        end if

      end repeat

      aOrderedList[aOrderedList.count+1] = aClosest
      aHitList.deleteAt(iPos)

    end repeat

    return aOrderedList
  else
    return [[false]]
  end if
end if

--return no collision
return [false]

end ghTraceRayOrSegment3D
-----

```



```

-----
--GetTriangleRayCollision
-----
--Purpose:
--a_aRay - an Array of 2 vectors, representing the start and
--end points of a ray in 3D space
--a_aTriangle - an array of vectors representing the
--points of the triangle in 3D world coordinates
--returns an array where the first element represents
--collision (1 if collide, 2, -1, or 0 if no collision) and
--the second element (if present) is the point of collision
--as a world-space vector
-----
on ghTriangleRayCollision a_aRay, a_aTriangle
--http://softsurfer.com/Archive/algorithm_0105/
--algorithm_0105.htm#intersect_RayTriangle()

--triangle vectors
vU = a_aTriangle[2] - a_aTriangle[1]
vV = a_aTriangle[3] - a_aTriangle[1]
vN = ghGet3DTriangleFaceNormal(a_aTriangle)

--test for degenerate triangle
if (vN = vector(0,0,0)) then return [-1]

--ray vectors
vDir = a_aRay[2] - a_aRay[1]
vW0 = a_aRay[1] - a_aTriangle[1]

fA = -vN.dot(vW0)
fB = vN.dot(vDir)
if (abs(fB) < 0.00001) then --ray is parallel to triangle
  if (fA = 0) then
    return [2] --ray is on the triangle plane
  else
    return [0] --ray is disjoint from plane
  end if
end if

--get intersection point of ray with triangle plane
fR = fA / fB
if (fR < 0.0) then return [0]
if (fR > 1.0) then return [0]

--intersection point of ray and plane
vI = a_aRay[1] + fR * vDir

--is I inside T ??
fUu = vU.dot(vU)
fUv = vU.dot(vV)
fVv = vV.dot(vV)

vW = vI - a_aTriangle[1]
fWu = vW.dot(vU)
fWv = vW.dot(vV)
fD = fUv * fUv - fUu * fVv

--get and test parametric coordinates (s,t)
fS = (fUv * fWv - fVv * fWu) / fD
--I is outside T ?
if (fS < 0.0) or (fS > 1.0) then return [0]

fT = (fUv * fWu - fUu * fWv) / fD
--I is outside T ?
if (fT < 0.0) or ((fS + fT) > 1.0) then return [0]

return [1, vI] -- I is in T

end ghTriangleRayCollision
-----

```

```

-----
--GetBarycentricCoordinates3D
-----
--Purpose: Solves for barycentric coords given a 3D triangle
--and a vector/point within that triangle. Returns a non-
--normalized vector
--a_aTriPts - array of 3 vectors that form the triangle
--a_vP - point that is inside the triangle on or its plane
-----
on ghGetBarycentricCoords3D a_aTriPts, a_vP

fU = a_aTriPts[2] - a_aTriPts[1]
fV = a_aTriPts[3] - a_aTriPts[1]
fW = a_vP - a_aTriPts[1]

fD = (power(fU.dot(fV), 2) - (fU.dot(fU))*(fV.dot(fV)))
fS = ((fU.dot(fV))*(fW.dot(fV)) - \
      (fV.dot(fV))*(fW.dot(fU))) / fD
fT = ((fU.dot(fV))*(fW.dot(fU)) - \
      (fU.dot(fU))*(fW.dot(fV))) / fD

fW = fS*fU + fT*fV --unique if P lies within the plane
fB0 = (1-fS-fT)
fB1 = fS
fB2 = fT

vBaryC = vector(fB0, fB1, fB2)
return vBaryC

end ghGetBarycentricCoords3D
-----

```

```

-----
--GetTexelValues
-----
--Purpose:
--a_aValues - array of 2D points, 3D points, or normals that
--are representative of values at the 3 points of a
--triangle.
--a_aWeights - barycentric coordinates of the point
--to solve for relative to vertex values
-----
on ghGetTexelValues a_aValues, a_aWeights

--compute X and Y averages based on barycentric coords
fR1 = (a_aWeights[1]*a_aValues[1][1]) + \
      (a_aWeights[2]*a_aValues[2][1]) + \
      (a_aWeights[3]*a_aValues[3][1])

fR2 = (a_aWeights[1]*a_aValues[1][2]) + \
      (a_aWeights[2]*a_aValues[2][2]) + \
      (a_aWeights[3]*a_aValues[3][2])

--if there is a third value, then calculate it also
if a_aValues[1].ilk() = #vector then
  fR3 = (a_aWeights[1]*a_aValues[1][3]) + \
        (a_aWeights[2]*a_aValues[2][3]) + \
        (a_aWeights[3]*a_aValues[3][3])

--return a vector if 3D triangle
vReturn = vector(fR1, fR2, fR3)
return vReturn
end if

--return 2D coord if 2D inputs
aReturn = [fR1, fR2]
return aReturn

end ghGetTexelValues
-----

```

Step into My Office

Our focus was to create a deeply immersive and visually stimulating Web site to represent AgencyNet as a cutting-edge interactive shop. After evaluating the objectives for our redesign, we determined that providing users with a first-hand look inside the day-to-day lives at our agency was of the utmost importance. We wanted to utilize the full capabilities of the medium and create an innovative and unique interactive experience that touched the users' senses.

I think we're all exceptionally proud of the site as a whole: the unique concept, the innovative creative and technical execution, the immersive visitor experience, and the fun and creative approach to content deployment. Most important, we achieved – actually exceeded – all of the objectives we set forth. www.agencynet.com





HostMySite.com

Built for ColdFusion Pros

by ColdFusion Pros

plans from

\$8.95 / mo.

FREE Domain Name*

FREE Setup

FREE 2 Months

- 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 for:

2 Months Free

and **FREE Setup on Any Hosting Plan***



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

James Kennedy
mbateam.com

*offer applies to any annual shared hosting plan

call
today!

877•248•HOST
(4678)





I am not the intranet

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

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

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