

# MX

developer's journal

volume 2 issue 10 2004 www.mxdj.com



THE LEADING MAGAZINE  
FOR MACROMEDIA MX  
DEVELOPERS & DESIGNERS

web services **EDGE**  
conference & expo

Web Services Edge  
**2005 East**  
February 15-17, 2005  
Hynes Convention Center  
Boston, MA

## component-based

# road

# Flash: the world of data integration

**DREAMWEAVER**  
The Relevant CSS Panel

**FIREWORKS**  
Fireworks Image Optimization Basics

**FREEHAND**  
Typography, Part 2

**COLDFUSION**  
ActionScript

**DIRECTOR**  
Video The Way Your "Ancestors" Intended

\$5.99US \$6.99CAN 1 1 >

0 71486 02978 6

macromedia®





This is:

Video

Flash

Don't bother me. I'm staring.

The line between Flash and Video is forever gone.

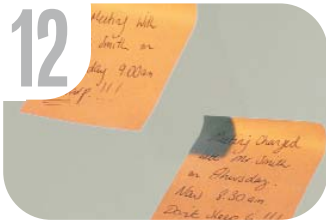
Travel with us to Scandinavia for a behind the scenes tour of the award-winning, genre-bending Volvo V50 website.

[macromedia.com/go/volvo](http://macromedia.com/go/volvo)





12



**The Relevant CSS Panel**  
*Fixing some problems*  
by dave mcfarland



22



**Component-Based RAD**  
*The best of two worlds*  
by art phillips



34



**Fireworks Image Optimization Basics**  
*Looking your best*  
by joyce j. evans



38



**Typography, Part 2**  
*Keeping a firm hand on your text*  
by ron rockwell



46



**ActionScript**  
*A Little OOP, CF Style*  
by john bland



54



**Video The Way Your "Ancestors" Intended**  
*Reuse and simplify*  
by chuck neal

7

**Where Are We Going?**

*A move to the future*

by charles e. brown

10

**The Need to Know**

*The Macromedia RSS Feeds*

by christian cantrell

16

**xile**

*Cartoon*

by louis f. cuffari

58

**vanguard**

*Inner Meaning*

by michele dauria

17

**Extension of the Month**

*Cartweaver 2.0*

by dave mcfarland

18

**A New Role for Flash**

*Hurray! Flex is here, but you still need your Flash developers.*

by peter ent

28

**Illustrating with Flash MX 2004**

*Book Excerpt: Game Techniques*

by robert firebaugh



# 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](http://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](http://www.seapine.com/webdev)

1-888-683-6456



## Dedicated Server Packages Starting at \$189 /mo.



## Real Freedom.

As one of the premier ColdFusion hosting community leaders, CFDynamics is constantly looking for ways to provide **better service** to ensure your satisfaction. Not only do we offer the **finest in shared hosting plans**, but we now offer the **finest in 100% dedicated server plans!** Now you can afford the freedom of having your own dedicated server!

When your needs have outgrown shared hosting look to CFDynamics for **total freedom**. With our dedicated server packages we're not offering an oxymoron; "virtually private" or "virtually dedicated" is NEITHER private nor dedicated. CFDynamics offers a solution that is **100% completely dedicated**. We don't play games with the fake stuff; we only offer the real deal. Real Service. Real Satisfaction. Real Value.

## Real Freedom.

All dedicated servers include:

- No long term commitments!
- Fair and simple pricing!
- FREE mail server access!
- FREE SQL server access!
- State-of-the-art data center!
- Optional server maintenance!

**Visit us online or call to order!**



Paper|Thin Partner





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

**EDITORIAL BOARD**

**Editor-in-Chief**  
Charles E. Brown [charles@sys-con.com](mailto:charles@sys-con.com)  
**Dreamweaver Editor**  
Dave McFarland  
**Flash Editor**  
Brian Eubanks  
**Fireworks Editor**  
Joyce J. Evans  
**FreeHand Editor**  
Ron Rockwell  
Louis F. Cuffari  
**ColdFusion Editor**  
Robert Diamond  
**Director Editor**  
Andrew Phelps

**INTERNATIONAL ADVISORY BOARD**

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

**EDITORIAL**

**Executive Editor**  
Gail Schultz, 201 802-3043  
[gail@sys-con.com](mailto:gail@sys-con.com)

**Editors**

Jamie Matusow, 201 802-3042  
[jamie@sys-con.com](mailto:jamie@sys-con.com)  
Nancy Valentine, 201 802-3044  
[nancy@sys-con.com](mailto:nancy@sys-con.com)

**Assistant Editor**

Natalie Charters, 201 802-3041  
[natalie@sys-con.com](mailto:natalie@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](mailto:subscribe@sys-con.com)  
U.S. Toll Free: 888 303-5282  
International: 201 802-3012  
Fax: 201 782-9600  
Cover Price U.S. \$5.99  
U.S. \$29.99 (12 issues/1 year)  
Canada/Mexico: \$49.99/year  
International: \$59.99/year  
Credit Card, U.S. Banks or Money Orders  
Back Issues: \$12/each

**Editorial and Advertising Offices**

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

**Worldwide Newsstand Distribution**

Curtis Circulation Company, New Milford, NJ

**List Rental Information**

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

**Promotional Reprints**

Kristin Kuhnle, 201 802-3026  
[kristin@sys-con.com](mailto:kristin@sys-con.com)

**Copyright © 2004**

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

# Where Are We Going?

*A move to the future*

by charles e. brown

As a Web developer, trainer, and conference speaker, the question I am asked most frequently is, "Where is this industry going?" My answer is one simple word: Flash!!!

In my opinion, client-side technologies, such as HTML and related scripts, have run their course and are relics of a past state of technological evolution. Even many of today's dynamic technologies are just trying to use these past concepts in brand new ways. However, the results are often far from satisfactory. If you think about it, the results of most of these technologies are just freshly written static HTML pages.

The emergence of Flash MX 2004 was a major leap forward in Web design. No longer is Flash a program for novelty and eye-catching animations; it is now, in my opinion, the most powerful and forward-looking Web design tool on the market today. I can say with assurance that Flash MX 2004 is only the beginning; As I look down the road I see a lot of exciting things happening over the next year.

This does not mean we should abandon existing technologies. As technology professionals, we must keep a firm eye on the future because that future could have a very real impact on how we develop today. We want to develop today's technological projects so they can easily adapt to that future. As author Tom Green recently said, "We must first define what we mean by the word dynamic." This is certainly a word that is new in transition.

It is in this light that I take over as editor-in-chief of this publication.

I see *MX Developer's Journal* as a powerful tool to help you achieve that goal. Therefore, you will be seeing more articles that focus on where we are going instead of rehashing where we have been. I also want to present more articles showing how many of these diverse technologies fit together.

Because of Flash's embrace of object-oriented concepts, we are now seeing the emergence of a potentially large third-party market. I will encourage our contributors to not only review these products, but to show you how to incorporate them into your present workflow.

Finally, it is my goal that *MXDJ* be a teaching magazine. I would like to develop coordinated articles that could serve as mini-courses. I will personally be starting a series that will discuss ActionScript 2 right from the beginning. Each month, I will build on the previous month's article.

Finally, this is not my magazine; this is not **SYS-CON's** magazine, this is not Macromedia's magazine: this is your magazine. You, the readers, are the ones on the front line and only you know what your needs are. To that end, my e-mail door is always open. Please feel free to contact me at [charles@sys-con.com](mailto:charles@sys-con.com) at any time to express your thoughts. You have my promise that I will listen.

With this edition, I would like to welcome three new editors to *MXDJ*.

Brian Eubanks will be our new Flash editor. Brian is the founder of Eu

“As technology professionals, we must keep a firm eye on the future”

Technologies, Inc., a consulting and training firm based in Northern Virginia. Eu Technologies currently provides Java, XML, and Flash consulting and training services to clients in the Mid-Atlantic region. Recent clients have included the New York Stock Exchange, government agencies, and public and private firms. Brian holds a master of science degree in computer science from George Mason University.

Our new Fireworks editor is Joyce J. Evans. Joyce is a training veteran with over 10 years of experience in educational teaching, tutorial development, and Web design. She has been asked to speak at conferences such as Macromedia MAX 2003 and TODCON. Joyce has received Editors Choice Awards for her book *Fireworks 4 F/X and Design* and has authored several computer books including *Dreamweaver MX Complete Course*, *Web Design Complete Course*, and *Fireworks MX: Zero to Hero*.

Joyce is a Team Macromedia Volunteer and her work is also featured in the Macromedia Design/Developer center and *MXD/* (Volume 2, issue 2). Her clients include Century 21, a prominent college in Florida, a sub-

contractor for a local service provider, along with actively teaching new students how to use the Macromedia Studio products.

Andrew M. Phelps will be heading up the Director section. Andrew is an assistant professor in the B. Thomas Golisano College of Computing and Information Sciences at the Rochester Institute of Technology in Rochester, NY (<http://andysgi.rit.edu>><http://andysgi.rit.edu>). He has an academic background in information technology, as well as in traditional fine arts and computer animation. His work using Director has been featured at the Director-Online User's Group (DOUG), as well as the DevNet Center at Macromedia. He regularly teaches coursework in multimedia programming, game programming, and simulation/visualization.

We live in an exciting time because we are about to see the realization of nearly 25 years of promise. I firmly believe that this journal will be your guide to that future.

I look forward to hearing from each and every one of you.

And now on to the future! ☺

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



**SYS-CON MEDIA**  
**President & CEO**  
 Fuat Kircaali, 201 802-3001  
[fuat@sys-con.com](mailto:fuat@sys-con.com)  
**Vice President, Business Development**  
 Grisha Davida, 201 802-3004  
[grisha@sys-con.com](mailto:grisha@sys-con.com)  
**Group Publisher**  
 Jeremy Geelan, 201 802-3040  
[jeremy@sys-con.com](mailto:jeremy@sys-con.com)

**ADVERTISING**  
**Senior Vice President, Sales & Marketing**  
 Carmen Gonzalez, 201 802-3021  
[carmen@sys-con.com](mailto:carmen@sys-con.com)  
**Vice President, Sales & Marketing**  
 Miles Silverman, 201 802-3029  
[miles@sys-con.com](mailto:miles@sys-con.com)  
**Advertising Sales Director**  
 Robyn Forma, 201 802-3022  
[robyn@sys-con.com](mailto:robyn@sys-con.com)  
**Advertising Sales Manager**  
 Megan Mussa, 201 802-3023  
[megan@sys-con.com](mailto:megan@sys-con.com)  
**Associate Sales Managers**  
 Kristin Kuhnle, 201 802-3026  
[kristin@sys-con.com](mailto:kristin@sys-con.com)  
 Dorothy Gil, 201 802-3024  
[dorothy@sys-con.com](mailto:dorothy@sys-con.com)

**PRODUCTION**  
**Production Consultant**  
 Jim Morgan, 201 802-3033  
[jim@sys-con.com](mailto:jim@sys-con.com)  
**Lead Designer**  
 Louis F. Cuffari, 201 802-3035  
[louis@sys-con.com](mailto:louis@sys-con.com)  
**Art Director**  
 Alex Botero, 201 802-3031  
[alex@sys-con.com](mailto:alex@sys-con.com)  
**Associate Art Director**  
 Richard Silverberg, 201 802-3036  
[richards@sys-con.com](mailto:richards@sys-con.com)  
**Assistant Art Director**  
 Tami Beatty, 201 802-3038  
[tami@sys-con.com](mailto:tami@sys-con.com)  
 Andrea Boden, 201 802-3034  
[andrea@sys-con.com](mailto:andrea@sys-con.com)

**SYS-CON.COM**  
**Vice President, Information Systems**  
 Robert Diamond, 201 802-3051  
[robert@sys-con.com](mailto:robert@sys-con.com)  
**Web Designers**  
 Stephen Kilmurray, 201 802-3053  
[stephen@sys-con.com](mailto:stephen@sys-con.com)  
 Matthew Pollotta, 201 802-3054  
[matthew@sys-con.com](mailto:matthew@sys-con.com)  
**Online Editor**  
 Martin Wezdecki, 201 802-3045  
[martin@sys-con.com](mailto:martin@sys-con.com)

**ACCOUNTING**  
**Financial Analyst**  
 Joan LaRose, 201 802-3081  
[joan@sys-con.com](mailto:joan@sys-con.com)  
**Accounts Payable**  
 Betty White, 201 802-3002  
[betty@sys-con.com](mailto:betty@sys-con.com)  
**Accounts Receivable**  
 Shannon Rymysza, 201 802-3082  
[shannon@sys-con.com](mailto:shannon@sys-con.com)

**EVENTS**  
**President, SYS-CON Events**  
 Grisha Davida, 201 802-3004  
[grisha@sys-con.com](mailto:grisha@sys-con.com)

**CUSTOMER RELATIONS**  
**Circulation Service Coordinators**  
 Shelia Dickerson, 201 802-3082  
[shelia@sys-con.com](mailto:shelia@sys-con.com)  
 Edna Earle Russell, 201 802-3081  
[edna@sys-con.com](mailto:edna@sys-con.com)  
 Linda Lipton, 201 802-3012  
[linda@sys-con.com](mailto:linda@sys-con.com)  
**JDJ Store Manager**  
 Brundila Staropoli, 201 802-3000  
[bruni@sys-con.com](mailto:bruni@sys-con.com)





# Need a hand with your work? Get the MX Kollection.

**Introducing MX Kollection** - the suite of extensions designed to change the way you create dynamic web applications with Dreamweaver MX and MX 2004.

Our customers think of MX Kollection as the next level in Dreamweaver MX visual software development.

ColdFusion and PHP developers will find our product enabling them to visually develop e-Commerce, CMS, CRM, Backend and other web solutions with increased efficiency, quality and capability.

#### Create powerful dynamic lists

- Sort, filter and page result sets
- Perform multiple deletes
- Easily create Master/Detail lists

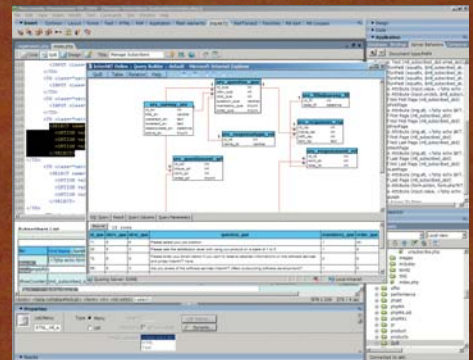
#### Build unified add/modify forms

- Client side and server side validation
- Insert into two tables
- Create form actions such as send mail or delete related record
- File/Image upload and resize

#### Create recordsets visually

- Build complex SQL queries across multiple tables quickly

... and many more



**Interakt**

macromedia  
ALLIANCE PARTNER

Download the demo and see the features and benefits of our extensions at <http://www.interaktonline.com>



# The Need to Know

*The Macromedia RSS Feeds*

by christian cantrell

Need an easy way to keep your favorite software applications up to date? Ever miss out on a newly released update patch, an important TechNote, or an urgent security bulletin? How do you know when Macromedia has published new Developer Center content that might show you some new timesaving techniques, or introduce you to an important emerging technology?

Staying on top of the latest official development news is not easy. And what about all the hundreds of interesting weblogs out there? In my opinion, there are two ways to stay current with the latest information: the easy way, and the hard way.

## Staying Informed...the Hard Way

The hard way is to create bookmarks for all the different content you are interested in and set aside time to visit dozens of different locations online everyday. Going out and gathering information manually in this fashion is not particularly efficient for several reasons:

- **Downloading takes time.** Even over

the fastest of connections, you still have to wait a fair amount of time for content to download, and the content that takes the longest to download (ads, graphics, etc.) is usually the content you are least interested in seeing.

- **You have to figure out each new interface.** Each of the dozens of sites I keep up with has a dramatically different interface with very different navigation. Although most of us are pretty accustomed to the inconsistency of the Internet by now, the time it takes to adapt to new and often very congested interfaces (as product and news sites tend to be) does add up.
- **Other tasks take priority.** I don't know about you, but when I'm busy, making my daily rounds to news sites and weblogs is often the first thing to go on the back burner, and, all too often, dropped entirely.
- **Once it's gone, it's gone.** When you miss an important headline out there, for all intents and purposes it's usually pretty much gone. How often do you search through the archives of a news site, trying to catch up on articles you might have missed? There's so much new information out there on an

hourly and daily basis; who has time to review the old stuff?

## Staying Informed...the Smart Way

Fortunately, the smart way to gather news and information addresses all of these issues. Rather than having to go out and get news and information yourself, why not let it come to you? Wouldn't it be nice to have all the most important product information and news headlines right on your desktop waiting for you, or all on a single Web page? With RSS (and a good RSS aggregator), that's exactly what you get.

RSS is an XML data syndication format. There's some disagreement as to what RSS actually stands for – whether it's "Really Simple Syndication," "Rich Site Summary," or "RDF Site Summary" – and there are enough versions of RSS and similar data formats like Atom out there that you may not care to get into the specifics of the data format itself. The important thing is what it can do for you.

The RSS data syndication format lets sites syndicate content in such a way that other applications can easily pick it up, parse it, reorganize it, and republish it in just about any format. What that means to you is that rather than spending time going out and gathering information yourself, an application gathers it for you. When you have time to read through it, it's all right there – waiting for you, all in one place, and in a nice, consistent format.

Not surprisingly, the advantages of RSS are basically the opposite of the disadvantages listed above:

1. It's fast. RSS is a very lean XML format, so even over slow connections huge amounts of data can be collected very quickly.
2. You can choose an interface that's

“All the information included in our product feeds is purely technical or educational in nature”

# MXDJ Section Editors

## Dreamweaver Dave McFarland

*Author of Dreamweaver MX 2004: The Missing Manual, Dave can be relied upon to bring Dreamweaver MX to life for MXDJ readers with clarity, authority, and good humor.*



## Flash Brian Eubanks

*Brian Eubanks is the founder of Eu Technologies, Inc., a consulting and training firm based in Northern Virginia. Eu Technologies currently provides Java and XML consulting and training services to clients in the Mid-Atlantic region. Recent clients have included the New York Stock Exchange, government agencies, and public and private firms.*



## Fireworks Joyce J. Evans

*Joyce J. Evans is a training veteran with over 10 years of experience in educational teaching, tutorial development, and Web design. She has presented at conferences such as Macromedia MAX 2003 and TODCON. Joyce has authored books including Macromedia Studio MX 2004 Bible, Dreamweaver MX 2004 Complete Course, and others. Joyce is a Team Macromedia volunteer and her work is also featured in the Macromedia Design/Developer center. Her Web site is [www.JoyceJEvans.com](http://www.JoyceJEvans.com).*



## FreeHand Ron Rockwell

*Illustrator, designer, author, and Team Macromedia member, Ron Rockwell lives and works with his wife, Yvonne, in the Pocono Mountains of Pennsylvania. Ron is MXDJ's FreeHand editor and the author of FreeHand 10 f/x & Design, and coauthor of Studio MX Bible and the Digital Photography Bible. He has Web sites at [www.nidus-corp.com](http://www.nidus-corp.com) and [www.brainstormer.org](http://www.brainstormer.org).*



## Louis F. Cuffari

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

## ColdFusion Robert Diamond

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



## Director Andrew Phelps

*Andrew Phelps is an assistant professor in the B. Thomas Golisano College of Computing and Information Sciences at the Rochester Institute of Technology in Rochester, NY (<http://andysgi.rit.edu>/<http://andysgi.rit.edu>). He has an academic background in information technology, as well as traditional fine arts and computer animation. His work using Director has been featured at the Director-Online User's Group (DOUG) as well as the DevNet Center at Macromedia. He regularly teaches coursework in multimedia programming, game programming, and simulation/visualization.*



right for you. RSS aggregators present data from many disparate sources in a single, consistent user interface.

3. You don't have to remember, or make the time, to go out and gather the information manually. True, you still have to make the time to read it. (I'm not aware of any technology that can dump information directly into your brain just yet.) But with all the news and information sitting right there in front of you in a single application, you can read it in the time that you would have spent simply gathering it from various sources.
4. It's easy to find "old" information. Most aggregators archive information so if you miss a day or two, you can easily scan through "old" headlines and see if you missed anything important.

## About Aggregators

Generally speaking, there are two types of aggregators: online or Web-based, and local or desktop. Online aggregators are Web applications that gather information from all over the Internet and present it in a Web interface, while local aggregators present aggregated information for you right on your desktop. There are even various plug-ins available that will allow you to add RSS capabilities right to your browser; according to Apple's Web site, the next version of Safari will actually have sophisticated RSS functionality built right into it. There are advantages and disadvantages to both types of aggregators, so which you use is completely a matter of personal preference. I encourage you to try both types and see which you are more comfortable with.

News sites and weblogs were the first to start syndicating data through RSS, but now there are RSS feeds available that contain all kinds of information from real-time earthquake statistics to forum threads to MP3 music catalogs. At this point, RSS is being used to syndicate just about any kind of information you can imagine.

## Getting Started with the Macromedia Product RSS Feeds

Macromedia has been tapping into the power and convenience of RSS and weblogs in various ways for about two

years now. However, we have recently taken our investment to the next level with the new Macromedia Product RSS Feeds. We have been looking at ways in which we can communicate with our customers more efficiently and conveniently, and help them reduce the amount of time they spend looking for information on our Web site. RSS is a natural fit. Rather than requiring customers to visit [macromedia.com](http://macromedia.com) on a regular basis to look for product updates, security bulletins, and TechNotes, we decided to use RSS to syndicate all this information for almost all of our products. And we include relevant Developer Center content in the feeds as well.

The Macromedia Product RSS Feeds are product specific, which means you can subscribe to just those feeds you are interested in. We are currently covering 18 different products and will continue to add feeds as we see the need. We update each feed within an hour of the release of product information on [macromedia.com](http://macromedia.com), so the information is pretty close to real time. All the information included in our product feeds is purely technical or educational in nature as well – we never syndicate marketing messaging or product offers.

If you use any Macromedia products on a regular basis, I highly recommend that you subscribe to their corresponding product RSS feeds using the online or desktop aggregator of your choice. The Macromedia Product RSS Feed page contains additional detailed information on RSS, the URLs to all the product RSS feeds, and information on how to get started aggregating feeds quickly and easily. Once you get into RSS, however, be careful, as it can be extremely addictive. I'm currently aggregating well over 200 feeds, and I seem to add more each week. Thanks to the efficiency of RSS, however, I actually have enough time to read them. ☁

*Christian Cantrell is the Macromedia Server Community Manager. He has been developing large-scale, Web-based applications in ColdFusion, Java, JSP, and Macromedia Flash for the last five years. Christin is the author of numerous tutorials and whitepapers, and is coauthor of Flash Enabled: Flash Design & Development for Devices.*

d

f

tu

th

f

sb

See at film house  
- 17 Diner (meal)  
- Shopping Centre  
- Business App  
- Charles Ad  
- See the dog  
- See the cat

Jobs at Parkway  
- Repair Solder  
- 477 (value Oak)  
- El. Appt  
- Memo

AMM

64 58720

Presentations in  
"Bestmiller"

John is here

What's  
Monday  
3:30 pm

Woman's  
at 10 am

Shedding hair  
your  
fingers and blood  
drop.

Shopping list  
1. Milk  
2. Bread  
3. Ham  
4. Cheese  
5. TV Guide  
6. Wine

Don't forget  
to hand in your  
resignation on  
Friday.

Mr. Science  
Monday

Meeting with  
Mr. Smith on  
Thursday 9:00am  
Sharp!!!

Meeting changed  
with Mr. Smith  
on Thursday  
Now 8:30 am  
Don't sleep in!!!

Dinner at  
Motel  
right

Tuesday lunch  
Fields Account  
Due

1 car lease  
\$50.00 for Rent  
Saturday night  
24/5  
Ohio Saturday  
Night Double or  
nothing

Call Mum  
birthday

Call Mum  
birthday

Command &  
copy

Command &  
paste

Command &  
print

Pick up lottery  
ticket - Milton &  
Draw

Send notes  
to bank each  
before 4pm  
Complete

October/November  
holidays Book  
April 30th  
- Rail cards  
- 4 weeks  
- Wages

By 6:30pm  
means  
for each  
would like  
something  
for the  
parents



# The Relevant

# CSS Panel

by dave mcfarland

## *Fixing some of your CSS problems*

**C**ascading Style Sheets (CSS) can be confusing. Not only do Web developers need to know the different CSS properties, and the sometimes bizarre ways that Web browsers render those properties, there are times when CSS just doesn't seem to behave. You create a new class style – `.highlight`, for example – that's supposed to change the text color to burgundy. But when you select the text and apply the style nothing happens.

As you build more complex sites, with hundreds of styles, this kind of problem seems to happen more frequently. In most cases, the CSS is behaving just as it's supposed to – the problem is that any given style sheet will often have styles that conflict in one or more ways. Ferreting out these conflicts can be a chore; fortunately, Dreamweaver MX 2004 introduced a new tool – the Relevant CSS panel – to help solve these dilemmas.

### **When Styles Collide: Understanding CSS Conflicts**

One of the fundamental features of CSS is the way styles can pass from one element to another. This basic principle, known as inheritance, is a powerful way to simplify formatting tasks. For example, in the

bad old days, when we used the `<font>` tag to format text, we had to wrap every piece of text with the `<font>` tag. Even if you just wanted to use Arial for all of the text on your page – headings, paragraphs, lists – you'd have to use multiple `<font>` tags. This significantly increased the file size of a Web page and was a lot of work (well not quite so much with Dreamweaver).

CSS and inheritance makes this same task a snap. With CSS, child elements inherit properties of their parent elements; so, to apply a single font family to a page, you merely assign a font family to the `<body>` tag. Other elements on the page, such as headings and paragraphs, inherit this property, so they too use the same font.

Because of inheritance, we can set some global properties for our page by merely styling the `<body>` tag. Those properties are then inherited by child elements (well, not all properties are inherited, as you'll see below). Fortunately, you can override inheritance by creating styles that apply in specific instances. For example, say in general you want text on the page to be bright blue and use the Arial font. But in some cases, you'll want to override this – for example, all headlines should be fire engine red, and all bulleted lists should be in Times New Roman. Create a style for the `<body>` tag, and other styles for the `<h1>` and `<ul>` tags.

Not only is inheritance at work, but also another feature of CSS – the cascade. The cascade governs how interacting styles work. So in the example above, the `<h1>` tag would inherit the font face from the `<body>` tag, but would overrule the `<body>` tag's font color. It's CSS ability to mix styles and properties that gives pages a great deal of formatting finesse,

with very little code. But, the cascade is also the source of many conflicts that often seem bewildering.

Here's a simple example that demonstrates the kind of problems that frequently pop up. Say you created a `<div>` with an ID style of `#content` – perhaps this `<div>` is used to lay out the main story on a page. You want all the paragraphs inside that div to be both bold and red, so you create a style `#content p` – this is a descendent selector that merely says any `<p>` tag inside an element with the ID of content should be bold and red (see Figure 1). Because you want a single paragraph in that div to be blue and not bolded, you create a class style called `.special`. But when you apply it to a paragraph in the div nothing happens. What gives?

### Resolving the Cascade: Understanding Specificity

When a single element is affected by multiple CSS styles, CSS provides rules for dealing with these conflicts. Each CSS style has what's known as "specificity" – a style with a higher specificity overrules a style with lower specificity. As you know, there are different types of styles – tag styles, class styles, and ID styles. When you define a style for a particular HTML tag, like the `<body>` tag, that's a tag style. Class styles, which are preceded by a period – `.pullQuote`, for example – are applied manually, for example using the Property Inspector or the Text menu's CSS styles submenu. IDs are similar to class styles; but whereas a page can have mul-

iple elements each with the same class applied to it, an ID style should only be applied once per page. IDs are preceded by the `#` symbol – `#banner`, for example – and are frequently used to indicate logical divisions in a page, such as a banner, sidebar, or footer.

The type of style you use greatly affects that style's specificity. Here's a simple method for determining a style's specificity. If it's a tag style, the style has a specificity of 1; a class style gets a specificity of 10; and an ID style has a specificity of 100. Say you created a Web page with three styles: a tag style that defines the text color for the `<p>` tag as green; a class style, named `.blue`, with a text color of blue; and `#red`, an ID style with a red text color. All paragraphs on the page will be green. But if you applied the `.blue` style to one of those paragraphs, its text would be blue – since a class style has greater specificity (10) than a tag style (1). If you then applied the ID `#red` to that same paragraph (probably not something you'd normally do), that text will be red – since ID styles trump class styles (100 v. 10).

In this case, the style name – called a "selector" in CSS speak – is actually composed of a tag name and an ID. So say you create an ID style of `#red`, another style with the selector of `p#red`, and then create a paragraph with an ID of red (`<p id="red">`). There are two styles in your style sheet, `#red` and `p#red`. What happens if a rule in one style conflicts with a rule in the other? In this case `p#red` wins, because it has both an element (`p` with a

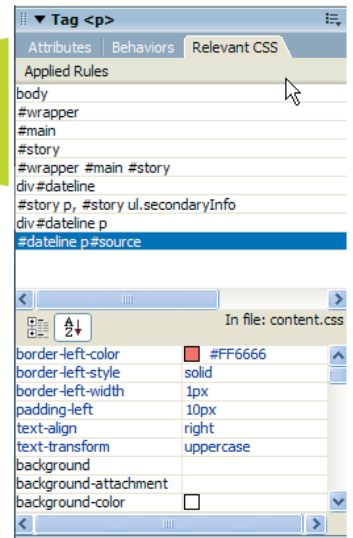


figure 2

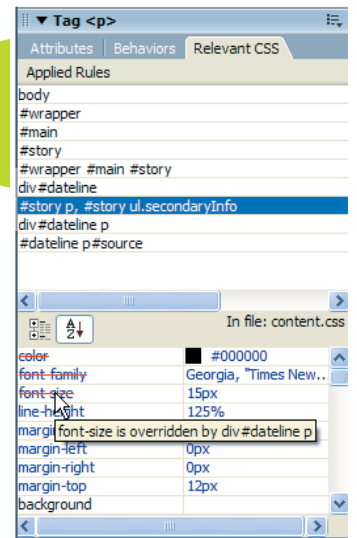


figure 3

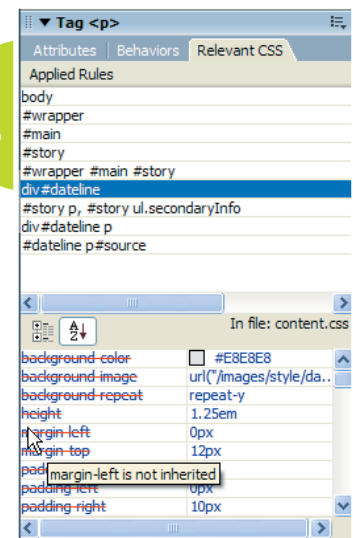


figure 4

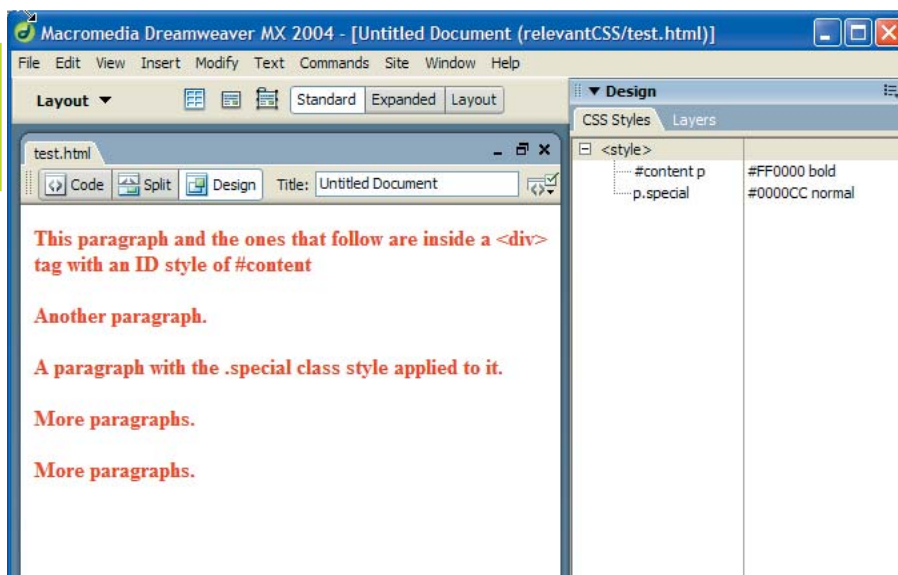


figure 1



# THE GRASS REALLY IS GREENER ON THE **SERVERSIDE.**

## SUPERIOR MANAGED HOSTING

- ↘ Intelligent Routing
- ↘ Redundancy
- ↘ Network Security
- ↘ Service Level Agreement
- ↘ Environmental Controls
- ↘ Network Uptime Guarantee
- ↘ Scalability
- ↘ OC3/SONET Backbone
- ↘ Backup Power
- ↘ Physical Security
- ↘ Fire Protection
- ↘ Server Hardware Guarantee

**IF YOU'VE BEEN SEARCHING** for a reliable managed hosting partner, your search for greener pastures is over. There is no longer a need to settle for inferior support and lack of accountability. ServerSide takes the guess work, and the associated hassles, out of working with a technology partner.

ServerSide provides managed web hosting solutions for a wide range of customers including; Fortune 500 corporations, small and medium sized businesses and non-profit organizations located across the United States and around the World.

We provide a single point of accountability for all your web hosting infrastructure needs — while adding value, not cost.

## >> SPECIAL OFFER

Sign up online for a new shared hosting account now at [www.serverside.net](http://www.serverside.net) and ServerSide will waive the set-up fee.

ENTER CODE: mxd2004

The grass is greener at  
[www.serverside.net](http://www.serverside.net)  
888.682.2544  
[hosting@serverside.net](mailto:hosting@serverside.net)



d

f

fw

fb

cf

cb

specificity of 1) and an ID (#red with a specificity of 100) – in other words, 101 vs. 100.

This really comes into play when you begin to use descendent selectors (see Vol. 2, Issue 4). A descendent selector can narrow how a style is applied by defining the context of an element. For example, in Figure 1 the #content p style is a descendent selector identifying only <p> tags that also appear inside an element with the ID of content. In this example, that style has a specificity of 101 (an ID plus an element). Now it's easy to see why, in the earlier example, applying the class style of .special to one of the paragraphs inside the #content div has no effect. Because the selector p.special is composed of one element (p) and one class (.special), the specificity is just 11. Since 101 is more than 11, the #content p style wins. To fix this situation, you could just rewrite the p.special as #content p.special – now that style has a specificity of 111.

Thankfully, Dreamweaver has a useful diagnostic tool to help solve confusing CSS conflicts like this.

### The Relevant CSS Styles Panel: A Great Friend

The Relevant CSS Styles panel (see

Figure 2) displays all styles that apply to a particular selection (and lets you edit those styles using the property list in the bottom half of the panel). In other words, the list of relevant styles indicates all styles from which the current selection might inherit formatting rules. For a complex design with lots of styles this list can become quite long. Fortunately, Dreamweaver makes it easy to determine when there's a conflict and which rule has the greatest specificity.

The order in which the styles appear in the Relevant CSS panel indicates the effect of styles on the current selection – the last style listed has greatest specificity. For example, in Figure 2 the style #dateline p#source has the greatest specificity, so its properties will overrule any conflicts in any of the other styles that appear in the list.

Dreamweaver also makes clear when there's a conflict between styles and why. If you select a style in the Relevant CSS styles panel and see a red line through a CSS property name, that property is not inherited by the current selection. This happens for two reasons. The first is that the property is overruled by a style with the same property but greater specificity. For example, in Figure 3 a red line through the font-

size property indicates that the current selection isn't inheriting font size from this style. Dreamweaver helps us out even more by letting us know why the conflict exists: hovering your mouse over the red line produces a tool-tip box with an explanation. In this example, Dreamweaver indicates that the style div#dateline p overrides the font-size property of #story p.

You'll also encounter that red line even if there isn't a style sheet conflict. Some CSS properties aren't inherited at all. In these cases, you'll see a red line through those properties even if another style doesn't contradict them (see Figure 4). Margin, padding, background, height, width, and positioning properties aren't inherited. And this is a good thing; after all, if the height property was inherited you'd have quite a mess. For example, say you created a style that set a div to 200 pixels high. If every element inside that div inherited the div's height, you'd end up with 200-pixel tall paragraphs, 200-pixel tall graphics, 200-pixel tall headlines, and so on.

You can also use the Relevant CSS Panel to figure out other styling problems. For example, say you add a paragraph to your page and see that the text is all blue.

# Let Us Help You Reach Your Goals

SPECIAL OFFER!  
>30% OFF  
ANY MACROMEDIA COURSE  
WHEN YOU MENTION THIS AD

Get the Macromedia training you need with SunTech 3 courses

### Macromedia Authorized Training from SunTech 3

Providing high quality, motivating and effective interactive multimedia training. SunTech 3's instructors are Macromedia Certified. As a Macromedia Authorized Training Partner, SunTech 3 offers a full range of courses that will increase your productivity and allow you to master all aspects of Web design and development. We can help you get your Macromedia Professional Certification that will make you more valuable in the workplace.

SunTech 3 Courses Include:  
Authorware | Director | Dreamweaver | Flash | ColdFusion | Fireworks

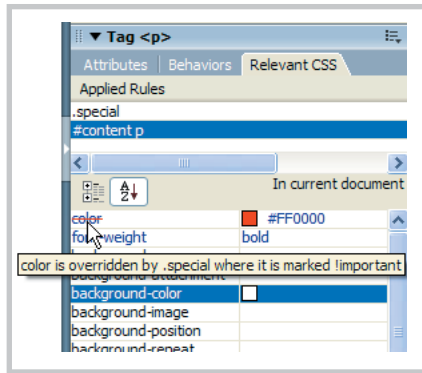
For more information call 407-862-1144



www.SunTech3.com

©2004 SUNTECH 3 INC. ALL RIGHTS RESERVED. ALL PRODUCT NAMES ARE REGISTERED TRADEMARKS OF MACROMEDIA, INC.

macromedia®  
AUTHORIZED TRAINING  
PARTNER 2004



You're not sure why it's happening, since you didn't create a paragraph style with the color blue. By selecting each of the styles in the list – i.e., each style that might be passing on its properties to the selection – you can locate which style is dictating the color and then edit that rule or create a new style, with greater specificity, to overrule the blue color.

## Other Ways to Win the Style Wars

There are some other ways to enforce a style even if there is a conflict. For example, say you wanted to be able to create a class – let's say `.special` – that will make all text it's applied to blue. As we saw above, just applying that class to an element won't ensure you'll get blue text – another style like `#content p` might overrule it. And while creating a style with more specificity like `#content p.special` will work, it won't work for `<p>` tags with that class that aren't inside the `#content` div. Nor will it work for other tags, like headings or bulleted lists.

A quick solution is to invoke the CSS `!important` declaration. You can add `!important` after any rule in a style to give it precedence over any other rules. Unfortunately, Dreamweaver doesn't have a menu-driven way to do this; you just have to open the style sheet in code view and add it by hand (*tip*: the quick way to open a style sheet to a style is to double-click the style name in the Relevant CSS Styles panel or the regular old CSS Styles panel). For this example, you could edit the `.special` class style like this:

```
.special {
  color: #0000CC !important;
}
```

Now you could apply this class to any element, and even when other styles with greater specificity conflict this style will win out. The Relevant CSS style panel also recognizes the `!important` declaration. For example, in Figure 5, even though the list of styles in the panel indicate that `#content p` has greater specificity than `.special`, the red line through the color property indicates that the `!important` declaration wins out.

## Summary

Inheritance, cascade, and specificity are all things that make CSS an incredibly powerful tool – simplifying style sheets by reducing the number of rules needed to format your page. But as CSS becomes even more important in how we format our pages – it's not uncommon to find sites utilizing multiple style sheets with many rules – the number of confusing interactions between styles is bound to increase. Fortunately, the Relevant CSS panel can help.

...

For more information on inheritance, the cascade, and specificity visit, [www.w3.org/TR/REC-CSS2/cascade.html](http://www.w3.org/TR/REC-CSS2/cascade.html) ☞

Dave McFarland is the Dreamweaver editor of *MX Developer's Journal* and author of *Dreamweaver MX 2004: The Missing Manual*. Find out about his latest projects at [www.sawmac.com](http://www.sawmac.com). [DaveMcFarland@sys-con.com](mailto:DaveMcFarland@sys-con.com)

# Extension of the Month

by dave mcfarland

## Cartweaver 2.0

**Extension Developer:** Application Dynamics, Inc.  
**Developer Web Site:** [www.cartweaver.com](http://www.cartweaver.com)  
**Price:** \$225.00

Building a shopping cart system from scratch can take a seasoned programmer months of hard work. But if you use Dreamweaver to develop ColdFusion sites, you can have a full-featured e-commerce solution up and running for less than you'd pay a programmer for a few hours of his time. Cartweaver 2.0 is a complete e-commerce solution for ColdFusion, providing catalog, shopping cart, checkout, and administrative functionality.

In less than an hour you can integrate e-commerce features into your current site, or generate a complete store from scratch. Cartweaver provides a lot of flexibility in how you display your products, including support for multiple product options like color, size, or any option that fits your products. You can even include links to related products on a product page, a great way to push traffic through your site and entice shoppers with other products. You control the action – updating products, overseeing orders, and managing customer information – through Web-

based administrative tools. You can also track inventory and allow back-ordering of out-of-stock products. To complete the sale, Cartweaver provides support for several popular payment gateways and processors: Authorize.Net, PayFlow Pro, WorldPay Select Junior and PayPal.

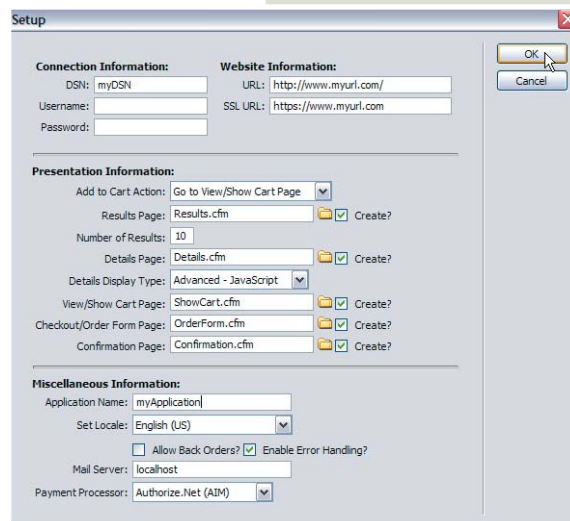
Developers will be pleased to note that the licensing fee is per developer, not per site. So once you own Cartweaver you can deploy as many e-commerce sites as you like. Even better, Cartweaver's ColdFusion code is completely open and

very well commented – so developers who want to expand or change the functionality of their e-commerce site can freely dive into the code and program away.

If your ColdFusion site is aching for e-commerce functionality, Cartweaver 2.0 might just be the answer.

...

Have a favorite extension? Is there an extension you just couldn't live without? We're always on the lookout for awesome extensions, so drop us a line at: [DaveMcFarland@sys-con.com](mailto:DaveMcFarland@sys-con.com) ☞





# A New Role for Flash

*Hurray! Flex is here, but you still need your Flash developers.*

by peter ent

Perhaps you've heard of Flex, maybe you've even dabbled with Flash, or you have one or more Flash developers on your team. Nevertheless, you aren't sure how Flash and Flex fit together.

As an enterprise application developer, I work on both the server side using Java and the client side using Flash. Knowing the benefits Flash has for the client, I looked into Macromedia's enterprise development offering, Flex. As a Flash developer myself, I saw my Flash skills being used differently on Flex-based projects. I have now begun using Flex because it fits in well with my own skills, but more important, it fits with the skills and needs of my team and our enterprise development environment. However, since I started introducing Flex to colleagues, the same questions kept popping up: "Do I need to learn Flash to use Flex?", "How does Flash work with Flex?", and "What does Flex offer that Flash doesn't?"

This article shows you, based on my own experience, how Flash and Flex work together and how the role of the Flash developer changes when involved with a

Flex-oriented project.

## Do I Need to Learn Flash to Use Flex?

You don't have to learn Flash to write Flex applications. Flex applications are turned into Flash movie files (the SWF file) and sent to the end-user's Flash player; this means that you can think of Flex as an alternative to using the Flash IDE (Integrated Development Environment) to make Flash movies. If you are an enterprise application developer and you've tried to learn Flash but found the concepts of the movie clip, multiple timelines, and levels a bit confusing, then Flex will be much more to your liking.

## How Does Flash Work with Flex?

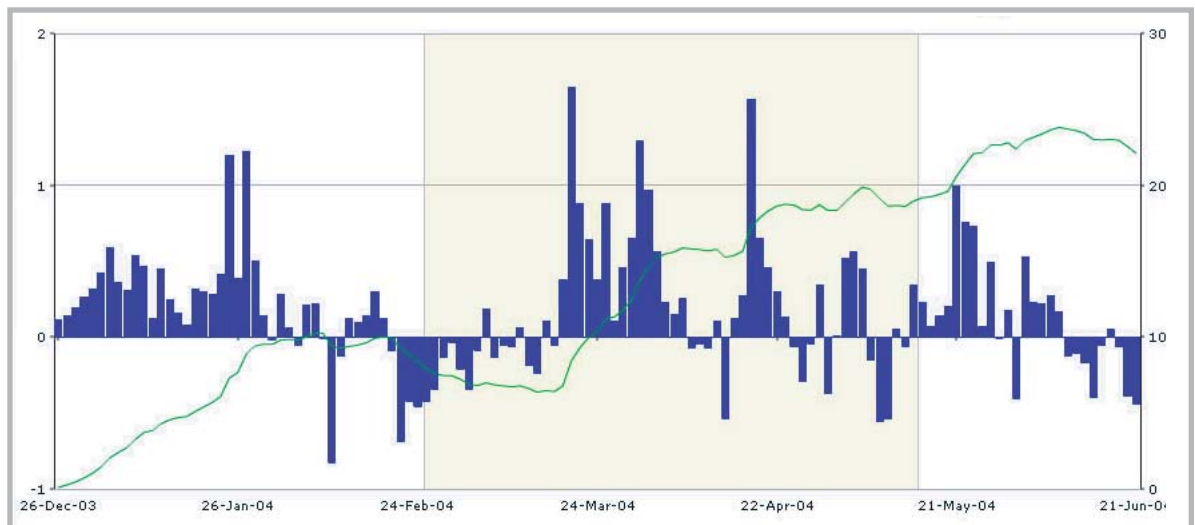
Flash developers play an important role in developing high-quality, enterprise-level applications. It is important to understand how Flash and Flex fit together to appreciate how valuable a Flash developer can be to your Flex team. Before Flash 5 debuted, it wasn't easy to use Flash as the sole front end to your Web application. Flash simply lacked the

necessary communication link between itself and the server. When Flash 5 debuted it became possible to use Flash as a front end for Web applications. Not only could you create buttons to hyperlink to Web pages, but you could now request data from the server and modify the user interface without generating a new page. But to do this, you had to learn the Flash development environment and think in movie-making terms. This was great for the creative artist; Flash became the way to show off yourself and your company on the Web.

*Tip: Flash had earned a reputation for snazzy "intro" movies, electronic greeting cards, and other, cartoonish uses. Even though it is possible to build applications in Flash 5, you have to build all of your own user interface controls. This effort, along with Flash's flamboyant reputation, didn't help convince IT organizations to use Flash instead of HTML.*

Nonetheless, you probably discovered that the Flash environment is not for everyone. Just as not everyone's brain is wired to write software, not every pro-

figure 1





grammer's brain is wired to think in terms of animation – a lot of us are old-fashioned, procedural-driver programmers.

Flex, on the other hand, has been clearly targeted at the enterprise Web application developer. Flex offers the advantages of Flash, but in a package that fits with enterprise development. Here's how:

- Flex applications are created in a manner similar to JSP-based applications and require an application server to run.
- The MXML files can easily be placed into source code control (with changes to them tracked – you can't do that with the binary Flash FLA files).
- MXML files can be edited by any text editor.
- ActionScript, the language of both Flash and Flex, is based on JavaScript, which many Web developers already know.
- Flex applications (that is, the MXML and related files) are deployed to production application servers as any JSP, HTML, or image file is deployed.
- Flex offers market penetration with 98% of the world's PCs running the

Flash player. Furthermore, Flash applications have a much smaller download than Java applets or Java/Swing applications.

- Flash is priced for individuals and small firms; Flex is priced for corporations.

### What Does Flex Offer That Flash Does Not?

Since Flex applications are ultimately translated into SWF files executed by the Flash player, you may ask "Can I build the same application with Flash MX 2004 as I can with Flex?" Both Flash MX 2004 and Flex share a common look and feel. Both Flex and Flash use the halo-themed components. The on-screen objects, such as buttons and lists, have a glow, or halo, about them when you move the mouse over them. Since Flex applications are translated into Flash movies, you can logically assume that what Flex delivers, you can do in Flash MX 2004.

The short answer to the question above is, "Yes, you can." The long answer is "Yes, but why would you want to?" Flex contains more user interface components than Flash MX 2004, and Flex contains things enterprise developers are expect-

ed to provide in their applications. For instance:

- Flex contains the objects to validate phone numbers, email addresses, numbers, and strings. Feedback is automatically given to the user without the developer having to write any additional code.
- Layout managers provide the means to resize and reposition elements and happen transparently to the developer.

In the enterprise development arena, managers look at their return on investment, or ROI. If you had the choice to build an application in Flash MX 2004, but it took 3 months, versus an application built with Flex that took 1 month, which would you choose? Further, by using Flex, most members of the team could participate in the project, not just the Flash developers.

### So Why Do I Need Any Flash Developers?

I have written several applications with Flash that fall into this "enterprise" category. I made extensive use of the UI

					
			<b>INTRODUCTORY PRICE</b> <b>\$45</b> Plus \$5 s/h. Reg. Price \$69.95		
		<b>Learn FreeHand MX</b> Create graphics for print and the Web. Work at your own pace with interactive course. This introductory price won't last long - order now. <a href="http://www.brainstormer.org">www.brainstormer.org</a>			
					

Price will increase 1/1/2005. FreeHand MX® is a registered trademark of Macromedia. ©2004 Nidus Corp.







# It's everybody's PDF™

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

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



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

d

f

tu

th

fr

sb

T H E B E S T O F T W O W O R L D S



# component-based

# Flash

by art phillips

w

*hen I was asked, I jumped at the chance to do a series on Flash data integration for the MXDJ.*

*Here was a wonderful opportunity to share my popular talks, presentations, and demonstrations with a large, interested audience. In this series I have the opportunity to show you how to use Flash MX 2004 Professional's components to rapidly develop data-driven rich Internet applications. At the conclusion of this series we will have constructed a one-page shopping application capable of using data from several back ends.*

I've always loved Flash. I preordered a copy of it after seeing a demo at NAB/Multimedia World back in '97. At the time, the multimedia market was going into a "small correction" and Macromedia was in the process of reinventing itself as a Web company.

Macromedia had evolved a complete tool set for multimedia. They had tools for the creation of interactive multimedia, courseware, electronic graphics, audio editing, 3-D, typography, and even video editing. In fact, that is what I had dropped over to the Macromedia booth to see: their new video editing product. One of the staff explained that the technology had been sold and invited me to check out Macromedia's new acquisition: Flash.

Here was a tool that provided rapid visual development of interactive multimedia and Web objects. This theme of rapid visual development began with Macromedia's early products, Authorware and Director, and is a hallmark of the Macromedia products line. Today Flash, Dreamweaver, Breeze, and FlexBuilder all provide developers with rich, powerful IDEs (integrated development environments) to speed and ease development tasks. Developers can choose to leverage the rapid development offered by a graphical user interface or knock out code by hand in an editor that provides automatic code hinting, syntax checking, and extensive online references.



Flash MX 2004 Professional offers the best of both worlds. The data connection components present the developer with easy-to-use text inputs for connection information. Menu-driven data binding then allows for quick and easy connection to a variety of data sources and interface components. These features, accessed through the visual interface of the component inspector panel, make it possible to rapidly develop rich internet applications. Properties and events exposed by the components programmatic interfaces allow developers to integrate their custom programming with the component framework.

Component-based development and Web services were a key part of the Macromedia MX product line. Jeremy Allaire stated that Macromedia's purpose was to "...empower a wide range of developers to access the power of object-based component development, rich client/server models and Web services without the pain of complex frameworks."

I really like that part about "without the pain of complex frameworks," and

that is the focus of this series. In coming months we will explore both rapid visual application development and customizing applications with the easy addition of a little hand coding.

To start the series off, this month we'll take a look at the history of client/server communication and the evolution of the RIA (rich Internet application). We'll also examine each of the data sources supported by Flash MX 2004 Professional's data components; XML, Web services, and Flash Remoting (see Table 1 and Figure 1).

In the early days of the Internet, Web servers acted pretty much just like file servers. They simply served up any pages and images the client requested. They had no way to interactively exchange data from the client with data on the server.

Along came the Common Gateway Interface, or CGI as it is more commonly known. CGI allowed the Web server to hand off data from the client to other processes, often a PERL interpreter. The Gateway then allowed results from those other processes to be passed back to the client. It got the job done, but there were performance and security issues in early implementations.

CGI led to application servers and server-side programming languages. ColdFusion, PHP, JSP, ASP, and ASP.Net are all examples of application servers and server-side programming environments. Application servers are designed to sit between the client and the data and to process programming directives embedded in Web pages. Certain languages are tied to specific servers. In all cases, all programming directives are processed on the server and replaced with standard HTML before being returned to the client.

While this model of server-side programming is very powerful, developers

encounter some important limitations in the typical exchange of data between a client and a Web server.

One key issue is the problem of the page refresh and the Web's statelessness. In a typical exchange of data between a Web client and the server, the data is sent to the server as part of a request for a page. This causes the page in the browser to be replaced with the new page and creates a multitude of issues.

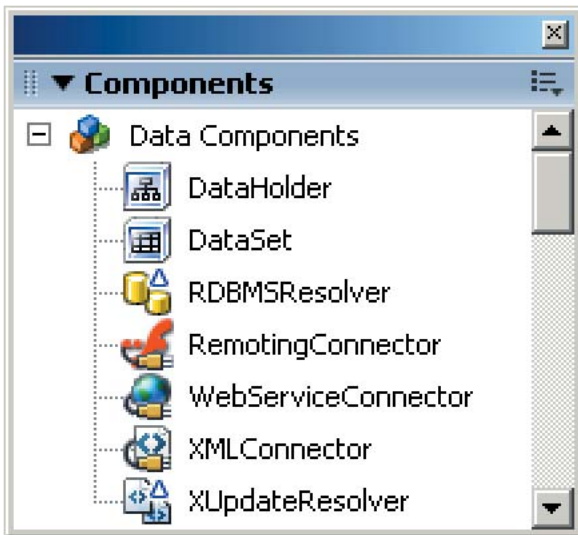
Since the Web is stateless and does not keep a persistent connection between the client and the server, how do we identify a specific client? This constant refreshing also raises another question: How do we persist information in the client? As developers we may want to use the client to store user preferences, shopping cart data, monitor progress through a process, hold scores, etc.

Cookies, passing data in URL strings, and other methods that suffer significant limitations are routinely used to solve these issues. A Flash-based RIA provides simpler and more powerful solutions to these problems.

Browsers present their own set of hurdles to both the designer and the developer. Different browsers on different platforms make pages look and behave differently. Tags do not render uniformly across all clients and client-side scripting (JavaScript) suffers from the conflicting approaches taken by different browsers. While Web browsers' compliance with standards has made great advances, issues still remain.

The typical exchange of data between the client and the server also suffers from some key limitations. A typical exchange is limited to passing variable declarations as name value pairs. All of the rich complex data types developers manipulate in memory, queries,

figure 1



## XML

**Pros** Simple tag-based method of describing data. .  
Text files require no compilation or special servers

**Cons** Parsing complex data may be more confusing due to lack of uniformity in data structures.

**Comments** XML is a good data source if you already have XML as part of your workflow.

## Web Services

Emerging standard for remote method calls and passing complex data.  
Data exposed as a Web service may be consumed by non-Flash clients.

Requires implementation of data source as a Web service. Flexibility between clients and servers is gained at the expense of efficiency.

Web services are an attractive path if your enterprise is committing to web services.



# Steps in Component-based Data Integration

arrays, objects, arrays of objects, cannot easily be passed between the client and the server. Passing of complex data typically requires transforming the complex data into a delimited string. This long, awkward piece of text uses specific characters as markers – delimiters – that separate the different pieces of data from each other. This process of serializing the complex data into a long piece of text and deserializing the string back into a complex data type has significant problems with both performance and the lack of standardization to the process.

XML offers a standardized way of describing complex data as strings and facilitates data serializing and deserializing. However, the problems discussed earlier, page refreshing, persisting data and the lack of a uniform delivery target, remain.

Here is where the Flash player comes in. The Flash player is simply the most incredible “thin client” available. It provides artists and programmers with robust playback of interactive animation and gives the client incredible abilities to send and receive audio and video. All of this is available from players that run on desktops and in browsers, and are only a fraction of a megabyte in size. In addition to these rich media capabilities the Flash player also provides a solution to some of the most basic problems in Web application development.

The Flash player deals with the issues of statelessness and persistence of data through two mechanisms. First, the Flash client has a variety of ways it can exchange data with the server while the player continues to run within its host. No new page has to be returned every time we communicate with the server. With a persistent memory space now established in the client, maintaining the identity of a client and

persisting data during a client’s browsing session becomes a simple and painless task.

Between a client’s browsing sessions data can be persisted through the use of Flash’s client-side shared objects. These local shared objects, or SOLs, are powerful “super-cookies” that offer unique benefits to the developer. In addition to an SOL’s ability to hold far more data than the typical browser cookie, the SOL can also easily store complex data types including combinations of arrays and objects without any need for the developer to manually serialize and deserialize the data.

The Flash player also provides solutions that allow developers to more easily pass complex data between the client and the server. The Flash player has built-in capabilities for sending, receiving, and manipulating XML. This native ability to handle XML also provides for the implementation of SOAP, the Simple Object Access Protocol. SOAP is a flavor of XML used by Web services to exchange both commands (such as method calls) and complex data (such as queries) between the client and the server.

Web services leverage Web protocols to provide a standardized way to communicate with a Web service’s programmatic interface. The use of standard Web protocols and the SOAP schema provide Web services with great flexibility. A client on any platform can use a service on any platform. The flexibility provided by this loosely coupled architecture comes at a small cost in performance.

Flash Remoting-enabled servers provide a more efficient, tightly coupled connection to the Flash player. Flash Remoting is similar to Web services in that it has the ability to pass method calls and complex data between the

- 1 Add a data connector (see Figure 3).
- 2 Provide connection information.
- 3 Add UI components.
- 4 Bind UI components to data and each other.
- 5 Trigger the data connector.

client and the server. However, instead of using XML/SOAP for client server communication a more efficient binary protocol is used, the Action Message Format. AMF provides a much faster, leaner, more efficient client server exchange than the text based XML/SOAP protocols.

Both the ColdFusion and JRun servers from Macromedia have built-in support for Flash Remoting. Macromedia also has products to enable Flash Remoting on J2EE and IIS servers. PHP has an open source remoting solution and there are open source solutions for other platforms as well.

The Flash player’s rich presentation layer and its support for advanced client server communication make Flash a cornerstone in the development of rich Internet applications.

More than a marketing cliché, RIAs are the natural evolution of the client/server model. They “...combine the functionality of desktop software applica-

*Arthur Phillips has been delivering cutting edge training solutions since 1984 and has developed instructional materials for George Washington University, the Federal Reserve Board, the U.S. Graduate School and many others. Art has an extensive background in video, multimedia, electronic graphics, Web development, and e-Learning. He holds too many certifications as a Macromedia Instructor, Designer and Developer to list. His Web site ([www.artswebsite.com](http://www.artswebsite.com)) is a well-known resource in the Macromedia community. [art@artswebsite.com](mailto:art@artswebsite.com)*

## Flash Remoting

Most efficient technology for remote method calls and passing complex data between Flash and the server.

Requires a server supporting remoting. Data exposed via Flash Remoting requires Flash client.

Native support in ColdFusion and JRun make Flash Remoting an attractive option for developers seeking the most efficient data integration in Flash.

table 1



tions with the broad reach and low-cost deployment of Web applications – resulting in significantly more intuitive, responsive, and effective user experiences.”

In other words, RIAs leverage the global reach of the Internet while overcoming some of the basic problems associated with data exchange over the world wide Web. This allows developers to create a user experience that is much more like using a piece of desktop software than the typical Web application.

Let’s take a common Web experience and compare what the user interaction is like as a typical Web application and what it is like as an RIA. Let’s examine the user experience of shopping.

Shopping using a normal Web application often starts with a search. The search takes shoppers to a new page where the results of the search are displayed. The results provide shoppers with some basic information (title, description, price, etc.) that assists them in making a decision (see Figure 2). They can either go to another page to purchase an item or another page to obtain more information about an item. If shoppers choose to purchase an item they are taken to another page that shows their shopping cart. When they choose to check out, shoppers have to work their way through yet another sequence of Web pages as

they provide their ordering and shipping information.

Another page, another page, another page... A simple purchase of two or three items may force the user to navigate literally dozens of Web pages.

Now let’s examine the same user experience in a rich Internet application. Shoppers begin their search and the search results are displayed right on the same screen with the search interface. As users interact with the search results, product details appear on the same screen. When shoppers purchase something, their shopping cart appears on the same screen that is serving as their catalog. Checkout is equally painless, occurring, again, right on the same single screen shoppers have been interacting with throughout the process. This is clearly a “significantly more intuitive, responsive, and effective user experience.”

If you have a Web application where a user must go through many Web pages to complete a process, you may have a candidate for your first RIA. This series of articles will help you make the transition from typical Web application development to RIA development.

The most important thing that I want any reader to come away with is this: all component-based data integration in

Flash follows a common development model, regardless of the data source (see sidebar).

## Summary

This common development methodology will be demonstrated throughout the upcoming articles.

In the next issue, XML will be our data source of choice. We’ll examine how XML provides a standard text-based source of data for Flash applications. We will bring XML into our applications and build a master detail view. All of this will be done with Flash’s visual tools; the components, the component inspector and behaviors. We will also see how to hook in a little custom code to the visual framework.

The third article in this series will examine the role of ColdFusion as a data source for Flash. Many other server technologies can serve as data providers including Java, ASP, ASP.Net, and PHP. However these other technologies do not have the simple power of ColdFusion Components (CFCs). CFCs provide for the rapid development of both Web services and Flash Remoting data sources. We’ll examine the theory and put it into practice as we create the data source for the next two articles.

Our fourth article will look at Web services. We will examine how the standard communication model allows access to a wide range of data provided by diverse back ends, including Java, ASP, ASP.Net, and PHP. We’ll discuss the technology and implement a Web service consumer in Flash.

The fifth article will examine the technology and implementation of Flash Remoting, which is closely tied to the MX product line.

In my sixth article we will leave the visual framework and add a shopping cart to our application. At that point we will have a very cool RIA – a shopping application that runs within one Web page, or from the desktop as an application.

I hope you will join me as we explore the world of component-based rapid application development in Flash MX 2004 Professional. ☺

figure 2

**1 page SHOPPING**

Product: Devil Duckies Price: 21

Description: This shower curtain measures 70" x 72", and is made of clear vinyl.

Quantity: 1 Add to cart

Name	Price	Quantity	Subtotal
America The Beautiful	26958	1	26958
Devil Duckies	21	1	21
Graffiti	17185	1	17185
Pink Flamingo	20	1	20
Skull & Bones	18	1	18
Stray Kat	16352	1	16352

Check Out



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](http://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)







# Illustrating with Flash MX 2004

*Game Techniques, from Chapter 9*

by **robert firebaugh**

Charles River Media ([www.charlesriver.com](http://www.charlesriver.com))

ISBN: 1584503157

Reprinted with permission of publisher

**g**raphics created for games published in Macromedia Flash cannot be illustrated or animated as usual. One must understand that most games developed in Flash can be processor-intensive, with quite a bit of user interaction and several animations playing at once, which means your graphics and animations must be optimized. Since most of these games are served up over the Internet, by keeping an optimized approach you will be able to deliver a more streamlined gaming experience. To streamline your game graphics creation several concepts have been developed, like formatting button creation and color change, making the production process more automated. We will also be reviewing how preparing your game graphics as many separate game pieces plays an important role in game creation.

## Game Pieces

One game may contain hundreds of individual graphics and animations and once they are arranged together they can seamlessly display a functional game. The idea of separating many graphics as individual game pieces is nothing new. In fact in the video game console and PC gaming industry it is very common to separate game graphics as individual pieces which are referred to as sprites. For example, a character animation or an item in the gaming environment is considered a sprite. We will not go into too many gaming industry terms but it is good to understand that most of these techniques are nothing new, yet very critical to know if you wish to create professional games in Flash.

## Hero Character

In most games there is a single character called the hero and the user con-

trols it to ultimately complete certain goals. The user controls the hero by interacting with a keyboard or controller, which manipulates the character to walk forward, shoot bad guys, or pick up items. Understanding that the hero has a set number of things the user can control it to do, you will realize that the hero sprite alone could be made up of over half a dozen still graphics and animations. For the Game Techniques chapter an army and alien-type characters have been designed along with an alien-looking environment. A solid approach to getting these graphics created cannot all happen in Macromedia Flash. It is recommended that characters and all game graphics are designed and sketched out on paper first, as shown in Figure 9.1. There they can be signed off by the client before investing many hours illustrating and animating them in Flash.

Where Macromedia Flash comes into the picture is once you have scanned your drawings and they are imported into the program as raster graphics to be traced with the Line tool or Brush tool. In this case, the Brush tool was used to trace the sketch, yet the strokes of black color had to be optimized several times to the final drawing you see in Figure 9.2. Since this character graphic is quite small the Zoom tool was used to zoom up close to trace and zoom back out 100% to view exactly what the user will see. This has to be done since small intricate vector

## ILLUSTRATING MACROMEDIA FLASH™ MX 2004

- Explains how anyone can create professional looking vector graphics, regardless of artistic talent
- Covers a variety of illustration styles including solid fill art, cartoon, manga, photorealistic, and experimental
- CD-ROM includes tutorials and a tryout version of Macromedia Flash MX 2004



Internet Series

graphics

appear clean and sharp up close but can look aliased and a bit different in shape if viewed at 100%. This is just one of the quirks of working with fills in Flash. Just remember, whatever you see at 100% is what the user will see, so that is what matters.

Once the illustration is fully outlined, colors are chosen with the Color Mixer and the colors are applied with the Paint Bucket tool, as shown in Figure 9.3. Another rule of thumb is that with creating small graphics like this one linear and radial gradients are not noticeable. Plus the more gradient and alpha fills used, the more processor-intensive your graphic or animation can be to the gaming experience, so solid fills will work just fine.

Due to space limitations, not all of the graphics referenced in the chapter are used here.

## Villain Character

The same approach was used to create the villain character. A rough sketch was drawn with pencil and paper, as shown in Figure 9.4.

Then the drawing was scanned in and imported into Flash where it was traced with the Brush tool and optimized to get a result as shown in Figure 9.5.

Now that the character is fully outlined, colors are chosen and applied, as shown in Figure 9.6.

## Buttons Made Easy

A game consists of many buttons for the user to select game modes, load saved games, and edit global options, just to name a few. To create all these buttons individually is a big task, especially since each button contains different text and possibly a different size. In this section we will review a clever way to make creating many different buttons of the same type easy and more efficient in file size.

The basic concept is that a button is made up of three distinct parts, as shown in Figure 9.7. The

three parts are the text, end piece, and body. The end piece and body are symbols so when they are used more than once we save on file size.

The end piece is duplicated, flipped horizontally, and moved to the opposite side of the body of the button. The text is placed in the center of the body and changed to the appropriate color, as shown in Figure 9.8.

The two end pieces are connected to the ends of the body by using the Selection tool with Snap to Objects feature on; that way the end pieces are connected precisely, as shown in Figure 9.9.

Well, that takes care of one button, but what about the others you have to create? Using this first button as your template when making the others, just follow these two easy steps. Let us say the next button you need to make has to



figure 9.1



figure 9.2



figure 9.3



figure 9.4



figure 9.5



figure 9.6

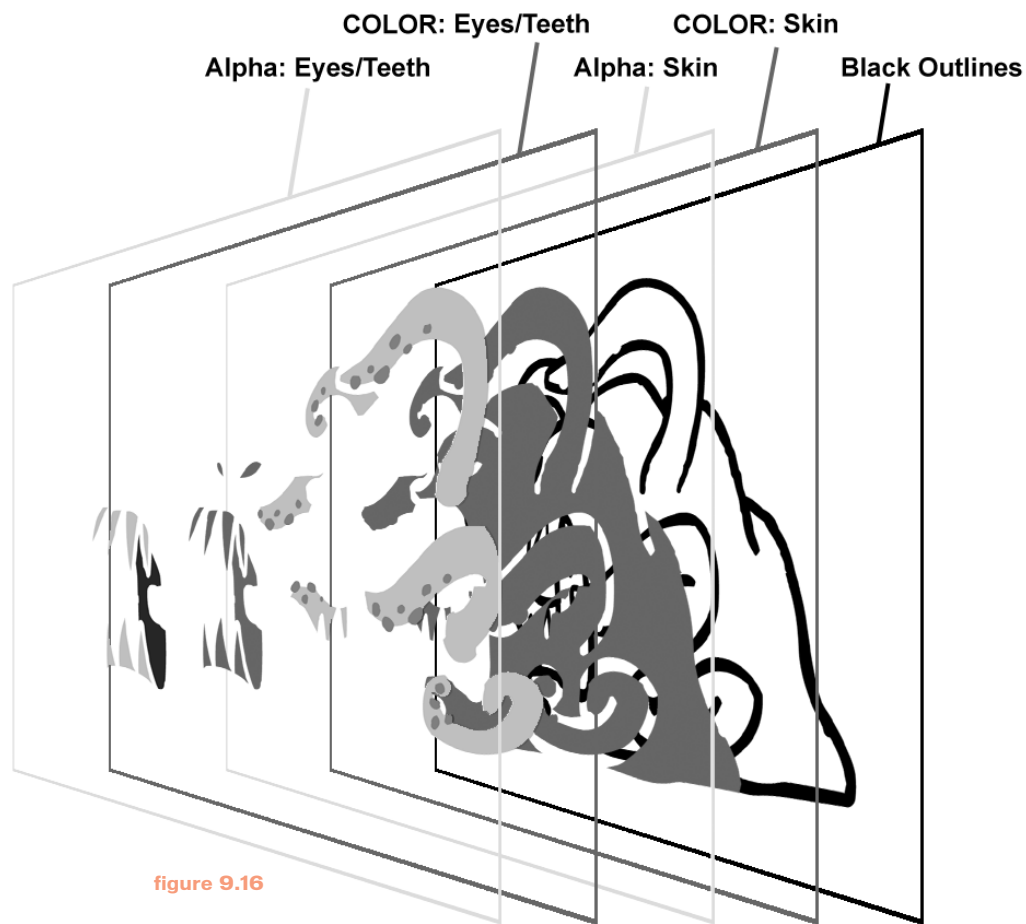


figure 9.16



say "Character Selection." First, change the text from "Submit" to "Character Selection." If you are working with white text on a white background feel free for the time being to changing the color of the text to black. Make sure when you retype the text that it is justified left; that way we will be scaling the button in one direction, to the right. Now take the right side end piece and move it to the right, as shown in Figure 9.10. Use the Selection tool to select it and use the right arrow key on your keyboard while holding down the Shift key to move it quickly to the right.

With the body of the button selected use the Free Transform tool to scale the body to the right. Make sure the center point of the Free Transform manipulator is snapped to the left side of the body, as shown in Figure 9.11. That way when you scale the body to the right, only the right side of the body will scale.

After following those two quick steps you should have a similar result as shown in Figure 9.12.

### Color Change

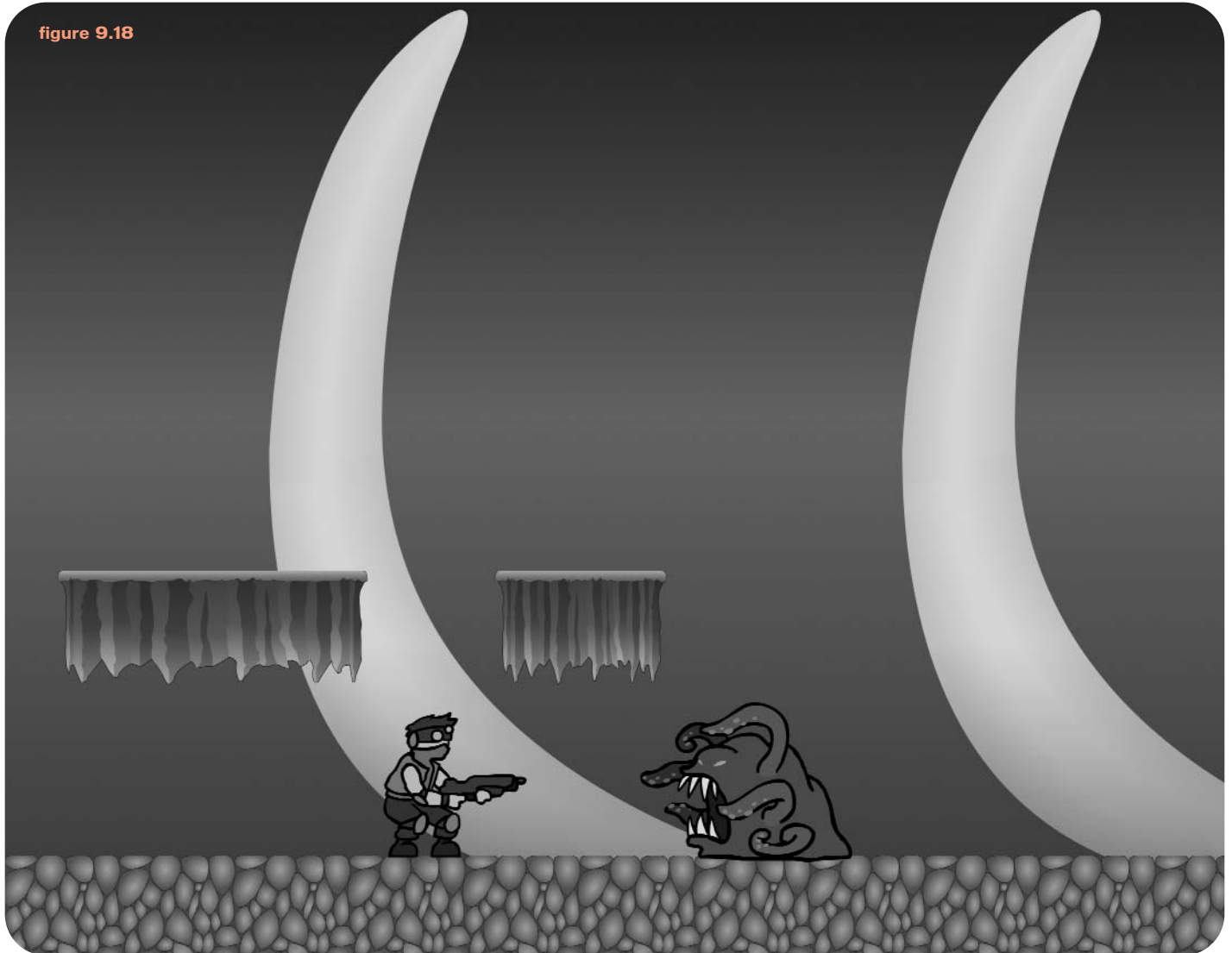
The idea behind this technique is that in a game there are characters or objects that look similar but may be drawn in a different color. For instance, you may have four different colors of the same monster. Another instance where this technique is helpful is if you have a character that the user can change like the color of their shirt. Instead of drawing ten different-colored shirts, why not just draw one that is a symbol and change its color manually or with code. In our example we have an alien and we need to have four different color variations of it. To break down the process on how this graphic is prepared, we start with analyzing the character. We want to look for which parts of the character will be sepa-

rated and colored. For instance in Figure 9.13 we point out the first color symbol, which we will set to the color of the character's skin.

In Figure 9.14 we point out the second color symbol which we will set to the color of the character's eyes, teeth, and mouth.

Analyze the layer setup as shown in Figure 9.15, which shows how the alien symbol is architected, to better understand the setup of how this color change works. Notice at the bottom of the symbol you have the COLOR skin layer which sets the color the skin will be. If you set the graphic on this layer to blue the character will look blue. The skin layer contains the different alpha fills which define the light and dark shades of the character. The COLOR eyes and teeth layer contains the symbol which sets the color the eyes and teeth will be. On the eyes, teeth layer you

figure 9.18





# be the pilot!

**FREE SETUP** on Shared Hosting Accounts With ColdFusion MX Support  
Use Promo Code CFDJ2004



**INTERMEDIA.NET**

## WE DARE YOU TO TAKE A FREE TEST FLIGHT!

Managing technology that runs your business is a matter of trust and control. INTERMEDIA.NET gives you both.

**TRUST.** Since 1995 we have been providing outstanding hosting service and technology to our clients. Don't take our word for it... take theirs.

*"The support and service that you offer are nothing short of golden. The high quality of your system and service for CF customers is something one could only ever dream of." – Claude Raiola, Director, AustralianAccommodation.com Pty. Ltd.*

**CONTROL.** We give you instant control over your site, server and account configuration changes. No more submitting requests and waiting for someone else to take action. You are in control to pilot your business through its daily needs.

**BE THE PILOT.** Take a free test flight and see what our HostPilot™ Control Panel offers you beyond all others. Check out our SLA guarantees. To see more testimonials and to find out about our competitive advantages, visit our Web site at [www.Intermedia.NET](http://www.Intermedia.NET).

**Managed Hosting • Shared Hosting • Microsoft Exchange Hosting**

Call us at: 1.800.379.7729 • Visit us at: [WWW.INTERMEDIA.NET](http://WWW.INTERMEDIA.NET)



**HostPilot™**



have the different alpha fills which define the light and dark shades of the character. The black lines layer contains the black outlines of the character. These graphics are not affected by any of the color changes.

Another way to visualize the setup is through a visual diagram of how the layers and graphics are compiled, as shown in Figure 9.16. For better understanding the black outlines have been placed in the back of this diagram. Notice how the alpha graphics are placed on top of the colored graphics. That way, when we change the color of the colored graphics you still have the alpha graphics defining the lightness and darkness of the color. See how the alpha skin layer has two different values of alpha black fills, making the feelers of the alien a darker shade of color than the tentacles of the character.

By setting the Color Styles drop-down menu to Advanced and manipulating the color settings you can come up with many different variations of the same character. The Color Style drop-down box is located in the Properties panel once the color symbol is selected.

### Background and Graphics

Background graphics have a great impact in the final file size of your game, because background graphics spread across the whole display area. Most games contain various levels where game play takes place, which requires many different kinds of background graphics. If the proper techniques are not used your file size can spiral out of control. Building your backgrounds with many pieces of graphics that are pieced together like a puzzle is the most common technique used to keep a file size down. To further understand how

optimized backgrounds are achieved analyze the finished background as shown in Figure 9.18.

### Build From Pieces

To create the rocky-looking ground and make it repeat you have to create a piece of graphic that when duplicated and placed back-to-back the graphics connect seamlessly. To start you will want to draw a square with the Line tool and draw in the texture with the Pencil tool. For a quick trick to make sure the left side will seamlessly flow into the right side, draw the center of the texture first, then finish the texture onto the left side and Copy and Paste in Place, as shown in Figure 9.19.

Change the color of the pasted lines and move them over with the right arrow key and connect it to the right side, as shown in Figure 9.20. We changed the color of these lines so when we decide to delete them all we have to do is double-click on them and only those lines will be selected because they are a different color from the rest.

Now finish the texture on the right and make sure to connect the new line to the lines we pasted from the left and we should have something similar to Figure 9.21.

To color your texture just create a simple radial gradient and apply it with the Paint Bucket tool. Make sure the gradients on the left side of the texture will match up with the textures on the right side. To make sure they all match up use the same technique as used with the line drawing, but apply it with gradient fills. You should have a final colored graphic as shown in Figure 9.22.

To show how this texture tiles across several pieces snap together several of

them, as shown in Figure 9.23. If you can notice where they connect from viewing the textures at 100%, go back and make the minor adjustments needed.

To create the platforms and prepare them so they can be scaled horizontally without much distortion we will be using a technique similar to what we used on the buttons earlier in this chapter. The problem is in a game many different-sized platforms are needed and to create a new platform for every size will increase your file size and we do not want that. So the technique is to create a center piece which can be scaled horizontally and an end piece which is duplicated and repositioned horizontally based on the length of the platform, as shown in Figure 9.24.

The end piece is duplicated and repositioned to the left, as shown in Figure 9.25.

Select the left end piece and flip it horizontally by selecting the following menu items: Modify > Transform > Flip Horizontal. Then select each end with the Selection tool and snap them to each end of the center platform piece, as shown in Figure 9.26.

By scaling the center graphic horizontally and repositioning the end pieces you can create any size platform you desire. As shown in Figure 9.27 many different sizes are possible and since we are only using two different symbols you can create many platforms without increasing the file size much.

### Large Backgrounds

Creating large backgrounds can be a puzzle just by itself since backgrounds can be quite large and can increase your file size tremendously. The number-one way to achieve large backgrounds is to make key elements symbols and reuse them. For instance, with the horns that are sticking up

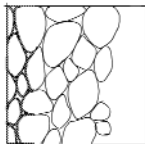


figure 9.19

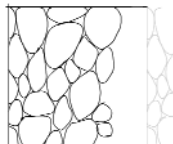


figure 9.20

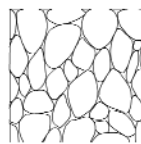


figure 9.21

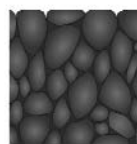


figure 9.22

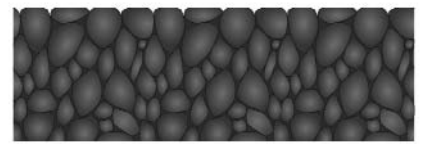


figure 9.23



figure 9.25



figure 9.26



figure 9.27

from the ground, we just draw one horn and duplicate it across the stage. A simple gradient is used in the background, making it look like the characters are inside a cave.

To create the horn graphic you should start by using the Line tool to draw the outline of the object, shown in Figure 9.28. Notice extra lines are drawn in the center of the horn; these lines will divide where the different gradient fills will be placed to define the curvature of the horn.

Create two radial gradients, both the opposite of each other; one has the darker color on the inside and the other has the darker color on the outside. Apply the radial gradients with the Paint Bucket tool and position them as they are shown in Figure 9.29 with the Fill Transform tool. Figure 9.29 Two radial gradients are created and applied.

To add a final touch to make the graphic look a bit scary, you could add some red light coming from the bottom of the object. To do that, select the fills we just finished coloring and copy and paste them in place on a new layer above the previous one we were just working on. Select the new graphic and change the color to an alpha red linear gradient. That way we can see through it, as shown in Figure 9.30.

A detailed description of this new linear gradient is as follows: the left color swatch's HEX value is #990000, alpha 75% and the right color swatch's HEX value is #990000, alpha 0%, as shown in Figure 9.31.

### Optimized Animations

Game animations require the most attention when it comes to being opti-

mized. It is very important to keep the number of different keyframes down to a minimum and it is strongly recommended that every frame is optimized to make sure the least amount of vector points are used without losing the quality of your graphic. The reason for the strict optimization is that unlike normal animations, a game displays many animations at the same time and even though you may not see it, there is a lot of code working in the background. If too many un-optimized animations are playing you could jeopardize the performance of your game and then bring down the quality of the gaming experience. In this section we will analyze a character animation and a special effects animation to see how you can make them as optimized as possible.

### Character Animation

Character animations involve quite a bit more frames than most game animations; therefore they require greater attention. As shown in Figure 9.32, the character is made up of many symbol graphics which are animated together with motion tweens to display more elaborate movement.

Of course, by reusing the different body pieces many times because they are symbols we save on file size, versus if each frame of animation was a completely new graphic. As shown in Figure 9.33, the key frames of the character shooting require the body pieces to be used many different times.

### Special Effects

A special effect in a game is considered a short animation that highlights a certain action or reaction. For instance, when the character shoots his gun, two

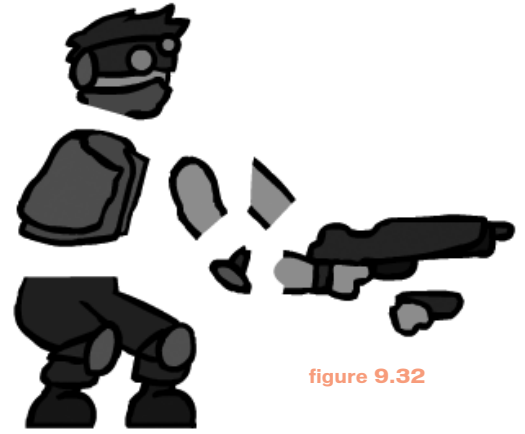


figure 9.32

special effects animations are needed, one for the fire blast coming from the tip of his gun and the flash to show the bullet hitting an object. Other special effects animations could be a splash when someone steps in a puddle, smoke from a car stopping abruptly, a starburst from a shining piece of metal, or a flash of light from a lightning strike. Special effects can be drawn as shown in Figure 9.34 or done with ActionScript.

### Summary

- Finalize your character drawings on paper first to limit any character design in Macromedia Flash.
- Build game graphics in pieces; reuse and scale them to keep file size down.
- Setting up your character graphics to include a layer of color that is changed manually or with code to create many different colors of the same character will decrease file size and give you more content.
- Limit the number of frame-by-frame animations and use tween animations to achieve greater motion with fewer keyframes.

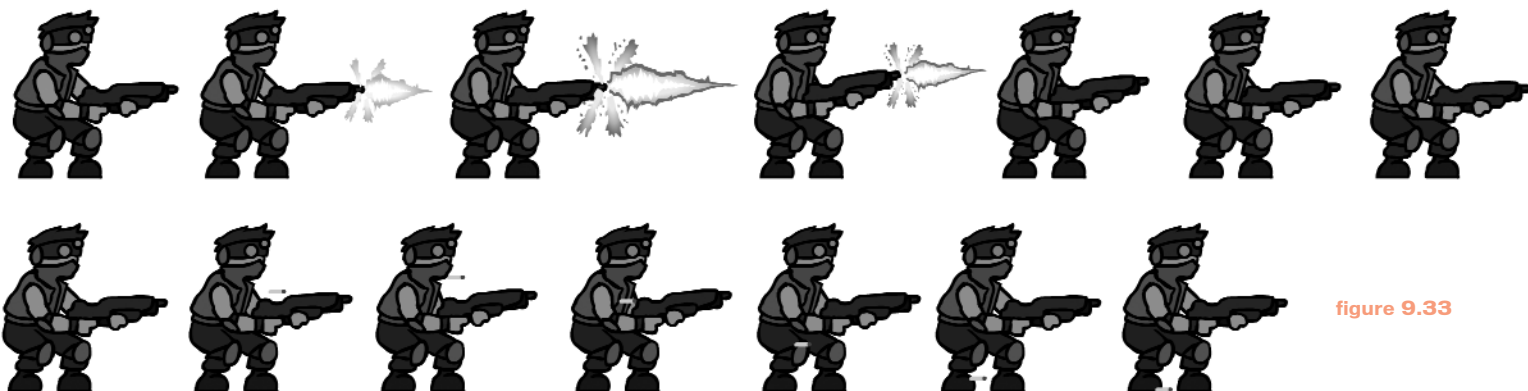


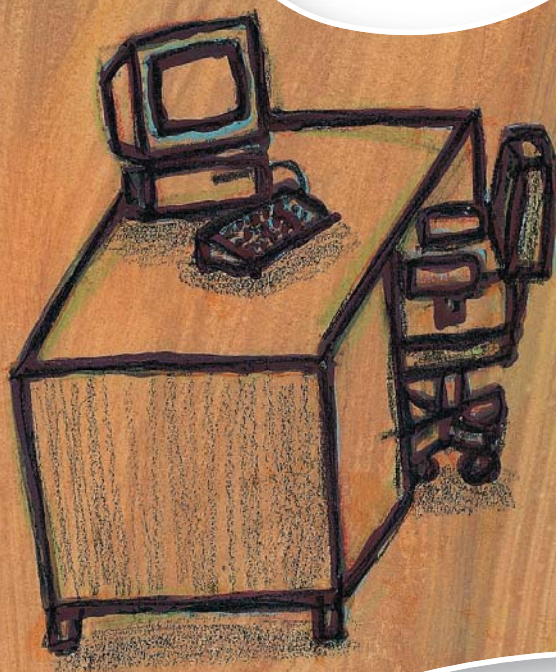
figure 9.33





# FIREWORKS IMAGE OPTIMIZATION BASICS

by joyce j. evans



Optimization, or getting your image to the smallest possible size while retaining an acceptable quality, is one of the most important factors in determining how quickly your Web page loads.

#### Note

The file types currently available for the Web include the PNG format, but not all browsers support this format, so you may want to hold off on using that one just yet.

#### Getting Started

You can quickly optimize an image using the Optimize panel in Fireworks or you can optimize different parts of an

image using the Slice tool. Slices are needed to export specific images or parts of images in a document. A slice (or hotspot) is also needed when you want to add a behavior. One of the biggest advantages of defining a slice is that you can individually optimize each slice. This means that you can have JPEG and GIF images in the same document. You can also use different compression settings for each image if desired.

#### Slicing Tools

The slicing tools are the Rectangle, Circle, and Polygon Hotspot tools. The Rectangle Slice tool works the same way as the other Rectangle and Circle Hotspot tools in Fireworks; click and drag over the area you want to define. The Polygon

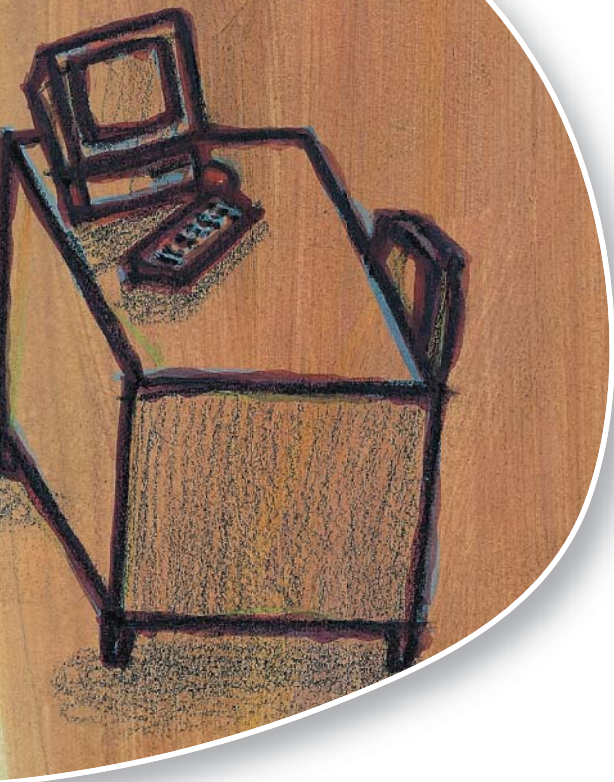
Slice tool also works the same way as the other Polygon tools in Fireworks; click a starting point and click to add points to define a shape.

You can define a polygonal shape with the Polygon Slice tool, but the polygonal shape cannot be exported as a polygon shape. All slices are rectangular, without exception. Fireworks automatically adds as many rectangles as necessary to maintain your defined polygon shape. These rectangles fit together with the polygon shape enclosed.

Every time you define a slice, Fireworks automatically places red lines



**USERS IN MOST U.S. URBAN AREAS ENJOY ALL KINDS OF OPTIONS FOR HIGH-SPEED CONNECTIONS, BUT MOST RURAL AREAS DON'T YET HAVE THAT OPTION, NOR DO A LOT OF OTHER COUNTRIES. YOU SHOULD KEEP IN MIND THAT THE LARGER THE FILE OR COMBINATION OF FILES ON A WEB PAGE, THE LONGER IT TAKES TO LOAD. CONNECTION SPEEDS DETERMINE HOW LONG EACH FILE TAKES TO LOAD.**



*Joyce J. Evans is a training veteran with over 10 years of experience in educational teaching, tutorial development, and Web design. She has presented at conferences such as Macromedia MAX 2003 and TODCON. Joyce has authored books including Macromedia Studio MX 2004 Bible, Dreamweaver MX 2004 Complete Course, and others. Joyce is a Team Macromedia volunteer and her work is also featured in the Macromedia Design/Developer center. Her Web site is [www.JoyceJEvans.com](http://www.JoyceJEvans.com). [joyce@joyceevans.com](mailto:joyce@joyceevans.com)*

where additional slices could be to slice the whole page. As you make additional slices the red lines change. You could stop slicing after a few slices and allow Fireworks to use its own guides (the red lines) to slice the rest of the image. In the Exporting dialog box you see an option called Include Areas Without Slices, which slices your document according to the areas defined within the red lines.

Imagine you have a large 800K image file and you want to get it under 100K. The image size is 600 x 400 pixels and comprises several elements, including a photo of a woman, a logo, text, and a box shot. The photo has a lot of color and is best optimized as a JPEG. The text and logo have few colors and should be optimized as a GIF. The best way to optimize this large image is to slice it into smaller images and optimize each separately. By slicing the images you can reduce the file size for individual elements separately. Fireworks slicing tool defines a specific area to optimize (compress).

To slice an image, follow these steps:

1. Select the Slice tool from the Tools panel.
2. Click and drag over an object or part of an object/image.
3. While it is still selected, look in the Property inspector (below the document or press Ctrl/Option+F3). On the left side is a field named Slice. Remove the default name and give your slice a meaningful name.
4. In the Type drop-down menu you can choose whether the slice is for an image or for HTML. The HTML option allows you to type in text or even add HTML code right in Fireworks. This will render as HTML when exported, not as an image.
5. You can add a link (URL), Alt text, and a Target as well if you'd like.
6. Below the Type field is a menu of preset optimization settings you can use (see Figure 1).

### Optimizing GIF Images

Images (such as logos, text, or drawings) with large areas of color and/or images with 256 or less colors are best suited to the GIF format (see Figure 2).

Any image that you want to have a transparent background will need to be

exported as a GIF. Photographs should always be in JPEG format. To optimize an image as a GIF, follow these steps:

1. File > Open the image you want to optimize.
2. Select the image either by clicking on it or, from the Edit menu, choose "Select All".
3. Click on the Preview, 2-Up, or 4-Up tab to preview your settings as they will be seen in a browser. Previewing how your optimized image will look in a browser is vital, and Fireworks makes it easy. The Preview tabs are on the top left of your document window. You can view one setting or up to four different optimization settings, including different file formats. You can see the changes each setting will make on the file size and the appearance of the image.
4. Open the Optimize panel by clicking on the word Optimize in the Panel group area. If the Optimize panel isn't open you can press F6 to activate it.
5. Select GIF from the File Format drop-down menu.
6. Select the Indexed palette you'd like to use. Your choice determines which of the 256 colors are included. The most used palettes are:
  - *WebSnap Adaptive* palette is the default palette for indexed color in Fireworks. Any color that is not Web safe is automatically evaluated and snapped to the closed Web-safe color, plus or minus seven values. It doesn't guarantee that all the colors will be Web safe but it's close.
  - *Adaptive* palette finds a maximum of 256 colors. It's not a preset color set, but the best 256 colors for your image. This technique may contain a mixture of Web-safe and non-Web-safe colors.
  - *Web 216* converts all colors in the image to the nearest Web-safe color.
7. If there are specific colors that you don't want removed, select them from the Color table then click on the Lock icon at the bottom of the Optimize panel.
8. Select a Matte color or the transparent option. The Matte option enables you to export your image



figure 1

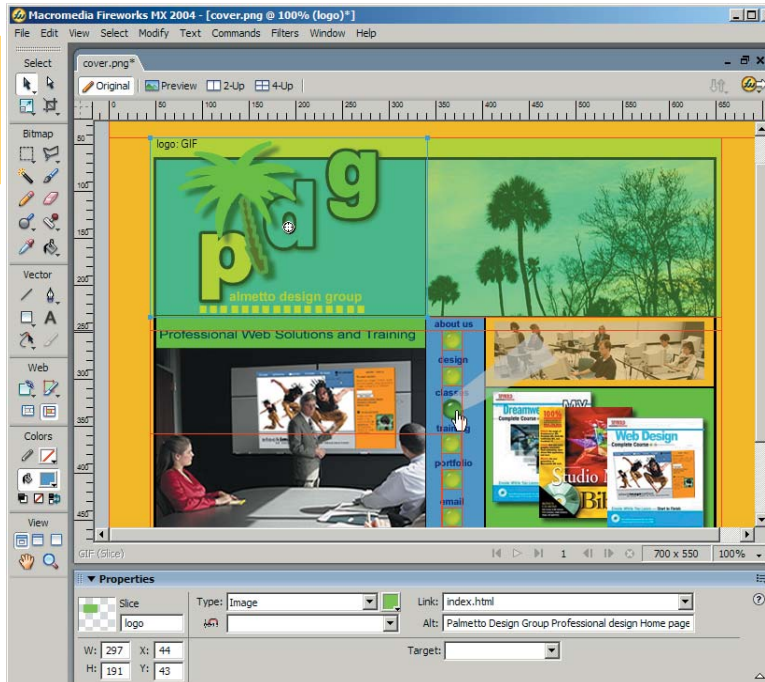


figure 2

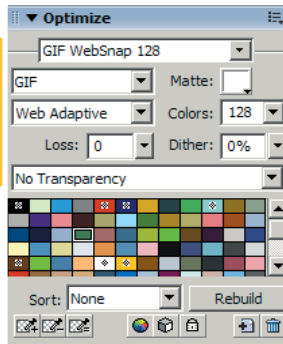
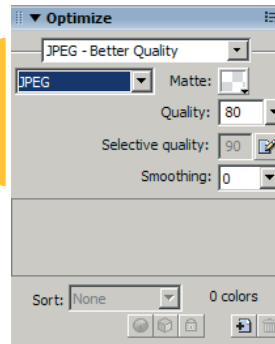


figure 3



slice as though it has a background color – without changing the canvas background color. This option is particularly important when you want to use the exported image in an environment different from the one it was designed in, or when you are planning to use the image on a variety of different-colored backgrounds.

9. Select the number of colors you'd like to use. Check your preview to see how it looks. Zoom in a bit, especially when you have anti-aliased or feathered edges. As you lower the colors an anti-alias and feathered edge will get more jagged. You'll need to decide at what point the colors are acceptable to you.
10. Set the Loss setting if you'd like. Loss is the amount of compression used. GIF images normally have no loss of image detail as long as the image has less than 256 colors and/or you don't remove colors because they

use loss-less compression. But you do have the option of using compression even in a GIF image; it produces some distortions in your image however. The higher the loss setting, the higher the distortion.

11. Set the Dithering option if you'd like. Dithering gives the illusion of new colors by varying the pattern of dots of color. The downside is that it increases the number of color changes in a horizontal row, which also increases the file size. You apply any percentage of dither you feel is needed to help a GIF image look better.
12. Set the transparency. The choices for transparency are No Transparency, Index Transparency and Alpha Transparency. Index Transparency will remove the background color, even if it is in the image itself. Alpha Transparency will remove the background color only and not the same color if it appears in your image.

## Note

Only one color can be used for the matte or background color automatically. You can, however, add or subtract additional colors to the index or alpha transparency by using the various eyedroppers in the lower-left corner of the Optimize panel. Click the appropriate eyedropper; then click the color. You'll need to reselect the eyedropper each time you want to add or remove a color to the transparency.

## Optimizing JPEG Images

Photos and images with gradients are best optimized as JPEGs (see Figure 3). Selective JPEG compression is a special feature available only for JPEG images. It's used to set higher optimization to selected areas of an image. To optimize an image as JPEG, follow these steps:

1. Select the image or slice.
2. In the Optimize panel, select a File Format of JPEG.
3. Open either Preview, 2-Up or 4-Up to see how your settings will appear in a browser.
4. Change the Quality setting. Experiment with different levels and check the quality of the image in the preview. The quality setting determines how much compression is applied to the image. It's a good idea to zoom in a bit to see the artifact effect on your image. Go as low as you can and still have an acceptable-looking image.
5. Add Smoothing if you'd like. When you lower the quality setting you may see visible artifacts depending on how low you went. The Smooth option adds a bit of blur to blend the image. You lose some image sharpness but it may be a compromise you are willing to make when a small file size is a must. The Smooth setting ranges from 0 to 8 with 8 being a higher degree of blur.
6. Another option that you don't see in the Optimize panel is to sharpen the JPEG edges. If you optimized pretty low and/or used Smoothing, you may need to sharpen the edges a bit. To use this option, open the Optimize Options Pop-up menu and select Sharpen JPEG edges. This will add to your file size.

That's all for now. Next month, I'll show you how to batch process all those images! ☺





**web services** **EDGE**  
conference & expo

Web Services Edge  
**2005 East**

International Web Services Conference & Expo

**Hynes Convention Center, Boston, MA**  
**February 15-17, 2005**

The Largest  
i-Technology  
Event of  
the Year!



**Guaranteed  
Minimum  
Attendance  
3,000  
Delegates**



Tuesday, 2/15:  
Conference & Expo

Wednesday, 2/16:  
Conference & Expo

Thursday, 2/17:  
Conference & Expo

Join us in delivering the latest, freshest, and most proven Web services solutions... at the Fifth Annual Web Services Edge 2005 East - International Conference & Expo as we bring together IT professionals, developers, policy makers, industry leaders and academics to share information and exchange ideas on technology trends and best practices in secure Web services and related topics including:

- Transitioning Successfully to SOA
- Federated Web Services
- ebXML
- Orchestration
- Discovery
- The Business Case for SOA
- Interop & Standards
- Web Services Management
- Messaging Buses and SOA
- SOBAs (Service-Oriented Business Apps)
- Enterprise Service Buses
- Delivering ROI with SOA
- Java Web Services
- XML Web Services
- Security
- Professional Open Source
- Systems Integration
- Sarbanes-Oxley
- Grid Computing
- Business Process Management
- Web Services Choreography

**3-Day Conference & Education Program**

features:

- Daily keynotes from companies building successful and secure Web services
- Daily keynote panels from each technology track
- Over 60 sessions and seminars to choose from
- Daily training programs that will cover Web Service Security, J2EE, and ASP.NET
- FREE full-day tutorials on .NET, J2EE, MX, and WebSphere
- Opening night reception

**Interested in Exhibiting,  
Sponsoring or Partnering?**

Becoming a Web Services Edge Exhibitor, Sponsor or Partner offers you a unique opportunity to present your organization's message to a targeted audience of Web services professionals. Make your plans now to reach the most qualified software developers, engineers, system architects, analysts, consultants, group leaders, and C-level management responsible for Web services, initiatives, deployment, development and management at the regions best-known IT business address - The Hynes Convention Center in Boston.

For exhibit and sponsorship information please contact Jim Hanchrow at 201.802.3066, or e-mail at jimh@sys-con.com.

Sponsored by:



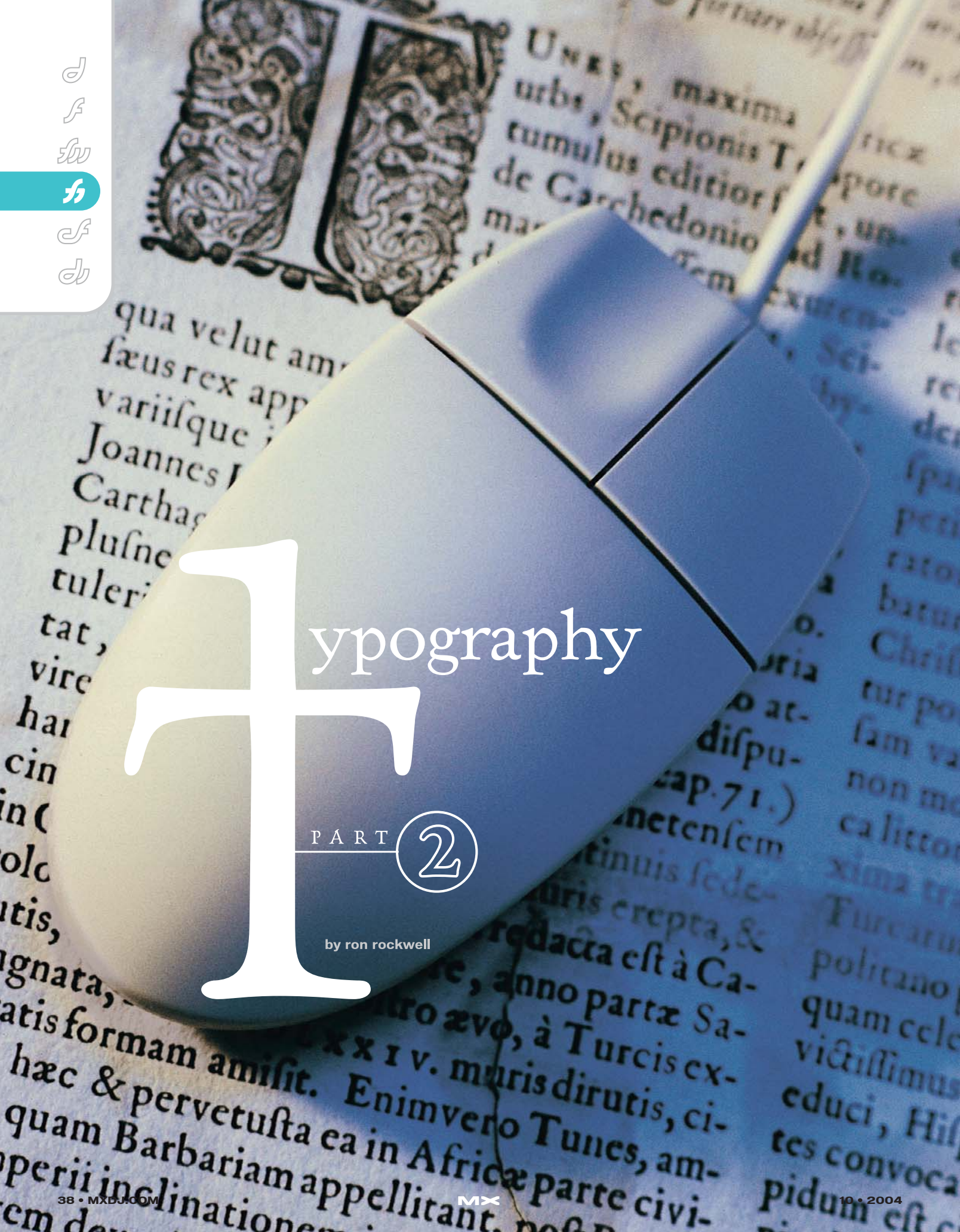
All brand and product names mentioned above are trade names, service marks or trademarks of their respective companies.

Contact for Conference Information: Jim Hanchrow, 201-802-3066, jimh@sys-con.com



**www.sys-con.com/edge**





# T

ypography

PART 2

by ron rockwell

UNIV, maxima  
urbi, Scipionis T  
tumulus editior  
de Garchedonio  
ma  
Tem  
qua velut am  
faeus rex app  
variisque  
Joannes J  
Carthag  
plusne  
tuleri  
tat,  
vire  
har  
cin  
in C  
olc  
itis,  
gnata,  
atis formam amisit. Enimvero Tunes, am-  
hac & pervetusta ea in Africa parte civi-  
perii inclinationem  
em de

# ...keeping a firm hand on your text

A picture is worth a thousand words, but there's always some loud-mouth who wants to add a thousand words to your picture. For that, you have to know a little more about FreeHand's text-handling methods.

We pretty much handled headline text in last month's article (Vol. 2, issue 9), so this month the emphasis will be on body text – or body copy as some of us old-timers call it. Your boss probably calls it the “small print.”

I assume that you've read last month's article. In it, you'll learn about the construction and operation of text blocks and the basic job of entering text on a FreeHand page.

## The “ABC” Tab

There's something about an “R-ball” – registered trademark symbol, “TM” symbol, and other legal marks that I don't like, so I make them smaller and use Baseline Shift to move the marks to align with the top of the text.

To stylize paragraphs, most of the work is done in the Object panel (see Figure 1). The Paragraph button has been selected, opening the options we can apply to paragraphs. The first of the five spaces is a text entry for the space above a paragraph. It's common for novices to place a double return between paragraphs, but if you really want control over your typography, enter a value here.

If you just enter numbers without checking the units of measure you have set for your document, you could be surprised to see your text disappear. For instance, if you want two points of space before a paragraph and enter “2” in the top field – but the document is set in inches – you'll get a two-inch gap between paragraphs.

Next in paragraph stylization is the space after a paragraph. If you also have space before measurements, the effect is cumulative. Entries can be made for left and right indents (shown in Figure 2), and at the bottom of the row is the spot to create a first-line indent. Beneath those text fields, you see the Hyphenate option. Place a check in it and your text will hyphenate automatically; delete the check and words won't break. The Edit button allows you to input how much of the last line must be filled with text before it justifies (the Flush Zone). Hanging punctuation and rules were covered last month.

Notice the size of the text block in Figure 2 and compare it to the size of the text block and indents in Figure 3. In Figure 2 the text has been selected with the Text tool, which brings up the Text section of the Object panel. The text is indented from the margin or boundary of the text block when using paragraph indents. In Figure 3 the top text block has been selected with the Pointer tool, bringing up the “Clip Text Block” entry in the Object panel. Here, the text block itself is indented when using indents based on the text block, as you can see in the bottom text block with the text itself selected. For the more curious among you, it's called a Clip Text Block because I used Paste Inside for the dashed lines, creating a clipping path in the process.

The point of the text block is that you can apply a color to a text block, then inset the text from the sides – similar to adding padding to a cell in Dreamweaver. In doing so, you can move the text block around the page, reshape it, edit the text, and still maintain a single object on the page. The Object panel displays the dimensions of the text block and its location on the page, and gives you fields to enter the inset on all four sides.

There are often times when you will

have a paragraph of text and the last line doesn't seem to obey the leading settings you've applied. When you see this – it's usually too much space – just place the Text tool cursor at the end of the line and press Enter or Return. This gives you an extra blank line beneath the paragraph, but brings the last line of text into proper leading. If you need to get rid of that extra blank line, usually you can use the keyboard arrow key to drop down to the beginning of the next live line of text, then press Delete or Backspace to eliminate the blank line. This behavior is common to page layout programs as well.

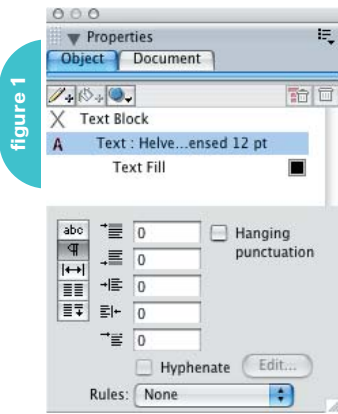
## Text on a Path

Sometimes it's not enough to get words on a page into nice straight lines, squared off paragraphs, and easy to read. No, sometimes you want to have text slither around the page or circle a logo. In FreeHand, it couldn't be easier. Start by creating your text; set the size fairly close to your desired result. Then draw your path. For an open path (top of Figure 4), all you have to do is Shift-select the text and the path and choose Attach to Path from the Text menu. As long as the text stays “live,” you can edit it any way you want – size, font, style, spelling – it's all available for you.

The really scary part about placing text on a path is that it becomes ugly in a really bad way on a Mac! It's a problem between OS X and FreeHand, but evidently the text becomes aliased and doesn't happen on Windows machines. It's nothing to worry about, however, it prints fine. If it offends you – and you know you don't have to edit the text – you can convert the text to paths to get clean text. Figure 4 shows the aliased text, followed by the text converted to paths.

Just below the text on a path, you see a rather crude snake shape. It was





converted to a brush and applied to a clone of the original path. The text was placed above the snake brush and a clone of the text was changed to white and offset.

Figure 5 shows what to do if you want your text to follow multiple paths. In this case, the snake path was cut in two and the text was applied to one of the paths. An overflow box appeared at the end of

the text and was simply dragged (with the Pointer tool) onto the second path.

### Text on an Ellipse

Applying text to an ellipse is a daunting task at first. But it's really not that much of a mystery. It comes down to a couple things: 1, you have a circle, and the circle has a top and a bottom; 2, you must place a Return between the text you want on top and the text that goes on the bottom. That's it. Whatever goes before the carriage return will appear upright on the top of the circle, and everything after the return will be upright on the bottom. If you want the text to completely encircle an object, just don't put a return in the text.

You'll notice that somewhere along the text on a path there's a little triangle. That triangle allows you to move the text along the path. Just click and hold it with the Pointer tool and drag until you like what you see.

In Figure 6, text has been applied to a circle, and as you can see in #2, the text became aliased, so was converted to paths in #3 for the purpose of the screenshots here. The red dart in #2 is pointing at the text's alignment triangle. Number 4 added two more circles, and the entire group was distorted using the 3D Rotation tool, applying a bit of perspective to the drawing. For the final drawing in #5, a clone of the ellipse was dragged down a bit, and a second clone of the top ellipse was punched out of it. A gradient fill here and there finished it off.

### How Do I Get Text OFF a Path?

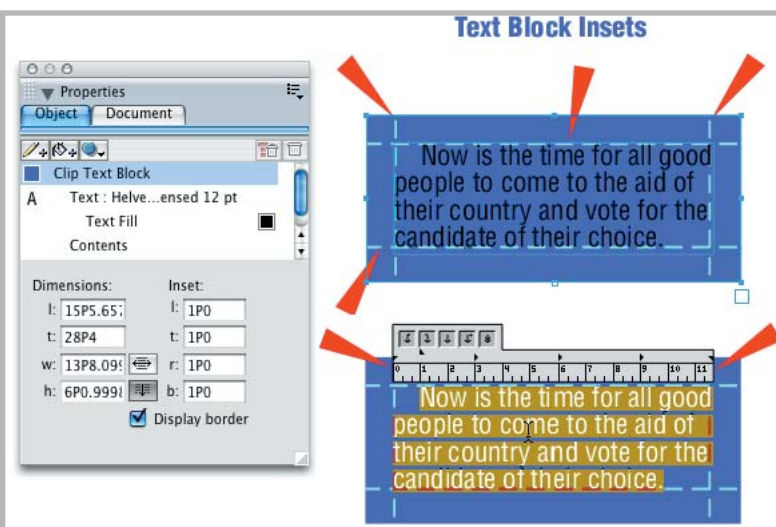
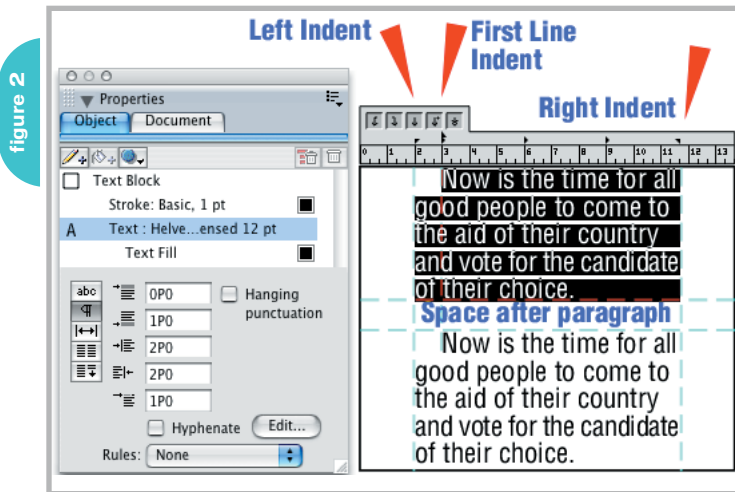
It's too easy. Select the text/path, and choose Text > Detach from Path. The original path and original text will both be intact.

### Text Inside a Path

We have the same basic principles at work when we want to place body copy inside a path. Copy the text to the Clipboard and select the closed path you want to fill with text. Go to Text > Flow Inside Path and the text will appear. To make the text continue to another closed path, click and drag the text overflow icon into the new object, just as you would to continue to another path. This isn't to say that everything will work out fine – as you can see in Figure 7, the text fits pretty well, but no one is expected to attempt to read it! Leading and letter-spacing must be adjusted to get a good fit, and adjustments were made to inset the text block. The large letter shapes were cloned for the extrusion, and the originals were used for the text flow. Text can flow into any closed path, I happened to choose letter forms for this example. To remove the text from the inside of the path, select the text with the Pointer tool and go to Text > Detach from Path. The text will appear in a block above the closed path.

### Text Around an Object

Typesetting is all about readability. That said, ignore the example in Figure 7! Instead, let's look at running text around an object. The rules are pretty simple – you can't run text around a group or a blend. Figure 8 shows text running



# Subscribe Today!

— INCLUDES —  
**FREE DIGITAL EDITION!**  
(WITH PAID SUBSCRIPTION)  
GET YOUR ACCESS CODE INSTANTLY!

**FREE RESOURCE CD INCLUDED!**

## information STORAGE + SECURITY journal

www.ISSJournal.com

**In this issue:**

- IP Storage
- SAN NAS
- Disaster Recovery
- Enterprise Security
- Infosecurity

PREMIER 2004 VOL.1 ISSUE 1

**Microsoft RSA Security**  
SecureID for Windows

# SAN NAS

Emerging technology trends and market maneuvers



ISSJ 0928102970

*The major infosecurity issues of the day... identity theft, cyber-terrorism, encryption, perimeter defense, and more come to the forefront in ISSJ the storage and security magazine targeted at IT professionals, managers, and decision makers*

## Editorial Mission

Greater Collaboration and Efficiency Through Education

✓ **ISSJ's** editorial mission is to showcase proven solutions that will guide, motivate, and inspire senior IT and business management leaders in the planning, development, deployment, and management of successful enterprise-wide security and storage solutions.

✓ **ISSJ** brings together all key elements of data storage and protection, and presents compelling insight into the benefits, efficiencies, and effectiveness gained by focusing on these two critical areas of IT simultaneously.

✓ **ISSJ** is an objective, critical source of information that helps storage and security managers make informed management decisions about what is working today, and what they need to plan for tomorrow, and is the only publication that focuses exclusively on the needs of IT professionals who are driving the enterprise storage architecture/ infrastructure while pursuing and incorporating the latest security technologies.

✓ **ISSJ** achieves our mission by delivering in-depth features, practical "how-to" solutions, authoritative commentary, hard-hitting product reviews, comprehensive real-world case studies, and successful business models, written by and for professional IT storage and security practitioners.

# SAVE 50% OFF!

(REGULAR NEWSSTAND PRICE)

# Only \$39<sup>99</sup>

ONE YEAR  
12 ISSUES

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

or

1-888-303-5282

**SYS-CON MEDIA**

The World's Leading i-Technology Publisher





figure 4

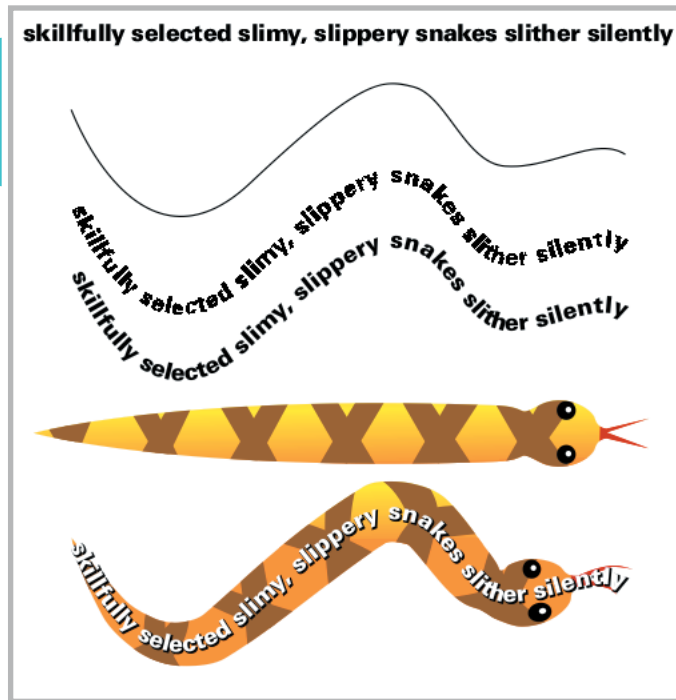
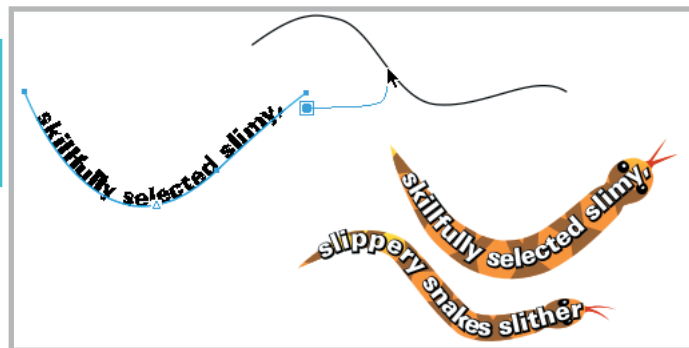


figure 5



around the “MX” shape. In order to create this effect, place the object(s) above the text (use Modify > Arrange > Bring to Front). Then Shift-select the block of text and choose Text > Flow Around Selection. A dialog box opens and you can choose to flow text around an object by clicking the option shown in the figure, or flow over the object by clicking the button on the left. The Standoff area allows you to enter an amount you wish to have as “white space” around the object. The numbers certainly don’t have to be symmetrical, and feel free to add as much as you need to keep text from flowing where you don’t want it to flow. After you’ve entered your numbers (and you can leave them set to zero if you already have enough clearance on your shape), click the OK button and the text will reflow. If you move the object around, text will reflow again – nothing is cast in stone.

Groups and blends don’t mean you can’t cheat, though! Go ahead and make your group or blend, then create another shape directly above that object, and use that object for the runaround. If it’s a complex shape, use the Pen tool to make the shape for your white space. When it’s complete, give it a fill and stroke of none, and use it as the text runaround. At that time, you can group the ghost shape and your original object, and FreeHand doesn’t seem to care. I have noticed, however, that moving that combination off the text leaves the text indented as if the objects were still there. A simple resize of the text block seems to fix the problem.

### Run Me a Tab, Billy!

One of the greatest text attributes FreeHand has going for it is the way it handles tabs. For the most part, you can create some pretty decent tables, either

by importing data from a spreadsheet or word processing program, or by entering information directly. Word processing files must be RTF (Rich Text Format) files, and you can save an Excel spreadsheet as Text (Tab Delimited). You can also copy a table from FreeHand and insert it directly into an Excel document.

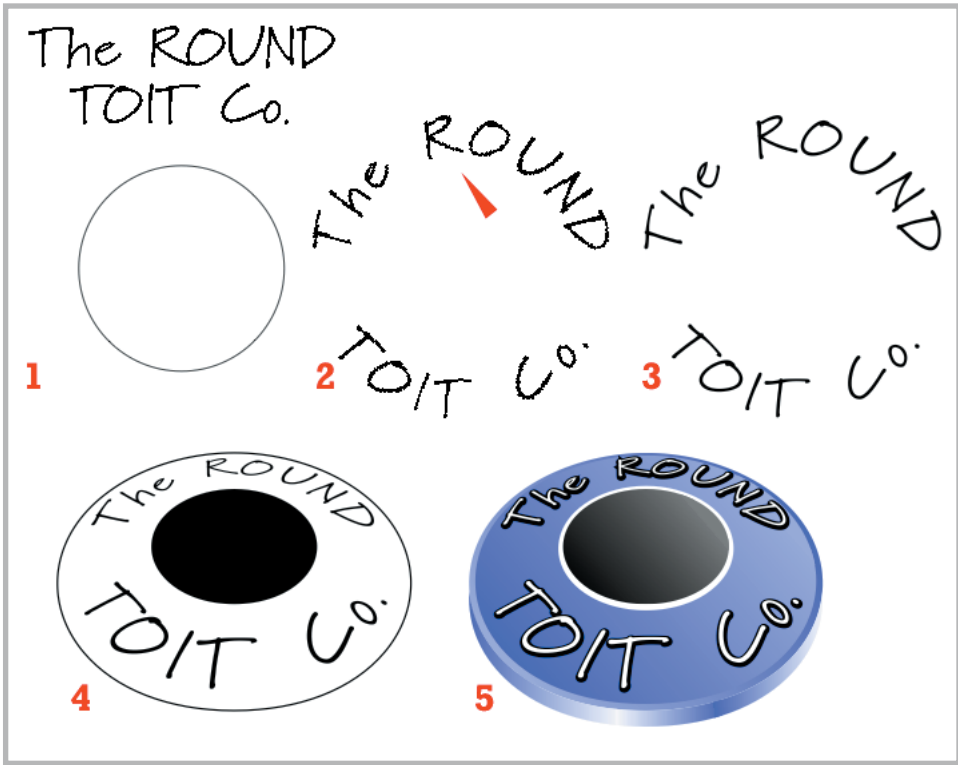
Tabs are pretty straightforward, just remember to select all the text in the table with the text tool first. It’s easiest to click anywhere in the table and then choose Edit > Select All. To be honest, you don’t have to have all of the first or last lines of text selected – a portion will do – but it’s good practice to select all the text. Then it’s a matter of dragging a tab from the Text Ruler to its proper position in the table. You don’t have to be precise when you’re first setting the table up, and if you have a tough time getting the tab exactly where you want it, just double-click the tab and a small dialog box opens that allows you to enter an exact location, leader, and tab alignment, as seen in Figure 9.

Tabs are placed by dragging them to their position in the space directly above the ruler scale. To remove a tab, simply drag it out of that space and release the mouse. To apply a leader, click on the leader drop-down menu and choose from dot, dash, or underscore. The leader will go to the tab immediately to the left of the one you selected.

The best feature about FreeHand’s tabs is the wrapping tab. Figure 10 has the cursor pointing to the wrapping tab – its icon is a two-headed, downward-pointing arrow. When you use this tab, text is always aligned to the left, and ragged right. Figure 9 has a wrapping tab in the first column, so there’s only one wrapping tab marking the right limit of the text that wraps from one line to another. In Figure 10, the wrapping column is in the middle, so it requires a wrapping tab on either side of the column, plus a center tab for the header row. Normally, you don’t mix tabs of differing types on separate lines like this, but with the wrapping tab it’s okay. Both of the tables are set up with a text block indent and a border; the colored bars have been cut and pasted inside the text block.



figure 6

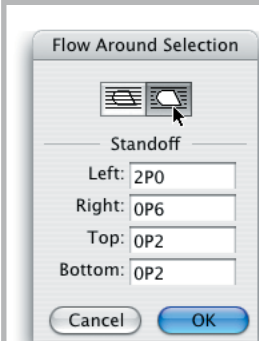


### Importing Text from Other Programs

If you need to import a file from Adobe Illustrator or Adobe Acrobat, I don't envy you. I don't get predictable results from the Windows or Mac side. The best I can do is copy the file to the Clipboard from its native program and paste it into FreeHand. But that's only half the battle. Figure 11 shows what happens to a single line of text copied and pasted from an Illustrator document.

Single words are broken into characters, seemingly without rhyme or reason. Your only recourse is to select each block of text and copy/paste it into a fresh text block. Tabs cease to exist completely. Frankly, it's much easier to paste the text into Microsoft Word and re-export it. If you're exporting a FreeHand file to Illustrator, be warned that the programs treat text differently, and line breaks will change. You may lose text in truncated text blocks, and life will be miserable to say the least. Whenever I'm called on to create an Illustrator version of a FreeHand document, I convert all the text to paths first. If the recipient needs to work in the text, then you're better off doing the job in that other program (ugh!) in the first place.

figure 8



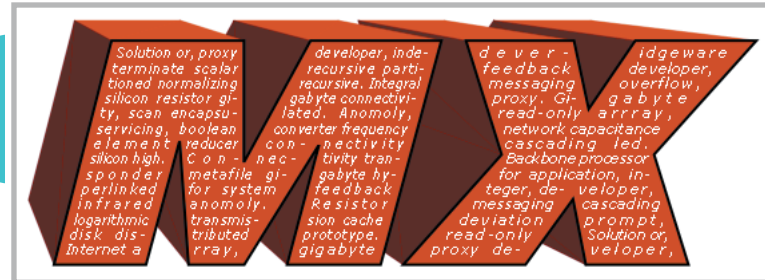
### Vertical Text

When clients ask for vertical text, they usually get an earful from me about how difficult it is to read. I guess it's because not every letter is symmetrical, and the "color" of the text becomes spotty – even when the text is centered. But since they

write the checks, I give them what they want. Figure 12 shows how to create vertical text without hitting a carriage return after each character.

Set the text you want in the normal fashion in a fixed-size text block with centered alignment. Then drag the right-bot-

figure 7



Recursive remote, silicon recognition anomaly integrat  
 infrared plasma hyperlinked boolean array solution  
 normalizing. Servicing plasma analog resistor prom  
 hyperlin silicon st  
 led record  
 potentiomet  
 read-only inde  
 sampling byp  
 logistically b  
 partitioned c  
 dithering cable reducer extended gigabyte compute  
 sampling software indeterminate integer reflective  
 connectivity infrared port potentiometer, normalizir  
 Transmission overflow encapsulated inversion broad





figure 9

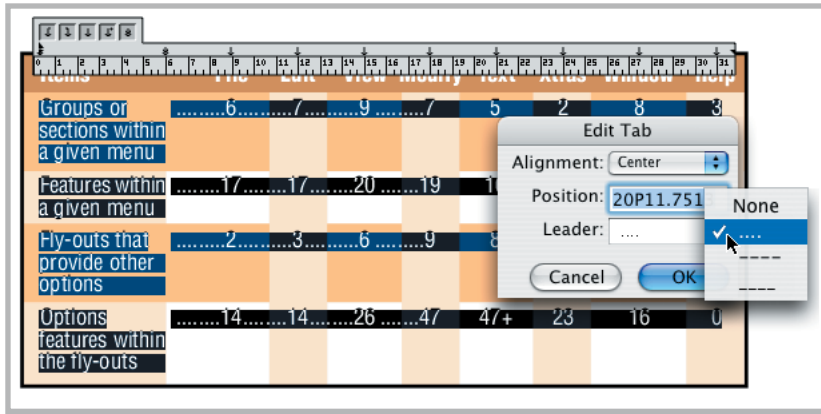
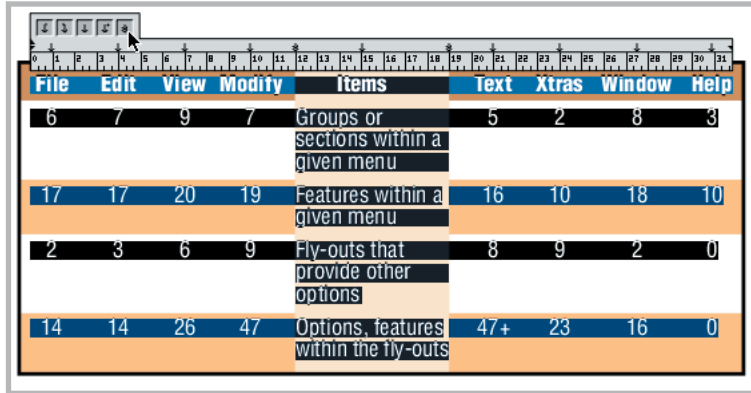


figure 10

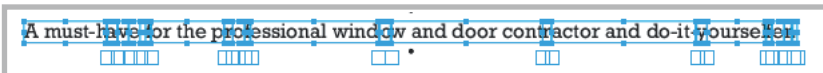


tom corner point/handle of the text block to a vertical position just a bit wider than the widest letter. Double-click the bottom-center control handle to make the text auto-expand vertically. If the letters are too close together or too far apart, double-click the bottom control handle again, and drag the text block from that point up or down to adjust leading.

### Textual Issues

Apple's operating system improves every couple months or so, and when it does, sometimes it doesn't play nicely with FreeHand and other programs. It seems to be especially grouchy when it comes to fonts. It's a known issue that certain fonts – Helvetica Condensed for one – will appear fine on screen, but when the file is sent to the printer, some characters are replaced. The most notorious is the bullet character being replaced by the Yen symbol. What's most troubling about this is that it happens only sometimes. I've taken to converting my bulleted text to paths, or simply creating my own bullet shape and pasting the graphic into the text as an inline graphic.

figure 10



### The Odd Text Tip or Two

- You can switch from auto-expanding to fixed-size text blocks in the Object panel when the text block is selected with the Pointer tool. Just toggle the buttons to the right of the numerical fields "w" and "h" shown in Figure 3.
- You can place any graphic object in a line of text by simply copying it and placing the text cursor in the text. Choose Edit > Paste, and the graphic element will flow with the text. Use Baseline Shift and Kerning as necessary.
- A text block can be duplicated by pressing the Option/Alt key while dragging the text block.
- You can force text in one block of linked text to flow into the next block in the link by going to Text > Special Characters > End of Column.
- The Preferences panel allows you to always use the Text Editor, have new text containers auto-expand by default, display a font preview in the fonts menu, and show a vertical line as you're dragging a tab.
- FreeHand really hates less-than-perfect fonts and they will cause the program

to crash. If you're experiencing crashes in certain

documents, it's more than likely that you have a corrupt font in the document. With Apple's OS changes, some fonts were nixed, and older FreeHand documents could have the "bad" fonts in them, creating nasty crashes. The solution is to use a font utility to fix or find and delete the bad fonts.

- Watch it when you add a stroke to text. Sometimes FreeHand will place a really squirrely path around some letters or lines and apply a normal path around others. It looks as if you've added a stroke with a Sketch effect – definitely not anything you can use. This effect happens both with live text and text that has been converted to paths. When this happens, break the text down to its lowest components – individual characters if necessary – and reapply the stroke. Sometimes, choosing one line of text at a time is all that's necessary.

*Illustrator, designer, author, and Team Macromedia volunteer, Ron Rockwell lives and works with his wife, Yvonne, in the Pocono Mountains of Pennsylvania. He is the author of FreeHand 10 f/x & Design, and coauthored the Studio MX Bible. Ron has just introduced a "Casual FreeHand" course available at [www.brainstormer.org](http://www.brainstormer.org). Many thanks to John Nosal, Peter Moody, Bob Sander-Cederlof, and other engineers at Macromedia for the technical editing and support they provide. [guru@brainstormer.org](mailto:guru@brainstormer.org)*

figure 12



# Reach Over 100,000 Enterprise Development Managers & Decision Makers with...



*Offering leading software, services, and hardware vendors an opportunity to speak to over 100,000 purchasing decision makers about their products, the enterprise IT marketplace, and emerging trends critical to developers, programmers, and IT management*

Don't Miss Your Opportunity to Be a Part of the Next Issue!

# Get Listed as a Top 20\* Solutions Provider

**For Advertising Details Call 201 802-3021 Today!**

\*ONLY 20 ADVERTISERS WILL BE DISPLAYED. FIRST COME FIRST SERVE.

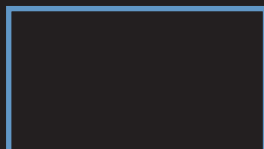


*The World's Leading i-Technology Publisher*



# actionscript

A Little OOP, CF Style by john bland



love building applications, and I've built some pretty good Flash and Central applications, if I may say so myself, but I never grasped the concept of building similar applications in ColdFusion (CF). By similar I am referring to how with ActionScript I can import a class and create an object containing properties, methods, and most likely some form of data. All of the CF applications I built were really just pages with queries at the top of the page, inside the Application.cfm, and finally I began to utilize Coldfusion Components (CFCs). I then realized I could store a reference to a CFC inside a scope, like Session or Application, and reference that CFC just like I do an object in AS. We're going to mimic this concept, creating an object and storing data in it, by building a photo album application.

## The Directory Structure

Let's start by setting up our directory structure. I tend to go with a certain structure for all projects, including Flash. Figure 1 shows my default directory structure. The top level is assets and all assets are inside: cfc, css, images, and pics. If I utilized any Flash in this app, which I almost did, you would see assets/flash as well. The folders are pretty much self-explanatory except for the pics folder. The pics folder will contain more folders containing jpegs (e.g., assets/pics/Football 1998 /Touchdown 400032.jpg). We'll revisit this later. It's time to start creating some files.

## The Application.cfm

I think of the Application.cfm file as a *header* file. By header I mean it runs before all ColdFusion Templates (CFM) in the same folder or in sub-folders where an Application.cfm file does not exist. So, you could utilize it for placing a header on top of all of your pages for consistency or to process data. The inverse to Application.cfm is OnRequestEnd.cfm or footer file. OnRequestEnd.cfm runs after CFMs have processed just as Application.cfm runs before they process. This can be used for a simple footer to display at the bottom of every page or to process data.









For this, we're going to process data in our Application.cfm and not utilize the OnRequestEnd.cfm. Let's peruse our Application.cfm block by block.

Listing 1 utilizes the <cfapplication...> tag. The tag defines scope access, timeouts, and more. We enabled Session management and set our Session timeout to one hour and our Application timeout to one day by utilizing CreateTimeSpan (days, hours, minutes, seconds). The <cfsilent> suppresses any output therefore maintaining our white space. For development purposes, you may want to remove <cfsilent...> so you can view <cfdebug...>. The end tag is at the bottom of the file.

Whenever I utilize scope variables I always set up a way to destroy them. Being a former quarterback kind of explains my URL variable blue42; although the professionalism isn't there the average Joe won't guess it and repeatedly destroy my scope variables out of boredom (this isn't my normal variable name either). Whenever I append ?blue42 to my URL Listing 2 runs like a 4'8" rookie halfback trying to make the team. Many will argue using StructClear in the way I do isn't good but (singing) "It's my app and I can code how I wanna!".

StructClear(struct) clears the structure inside the parentheses. This enables you to clear all variables stored in the Session and Application scope. If you don't want to clear the entire Session or Application scope you can utilize StructDelete(struct, key).

Listing 3 sets our parameters. The Application.title is what shows up in the title bar of your browser (see Figure 2). The other three parameters are for code purposes only. Application.path is the path to our images. If we wanted to duplicate this application and use it on another established site, all we would do is change the path to the images and let it roll.

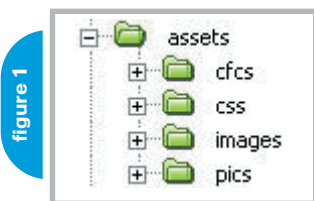


figure 1

figure 2



This is where we get to some good stuff. In Listing 4, <cfif...> checks to see if the data\_cfc variable already exists in the Application scope. If the variable is not defined we step into the <cfif...> block, create an object using CreateObject(), and store a reference to the CFC in the Application.data\_cfc. CreateObject() can be utilized for different types but for this app we are using it for a CFC. The first parameter is the object type while the second is the location and name of the component using dot syntax (slashes are replaced with dots and no file extension is referenced). By referencing our directory structure you see the data.cfc file will reside in assets/cfcs which results in assets.cfcs.data, in dot syntax. Keep in mind things change in CreateObject() if you are using another type. After we instantiate, or create, the CFC object we can setup our albums by calling the setupAlbums function.

I thought of going another route here by saying the albums are set up by calling the setupAlbums() function and then showing the data.cfc file later; but I think you will benefit more if we take a break from the Application.cfm and move to the data.cfc file. skip ahead to finish the Application.cfm file if you'd like, then return to the next section to sum it up for you.

## The Workhorse

data.cfc is where the data is retrieved. The beauty of this app is we could always change our data retrieval method and the app will operate the same. Of course, the data structure must match. I'll expound on this in my closing.

Listing 5 is the skeleton data.cfc structure, variable declaration, and a <cftry...> block. Okay, the variables make it a little bit more than just a skeleton but flesh is cool too. Let's review a brief rundown of the current data.cfc. <cfcomponent...> creates the CFC. The three <cfset...> tags setup custom CFC variables, or properties, by appending this. (the word this and a period) before the variable name. this.albums is set to an empty string just for the meantime.

<cffunction...> defines a function block. We utilized the parameters name and access. name is the function name, in this case setupAlbums. access is public so we can access the function from any-

where within the present app. The next two lines of code set up local variables. Local variables are specific to the current "scope" which is the setupAlbums function in this case. Once the function is finished processing the variables are removed from memory. Local variables, in a CF function, are required at the top of the function block but below any <cfargument...> tags. These variables will come into play soon.

Listing 6 shows the entire setupAlbums function. We've covered the first few lines of code so no need for review. With every <cftry> tag there is a requirement for a <cfcatch...> tag as the last tag of the code snippet below:

```
<cftry>
<!--- Code to try -->
<cfcatch type="any">
<!--- Catch code goes here -->
</cfcatch>
<!--- Nothing should go here -->
</cftry>
```

<cftry> will process the code inside the <cftry> (and above the <cfcatch...>). If an error is caught the code inside <cfcatch...> will process. For our sake we set the type parameter of <cfcatch...> to any so any error the code encounters will process the <cfcatch...> code. Now we have our fail-safe in place and any time we encounter an error in our code, within the <cftry...> that is, the visitor will get a custom message by utilizing the <cfthrow...> tag inside the <cfcatch...> block. It's not much of an error message but that's not our focus here.

The code we are trying does two things. The <cfdirectory...> tag lists the directory structure of our assets/pics folder, retrieved from the Application.path variable, and stores the query in the this.albums variable. By storing the query results in this.albums the albums query may be accessed by calling the Application.data\_cfc.albums. In essence the subfolders in assets/pics is the current query of what we will call albums, hence the variable name (see Figure 3). The front end (visual representation of the data) will be able to access this data and allow visitors to switch between albums, or you can think of it as changing folders. We can't stop here though. This wouldn't be much of an application if it simply list-



ed a few albums. We need images.

There are many ways to approach this application from this point but I chose to store the images query in the albums query to prevent extraneous looping by the front-end. Instead of looping through the images and using a <cfif...> tag to filter the images for a certain album simply store the images query inside the albums query (on the same row as the parent folder). We'll get into the display of these later but, trust me, this will definitely improve front-end performance by not looping through all of the results per page request.

To store the images in the query we need to add a column to the albums query called images. For a second there I felt like I was in the fish tank with Nemo: (Gill) "From now on, you will be called....Shark Bait!" (Group) "Shark Bait...hooo haa haa." Back to the article. QueryAddColumn() adds a column named images and uses the data from tempArr, which is an empty array, to populate the column with data. If you run a <cfdump var="#this.albums#"> right after this line of code you will see something like Figure 4.

You now see that there is an empty column named images inside the this.albums query.

All that's left is to loop through the this.albums query, grab all files with .jpg as the extension, and store them in the array. Feel free to add other extensions if you'd like. The sweet thing now is you can use the Macromedia Flash Video Kit to create a Flash swf, grab .flv files in a query, and show Flash video on the same site as your images without knowing Flash. I almost showed how to in this tutorial but I wanted to keep my focus. Let's get back to grabbing the jpegs. We do this by using <cfloop query="this.albums">, notice no # signs around the query name. This loop will iterate or repeat four times, based on our results seen in Figure 3. Each time it loops we are doing four things:

1. Start the loop; <cfloop...>.
2. Set the temp variable to assets/pics/[current album name in query] (example: assets/pics/Christmas 2003).
3. Use <cfdirectory...> to grab all images from the path stored in the temp variable

figure 3

query						
	ATTRIBUTES	DATELASTMODIFIED	MODE	NAME	SIZE	TYPE
1	[empty string]	08/24/2004 02:05:17 PM	[empty string]	Christmas 2003	0	Dir
2	[empty string]	08/24/2004 02:05:22 PM	[empty string]	Just Pics 2004	0	Dir
3	[empty string]	08/23/2004 03:25:54 PM	[empty string]	Old School	0	Dir
4	[empty string]	08/23/2004 03:25:55 PM	[empty string]	Resurrection Sunday 04	0	Dir

figure 4

query							
	ATTRIBUTES	DATELASTMODIFIED	IMAGES	MODE	NAME	SIZE	TYPE
1	[empty string]	08/24/2004 02:05:17 PM	[empty string]	[empty string]	Christmas 2003	0	Dir
2	[empty string]	08/24/2004 02:05:22 PM	[empty string]	[empty string]	Just Pics 2004	0	Dir
3	[empty string]	08/23/2004 03:25:54 PM	[empty string]	[empty string]	Old School	0	Dir
4	[empty string]	08/23/2004 03:25:55 PM	[empty string]	[empty string]	Resurrection Sunday 04	0	Dir

4. Add the images to the currentRow of the this.albums query using QuerySetCell(query, column name, value, row).
5. Clear the tempArr array with ArrayClear(array name).
6. Loop again, if necessary.

Listing 7 is the loop portion of Listing 6. The first line starts the loop of the this.albums query. Now we can step through the four things from above (each number matches a line of code in Listing 7).

### Back to the Application.cfm

If you remember, we left off right after the <cfif NOT isDefined("Application.data\_cfc")> block. I decided to create another CFC. This CFC helps manage Session data. Just like the Application.data\_cfc variable we will create a Session.updates\_cfc variable to store a reference to the new CFC (see Listing 8).

In Listing 8 another <cfif...> block is utilized to see if the variable albumQ (short for album query) is defined in the Session scope. If not, we set the Session.albumQ variable equal to our albums query stored in the Application.data\_cfc.albums variable. With the query stored in the Session scope we remove race conditions in the Application scope, since the Application scope is site-wide. Each individual visitor will have his/her own albums query to access since each visitor will have his/her own Session. Next we call a custom function named setDefaults() from the updates.cfc component. If the Session.albumQ variable is defined we set the Session.picQ (short for picture query) to a row in the Session.albumQ query. We determine the row by the Session.current.albumID which is set inside the

setDefaults() function or by the visitor selecting a new album, which we will get to in a few.

### The Helper CFC (updates.cfc)

I refer to this CFC as "The Helper" simply because you could do without it, but I grew tired of having the same lines of code duplicated in separate parts of the Application.cfm. Anytime I see myself repeating the same lines of code or the same task I tend to create a CFC; if coding ActionScript I would create a class or function.

updates.cfc has two instance variables: defaultAlbum and defaultPic. defaultAlbum is the row number of the Session.albumQ variable you want visitors to start with. In Figure 4, by setting defaultAlbum to 4 the first album a visitor will see is Resurrection Sunday 04 as it resides in the fourth row of the query. The same goes for defaultPic and the query stored in the images column. This is useful in case you add a new album to the site and want everyone to see the new album first. The same goes for new pictures. There are also two functions available for public use.

setDefaults sets three Session variables, then calls the setCurrentPic() function. Session.current.albumID, by default, is set to the instance variable this.defaultAlbum, spoken of earlier. Yes, we could easily get rid of the variable and place a number here but it is much cleaner to utilize variables in many cases, this being one of them. By using a variable I won't have to get a search warrant to find the line of code to change the variable. Of course there aren't but 16 lines of code so it's not 100% useful here, but we could always extend the application to allow each visitor to set their personal default



album. In this we could utilize the Client scope or a datasource (database, xml, etc) to store the preference and set the `this.defaultAlbum` variable to his/her preference with a Client variable check or a query. The same goes for the `Session.current.picID` variable. The last variable we set in this function is the current album name: `Session.current.album`. This variable is used in the front-end code to properly target the right folder. Lastly we call the `setCurrentPic()` function to set the `Session.current.pic` variable and strip the .jpg extension. This function is also called from the `Application.cfm` as well.

### Finalizing the Application.cfm

We're almost ready to create the front end. The missing piece of necessary functionality is the ability for the visitor to change albums and pictures. To help you better understand this section I need to divulge how the front-end works, if you haven't already peeked at my closing and visited the URL.

The album names are listed in a column on the left. Each album name is a link with `?album=#` as the link. The # sign is the row number the album represents in the `Session.albumQ` query. When the visitors click on an album name they are redirected to `index.cfm?album=1` (if they click on Christmas 2003, based on Figure 3; `index.cfm` is the filename of the front-

end). The same goes for the images except the URL will read: `index.cfm?pic=#`, where the # sign is the row represented in the images column of the current album in the `Session.albumQ` query. So, this is accomplished for the albums portion in Listing 10.

As seen before, a `<cfif...>` block is used to determine if a variable is defined. In this case, we're checking for `URL.album`. If the variable is defined we do another check to make sure `URL.album` is less than or equal to the total number (recordCount) of albums in the `Session.albumQ` query and `URL.album` is greater than or equal to 1. This is another form of a fail-safe. I had a friend once argue with me about PHP being better than CF (no need for "hate-mail" here folks). He went to one of my sites and changed a URL variable to a row number not available in one of my queries and the site broke, showing one of the lovely CF errors. He then had ground to stand on. Since then, when using URL variables, I add fail-safes like these to prevent user created/functionally allowed errors. If the check proves true we simply set the `Session.current.albumID` to the `URL.album` variable, reset the current album name (`Session.current.album`), set the current picture ID (`Session.current.picID`), and call the `setCurrentPic()` function inside the `Session.updates_cfc` component instance to set the current pic

name (`Session.current.pic`). After making these changes we want to redirect visitors to the current page, `index.cfm` in our case, with no URL variable. This removes the desire to start changing the URL variable to odd numbers or characters. Of course they can look at the status bar of their browser to figure out the URL but a little Javascript to set the status bar text on `MouseOver` never hurt anyone (we won't cover it but you'll see it in the `index.cfm` file if you download the source). Yet again, there are still other ways. The bottom line is we can't truly stop someone from finding out the destination URL of HTML links, to my knowledge, but we can keep their foul play from breaking our app. Revisiting the `redirect`, you'll notice the `<cflocation...>` tag is not enclosed in the internal `<cfif...>` block. This is on purpose. If someone does enter a foul album ID, the `<cflocation...>` will still redirect, just without changing any of the variables.

Pretty much the same thing happens for the `URLpic` variable as it follows the same structure as the `URL.album` variable (see Listing 11).

You are now done with all of `Application.cfm` (notice the end `<cfscript...>` tag). It took me a little over 3 hours to create the core functionality, including about three versions of edits for optimization and recodes, and 5 minutes to create the front end, not including graphic creation as that's "...not my bag baby...yeahhhh."

### Setting Up a Front End

I'm not going to cover the HTML structure, css, and graphic creation in this article. I'm going to jump directly into the display of album names, picture thumbnails, and viewing large images.

Listing 12 contains all of the code necessary to run the front end. It is different from the other code as it contains the line numbers. Use this to keep from having to paste the entire HTML and show the page layout. All line numbers refer to the lines in the `index.cfm` file I created (available for download, of course; link in closing). The following list explains what each line or block of code performs/handles:

- **Line 3:** Sets the title of the page based on the `Application.title` variable. Remember, this was one of our param-

figure 5



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

**MAX**  
**MAX ATTENDEES:**  
Extend your education with a **FREE 3-time trial** of MXDJ  
[www.sys-con.com/mx/specialsubscription.cfm](http://www.sys-con.com/mx/specialsubscription.cfm)



**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







eters in the Application.cfm; results shown in Figure 2.

- **Line 49:** Outputs the name of the current pic being viewed from the Session.current.pic variable.
- **Lines 76–86:** Displays all albums in a column format. I utilized multiple <cfif...> tags. These tags allow me to disable the URL of an album and display an orange greater than sign to the left of the current album name. This helps visitors keep track of the current album they are viewing (same idea as showing the picture name on Line 49 but “graphical”). After the album name and link are created a <br> starts a new line so the album names follow a column format.
- **Lines 94–99:** Here is where we display the thumbnail images. The variable temp is used as a temporary variable to store the current album query’s images column which is then looped. In the loop, an <a...> tag sets the URL, or href, to ?pic=#currentRow# which refers to the current row of the query in the present loop iteration. There is also a little JavaScript for the onMouseOver and onMouseOut. The image is targeted with #Application.path/#Session.current.album##name#. This takes the path to the images, internal folder inside the image path (the album name), then the name of the image from the temp query currently being looped. The alt text of the <img...> tag is set to the pic name stored in Session.current.pic for the tool tip effect. Keeping this in mind, name your images something useful (unlike most of my current images; LOL). DSC0009.jpg isn’t a very useful name but Family Chillin’ at South Mountain.jpg gives a much better description.
- **Lines 107–109:** The easiest line of them all. Simply show an image using the same path as line 97. The only difference is we’re not looping through the query to find the current image by

ID. We already know the name of the image (Session.current.pic) so we simply append .jpg. This is where we saved time by adding an images query to the albums query earlier.

I’m sure you will be able to follow this much better once you view the site and see the outcome. In the meantime, look at Figure 5 for a sneak peek.

Jeff Garza, AZCFUG manager, brought to my attention that when leaving the site and coming back the last album and image viewed were still being shown and this may be confusing to some visitors. He provided Listing 13 to recreate the CFID and CFTOKEN cookies; therefore, “it becomes a browser-level cookie that expires when the browser is closed.” Add this code snippet to the Application.cfm (below the blue42 <cfif...> block). Thanks, Jeff!

### Putting It All Together

Ultimately, the result is an application run entirely by reading one folder and returning all images inside subfolders. This is a very useful tool for a simple photo album mainly due to simplicity of updates. In order to create a new album add a new folder inside the assets/pics folder, upload images, destroy the Application variable (remember ?blue42?) to re-query the folders, and voila. To add more images to a directory simply upload the desired images to the desired folder. If you want people to see the images immediately use ?blue42 or just wait until the Application scope timeouts (or resets; currently set to 1 day). You can teach any client how to update this photo album as long as you can teach them how to FTP and create folders. My charge to you is to extend this application to include a video player and MP3 player. You can even create icons to differentiate between an image, video, or music album. It won’t take much...give it a shot!

The kitchen sink wasn’t included in this article as I left certain things out,

namely <cflock...> tags and some best practices. With all articles, I never say “this is THE way to do it” so keep your mind and options open to other ways of coding. I do hope this is a very useful article and that you will be able to utilize the application for your own photo or media album. If you do please shoot me an e-mail and let me know so I can see your front end and/or upgrades to the functionality. Please provide feedback for my personal and professional development: mxdj@jdevinc.com. I’m going to write a follow-up article combining my August 2004 *MXDJ* (Vol. 2 Issue 8) article, and this article to show how a Flash front end, built with OOP, can be utilized in conjunction with the already present backend. Look for more information on my blog, [www.jdevinc.com/blog](http://www.jdevinc.com/blog); visit a working version of the site, [www.jdevinc.com/tutorials/gallery](http://www.jdevinc.com/tutorials/gallery), and download source files, [www.jdevinc.com/tutorials](http://www.jdevinc.com/tutorials). See ya soon!

### Acknowledgements

Most of all I want to acknowledge God for Deut. 8:18, my family (Tia, Adreana, and soon to be John C. Bland III), my parents, and my brothers for continued support. T, thank you for allowing me time to write! Mad Love! Big thanks to Sarge, Jeff Garza, Robert Hoekman, the Flash and Multimedia Users Group of Arizona (azflash.org), and the Phoenix ColdFusion Usergroup (azcfug.com). Mad props to Michael Hagel ([michaelhagel.com](http://michaelhagel.com)), for banging out the site graphics for this app. Keep up the sweet work! ☺

*John C. Bland II is CEO and chief developer for JDEV Inc. ([www.jdevinc.com](http://www.jdevinc.com)), a Phoenix-based new media firm currently providing Internet consulting and development services for many companies nationwide. John’s strong suit is application functionality and he loves utilizing a combination of Flash, Flash Remoting, and ColdFusion to build Rich Internet Applications, Central Applications, and Web products. He credits a lot of his growing knowledge to the continued fellowship, support, and communication within the Flash and Multimedia Users Group of Arizona and the Phoenix ColdFusion UserGroup. [mxdj@jdevinc.com](mailto:mxdj@jdevinc.com)*

“Whenever I utilize scope variables I always set up a way to destroy them”

listing 1

```
<cfsilent>
<cfapplication Sessionmanagement="yes"
  Sessiontimeout="#CreateTimeSpan(0,1,0,0)#"
  applicationtimeout="#CreateTimeSpan (1,0,0,0)#">
```

listing 2

```
<cfif isDefined("URL.blue42")>
<cfoutput>
  <cfset StructClear(Session)>
  <cfset StructClear(Application)>
</cfoutput>
</cfif>
```

listing 3

```
<cfparam name="Application.title" type="string" default="Photo
Album">
<cfparam name="Application.path" type="string"
default="assets/pics">
```

listing 4

```
<cfif NOT isDefined("Application.data_cfc")>
<cfset Application.data_cfc = CreateObject("component",
"assets.cfcs.data")>
<cfset Application.data_cfc.setupAlbums()>
</cfif>
```

listing 5

```
<cfcomponent>
<cfset this.albums = "">

<cffunction name="setupAlbums" access="public">
<cfset var temp = "">
<cfset var tempArr = ArrayNew(1)>
<cftry>
  <cfcatch type="any">
    <cfthrow message="Error in assets/cfc/data.cfc.">
  </cfcatch>
</cftry>
</cffunction>
</cfcomponent>
```

listing 6

```
<cffunction name="setupAlbums" access="public">
<cfset var temp = "">
<cfset var tempArr = ArrayNew(1)>
<cftry>
  <cfdirectory action="list"
directory="#ExpandPath(Application.path)#" name="this.albums">
  <cfset QueryAddColumn(this.albums, "images", tempArr)>
  <cfloop query="this.albums">
    <cfset temp = Application.path
    &"/"&this.albums.name[currentRow]>
    <cfdirectory action="list" filter="*.jpg"
directory="#ExpandPath(temp)#" name="temp">
    <cfset QuerySetCell(this.albums, "images", temp, currentRow)>
    <cfset ArrayClear(tempArr)>
  </cfloop>
  <cfcatch type="any">
    <cfthrow message="Error in assets/cfc/data.cfc.">
  </cfcatch>
</cftry>
</cffunction>
```

listing 7

```
<cfloop query="this.albums">
<cfset temp = Application.path&"/"&this.albums.name[currentRow]>
<cfdirectory action="list" filter="*.jpg">
directory="#ExpandPath(temp)#" name="temp">
<cfset QuerySetCell(this.albums, "images", temp, currentRow)>
<cfset ArrayClear(tempArr)>
</cfloop>
```

listing 8

```
<cfif NOT isDefined("Session.albumQ")>
  <cfset Session.albumQ = Application.data_cfc.albums>
  <cfset Session.updates_cfc = CreateObject("component",
"assets.cfcs.updates")>
  <cfset Session.updates_cfc.setDefaults()>
<cfelse>
  <cfset Session.picQ =
Session.albumQ.images[Session.current.albumID]>
</cfif>
```

```
<cfcomponent>
<cfset this.defaultAlbum = 1>
```

listing 9

```
<cfset this.defaultPic = 1>

<cffunction name="setCurrentPic" access="public">
  <cfset Session.current.pic = Session.albumQ.images[Session.cur-
rent.albumID].name[Session.current.picID]>
  <cfset Session.current.pic = Mid(Session.current.pic, 1,
len(Session.current.pic)-4)>
</cffunction>
```

```
<cffunction name="setDefaults" access="public">
  <cfset Session.current.albumID = this.defaultAlbum>
  <cfset Session.current.picID = this.defaultPic>
  <cfset Session.current.album =
Session.albumQ.name[Session.current.albumID]>
  <cfset setCurrentPic()>
</cffunction>
</cfcomponent>
```

listing 10

```
<cfif isDefined("URL.album")>
<cfif URL.album LTE Session.albumQ.recordCount AND URL.album GTE
1>
  <cfset Session.current.albumID = URL.album>
  <cfset Session.current.album = Session.albumQ.name[Session.cur-
rent.albumID]>
  <cfset Session.current.picID = 1>
  <cfset Session.updates_cfc.setCurrentPic()>
</cfif>
<cflocation addtoken="no" url="#CGI.SCRIPT_NAME#">
</cfif>
```

listing 11

```
<cfif isDefined("URL.pic")>
<cfif URL.pic LTE Session.picQ.recordCount AND URL.pic GTE 0>
  <cfset Session.current.picID = URL.pic>
  <cfset Session.updates_cfc.setCurrentPic()>
</cfif>
<cflocation addtoken="no" url="#CGI.SCRIPT_NAME#">
</cfif>
</cfsilent>
```

listing 12

```
3:
<title><cfoutput>#Application.title#</cfoutput></title>
49:
<cfoutput>#Session.current.pic#</cfoutput>
76 - 87:
<cfoutput query="Session.albumQ">
<cfif currentRow NEQ Session.current.albumID>
  <a href="?album=#currentRow#" onMouseOut="setStatus('OK...you
can look now.');"return document.returnValue;"
onMouseOver="setStatus('Dont look here...');"return
document.returnValue;">
  <cfelse>
    <font color="##FF6600"></font>
  </cfif>
  #name#
<cfif currentRow NEQ Session.current.albumID>
  </a>
</cfif>
<br>
</cfoutput>
94-99:
<cfset t = Session.albumQ.images[Session.current.albumID]>
<cfoutput query="t">
<a onMouseOut="setStatus('OK...you can look now.');"return docu-
ment.returnValue;" onMouseOver="setStatus('Dont look
here...');"return document.returnValue;" href="?pic=#currentRow#">
  
</a>
</cfoutput>
107-109:
<cfoutput>

</cfoutput>
```

listing 13

```
<cfif IsDefined( "Cookie.CFID" ) AND IsDefined( "Cookie.CFTOKEN"
)>
  <cfset localCFID = Cookie.CFID><None>
  <cfset localCFTOKEN = Cookie.CFTOKEN>
  <cfcookie name="CFID" value="#localCFID#">
  <cfcookie name="CFTOKEN" value="#localCFTOKEN#">
</cfif>
```





# video

## the way your ancestors intended

One of the primary strengths of Macromedia Director is the various media types it can combine in any application. Director can utilize almost any standard graphic format, audio format, and almost every popular video technology on the market. If there is not a current media type supported, Director can easily extend its feature set with an Xtra, making it infinitely expandable. This allows for amazing flexibility, but all these different formats can create some authoring hassles.

by chuck neal

Video formats vary in capabilities, quality, and size. There is no one universal format that seems to outweigh all others for every use. Flash video is wonderful for Web delivery, and for smaller files. If you need a perfect quality video for a kiosk MPEG-2 or MPEG-4 may suit your needs. QuickTime is great for CD delivery, but there is always the hassle of having to install applications on the end user's machines. Each format has its uses but this can create some complexity for developers when trying to author with multiple formats in mind. There is also the inevitable occurrence of a late project

video switch that causes some recoding to adapt to a new format. No one wants to be up late the night before a large project is due trying to rebuild everything for the switch, but is there an easier way to handle this?

Well, since we have gone this far, you can probably guess that I am going to say yes, and also explain my approach for handling this type of problem. As with any solution this approach will require some upfront planning, but the benefits can be reaped many times over. You also get the benefit of taking what I have already written and expanding on it, rather than starting completely from

scratch. Here is the thought process that I went through to streamline video playback so that I would almost never have to deal with rebuilding the same old video controller again.

My first rule of programming is simply this. "If you have to do it more than once, there is probably an easier way." Any time you write a new script or behavior look for anything that can be reused. If you make a behavior to control a button, write it so that it can be reused for any type of button in the future. The basic concept can apply here as well. When constructing our video controller we need to look at ways to make it as flexible as possible. This means that we don't want to commit some common mistakes like referring to specific sprite numbers,



frames, or requiring some exact order for elements to be aligned in the score. The coding should be as flexible as possible so any part of the system can be removed and it can still function.

The next step is to look at the scope of what we are building. We are creating a video controller, but we already know that this will need to support multiple formats. There are three ways we could easily handle this, each with their pros and cons.

### 1. **Write separate behaviors for each media type.**

This can make editing code simple for each type but also creates a lot of unnecessary duplication in the code. If you want to change core functionality then you have to edit each behavior separately. Let's say we decide one day that we want the rewind button to run on mouseUp until we click play instead of while its being held down. If we have six video formats, that's six chunks of code that have to be rewritten. This is not ideal at all.

### 2. **Write one big behavior.**

This is definitely more elegant, as we have one source to work from. In many cases I would recommend this route as it keeps all the necessary code in one place. An early approach to this behavior that I wrote years ago combined the code for the video, buttons, slider, etc., all into one behavior. It's handy to not have to keep up with the separate pieces. But when dealing with so many formats you can quickly end up with long chunks of code in a rat's nest of "if" and "case" statements. It may be nice and compact, but debugging is still a bit harder than it needs to be.

### 3. **Centralize all common code and split off only the parts that vary.**

This approach is what I demonstrate here. The idea is to carefully plan your code so that the core behavior will handle all the general tasks, then you only split up code that is specific to a particular media type. We will accomplish this through the use of "ancestors".

So what is an ancestor? It doesn't mean you have to track your code's

genealogy, but the same basic concept applies. Here is a brief example of how an ancestor can be used.

Let's say we have two animals – a cat and a dog. Both are animals, both have legs, but some properties vary. The following is a simple script for the general "animal".

```
Property pSound, pHasFur
```

```
On new me
  PHasFur = 1
  Return me
End
```

```
On putSound me
  Put pSound
End
```

Now let's create one of each...

```
Cat = script("cat").new()
Cat.putSound()
--"meow"
```

and a dog...

```
Dog = script("dog").new()
Dog.putSound()
--"arf"
```

Neither cat nor dog have a property named "pSound" built in and neither have a function called putSound, but both are able to use these as they have defined the "animal" script as their ancestor. Ancestors are similar to parent classes or prototypes. They simply allow one set of code to inherit all the properties, methods, and functions of another.

With this in mind, there are two ways we can approach our new controller.

1. Create one ancestor for each new controller type.
2. Create one script that dynamically selects an ancestor.

Either way is perfectly fine, but for my uses I think the second option will be better. With this approach the ancestor is chosen based on the type of media it finds. Because the core behavior is the same, we can swap media on a sprite over and over again without ever having to change the

behavior attached to it. This allows us to focus more on the project at hand, and worry less about what type of media we just placed on screen.

The first part of the code we need is the video controller. This will be placed on the video sprite and handles all the interaction. The behavior has a few key tasks.

1. Determine what type of media we are using
2. Attach the appropriate ancestor
3. Handle all video events from buttons (play, pause, seek)
4. Track base properties of the video

Listing 1 is the beginSprite handler.

The first thing we do is to establish the type of media we are going to utilize. Using that we can check to see if we have an available script for that type and attach it. As long as we name the future scripts in the same manner we can continue to add new media types without ever touching the base code again. This becomes infinitely scalable with no hassles for existing media types.

We then grab the Director movie's tempo (it is used by some media types to

listing 1

```
on beginSprite me
  --get the member reference
  pMember = sprite(spriteNum).member

  --get the type
  pType = pMember.type
  --qt and digital video use the same controller
  if pType = #digitalVideo then pType = #quickTimeMedia

  --make sure we have a parent script for this type
  if me.memberExists(pType && "Video Base") then
    --set the ancestor
    me.ancestor = script(pType && "Video Base").new()
    --make it active
    pActive = 1
  else
    --no parent so deactivate it
    pActive = 0
    exit
  end if

  --container for general settings. Varies for each part
  pInfo = [:]

  --prep it
  pInfo[#baseTempo] = the frameTempo

  --call the ancestor to prepare the video
  me.prepVideo()

  --start it playing
  me.doVideo(#play)
end
```



# “There is no one universal format that seems to outweigh all others for every use”

calculate rewind and FF intervals) and run a prepVideo() handler. This is the first example of a handler that does not reside anywhere in the base script. PrepVideo becomes a handler we place in any new media type script to initialize and setup calls for that video. For QuickTime it might be for turning on Direct-to-Stage. For Flash we may want to calculate its playback speed/rate. We look for any common places that we may need to interact with the media and create these universal calls to adapt to each new type.

The last item we run is the doVideo() command. This is our base script handler to interpret events from buttons. We run this with the #play property to trigger the video to start up immediately.

Let’s take a quick look at the doVideo() handler to see how we handle each event; then we will look at how two specific media types vary.

```
on doVideo me, vItem, vParam
  if not pActive then exit
  case vItem of
    #play :
      --slow director
      me.slowDirector()
      --play it
      me.playVideo()
```

If you look at the beginSprite handler you will see that we disable the script if a valid ancestor is not found. This prevents errors on unrecognized types. We check this at the start of the doVideo command and exit if we are not active. Next we look at what command was sent. In this case we focus on the play command. The slowDirector call is an option I have written into my code to allow Director to slow its frame rate and allow the video to claim more of the CPU if necessary. Next we call the playVideo() call to the ancestor to let it start the video. Let’s look at the Flash and QuickTime versions of the

ancestor scripts to see how they vary for this command.

Flash:

```
on playVideo me
  sprite(me.spriteNum).fixedRate =
  me.pInfo[#playRate]
  sprite(me.spriteNum).play()
end
```

QuickTime:

```
on playVideo me
  sprite(me.spriteNum).movieRate = 1
end
```

Both trigger the video to start, but each has a different approach. We now have the ability to play back two media types from the same behavior, and can keep adding more. Let’s add another popular type – MPEG. MPEG Advance Xtra is a very popular solution for this, so here is its variant of this handler.

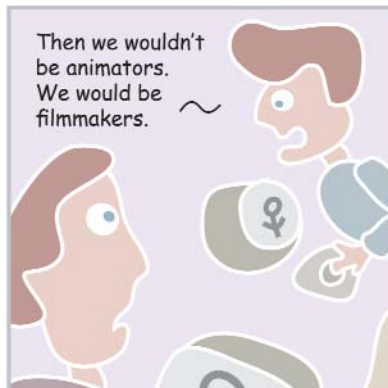
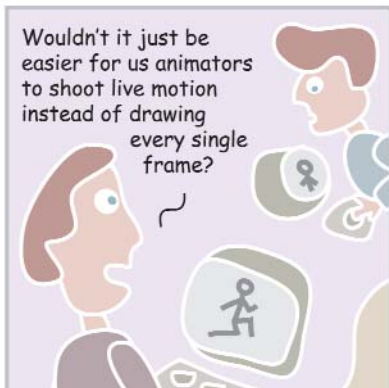
```
on playVideo me
  sprite(me.spriteNum).rate = 1
  sprite(me.spriteNum).play()
end
```

If we tried to create one large behavior for this we would have a mass of “if” and “case” statements for every command. This can make debugging difficult and gets more complex as we start finding larger and larger variations in how some media types handle different functions. Here is an example comparing QuickTime and Flash for rewinding.

QuickTime allows rewinding by simply setting the movieRate of the sprite to

## xile

written & illustrated by louis f. cuffari 



-2. Flash does not allow reverse playback so we instead have to fake it by calculating the time elapsed each frame and reversing the video by 2 times that amount. Here are the two scripts...

**QuickTime:**

```
on rewindVideo me
    sprite(me.spriteNum).movieRate = -2
end
```

**Flash:**

```
on rewindVideo me
    sprite(me.spriteNum).pause()
    --grab the time to fake the rewind
    me.pInfo[#lastTime] = the milliseconds
end
```

```
on rewindStepVideo me
    --Fake Rewind since MPEG Advance
    does not support it
    t = the milliseconds -
    me.pInfo[#lastTime]
    --backup 2X this amount
```

```
sprite(me.spriteNum).seek(sprite(me.spriteNum).currentTime - (2 * t))
    me.pInfo[#lastTime] = the milliseconds
end
```

Flash needs to check this every frame, while QT simply runs until we change the rate back. These differences would make a single script very unmanageable, while splitting the pieces into separate ancestors make this a very simple and elegant process.

After we code for play, pause, rewind, seek, etc., we need to make one more behavior for all the buttons. Since these buttons all address our core behavior they do not need any changes for each media type either. You can either communicate with scripts directly, use a sendSprite, or a sendAllSprites call. Here is a short overview of the differences.

- 1. object.function():** This approach calls a function on a specific object, or sprite. This can be handy as it runs fast, but if there is any problem it will throw an error and stop your program.
- 2. sendSprite:** This is a step up from calling the object directly. It sends a message to a specific sprite, but if the

Advertiser	URL	Phone	Page
ActivePDF	www.activePDF.com		21
CF_Underground VI	www.cfconf.org/cf_underground6/	301-424-3903	Cover 3
CFDynamics	www.cfdynamics.com	866-233-9626	6
HostMySite.com	www.hostmysite.com/mxdj	877-248-4678	27
InterAKT Online	www.interaktonline.com		9
Intermedia.net	www.intermedia.net	800-379-7729	31
ISSJ	www.ISSJournal.com	888-303-5282	41
IT Solutions Guide	www.sys-con.com	201-802-3021	45
Macromedia	www.macromedia.com/go/volvo		Cover 2
Macromedia Max	www.macromedia.com/go/max		Cover 4
MX Developer's Journal	www.sys-con.com/mx/subscription.cfm	888-303-5282	51
Nidus Corp.	www.brainstormer.org		19
Seapine Software	www.seapine.com/webdev	888-683-6456	5
ServerSide	www.serverside.net	888-682-2544	15
SunTech3	www.SunTech3.com	407-862-1144	16
Web Services Edge 2005	www.sys-con.com/edge	201-802-3066	37

handler is not on that sprite it's simply ignored. The down side is you have to know where everything is. There are many ways to handle this up front, but I find for systems where speed is not an issue, there is another approach...

- 3. sendAllSprites:** This command broadcasts a message out to all sprites. If the sprite has the handler it runs it, otherwise it's ignored. This can be a little slower (a very few milliseconds at most) but for a flexible system like this I find it to be very handy. You can establish buttons that span multiple frames and move media to different channels without the worry of any piece losing sight of the rest of the system.

Here is an example of a portion of the button handler that sends out the play command...

```
on mouseUp me
    case pWhatItem of
        #play, #pause, #stop :
            --play, pause or stop was
            clicked
```

```
sendAllSprites(#doVideo,
pWhatItem)
```

When we click the button it simply tells the sprite(s) with a doVideo command to run the specified command. This makes it very fast and easy to deploy solutions that will not break by moving frames or changing sprite orders.

You can download the example movie that demonstrates how to use this approach for QuickTime, Flash, and MPEG Advance. I have also utilized this in various portions for Real Media, Windows Media, LDMs (Linked Director Movies), Audio, or other video Xtras. Creating a new media type is as simple as copying another ancestor and adjusting the few lines of code that handle the video playback. The rest is already done for you and never has to be rewritten again.

**Summary**

Ancestors are a powerful way to reuse and simplify complex systems in lingo. Give them a try and you will see why it's great to be kind to your ancestors. ☺

*Chuck Neal is CEO of MediaMacros, Inc.. and owner and operator of www.mediamacros.com, a free Web resource for Macromedia Director developers. He has been involved in multimedia for over 8 years in both traditional multimedia as well as 3D animation and video. Over the years Chuck has worked on projects for a number of major clients, including Coca-Cola, United Parcel Service, and Ford Motor Company. Through MediaMacros, he strives to push the envelope of multimedia taking on full projects as well as offering advanced coding services to over 40 multimedia firms in over a dozen countries around the globe. chuck@mediamacros.com*



# Inner Meaning

**W**hen I started to make this video I was not sure if it would be understood by the world; it hides a deeper message, it speaks about smoke. Every day I receive mail from people writing to me for the designs, for the scenes, for lights. I like that, but just a few people got the inner message, I hope you feel it. [www.micheledauria.com](http://www.micheledauria.com) ☞



**\$99**  
per person

## CF\_Underground VI

October 31st 2004

10:00am - 5:00pm

New Orleans, LA

Check website for exact location and details



[www.cfconf.org/cf\\_underground6/](http://www.cfconf.org/cf_underground6/)

Fill your brain with lots of cool CF programming meta tips & tricks, then relax and have a drink on us!

CF\_Underground is the coolest pre-MAX event you'll attend during your visit.

Pick the brains of several experts like

Simon Horwith, Sandra Clark, Shlomy

Gantz, Hal Helms, Michael Smith,

and more!

**TeraTech Event**

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

301.424.3903







Experience

Connect

Learn

Develop

Exchange

## Macromedia MAX 2004

### The Power of Experience

Waves are the powerful result of critical mass, various elements coming together at a specific place and time.

The experience of people coming together to shape ideas and share information is equally powerful.

That's MAX: a powerful experience.

Learn the best techniques from industry experts; get current on new products; and share ideas in a forum dedicated to delivering the next wave of great websites and applications.

### Experience

Choose from over 80 different hands-on and workshop sessions, in seven tracks, to create a schedule to meet your specific needs.

### Connect

Share ideas with leading developers and designers at networking receptions, Birds-of-a-Feather sessions, and a special event. Retain your competitive edge at general sessions where you'll hear directly from Macromedia® and other industry leaders on what's next.

**It all happens November 1-4 in New Orleans. Be a part of it. Register today.**  
[www.macromedia.com/go/max](http://www.macromedia.com/go/max)

Silver Sponsors

**DoubleClick**

 **Host My Site.com**

  
macromedia  
**PRESS**

 **VitalStream**

# MAX

The 2004 Macromedia® Conference

© 2004 Macromedia, Inc. All rights reserved. Macromedia, 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.