

MX

developer's journal

volume 3 issue 1 2005 www.mxdj.com

THE LEADING MAGAZINE
FOR MACROMEDIA MX
DEVELOPERS & DESIGNERS

macromedia

captivate



getting
started

 **DREAMWEAVER**
Still Snowing

 **FLASH**
RIAs and Web Services

 **FREEHAND**
The Flow Between Fireworks & FreeHand

 **DIRECTOR**
Visual Debugging Tools for Shockwave

web services **EDGE**
conference expo

Web Services Edge
2005 East

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

P. 47



\$5.99US \$6.99CAN 02>



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





26



**The Flow Between
Fireworks & FreeHand**
*A little patience yields a
lot of rewards*
by ron rockwell



32



**Visual Debugging Tools
for Shockwave**
*Making it easier to "see"
what's happening*
by andy phelps



40



**Getting Started with
Macromedia Captivate**
Let us demonstrate for you!
by lisa heselton

7

**New Year,
New Look**
*What's coming from
MXDJ in 2005*
by charles e. brown

8

**Introducing the
Macromedia
Knowledge Base**
*An exciting new
development*
by deb dickerson

57

**Introducing
the "Making
Blackstone"
Article Series**
*Blackstone
heralds a new
dawn for
Macromedia
ColdFusion*
by damon cooper
& tim buntel

58

vanguard
*Commanders of
Ingenuity*
by mindflood



10

Still Snowing
Upgrading your snowman
by jim babbage



20

More OOP in Flash
Taking the photo album to Flash, Part 2
by john c. bland



14

RIAs and Web Services
An important role in Flash
by art phillips

CFDynamics

Dedicated Server Packages Starting at \$189/mo.

All dedicated servers include:

- ▶ FREE STATS SOFTWARE!
- ▶ No long term commitments!
- ▶ FREE SQL server access!
- ▶ FREE MAIL SOFTWARE!
- ▶ Fair and simple pricing!
- ▶ Optional server maintenance!

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 dedicated server packages they're not offering an oxymoron; "virtually private" or "virtually dedicated" is **NEITHER** private nor dedicated. CFDynamics offers a solution that is **100% completely dedicated**. They don't play games with the fake stuff; CFDynamics only offers the real deal. Real Service. Real Satisfaction. Real Value.

Real Freedom.



Visit us online or call to order!



Paper|Thin Partner



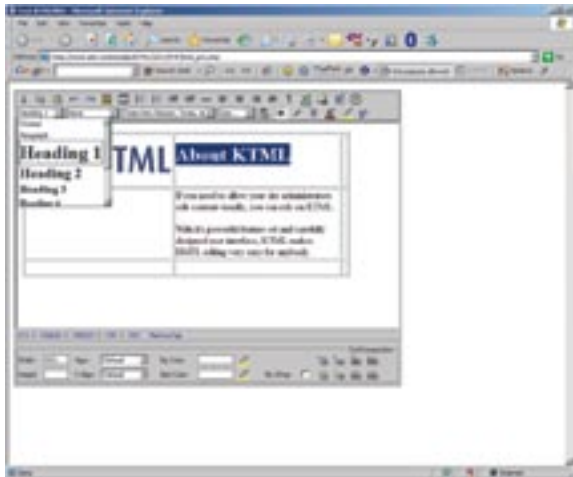
Web (r)evolution:



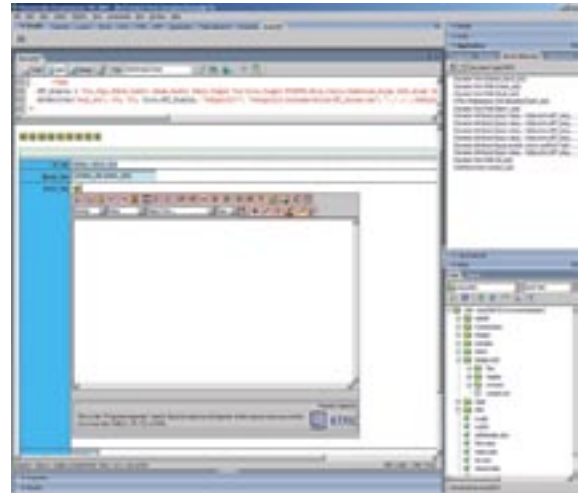
KTML

Let your clients edit their Web sites content through the browser

Visually use CSS styles (Word like)



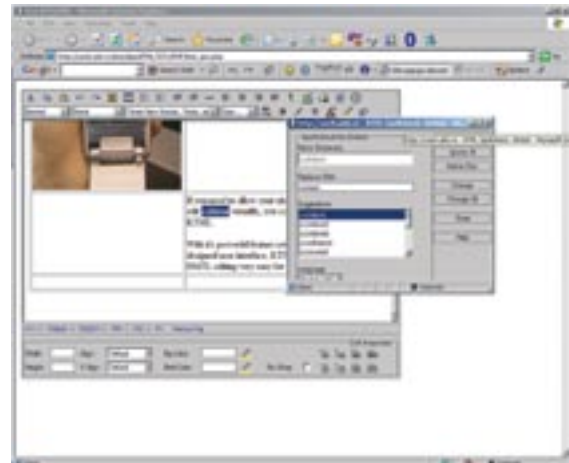
Native Dreamweaver MX integration



Easy-to-use image explorer



Multi-language spell-checker



KTML 3.5 - HTML online editor

See all the features, benefits and a live demo at:

<http://ktml.interaktonline.com/>

Interakt

macromedia
ALLIANCE PARTNER

Supported Technologies:



Group Publisher Jeremy Geelan
Art Director Louis F. Cuffari

EDITORIAL BOARD

Editor-in-Chief
Charles E. Brown charles@sys-con.com
Dreamweaver Editor
Dave McFarland
Flash Editor
Brian Eubanks
Fireworks Editor
Joyce J. Evans
FreeHand Editor
Ron Rockwell
Louis F. Cuffari
Director Editor
Andrew Phelps
Captivate Editor
Tom Green

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

EDITORIAL

Executive Editor
Gail Schultz, 201 802-3043
gail@sys-con.com

Editors

Jamie Matusow, 201 802-3042
jamie@sys-con.com
Nancy Valentine, 201 802-3044
nancy@sys-con.com

Assistant Editors

Natalie Charters, 201 802-3041
natalie@sys-con.com
Seta Papezian, 201 802-3052
seta@sys-con.com

Technical Editors

Jesse Warden • Sarge Sargent

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

Subscriptions

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

Editorial and Advertising Offices

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

Worldwide Newsstand Distribution

Curtis Circulation Company, New Milford, NJ

List Rental Information

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

Promotional Reprints

Kristin Kuhnle, 201 802-3026
kristin@sys-con.com

Copyright © 2005

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

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

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

MX
developer's journal

New Year, New Look

What's coming from MXDJ in 2005

by charles e. brown

It is the New Year, a time when we make resolutions and change (alright, I promise I won't talk about diets). It is also a time when we look back on the previous year. Publishing this journal is certainly no different.

It is hard to believe that my first quarter as editor-in-chief is now completed. I was overwhelmed by the number of very positive emails I received when the September issue came out. Most were in agreement about Flash being the technology of the future. However, with that agreement came many questions. We will be treating those many questions this year.

In the plans are four themed issues, one for each quarter, that will address many of the questions that were asked. Here is what I am projecting:

I received many letters asking how the pieces fit together. So the first theme issue, in March, will deal with coordinating the various Macromedia MX components. We will address issues such as how Fireworks works with Flash, using Dreamweaver with Contribute, Flash, and Captivate, etc.

The second theme, in June, will address an important new topic: the third-party market that has grown around MX products. We have started to devote more space to product reviews. This issue will examine how to incorporate these products into the workflow.

Next, we'll talk about training and education. We will examine the options for receiving and delivering the skills necessary in this profession. We will also examine the Macromedia certification process.

Finally, our year-end issue will offer greater coverage of the annual MAX conference. This symposium sets the theme and direction of this industry for the following year. The information coming from it is always exciting.

Another feature that will be appear-

ing in 2005 will be case studies. Currently, I am talking to the developers and designers of several major Web sites and they have agreed to write about their experiences in this journal. Unfortunately, I cannot get into specific details now, but trust me, they will be exciting. Expect a wealth of information to come from these articles.

Increasingly, as topics become more complex and feature rich, we are finding it difficult to contain all the necessary information in a single article. We will be seeing an increase in multipart articles. This trend began with Art Phillips and his series on RAD. I am continuing it with an ongoing, and open-ended, series on AS 2.0.

Since I took over in September, I have had a tough time fitting many articles into nice, neat little categories. For instance, should an article about using Dreamweaver with ColdFusion be classified as a Dreamweaver article or a ColdFusion article? What about Flash Video? Is that a Flash article or should it have its own classification? In light of that, we will discontinue the strict categories in this journal. I think this will give us a more fluid and flexible format to work with. Of course, I will be anxious to hear your feedback regarding this change.

I am also planning to promote greater interaction between the printed magazine and the Web site. It is tough to talk about multimedia and not be able to show it. It is my hope that we will have downloadable files to work through some of the tutorials and examples.

Like a Web site, this magazine is a developing work in progress. Next January it will be interesting to see how all of these plans went and where the roads led us.

In conclusion, I want to take this opportunity to thank everyone for your support and to wish you every success and happiness for the new year of 2005. ☁



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

Introducing the Macromedia Knowledge Base

An exciting new development
by deb dickerson

"I've been looking for that!"
"How did it know what I was looking for?"

If you've made any of these comments recently, then you've probably been using the Knowledge Base, a self-service guided search tool that allows you to access precise technical information on Macromedia products.

The Knowledge Base has an intuitive guided search tool that allows you to fine-tune your queries based on established categories and "best bet" choices. This tool is free for all macromedia.com visitors.

Guided Search

Let's face it – in most cases, you'll find one of two scenarios when searching for information: too many results, or not enough. The new guided search aims to eliminate that problem. Once you've entered your initial query, the search tool provides you with various areas to further focus or expand your results:

- **Relevant search topics:** When Macromedia publishes a new document, the Knowledge Base analyzes the content and maps it to appropriate categories: Activity, Symptom, Product Versions, or Environment. You can focus your search within any of these topics to filter your search results.
- **Search within a search:** If you feel the list of documents is still too large, you can further restrict your results by typing additional search terms in the "Search within your results" field. Rather than searching through the entire Knowledge Base again, this action searches within the documents already returned in the previous search.
- **Best bets:** In some cases, you will also notice a section labeled "Best Bets." You will see this option when we think a particular link may be useful for you, based on the search terms you entered.
- **Advanced options:** Using the Advanced

Search option, you can also filter your search based on criteria not previously available.

I Say ColdFusion, You Say CF

In an environment where acronyms rule, everyone has their own way of referring to something. In the past, this would have been a problem. Now, with an extensive dictionary of technical terms available, it's no longer an issue. The Knowledge Base is aware of the variations of those technical terms and will find the content you're searching for. Each search a customer performs automatically provides structured feedback on the technical quality of the Knowledge Base to Macromedia, which allows us to continually add new technical information to make the Knowledge Base more effective.

Yes, You Can Teach an Old Dog New Tricks

The Knowledge Base content currently includes technical information (TechNotes), Macromedia Developer Center articles, and Macromedia customer service notes on policies and procedures (service notes). Again, because each search of the Knowledge Base provides information regarding success or failure of the results, we can be smarter about providing the information you need, so you can succeed in using our products. The information gathered also provides us with valuable insight to fill content gaps, improve relevance, and increase the effectiveness of the system.

If you've used the new Knowledge Base, we'd love to hear your feedback (www.macromedia.com/bin/kb_feedback.cgi). If you haven't, we've provided a FAQ (www.macromedia.com/support/knowledgebase) with additional information and a short demonstration of how to use the guided search, as well as general search tips (www.macromedia.com/support/knowledgebase/searchtips). ☺

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

ADVERTISING
Senior Vice President, Sales & Marketing
Carmen Gonzalez, 201 802-3021
carmen@sys-con.com
Vice President, Sales & Marketing
Miles Silverman, 201 802-3029
miles@sys-con.com
Advertising Sales Director
Robyn Forma, 201 802-3022
robyn@sys-con.com
Advertising Sales & Marketing Manager
Dennis Leavey, 201 802-3023
dennis@sys-con.com
Advertising Sales Manager
Megan Mussa, 201 802-3023
megan@sys-con.com
Associate Sales Managers
Kristin Kuhnle, 201 802-3026
kristin@sys-con.com
Dorothy Gil, 201 802-3024
dorothy@sys-con.com
Kim Hughes, 201 802-3025
kim@sys-con.com

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

SYS-CON.COM
Consultant, Information Systems
Robert Diamond, 201 802-3051
robert@sys-con.com
Web Designers
Stephen Kilmurray, 201 802-3053
stephen@sys-con.com
Matthew Pollotta, 201 802-3054
matthew@sys-con.com
Online Editor
Martin Wezdecki, 201 802-3045
martin@sys-con.com

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

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

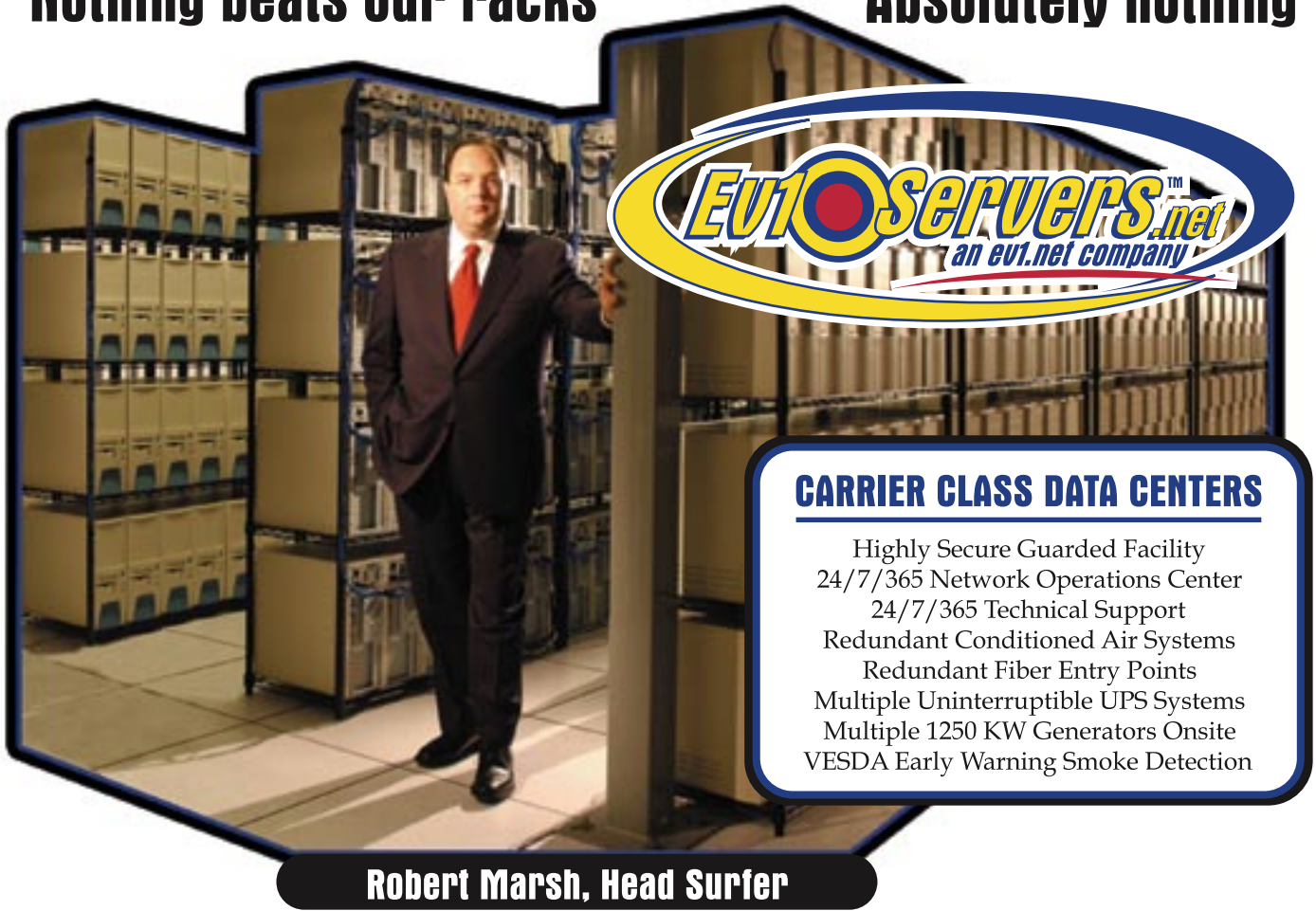
CUSTOMER RELATIONS
Circulation Service Coordinators
Edna Earle Russell, 201 802-3081
edna@sys-con.com
Linda Lipton, 201 802-3012
linda@sys-con.com
Monique Floyd, 201 802-3082
monique@sys-con.com
JDJ Store Manager
Brundila Staropoli, 201 802-3000
bruni@sys-con.com

SYS-CON
MEDIA

Debbie Dickerson is a Knowledge Base Program Manager and has been working at Macromedia since 1999.
ddickerson@macromedia.com

Nothing beats our racks

Absolutely nothing



CARRIER CLASS DATA CENTERS

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

Robert Marsh, Head Surfer

START YOUR OWN WEB HOSTING BUSINESS TODAY!

from **\$299*** **Dedicated Server**
Dual Xeon 2.4 GHz
2 GB RAM • 2 x 73 GB SCSI HD
Remote Console • Remote Reboot
2000 GB Monthly Transfer Included
Instant Activation!

Over 20,000 Servers!
1-800-504-SURF | ev1servers.net

PLESK7
RELOADED
Preferred Control Panel

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

Still Snowing

In “Let It Snow, Man” (*MXDJ* Vol. 2, issue 12), I used Fireworks to create all the essential ingredients for our frozen friend. We then exported the pieces using Export as CSS Layers. This gave us a starting HTML page (generated by Fireworks) and all of our graphics in absolutely positioned DW layers (DIVs).

In this article we will work in Dreamweaver and use the Drag Layer behavior to create a Web page where you can move all the snowman pieces together and build your own snowman online. The Drag Layer behavior uses Dynamic HTML (DHTML) to make your page interactive.

This concept could be applied to many different uses: visual quizzing, jigsaw puzzles, and other holidays, just to name a few.

Layers Can Be Such a Drag . . . or Not

We have a certain amount of prep work to do here before we create our draggable layers. To begin with, due to old coding there are some issues with the Drag Layer behavior in itself, as pointed out by CMX’s Danilo Celic in one of his CMXtraneous Blogs, Dreamweaver Drag Layer behavior broken in Mozilla with valid doctype. In a nutshell, the Drag Layer behavior doesn’t work in standards-compliant Mozilla browsers. With a big thank you to Danilo, here’s what you need to do in order to fix the problem.

You need to locate the Drag Layer JavaScript file (`{dw install folder}/Configuration/Behaviors/Actions/Drag Layer.js`). On my Windows XP Pro system, this is found in: `C:/Program Files/`

Upgrading your snowman
by jim babbage

Macromedia/Dreamweaver MX 2004/ Configuration/Behaviors/Actions/Drag Layer.js. If you’re not on Windows XP Pro, Danilo has also kindly blogged the Dreamweaver Configuration files locations.

1. In Drag layer.js go to line 161 and change it from:

```
else if (NS6){style.left = newLeft; style.top = newTop;}
```

to:

```
else if (NS6){style.left = newLeft+'px'; style.top = newTop+'px';}
```

2. Then go down to line 174 and change it from:

```
else if (NS6) {style.left = MM_targL; style.top = MM_targT;}
```

to:

```
else if (NS6) {style.left = MM_targL+'px'; style.top = MM_targT+'px';}
```

The first change allows the specified layer to be dragged, and the

figure 1

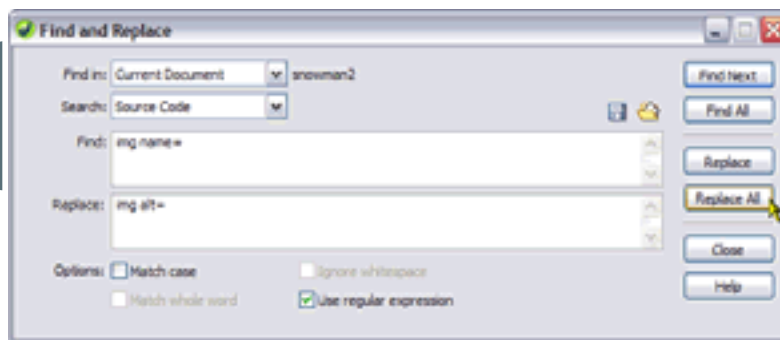


figure 2

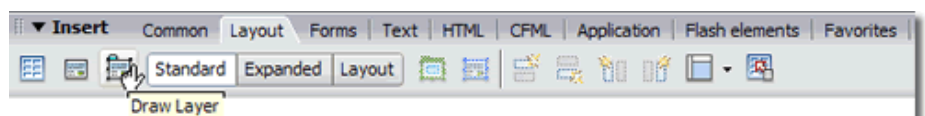
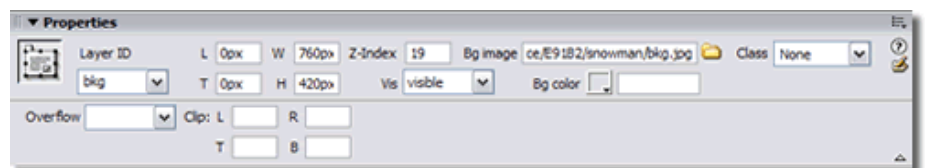


figure 3



second change allows the layer to snap to a location when dropped if that option is selected within the Drag Layer behavior dialog. Notice that in both cases all we added was the unit of measure (px) to the values being assigned.

3. Close and restart Dreamweaver and it should happily add in the correct code to run in valid pages for Mozilla.

With that little task out of the way, let's do a little under-the-hood work on the HTML page created by Fireworks.

Please note that even *with* these changes, the Drag Layer behavior does *not* function in the Safari browser.

Turning a Fireworks HTML Page into an XHTML-Compliant Page

While Fireworks can generate HTML, it doesn't do the best job in the world. First, the page doesn't have a doctype. Also, Fireworks will generate `img` name attributes for all our graphics based on the name of each layer. The `img name=""` attribute causes Internet Explorer a bit of grief, and as a result the page won't work as we want it to. Rather than start from scratch, we can touch up the Fireworks-created Web page and make it suitable for our needs.

Note: Due to a bug in Internet Explorer 6 the name attribute causes the layers not to be draggable, even though they are in some other browsers. By replacing the name attribute with the alt attribute, you eliminate this problem and make your page more accessible.

Before we go any further we will do a Find and Replace for this attribute, and change it from name to alt. XHTML compliance requires that all images have an alt attribute, even if it is left blank.

If you wish, you can edit the alt attributes later, giving each graphic a better description. However, for the purposes of this page – which is really a visual experience – I think the basic alt text will suffice.

1. In Dreamweaver, go to Edit>Find and Replace. Use these settings.
 - **FIND IN combo box:** Current document
 - **SEARCH combo box:** Source code
 - **FIND text box:** `img name=`
 - **REPLACE text box:** `img alt=`

2. Choose Replace All and DW will cycle through your HTML code, making the changes for you. This is a whole lot easier than searching on your own!
3. With basic alt tags set, let's now convert this page to an XHTML-compliant page. Dreamweaver makes this step quite easy too.
4. Switch to Code View for a moment to see what happens when you do the next couple of steps. Look at the beginning of the code. It should resemble what you see in Listing 1, line spaces and all.
5. Notice that there is no doctype. Go to File > Convert > XHTML. Look at the code again. Line 1 of the code now begins with a transitional doctype:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD
XHTML 1.0 Transitional//EN" "http://
```

```
www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<HTML xmlns="http://www.w3.org/1999/
xhtml">
```

6. Our last step is to remove the line spacing in the `<head>` area. Go to COMMANDS > APPLY SOURCE FORMATTING. The blank lines in your code disappear.
7. Save your page.

Tweaking the Page

You may recall that in my previous article I had you export a JPG slice for our gradient background. If you look at the current page in Dreamweaver, you will notice that the slice is not currently being used. Our snowman sits in a winter wonderland of whiteness. Let's use that background slice. We can do this in a couple of ways:



- We can create a new DW layer (AP div), set the Z-index so the new layer is at the bottom, and then apply the background slice as a repeating image.
- We can set the background image for the page to use this slice.

The main difference between these two choices is that the first option would contain the background within the div/layer. The second option would fill the entire page, and continue to repeat both vertically and horizontally unless we use CSS to limit the repetition.

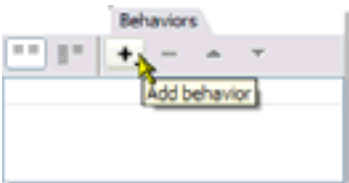
I also envision this background to be the frame for our winter picture, so in my case I will use the background slice within a newly created layer.

1. Select the Layout tab from the Insert Bar (see Figure 2).
2. Choose the Draw Layer icon.
3. Move your cursor onto the page, then

"The Drag Layer behavior uses Dynamic HTML (DHTML) to make your page interactive"

- click and drag to drag an empty layer. Don't worry about size or position.
4. When you have drawn the layer, click on its border to select it. The Properties Inspector will display attributes for the layer.
 5. Set the T and L values to 0.
 6. Set the W value to 760px.
 7. Set the H value to 420px.
 8. Set the Layer ID to bkg.
 9. Browse for the bkg.jpg image slice we created in Part 1. Figure 3 shows the setting I have in my Properties Inspector.
 10. Go to the Layers Panel in DW. If it is not visible, go to Window > Layers (F2). Note that our new layer is at the top of the Z-index, obscuring all of our other snowman pieces.
 11. Click on the bkg layer and drag it all the way down to the bottom of the Layer Panel. Now when the layer is

figure 5



deselected our snowman will appear on top of the background.

Snowman Goes to Pieces - Full Story at 11

With our background layer in place, it's time to take Frosty apart. One of the reasons I wanted to use the FW-generated HTML page is that it gives us the final position of each snowman piece and has all the layers in place with the right Z-index. We could have done all this from scratch – creating all the necessary layers (as we did with the background layer) – but why make all that extra work for ourselves when we don't have to?

What we will do next is set the Drag Layer behavior for each layer, and then move the pieces underneath or to the right of the background.

1. To start, select the body tag from the Tag Selector. You must apply this behavior to the body tag, rather than to the individual layers (see Figure 4).
2. Go to the Behaviors panel (see Figure 5) and click on the Add Behavior icon.

listing 1

```
<HTML>
<head>
<title>snowman2</title>
<meta http-equiv="Content-Type" content="text/HTML;iso-8859-1">
<meta name="description" content="FW MX CSS Layer">
</head>
```

3. Choose the (yes, you guessed it) Drag Layer behavior.
 - If the Drag Layer behavior is greyed out, you will need to choose the Show Events for menu item, and choose IE 6.0. This will make the drag behavior accessible.
4. When the dialog box opens, select the hat layer from the Layer dropdown menu.
5. Click the Get Current Position button to have DW apply the hat's x and y coordinates.
6. Optional: Set a value of 20 pixels for the Snap within the option if you want the snowman pieces to snap into place when they get near your original location (see Figure 6).
7. Click on the Advanced tab.
8. For the While Dragging option, make sure the Bring Layer to Front box is

checked and then choose Restore Z-index from the combo box.

9. Click OK (see Figure 7).

I recommend that you not set the Restore Z-index option for the eyes. They are rather hard to reselect once they've been placed because the hat and nose layers overlap them somewhat.

You can add your own custom JavaScript message boxes to encourage your users as they move each piece into place.

10. Select the hat layer and drag it off to the right of or underneath your background layer.
11. Select the body tag again and repeat the above procedure. You will need to do this for each layer that you want to make draggable.

You don't have to reposition the layer right after adding each iteration of the behavior. Go ahead and add all the Drag Layer behaviors, then jumble up the pieces of your snowman afterwards. ☹

figure 6

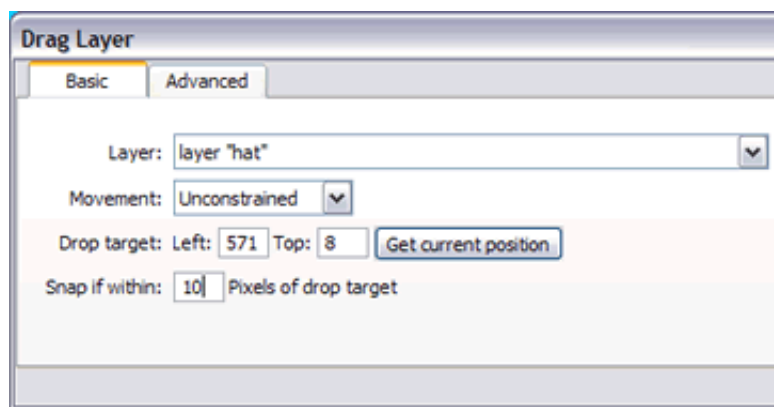
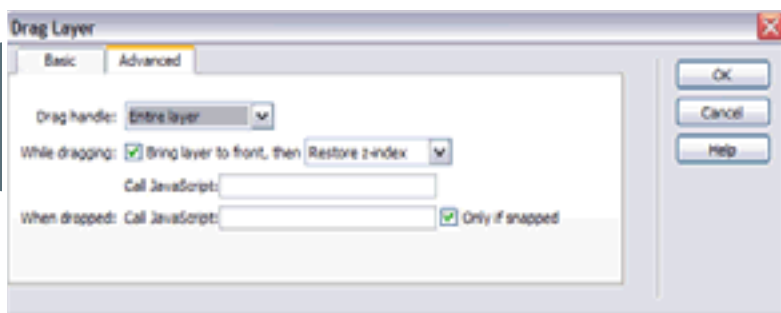


figure 7



Jim Babbage (contributing from CommunityMX) comes from the photographic world and has spent many years as a professional studio photographer. His involvement with the Web began in the mid '90s when the company he worked for had just gotten online. Born and raised in Toronto, Jim teaches imaging, Web design, and photography at Centennial College's Centre for Creative Communications (www.thecentre.centennialcollege.ca). He is a partner in Newmedia Services (www.nms123.ca), a small communications company specializing in the things he teaches. He is a regular contributing partner to Community MX (www.communitymx.com) where he's written many articles and tutorials for Fireworks, Dreamweaver, and other

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

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

Surround SCM lets you...

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

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

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

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

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

Remotely access your source code repository from Dreamweaver MX.

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

Features:

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

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

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

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

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

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

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

Want to learn more?

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



www.seapine.com/webdev

1-888-683-6456



RIAs and Web Services

An important role in Flash
by art phillips

This month the topic is Web services. Previous articles looked at the RIA, component-based development, XML, and the role of ColdFusion as a data source. In this issue we examine the role of Web services in RIA development.

What is a Web service? Two years ago I really didn't have a clue. Sure, I had heard some of the buzzwords such as SOAP and WSDL, but back then I couldn't have told you what they meant. Then Macromedia asked me to speak at their developer's conference: MAX. Knowing of my passion for Flash and data integration they offered me a great topic: Flash and Web services. It was a wonderful opportunity and a great eye-opener for me. Just what is a Web service? Let's break it down. Web. Service.

The Web refers to an immense information space and network that enables unprecedented access to resources. A Web resource is some type of electronic construct, such as a file, network, proces-

sor, application, or service. Every Web resource is identified by its URI and accessed via Web protocols.

A service is a resource that exposes its functionality through a programmatic (rather than a graphical) interface. The method of calling the service, and the possible results of that call, are exposed via documentation (the schema). Developers only need to concern themselves with this programmatic interface.

A Web service, therefore, is a resource that is identified by a URI and accessed by applications using standard Web protocols in accordance with the schema that describes its programmatic interface.

Applications can be assembled from these services.

The service provides an interface that can be called by another program. This interface acts as an adaptor between the Web and the actual application logic that implements the service. The programmatic interface provides a layer of abstraction that makes the connections

flexible and adaptable. This application-to-application programming interface can be invoked from any type of application client or service

What Role Do These Web Services Play in RIA Development?

Web services provide a standardized method of passing commands and complex data between the client and the server.

Macromedia recognized the power of this technology and expressed a serious commitment to it with the MX product line.

Jeremy Allaire succinctly stated Macromedia's position: "The convergence of rich clients, Web services and the need for a more scalable Web application development and deployment model have driven Macromedia to deliver components and Web services as part of Macromedia MX. ...empowering 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. The Macromedia MX product family combines ColdFusion Components with the rich client capabilities of the Macromedia Flash Player, the development capabilities of Macromedia Dreamweaver MX, and the openness and interoperability of SOAP and XML Web services."

This series of articles has explored that vision. We have seen how Flash's data and UI components provide for rapid visual development. In the last issue we saw how ColdFusion provides an extremely simple method of creating a rich data source. User-Defined Functions (UDFs) in a ColdFusion Component (CFC) can make data sources for both Web services and Flash Remoting technologies.

View the full list.

XMethods Demo Services

Style	Service Name	Description	Implementation
RFC	Top: AllData	Show SOAP Service's data source	Class
RFC	Top: XMethodsComponent	Provides a SOAP RFC interface to XMethods for query operations	Class
RFC	Top: Weather - Temperature	Current temperature in a given U.S. zip code region	As of the SOAP
RFC	Top: Customer Feedback Form	Interchange rate between customer information	Class
RFC	Top: Customer Traffic Counter	Calculates highway roadblock	As of the SOAP
RFC	Top: Customer Stock Quote	20 minute delayed stock quote	Class
RFC	Top: Web Page Monitor	Checks current hitpage of web page service	As of the SOAP

XMETHODS Home Interface Tools Implementation Manager Engine Tables About

Weather - Temperature

[Top: All](#)

WSDL	http://www.methods.com/soap/WeatherTemperature.wsdl	XML Schema	See RFC: http://www.methods.com/soap/WeatherTemperature.wsdl
SOAP Binding	TemperatureInfo		
Key	url=http://www.methods.com/soap/WeatherTemperature		
Owner	methods.com		
For more info			
Description	Current temperature in a given U.S. zip code region.		



figure 1

figure 2

“The convergence of rich clients, Web services, and the need for a more scalable Web application development and deployment model have driven Macromedia to deliver components and Web services as part of Macromedia MX.”

-jeremy allaire, april 2002

The standardization that a Web service brings to distributed computing is based on its use of standard Web protocols. The most notable are HTTP and XML. Domain Name Service (DNS) technologies also have a role.

So, how do we use all this in Flash?

The first thing we need is the Web service's WSDL file. WSDL stands for Web Service Description Language. It is one of the flavors of XML that Web services use.

Please be very clear on one thing; the WSDL file is not the Web service. The WSDL file is a description of the Web service. Think of the WSDL file as the metadata for the service. It describes what the service does, how it communicates, and the actual Web address of the service.

The WSDL file describes the service's programmatic interface. It is a contract. It promises us that if we do this, we will get back that.

The good news is that we should never have to either read or write a WSDL file ourselves. The WSDL file should be automatically generated by the technology in which we created the service. When we need to use a Web service, the technology we are using should be able to read the WSDL and help us consume the service.

Flash, Dreamweaver, and ColdFusion all provide that level of support.

In ColdFusion the WSDL is generated when we append ?wsdl to the end of our ColdFusion Component's address. If

our component's address is www.mydomain.com/mycomponent, the address for the WSDL is www.mydomain.com/mycomponent?wsdl.

From this WSDL a proxy object can be created. The proxy object is an object we instantiate on our server, or for our purposes, in the Flash client, that has all of the methods and properties of the service described by the WSDL. We can then use all of the functionality of the remote service right on our local proxy. Commands and data are passed between our proxy and the actual service using another flavor of XML; SOAP.

SOAP (the Simple Object Access

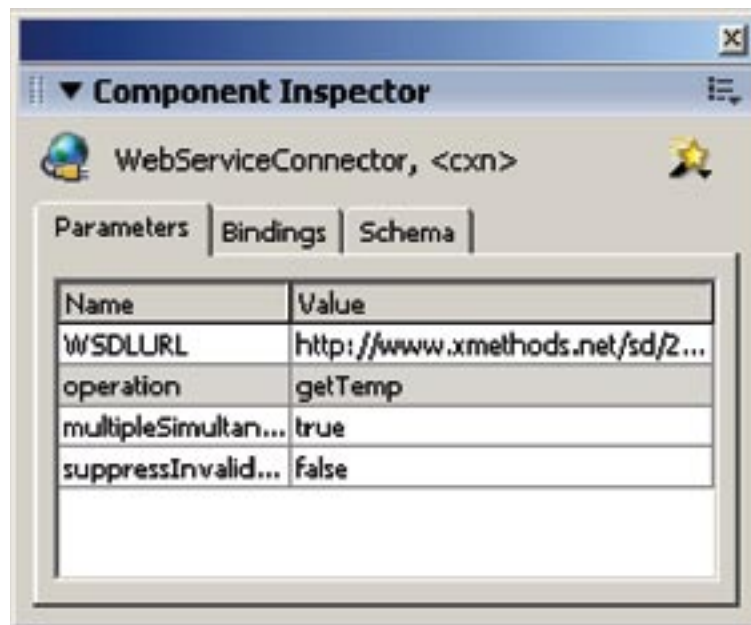
figure 3



figure 4



figure 5





Protocol) is the XML schema that provides a standardized way of exchanging data and commands between a Web service and a Web service consumer. SOAP is independent of the platform, object model, and programming language used.

This standardization provides for great flexibility. Any client on any platform can communicate with any service

on any platform. Such exchanges are said to be "loosely coupled."

This broad flexibility means that the exchange may be slightly lacking in efficiency. If we were dealing with a specific service and a specific client we could format the exchange in the most efficient manner without regard for other clients and services. This is what Flash Remoting does.

Flash Remoting requires a Flash Remoting-enabled server and the Flash client. The exchange between the service and the client is accomplished by exchanging binary data in the Action Message Format (AMF). This binary is leaner and more efficient than the text-based markup of the SOAP XML. Flash Remoting represents the most efficient way of exchanging commands and data with a service. Flash Remoting will be the subject of the next article in this series.

Let's look back briefly at last month's article on ColdFusion as a data source. We saw that by simply setting the access attribute of a user-defined function to remote we could create a ColdFusion data source that functioned both as a Web service and a Flash Remoting data source.

This flexibility allows users in other frameworks (ASP.net, Java, PHP) to consume our ColdFusion data source as a Web service using WSDL and SOAP XML. The same data source can then be used by Flash clients as a Flash Remoting data source using the more efficient AMF binary. Macromedia has provided for efficiency within their products without excluding other technologies.

We have discussed the role of XML in Web services, but earlier I mentioned that Web services also use domain name service (DNS) technology. Part of the bigger picture of Web services is the idea of directories where we can browse for services to assemble applications from. These directories are called UDDIs (Universal Description, Discovery and Integration) and they are where Web services use DNS technologies.

While an optional part of the Web services framework, UDDIs present a very interesting development model. Developers will search these directories by functionality, or by industry, to find services to build applications from. Inventories, pricing, shipping information, and other business information will be able to be found in UDDIs.

UDDIs can be public or private. There are several public UDDIs, including those at IBM and Microsoft. One that has been very popular with the Macromedia community is XMethods (www.XMethods.com). The famous and popular Babel Fish service was once available there.

figure 6

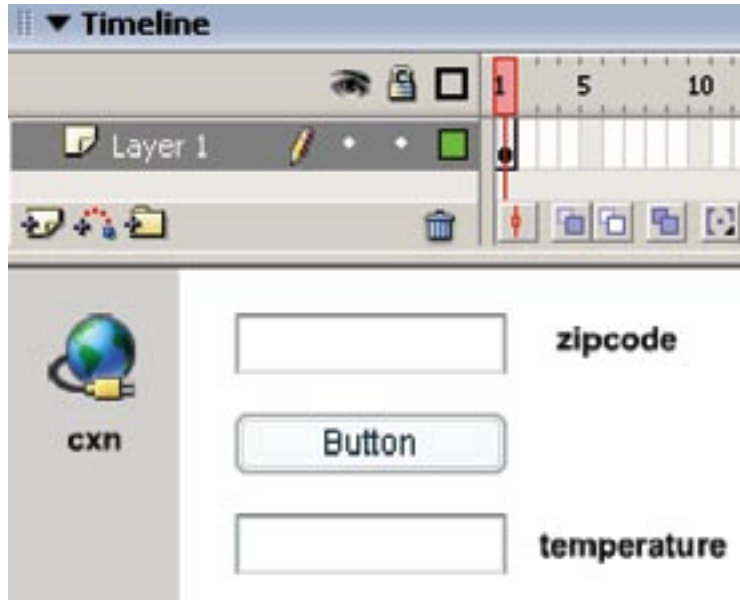
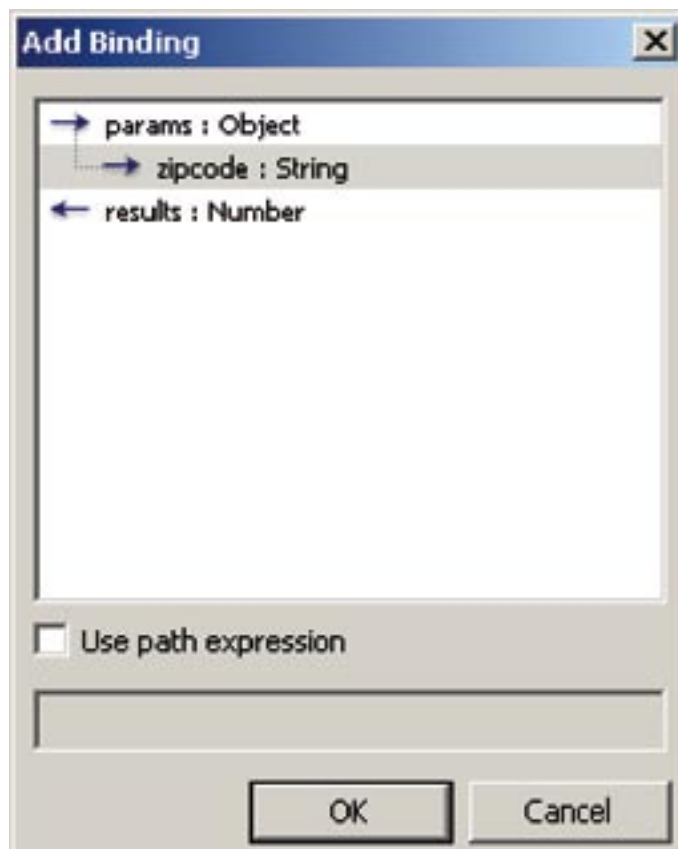


figure 7



CommonSpot™

Efficient Content Management

fast. easy. affordable.



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

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

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

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

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

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

Paper | Thin



XMethods provides a listing of Web services as well as a list of clients using the services. In addition, through a partnership with Mindreef, Inc., XMethods provides a means of testing the service and seeing the underlying exchange of SOAP.

XMethods' home page lists the most

recent additions to the directory. At the bottom of the home page there is a list of XMethods demo services. While you are on XMethods' site don't overlook the links for the full list of all the Web services listed at XMethods.

Our application will use a simple service from the XMethods demo services

at the bottom of the home page. The Weather – Temperature service will give users the temperature when a ZIP code is submitted.

At the bottom of XMethods' home page, in the Demo Services section, find and follow the link for Weather – Temperature (see Figure 1). This will lead us to a page with information and links related to the service. Take a minute and look at the information on the page; the only thing we will need to use this simple service for is the address for the WSDL (see Figure 2). If we follow the link near the top of the page we can see what the WSDL for this service looks like. *Note:* This illo is *very* optional (see Figure 3). If we copy the address from the address bar we are ready to open Flash and build our Web service consumer.

Create a new Flash document. Add a new WebServiceConnector from the Components panel (see Figure 4). Like the other connectors, this component does not have a visual presence in the final SWF. It is often placed off stage, out of the way of visual elements. Give it an instance name of cxn.

Open the Component Inspector and go to the Parameters tab. Paste the WSDL into the WSDLURL field. Flash will go out, get the WSDL, analyze it, and use the WSDL information to assist us in creating our application. After a few seconds we can click the operation field underneath the WSDLURL field. This will show us all of the functions that we can run from the service. The Weather – Temperature service has one operation, getTemp. We'll leave the other settings at their defaults. The finished Parameters settings will look like Figure 5.

With the connection configured, we can build the user interface for the application. We'll add two text inputs from the UI Components to the stage, one above the other. We'll give the top one an instance name of zipcode and the bottom one temperature. In between them, we'll place a Button component. The Button does not need an instance name, as we will apply our actions directly to it. The finished set up will look like Figure 6.

With our components in place, we are ready to bind them together. We'll start by binding the user input (zipcode) to the service.

Select the WebServiceConnector on

figure 8

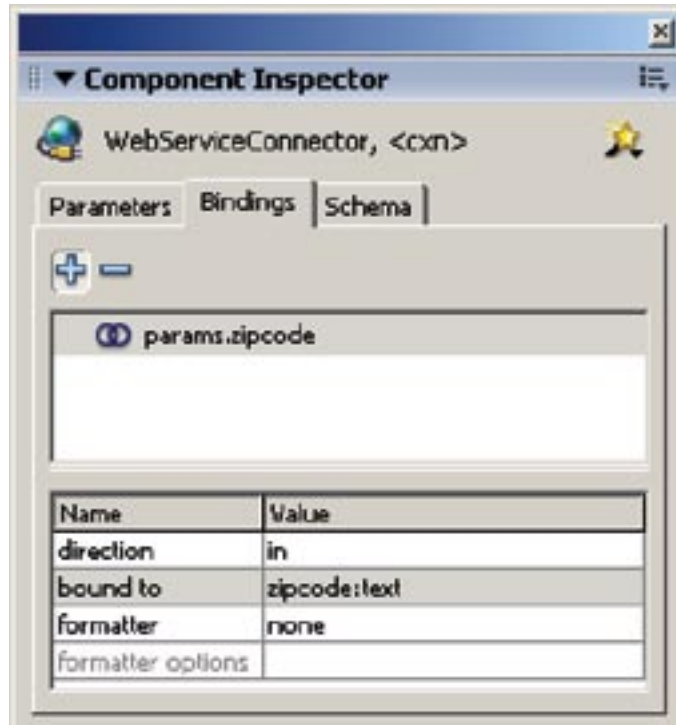
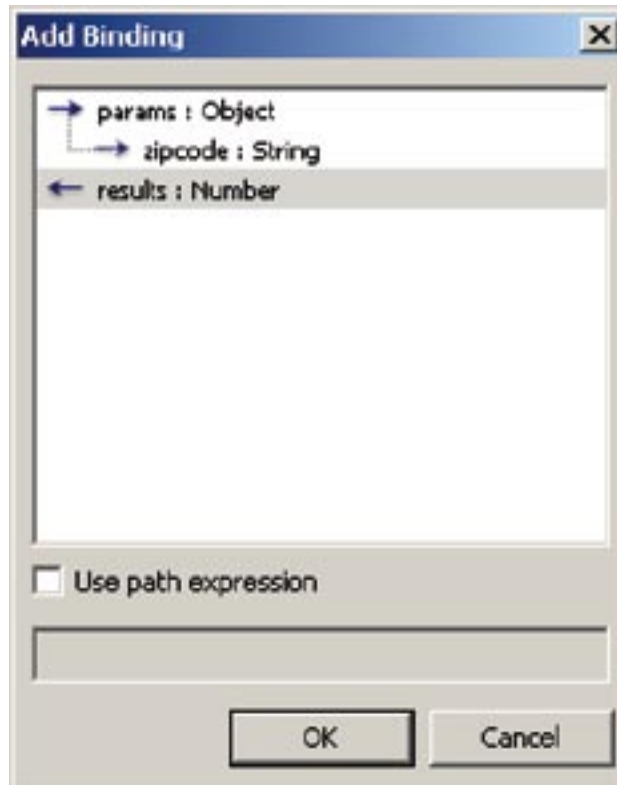


figure 9



stage. In the Component Inspector panel move to the Bindings tab. Click the plus sign to add a new binding and select zipcode from the params (see Figure 7). Back in the Component Inspector, move down to the bound to field and choose TextInput, <zipcode> from the Bound To window. The finished binding will look like Figure 8.

Now, we'll bind the results to the interface. Keeping the WebServiceConnector selected and staying in the Bindings tab, click the plus sign to add the final binding. In the Add Binding window, select results (see Figure 9). Again, move down to the Bound To field. Open the Bound To window and select TextInput, <temperature>. The final binding will look like Figure 10.

All that remains is to program our button to trigger the call to the service. We will do this by adding a behavior directly to the button. Start by selecting the button on the stage. Open the Behavior panel and click the plus sign to start adding the behavior to our button. Choose Data > Trigger Data Source (see Figure 11).

From the Trigger Data Source window, be sure to select the WebServiceConnector; cxn. The applied behavior will look like Figure 12.

Test your movie. When the SWF opens you should be able to enter a valid ZIP code, click the button, and receive the temperature for that area.

One important thing to note: this application will work fine from the SWF player. It will not work if we put it on our server. Web services use XML in the exchanges of commands and data. When running from a Web server, Flash's XML class does not allow data to be loaded from other domains without a permissions file on that server. This will be examined with other security issues in a future article.

Summary

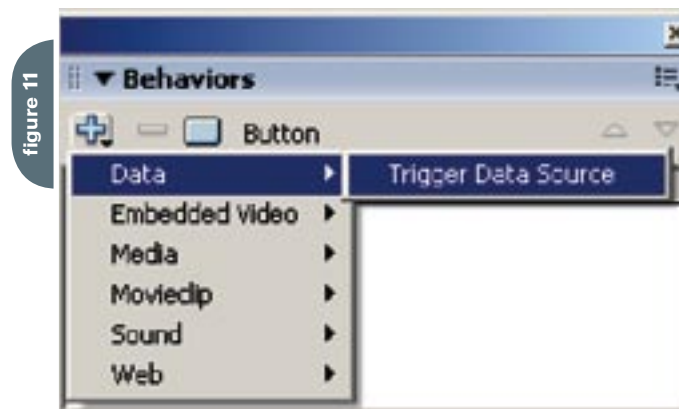
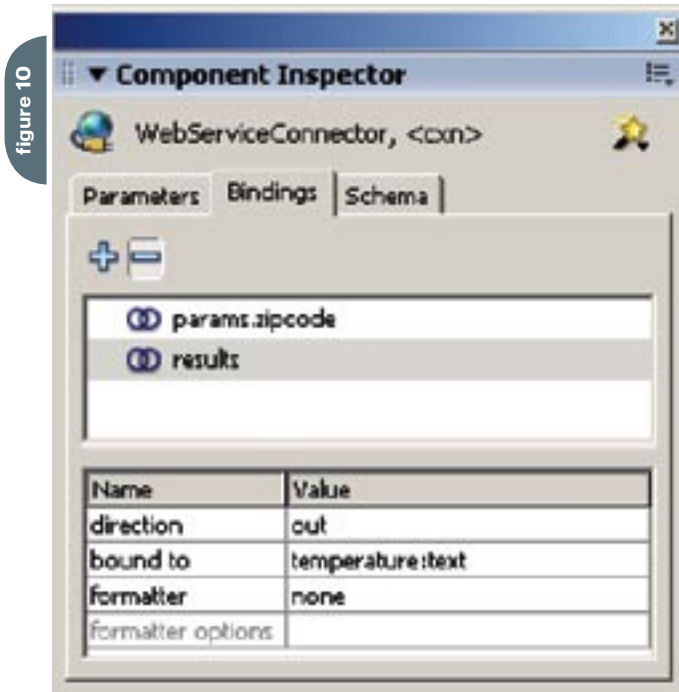
Remember that you can use last month's article to create your own Web service data source. Then your data source and Flash application can come from the same domain and there will be no security issues to prevent the loading of data.

Have some fun and play around with some of the other Web services at

XMethods. Remember that some services may return objects to us that we cannot use. Any service that returns a string or number is a safe bet.

That's it for this month. Drop back next month as we examine and build an application based on Flash Remoting. ☺

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.artwebsite.com) is a well-known resource in the Macromedia community. art@artwebsite.com





More OOP in Flash

Taking the photo album to Flash, Part 2

by john c. bland

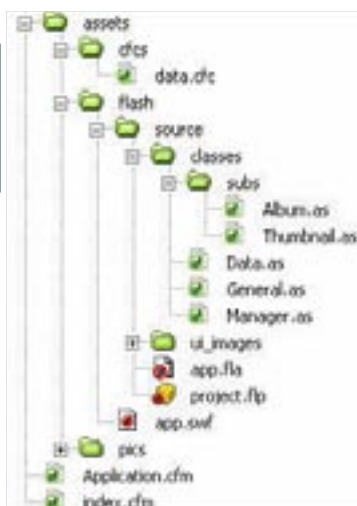
Hopefully you read last month's issue (*MXDJ*, Vol. 2, issue 12). This is a continuation of that article, so please have it handy for reference purposes. As a small review, currently we have our backend coded and the design implemented. It is time for some real work now.

Real Work

What is real work? For the purpose of this article it is Actionscript class development, not to knock designers or design work. Before I start a class I always create my directory structure. Figure 1 shows my common directory structure. Everything is the same as in the October issue except we have a Flash directory and we are missing two directories. Let's explore the flash/source directory.

The flash/source folder is strictly for my Flash source files. I will typically cloak them in Dreamweaver so they don't upload every time I decide to publish the entire site. The main thing to notice is where the app.flc file resides in reference to the classes folder. This is key for class development as you must specify the exact location in dot syntax. The classes folder also has a subfolder called subs, where I store all subclasses. Sometimes I'll even have subfolders here depending on the application/class needs. Okay.

figure 1



Now we can start developing. I'm going to walk you through the exact steps I took to develop this application.

Data.as Class

This class handles the data retrieval. I started here because without data nothing will work. The first step was to import the necessary classes.

```
import mx.remoting.NetServices;
import mx.remoting.debug.NetDebug;

import mx.events.EventDispatcher;
```

These are the basic imports for Remoting (NetServices and NetDebug) as well as the EventDispatcher class. While developing the app I thought about a recent project I did for a company and how I could have improved it. Well, using the EventDispatcher would have been one way so I figured I'd use it here (see Listing 1). I'll go into a little more detail later.

The next task is to create the class variables and code the constructor function. I always put my variable declarations at the top of my class block with my class constructor, if I use one, right after my variable declarations; structurally it is better for me. Here the constructor, Data(), merely calls the init() function, my typical constructor setup.

Our init() function, by line, initializes Remoting debugging (you can now use the NetConnection Debugger to monitor your Remoting calls), the EventDispatcher, and stores a reference to the data.cfc component in the data_cfc variable created earlier. I don't want to spend a lot of time on Flash Remoting since that is not the core topic here, but I will touch on a few things.

What we are doing is creating a gateway connection to the string stored in the remotingURL variable by using NetServices.createGatewayConnection(remotingURL). On top of that we use the getService(serviceAddress,

[default responder]) function from the NetServices class, which "Creates a Flash Remoting NetServiceProxy object that allows access to application server functions" (Flash Help Docs). The serviceAddress parameter is the location of the CFC, after the remotingURL, using dot syntax. So, the remotingURL points to the server www.jdevinc.com. The dot syntax to the component is projects.flashgallery.assets.cfcs.data. The full URL to the component is www.jdevinc.com/projects/flashgallery/assets/cfcs/data.cfc. Notice the .cfc is left off of the dot syntax for the component. For you CFers out there, think of this just like you would CreateObject("component", "projects.flashgallery.assets.cfcs.data") if you called it from the root directory.

The final parameter is the default responder. You can use an object as the default responder, and in our case we used *this* to reference "this" class as the default responder. I prefer doing it this way but it is not, by any means, the only way to do it. Remoting offers multiple avenues so I strongly suggest you explore all of them and then come up with your own comfortable way of setting it up. We now have a reference to our data.cfc that we will call later to make our Remoting call (see Listing 2).

The first three functions are blank on purpose. I read Kenneth J. Toley, III's article before stepping into using the EventDispatcher and it raised an eyebrow as to why three empty functions would sit there. Well, the answer was obvious enough when I opened the EventDispatcher class. To keep it brief, when we initialized the EventDispatcher class we passed it the parameter *this*. The initialize(object:Object) function sets the object's addEventListener, removeEventListener, dispatchEvent, and dispatchQueue functions to a reference of the EventDispatcher class functions. So, when you call the addEventListener() function in the Data.as class you are, in essence, calling the function in the

EventDispatcher class.

Moving on, we have the trigger function. Here we simply make the Remoting call `data_cfc.getAlbums()`. Now, why do we call `getAlbums()`? Simple. That is the name of the function in the `data.cfc` component. Remember, CFers, it is just like using `CreateObject()` in ColdFusion. After making our Remoting call we need a responder function setup for the `_Result` and `_Status` returns.

`getAlbums_Result(result)` is the result function. All we are doing here is dispatching an event with the type of `datalsHere` and the data equal to the result parameter. The same goes for the `getAlbums_Status` function with a different type and parameter name. The `_Result` function receives successful returns while the `_Status` receives any status errors (search for About handling errors for a Connection object in the Flash Help docs). The idea for this class is merely "Get the data!" We're not storing any information in the class nor are we actively performing any tasks. The purpose here is to do something only when told to just as parents used to say "speak when spoken to!" LOL.

That's it for the `Data.as` class. In a nutshell, we set up Remoting, the EventDispatcher, and a way to trigger the Remoting call. Now we need something to trigger the Remoting call.

Manager.as Class

What is the Manager class? Well, truth be told, it's the manager of the application, hence the name. You could also think of this as the controller, in a sense, for the Model-View-Controller (MVC) architecture. This app is not set up strictly to the MVC architecture as this class does some of what the view would do.

```
import classes.Data;
import mx.remoting.RecordSet;
```

These are our import statements for the class. We will use the `Data.as` class we just created as well as `RecordSet` functionality, although not much (see Listing 3).

As you can see, the setup here is the same as the setup in `Data.as`. The variables will be covered as we access or set them up, whichever comes first. Everything starts with the Constructor

function. `Manager()` receives one parameter, which is stored in the root variable. root is easy to confuse with `_root` but don't. You can rename the var `timeline` if that is more comfortable for you. I like root and I am comfortable using it so it is my preference.

Now we run the `init()` function. In `init()` we are creating an empty `MovieClip` on the stage at coordinates (35, 120). On the stage we have two `ScrollPanels` (`thumbs_sp` and `large_sp`). The two variables `thumbsSP` and `largeSP` hold a reference to the content of the `ScrollPane`, not the `ScrollPane` itself. At first I had a reference to the `ScrollPane` but realized I don't specifically reference the `ScrollPane` as much as I would the content so there is no need to use two sets of variables.

Speaking of content, we need some to work with. After setting up the references to the `ScrollPane` content we instantiate the `Data.as` class and store it in the variable `dataClass`. At this point `Data.as` has called its Constructor function, which ran the `init()` function. Since that is done we can now subscribe an event listener to the class to let us know when the data is here and when the data fails, hence the name I chose for the type. The listener object is this class object so we can set up a normal class function instead of creating a new `Object()` to listen for the event. All we need now is to get the data by calling the `Trigger()` function.

At this point the Remoting call will take place and a result or status error will be returned. Once the Remoting call is back one of two events will happen: `datalsHere` or `dataFailed` (see Listing 4).

These are the Event Listener functions. As you can see, the names are exactly the same as the event types seen in the `_Result` and `_Status` callbacks (see the `Data.as` class). If you are familiar with

event listeners already you understand the parameters received by both of these functions. I like to call mine `ev`, but the help docs and other places I've seen name it `evt`. It doesn't matter though. Name it whatever you want. When I first started with event listeners I was John "Stick Strictly to the Help Docs" Bland, but soon realized it is only a parameter so I have control! The parameter is typed as an object. In the object, with the current setup, it contains type and data (see `Data.as` class). The data will actually have the result of the Remoting callback.

The idea of the `datalsHere` function is to set up our album names in the left column. So, what we initially do is set a few variables we need. Since we will duplicate `MovieClips` vertically we need to manage our `_y` value. This is stored in the local variable `y` and the `RecordSet` length is stored in the local variable `len`. For those of you running a for loop like this: `for(var i:Number = 0; i < ev.data.length; i++){}`, start setting `ev.data.length` to a variable, preferably local if you are in a class and you aren't going to need it outside of that particular function. It may seem insignificant but each time the loop reiterates the length of the `ev.data` `RecordSet` has to be retrieved. This may not destroy your application or Web site but it could slow it down just slightly. Our variables are set up and we are ready to add our album `MovieClips` to the stage.

The for loop is pretty simple in that it doesn't do anything ultra special. The first thing we do is attach the album `MovieClip` from the library to the root. albums `MovieClip` we created in the `init()` function. Everything here is normal `attachMovie()` stuff except for the `initObject`. This is the parameter after the depth and is an object. The first two parameters are self-explanatory: `_x` and `_y`. The next two are class variables used in our

"What is real work? For the purpose of this article it is Actionscript class development"



Album.as class, coming to a paragraph near you. myData is the album data. If you remember how the query looked in a cfdump, from October's article (MXDJ, Vol. 2, issue 10), we have a row in the query with a custom column for images which stores another query. Well, cfdump isn't available in Flash but we do have the NetConnection Debugger. Figure 2 shows the Debugger with the result of our Remoting call. Notice the _items object has the possibility for multiple records although only the 0 record is shown here. You see all of the information for the directory Christmas 2003 as well as another object named images which is actually another RecordSet (or query for you CF'ers) with its own _items object. This 0 record, and all subsequent records, is the data being passed in the myData parameter of the attachMovie(). So as the loop iterates through each of the _items (0, 1, 2, etc.), that block of data is sent to the new album being attached to the stage. The last parameter is just a reference to this instance of the Manager.as class.

One thing I had to resort to was making the album name textfield multiline. This was critical in my eyes. Without a multiline textfield the album name could be cut off at the end. That's not good enough for me. In light of this we have to compensate for the fact that each album could be a different height. That is what the next line of code handles. It simply takes the _height of the newly attached MovieClip and adds it to the current running _y total stored in the y variable. To give you an idea of what it does, trace the y variable right after this line. You will see it increment by the _height of the newly attached MovieClip plus two, for spacing. We're all done here. Our albums are now on stage and displayed properly (see Figure 3).

Just to show how to handle the data-Failed event I added the function to the class. Currently it doesn't do anything because I'm not building this application for production use, but you could make a modal popup display to let the user know the data wasn't retrieved properly, or do another Remoting call in case it was a fluke (be careful though as you could ultimately have an endless loop if _Status callbacks keep occurring).

I think it is fitting we continue here

and cover the rest of this class, instead of jumping into our Album.as class. There isn't much left so, let's keep rolling (see Listing 5).

This is pretty much what the user is waiting for...some images. Our createThumbnails function is called from the Album.as class. Once called, the main objective is to create the thumbnails by looping over our RecordSet. First thing we do is refresh our ScrollPane so every time this function is called a fresh MovieClip the content of the ScrollPane is reloaded (which is an empty MovieClip from the library). ScrollPane.refreshPane(); "refreshes the scroll pane after content is loaded" (Flash Help Docs). Notice we used _parent instead of just thumbsSP.refreshPane(); If you trace(thumbsSP); you will see it is a reference to a MovieClip one (child) level into the ScrollPane. Just like referencing any other parent MovieClip, we use _parent to access the ScrollPane itself. This is the only time we need to access any of the ScrollPanes directly. Now I'm sure you can see why I didn't care to store a reference to the actual ScrollPane. Next we set up variables.

We are doing the same exact thing here that we did in the dataIsHere function with a few subtle differences. The first difference is that this is a horizontal list so we are tracking a variable called x instead of y, with obvious technical differences. The first parameter we receive in this function is a RecordSet

named rs. This variable holds the images RecordSet from the parent album and is what we will loop through to create our thumbnails. To loop we, of course, need to create our len variable, which grabs the length of the rs variable. Now we can loop.

As previously noted, we aren't doing anything new here other than referencing different destination MovieClips (thumbsSP), attaching a different MovieClip (Thumbnail), and changing the _x value instead of the _y. There is one extra parameter in the initObject. This is merely the name of the parent album, which is the second parameter of this function, album. The next thing to do is increment the x variable by the _width of the newly attached MovieClip plus two pixels. The two pixels add a little extra space between each thumbnail so it doesn't look like one long image. This is also the extent of my design consideration. Just kidding...

Well, we can create thumbnails (see Listing 6).

This is a very simple way to load an image. Now, by using a ScrollPane we could have simply set the contentPath of the ScrollPane to the location of our image and it would have worked like a charm. The reason I stuck with loadMovie() is to follow suit with the thumbnails setup. This also leaves you the ability to ditch the ScrollPane and load it into your own location or use the MovieClipLoader class, as we will see later.

figure 2

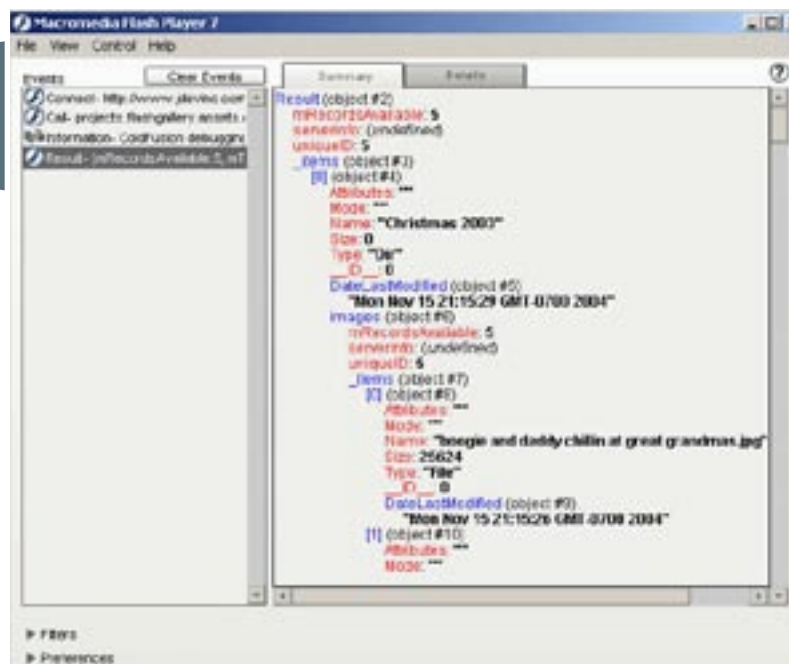
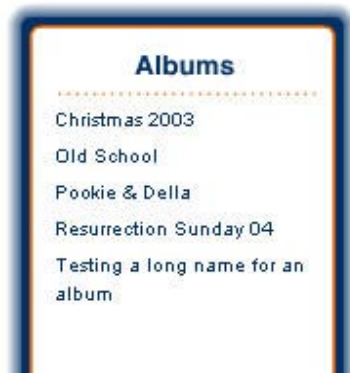




figure 3



Our first task is to refresh our pane, just like the createThumbnails() function, then create our new MovieClip in the largeSP variable (remember this is a reference to the root.large_sp's content). Next we loadMovie(), our file parameter, into our new largeSP.large MovieClip and we are done loading content. Of course we could do a nice and pretty preloader but I figured that is an upgrade you can implement, if you so desire. In the ColdFusion version we would display the name of the current image after "Pic :: ". We'll do the same here by setting the text property of root.image_txt to the name parameter we will receive from the thumbnail.

That's pretty much the bulk of the application as this class is by far the largest, although small itself. If you were to set up the FLA right now and test, you would see the albums and that is it. The functionality is there to create the thumbnails and load the large image but we haven't given any life to the album MovieClips.

Well, just like a good soap opera I'll leave you in suspense. Hopefully you don't do as I did and give up on waiting for the next episode. We are almost done though, so hang in there. In my next article we will cover the thumbnail images and displaying the large images. See you soon! :-)

Resources

I thought I would highlight some of the resources I used while writing this article as well as when developing the app. Of course the Flash Help Docs were referenced numerous times for precision but other than that, here are some more resources:

- *Information and articles on Flash Remoting:* www.macromedia.com/devnet/mx/flashremoting

- *Information on implementing the EventDispatcher class, Kenneth J. Toley, III's article:* www.macromedia.com/devnet/mx/flash/articles/creating_events.html
- *Information on the MVC architecture:* www.macromedia.com/devnet/mx/flash/articles/mv_controller.html.

Acknowledgements

Most of all I want to acknowledge God for Deut. 8:18, my family (Tia, Adreana, and the newborn John C. Bland III [bday 1/10/2005]), my parents, and my brothers for continued support. T, thank you for allowing me time to write! Mad Love! I want to send big thanks to Sarge (sargeway.com/sarge), Robert Hoekman (widgetmaker.net), The Flash and Multimedia Users Group of Arizona (newly moved to gotoand-stop.org), and The Phoenix ColdFusion Usergroup (azcfug.com). I also have to give mad props to Michael Hagel (michaelhagel.com) for banging out the site graphics for this app and for writing the design segment. Keep up the sweet work! Oh, and I REALLY, REALLY listened...for real! ☺☺

CMS300 View In: English (US) Add: -select language-

You find the perfect content management solution.



It has all the features you want at a fraction of the cost expected.

It deploys in hours, not weeks.

Everyone loves it. Everyone uses it.

Everyone upstairs pats themselves on the back.

Download a 30-day free trial at www.ektron.com/cfdj


```

class classes.Data{
    private var data_cfc:Object;
    private var remotngURL:String = "http://www.jdevinc.
com/flashservices/gateway";

    public function Data(){
        init();
    }

    public function init():Void{
        NetDebug.initialize();
        mx.events.EventDispatcher.initialize(this);

        data_cfc = NetServices.createGatewayConnection(remo
tingURL).getService("projects.flashgallery.assets.cfcs.
data", this);
    }

```

```

function dispatchEvent():Void{
}
function addEventListener():Void{
}
function removeEventListener():Void{
}

public function trigger():Void{
    data_cfc.getAlbums();
}

private function getAlbums_Result(result) :Void{
    dispatchEvent({type:'dataIsHere', data: result});
}

private function getAlbums_Status(status) :Void{
    dispatchEvent({type:'dataFailed', data: status});
}
}

```

```

class classes.Manager{
    private var dataClass:Data;

    public var root:MovieClip;

    private var thumbsSP:MovieClip;
    private var largeSP:MovieClip;

    public function Manager(r){
        root = r;
        init();
    }

    private function init(){
        root.createEmptyMovieClip("albums", root.getNextHigh-
estDepth());

```

```

        root.albums._x = 35;
        root.albums._y = 120;

        thumbsSP = root.thumbs_sp.content;
        largeSP = root.large_sp.content;

        dataClass = new Data();
        dataClass.addEventListener("dataIsHere", this);
        dataClass.addEventListener("dataFailed", this);
        dataClass.trigger();
    }

```

```

private function dataIsHere(ev:Object):Void{
    var y:Number = 0;
    var len:Number = ev.data.length;

    for(var i:Number = 0; i < len; i++){
        root.albums.attachMovie("album", "album" + (i+1),
root.albums.getNextHighestDepth(), {_x: 0, _y: y, myDa-
ta: ev.data.items[i], Manager: this});
        y += root.albums["album" + (i+1)]._height+2;
    }
}

private function dataFailed(ev:Object):Void{
    //do failed stuff here
}

```

```

public function createThumbnails(rs:RecordSet, album:
String):Void{
    thumbsSP._parent.refreshPane();

    var x:Number = 0;
    var len:Number = rs.length;

    for(var i:Number = 0; i < len; i++){
        thumbsSP.attachMovie("Thumbnail", "thumb" + (i+1),
thumbsSP.getNextHighestDepth(), {_x: x, _y: 0, album:
album, myData: rs.items[i], Manager: this});
        x += thumbsSP["thumb" + (i+1)]._width+2;
    }
}

```

```

public function loadLarge(file:String, name:String):
Void{
    largeSP._parent.refreshPane();
    largeSP.createEmptyMovieClip("large", 1);
    largeSP.large.loadMovie(file);

    root.image_txt.text = name;
}
}

```



The Flow Between Fireworks & FreeHand

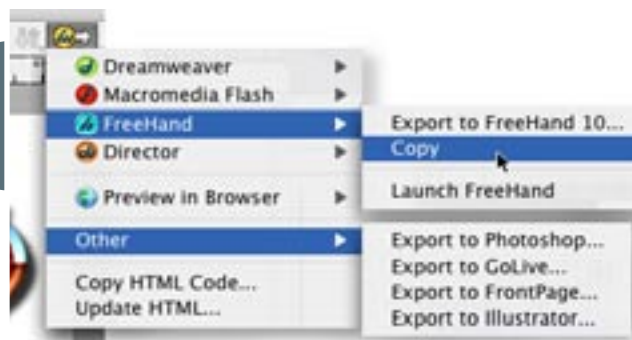
by ron rockwell

These days it's a slippery slope we wander between graphics for onscreen viewing and graphics designed for printing. Depending on your own background, you may be more prepared for one application than the other. Working with **FreeHand** and **Fireworks** can be slightly daunting, but immensely rewarding.





figure 2



Macromedia defines Fireworks as “the easiest way to create, optimize, and export interactive graphics in a single, Web-based environment.” FreeHand is described as a vector-based drawing application used to “create print and Web graphic illustrations such as logos and advertising banners.” If I may, I’d like to add a little information to those descriptions.

Fireworks works primarily and ultimately in bitmaps, or rasters as they are also called. By definition, these files are created for viewing on a computer monitor in one way or another, usually via the Internet. Vectors are used in some drawing procedures, but in the end are converted to bitmaps. The drawing tools are adequate, but not nearly as precise or numerous as those contained in FreeHand. Conversely, FreeHand uses vectors for the majority of the work done, and has a few bitmap effects that both soften and amplify the vector art.

Fireworks isn’t the program to use if you are ultimately going to print your artwork. That’s not to say that you cannot design and complete a logo in Fireworks with the hopes of getting it on paper; it only means that there’s a bit more work to be done – in FreeHand. Alternately, you shouldn’t spend a lot of time creating Web graphics in FreeHand that could just as easily be done in Fireworks. Instead, general or precise shapes can be created in FreeHand and exported to Fireworks for Web preparation.

Beyond obvious differences in the tools involved in creating artwork in the two programs, there are major and minor inconsistencies between the programs guaranteed to irritate or confuse the best of us, if not drive us mad! First we’ll discuss the ins and outs of Fireworks, then the same situations in FreeHand in order to help you create a smooth and efficient workflow. Each of these programs is good in its own right, but when they’re used in combination, they become downright powerful.

Fireworks

To begin, you must remember that finished artwork from Fireworks is bit-mapped, and that means it is resolution dependent. If you enlarge the artwork, you’ll be embarrassed by jaggies and crude imagery. One-size-fits-all does not apply for Fireworks. What all this means is that if you want a logo to fill a 300-pixel by 150-pixel space, you must create the logo at exactly that size. Yes, you can create it larger and reduce it, but you’re better off doing the artwork on a 1:1 basis.

The resolution in a Fireworks document is 72 ppi (pixels per inch, sometimes referred to as dots per inch) by default, but that number can be changed in the Modify>Canvas>Image Size menu. Figure 1 shows the difference in resolution from 72 ppi to 144 ppi to 300 ppi. Briefly, increasing the resolution increases the image’s physical dimensions. What is happening is that your monitor displays

72 ppi, and doubling the image’s resolution requires the image to double in size – on the monitor – in order to maintain the 72 ppi view. If you notice the file dimensions, you’ll see that the number of pixels will increase as the resolution increases, while the physical dimensions of the object remain the same. If you look a little more closely at Figure 1, however, you can see that the crosshatched patterned fill remains the same size throughout the resolution range. If you did a logo such as this at 300 ppi and saved it as a TIFF file, the pattern would be pretty much negated when ink hit the paper. By the same token, if you did the logo at 300 ppi and changed its resolution to 72 ppi, you may not be pleased with the way some effects appear. So, to summarize, it’s best to do your Fireworks Web-bound artwork at same size or 1:1 to keep your sanity.

Fireworks has many nice bitmap effects, such as shadows, bevels, glows, transparency, and so on, but depending on the file type or method of export, not all of these effects will be retained. As long as your art is headed for the Web, there’s nothing in Fireworks that won’t work just fine.

Export to Freehand

The rub can happen when you remove an object from Fireworks for use in another program. This article will only describe what happens between Fireworks and FreeHand.

Raw File Transfers

If you have both FreeHand and Fireworks running, it’s pretty simple to select the Fireworks artwork and drag and drop it onto the FreeHand page. Be sure to save both documents before attempting any file transfers! Crashes usually happen when you can least afford them. You can also copy the Fireworks object and paste it into FreeHand.



However the most convenient method is to select the object or objects and click on the Quick Export button at the top right corner of the Fireworks document window. That menu offers exports to FreeHand, Director, Dreamweaver, and Flash. There's an extended "Other" menu that allows you to export to other programs as shown in Figure 2. Any of these methods will give you the optimum file transfer for the next intended use of the graphic. In the case of FreeHand, copy and paste, drag and drop, or copying from the Quick Export menu, all provide an editable illustration. Depending on the settings you apply in the various dialog boxes that will appear, most of the artwork will remain editable – or not, as you see fit – in vector form.

Some attributes may be lost in the translation. For instance, if you have a patterned fill, it will not appear when it is exported to FreeHand (see Figure 3). At first blush, you may think that you could simply apply the appropriate pattern within FreeHand, but be advised that patterned fills in FreeHand are not recommended for high-quality printing as they default to 72 ppi. If you wish to have a patterned fill, you're best off creating a tiled fill or otherwise creating the fill manually. Fireworks cannot have a stroke less than a pixel due to its resolution-dependent makeup – after all, you can't show half a pixel at any resolution. So you'll have the advantage of being able to fine-tune stroke widths once the graphic is in FreeHand. But the main thing you'll notice is the color shift. Fireworks is RGB, period (WMBP notwithstanding). Even when you convert a piece of art to a Grayscale GIF or use just two colors in the document, it's still an RGB file. Disregarding black and white files, FreeHand will use its RGB rendering to display the image, and you can bet that colors will look completely different. The colors are correct, however. If you want to assure yourself, use the eyedropper tool in either program and drag a color selection to the Color Mixer panel. But... if the image is going to be printed, then it will be converted to CMYK and you will have a color shift, usually a dulling or desaturation of colors. There's nothing you can do about that. To allay your fears or concerns in FreeHand, you can choose Xtras>Colors>Name

All Colors, then in the Swatches panel convert all those colors to CMYK. You'll see an immediate difference on-screen. Print the document and make any adjustments you think are necessary.

Another way to get graphics into FreeHand is to use Fireworks' Export menu. You can choose from PNG, JPG, GIF, TIFF, BMP, WBMP, and PICT. However, if you take this route you won't be able to edit the file in FreeHand because the file becomes bitmapped (see Figure 4). You will be able to use the Fireworks roundtrip editing button and edit the file back in Fireworks, though. The resolution of an exported object will remain the same. On the other hand, if you've dragged and dropped, copied and pasted, or used the Quick Export options, the artwork will take on the resolution of the FreeHand document. They don't make it simple, do they?

Export to Fireworks

Okay, now you have to switch hemispheres in your brain – we'll be talking about vectors originating in FreeHand and being introduced into Fireworks. In this case, you've designed a logo in FreeHand and now you want to place it on the Web. Going in the opposite direction, you can drag and drop, or copy and paste FreeHand artwork into an open Fireworks document. You can also choose File>Export, and choose from many different formats, but PNG, JPG, or GIF would work best if you're going straight to the Internet with the file. Our original FreeHand logo is shown in Figure 5. Because of the greater number of tools and tricks available, the logo is a bit more complicated, with tiled fills, a drop shadow made from a blend, and many layers.

You must pay attention to the condition of the artwork when you move it from FreeHand to Fireworks. For instance, if everything is ungrouped and you copy and paste or drag and drop, the art will land in Fireworks pretty much as it was in FreeHand. There's quite a bit of anti-aliasing that goes on, and any strokes that are thinner than 1 point become a full point in width – which can come as quite a surprise on detailed drawings or logos! You will be able to move, edit, or otherwise manipulate virtually all the FreeHand objects in your Fireworks document.

Figure 6 shows a drag-and-drop ver-

figure 3



figure 4



figure 5



figure 6



sion of the FreeHand logo in Fireworks. Notice that a half-point white stroke around the text has gained weight – and doesn't look half-bad. Oddly enough, the text is softer, but the tiled background is harsher. Regarding the tiled fill, in FreeHand the tiled fill is simply applied to the ellipse, and there's no "overflow" of the fill. But Fireworks sees the entire tiled fill as a graphic inside a mask, so the fill object is much larger than the actual graphic.

Bitmap and vector effects carry over with the FreeHand artwork when you copy and paste or drag and drop. The effects are editable, but as you might expect, attempting changes to some objects with effects can give you a surprise or two. Those surprises can be remedied easily enough. One instance is a gradient fill that has a drop shadow. Changing a feathered effect to a hard edge results in the gradient's changing directions. It's nothing that can't be fixed quickly, but it's something to be aware of.



If you have increased the Raster Effects Setting in FreeHand to 300 ppi to accommodate drop shadows or other effects, those effects will acquire the resolution of the Fireworks document. If it is a brand new Fireworks document, that means the effects will drop to 72 ppi. You can enlarge the vector-based artwork in Fireworks with complete confidence that the image will not be distorted as a bitmap would. Any vector or raster effects you've applied to the path or object import fine, and will be scaled properly with the object.

Layered FreeHand artwork will retain the layers, but layer names are lost. Instead, objects from individual layers are given generic names such as "Composite Path" or "Group."

There's a big difference, though, if you group the artwork before placing it in Fireworks. It becomes a single flat-

tened bitmap graphic that can only be reshaped or drawn over, and tiny details like thin strokes also disappear. You will not be able to edit any of the vectors that created the object, but bitmap editing is possible.

Other differences occur when you have an open Fireworks document and you choose to import a FreeHand document. The first thing you're greeted with is the Vector File Options window shown in Figure 7. Obvious and not-so-obvious choices are available and both are convenient and time-saving. For instance, if you know you'll want the FreeHand object to be smaller in Fireworks, you can change the scale right off the bat. You can also choose to distort the object's size by inputting width and height values. The resolution can also be modified. As a default, the FreeHand document will be at 300 ppi (except for any raster effects – keep your mind clear!). There are choices for Anti-Alias effects on paths or text, and the level (Smooth, Strong, Crisp, or System).

Then you can decide how you want the file converted, starting with the page or pages you wish to import. If you want a specific page in a multipage document, select the page number from the dropdown menu. You can choose instead to open pages as individual frames.

The next dropdown menu concerns layers; you can ignore them, remember them, or convert the layers to frames, depending on what you want to do in Fireworks – there's no magic bullet for all jobs. Last, you can choose to include layers that you've turned off in FreeHand (invisible layers), or to include background layers or not.

Then you're down to the Render as Images section, where you choose the level of simplification you wish the artwork to have. By setting an amount in the text fields, you tell Fireworks to convert the vector object into a bitmap – thereby minimizing your editing choices. You can set these numbers high enough to keep everything editable, but at the expense of a sluggish Fireworks document.

If you select a compound path (masked object) in Fireworks and place it into FreeHand, you'll be surprised to see the object as if you'd chosen "Cut Contents" from the Edit menu. What happens is that the objects involved in the compound path are separated, but grouped. To restore order, simply ungroup the group, cut the appropriate objects and use Edit>Paste Inside (Paste Contents).

On more than one occasion (today in fact), I have imported an ungrouped, fairly complex FreeHand document and seen some objects rendered at different scales than others. That frankly scares the heck out of me, so I'm extremely watchful.

Text Issues

There's one minor trouble area when it comes to dealing with text, and that's kerning. If you apply a -10% kerning to a block of text in FreeHand, it will read as -8% in Fireworks. The physical change between the letters will remain the same, but they pick up the differing values. I bring this up in case you're working with text back and forth between the two programs and want to remain consistent. Figure 8 shows the differences. As you can see, a -10% kern in Fireworks will probably not be acceptable in most cases – at least in this particular font.

Editing Fireworks Objects from Freehand

Just as Fireworks has its Quick Export button to get a document into another program, FreeHand allows you to edit bitmaps directly in Fireworks through the Edit in Fireworks button at the bottom of the Object panel. Simply select a bitmap object – it could be a digital photo, or have been created in Fireworks – and click the Edit in Fireworks button. Fireworks will start up if it's not open, and you'll be greeted with a Find Source for Editing window as shown in Figure 9.

figure 7



figure 8

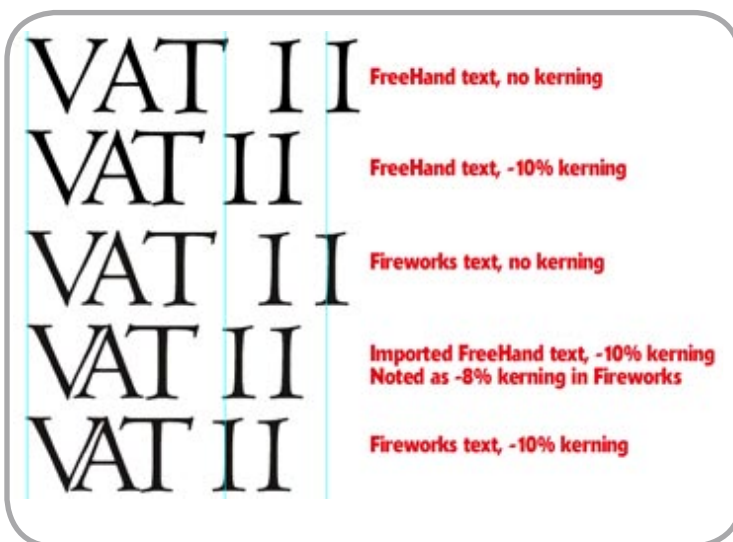


figure 9



The Find Source window is much more important than you may at first think, because you can cause irreparable damage to an original file if you're not paying close attention. Here's how it works: you are given the immediate choice of selecting a PNG or the actual file to edit. If you select Use This File, then you will be editing the original document, just as if you had opened the original in Fireworks (or Photoshop) and done the editing. Changes are permanent. On the other hand, you can choose Use a PNG. At that point a navigation window appears so you can find the original PNG to work on. If you don't have one, the original file opens anyway... scary stuff. When you're not sure whether you want to make lasting changes to an image,

it's best to do a Save As and work from a new copy of the image just to be safe.

At the bottom of the window is the Fireworks Source Files dropdown menu, and you have three choices: Always Use Source PNG, Never Use Source PNG, or Ask When Launching. The last choice speaks for itself; Always Use Source PNG will open the Fireworks PNG for the placed image. The source PNG and the placed image will both be updated – changes are permanent. Never Use Source PNG opens the placed image, whether a source PNG can be found or not, and the editing only applies to the placed image – the original (wherever it may be) is not touched. Making a selection from this menu will apply to all future edits, and will pre-empt the Find Source window.

When you're done with the editing, click the Done button. You'll be returned to FreeHand, the image will take a few seconds to redraw, and you can get back to work.

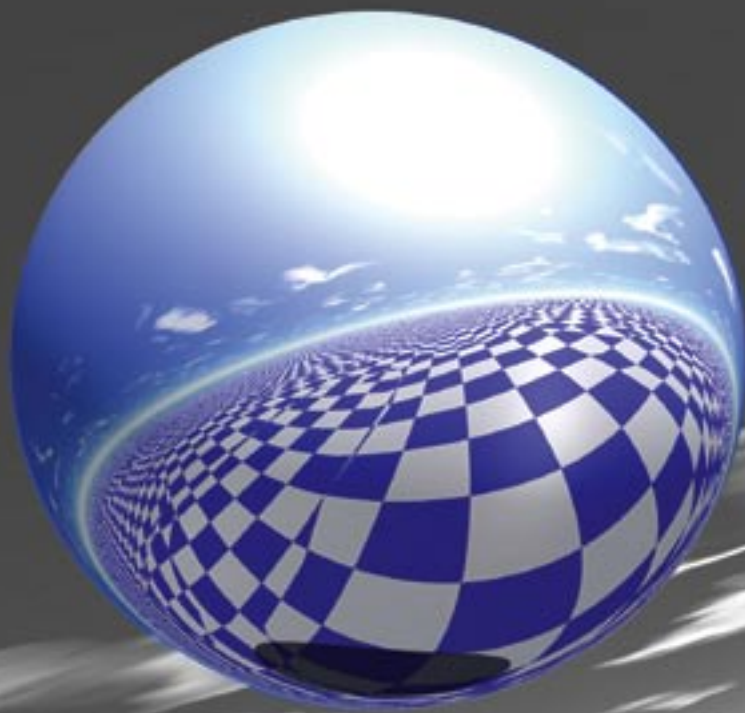
If your bitmap has been placed from Fireworks in the first place, you'll bypass the Find Source window and the original Fireworks document will open right away.

I hope I've shed a little light on working between and with both Fireworks and FreeHand. With a little patience, you can get super results for print from FreeHand, and for the Web with Fireworks. ☺

Illustrator, designer, author, and Team Macromedia volunteer Ron Rockwell lives and works with his wife, Yvonne, in the snowy Pocono Mountains of Pennsylvania. He is the author of FreeHand 10 f/x & Design, and co-authored the Studio MX Bible. He has just introduced a FreeHand course available at 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

					
			<p>INTRODUCTORY PRICE</p> <p>\$45</p> <p>Plus \$5 s/h. Reg. Price \$69.95</p>		
		<p>Learn FreeHand MX</p> <p>Create graphics for print and the Web.</p> <p>Work at your own pace with interactive course.</p> <p>Great gift idea - order now for the holidays!</p> <p>www.brainstormer.org</p>			
					

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



VISUAL DEBUGGING TOOLS FOR SHOCKWAVE

by andy phelps

Recently, I have been working in Shockwave3D in Lingo and JavaScript syntax. Regardless of what I am writing, I keep running into issues that can be summed up by the following phrase: I don't know exactly what I'm doing. More precisely, I don't know exactly what I am doing when I am doing it. I'm figuring it out. And that's what programming interactive things is all about: finding a way to do what you need to do, so that the users will have the experience you want them to have. Unfortunately, it is sometimes difficult to tell exactly what your code is doing, and this is particularly true in 3D environments.

3D



So with these issues in the back of my mind, I was working on my lightmap generation tool, and I was really getting stuck. A "lightmapper" is a tool that pre-generates, through raytracing or other means, a series of maps that represent the lighting in a 3D scene as textures on the individual objects. This is a very popular technique in game-level design, as Brian Robbins noted at MAX, and as several authors have noted in the game development community. Figure 1 shows a few sample renders from my tool in progress.

The problem I was having was in projecting the shadows, particularly in figuring out the angle to each light from the points along the surface. I was never really "sure" of exactly where the ray was that I was checking against. This is relatively easy to figure out for a single light, but I was getting very confused when calculating multiple light sources (see Figure 2). In order to figure out just what was going on, I used a strategy that I have used in the past: build a visual "prop," or "stand-in" of the ray itself. The only complication is

that Shockwave3D has no #line primitive. No problem! Just make a very long, skinny triangle. Listing 1 shows a Lingo handler that creates a "connector" object: it has a triangle that it uses as a "line," which it can snap between a beginning and end point. Additionally, it can color each end of the "line" a different color, and blends the two together along the triangle face.

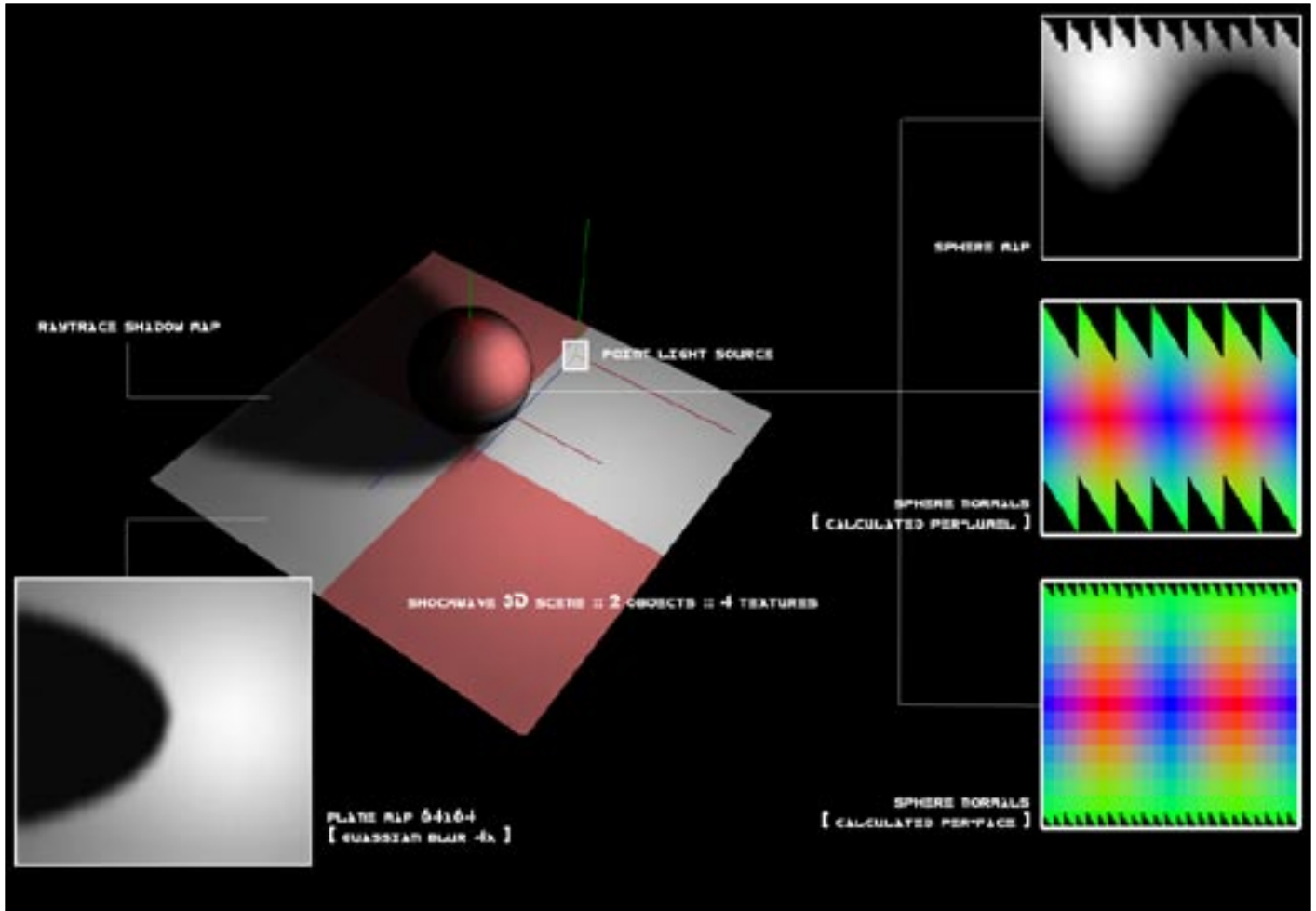
Using these lines, I was able to trace out each and every light path in the scene, and get a sense of whether or not it was doing what I wanted it to do (see Figure 3). This was very handy, but this tool isn't limited to that specific use. I've used it in the past to represent surface normals (something I wish S3D had a #debug flag for), direction vectors, a "point at" vector between two objects, and even rotational axes when I haven't been able to see the ones drawn through the #debug flag. (For some reason, the axes generated by Director are all black when drawn with the DirectX7_0 renderer on my nVidia cards). Because each end of the "line" can be colored separately, they can be used to repre-

sent directional vectors, rather than just a straight connection between two points.

In order to get the "lines" to show up well in a debug environment, I generally create a custom shader that ramps up the emissive and ambient qualities (and sets some flags so that the vertex colors actually have an effect). For the script in Listing 1, I used the shader props set in Listing 2. I've also used "lines" in a similar way in JavaScript syntax.

Being able to "see" what is actually going on has proved invaluable for this and several other projects. Using color and shape makes it much easier to tell what the code is doing than an iterative 'run and see' approach. By thinking of debugging visually, tools can be constructed that make it easier to see just what is going on. If you have either already built tools, or are thinking of writing some after seeing these simple examples, I encourage you to share them with the community as you are able, to build up a library of visual aids that help us in our daily work. ☺

figure 1





HostMySite.com

Built for ColdFusion Pros

by ColdFusion Pros

plans from

\$8.95 / mo.

FREE Domain Name*

FREE Setup

FREE 2 Months

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

Visit www.HostMySite.com/mxdj for:

2 Months Free

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



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

James Kennedy
mbateam.com

*offer applies to any annual shared hosting plan

call
today!

877•248•HOST
(4678)



figure 2

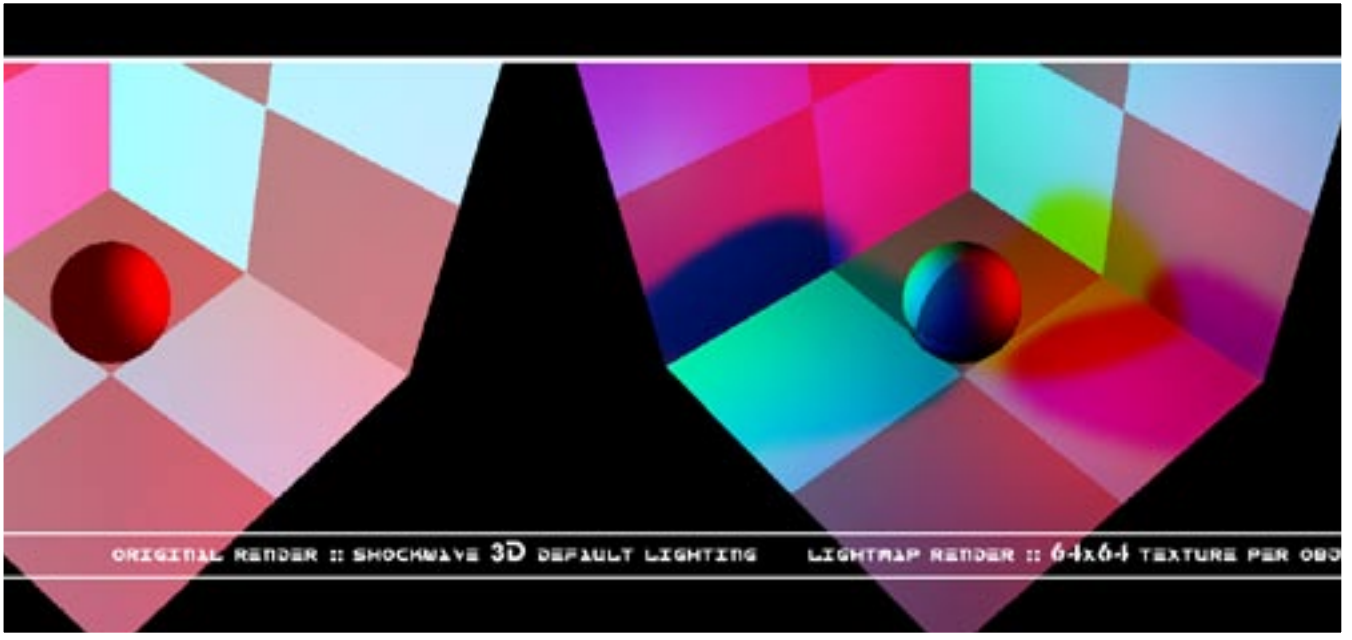
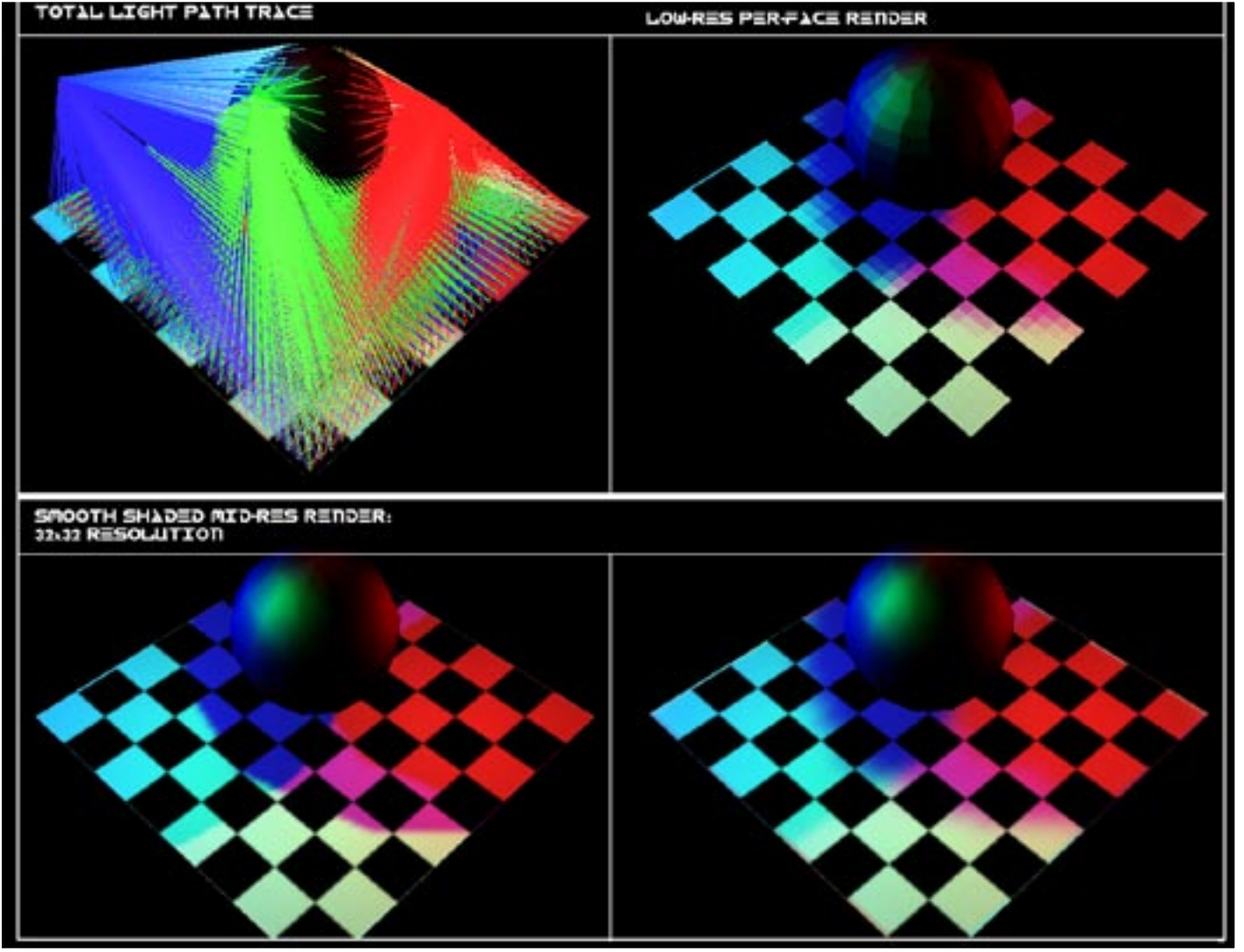


figure 3



Subscribe Today!

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

FREE RESOURCE CD INCLUDED!



Microsoft
RSA Security
SecureID for Windows

SAN NAS

Emerging technology trends
and market maneuvers



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 \$39⁹⁹
ONE YEAR
12 ISSUES

www.ISSJournal.com

OR

1-888-303-5282

**SYS-CON
MEDIA**

The World's Leading IT-Technology Publisher



listing 1

```

-----
--3D DEBUG LINE
-----
property p_vPosA    -- position A of line
property p_vPosB    -- position B of line
property p_mshMesh  -- line mesh
property p_mModel   -- line model

-----
-- ThreeDLine::New()
-----
-- Purpose:
-- a_sName      - string name of the line to be created
-- a_aColor     - array of 2 rgb colors [start, end]
-- a_vPoint1    - vector start of line
-- a_vPoint2    - vector end of line
-- a_shShader   - shader to be applied to the line
-----
on new me, a_sName, a_aColor, a_vPoint1, \
    a_vPoint2, a_shShader, a_3dWorld

    me.p_vPosA = a_vPoint1
    me.p_vPosB = a_vPoint2

    --create a mesh for this connector
    if voidP(a_3dWorld.model(a_sName)) then

        --continue
    else
        return 0
    end if

    me.p_mshMesh = a_3dWorld.newMesh(a_sName,1,3,3,3,0)

    --set color list
    me.p_mshMesh.colorList = [ a_aColor[1], \
                                a_aColor[2], \
                                a_aColor[1] ]

    --set vertex list
    me.p_mshMesh.vertexList = [ vector(0,0,0), \
                                vector(0,100,0), \
                                vector(100,100,0) ]

    --set normal list
    --NOTE: this is a hack, normals here make ok
    --lines, but are not technically correct for 3D
    --lighting...

    me.p_mshMesh.normalList = [ vector(1,1,1), \
                                vector(1,1,1), \
                                vector(1,1,1) ]

    --set the vertices and colors into the mesh
    me.p_mshMesh.face[1].vertices = [1,2,3]
    me.p_mshMesh.face[1].colors = list(1,2,3)

    --build the triangle
    me.p_mshMesh.build()

    --create a model from our triangle
    me.p_mModel = a_3dWorld.newModel(a_sName, \
        me.p_mshMesh)

    --set render ops for best debug lines
    me.p_mModel.visibility = #both

    --set shader on this model
    me.p_mModel.shader = a_shShader

    me.mUpdate(me.p_vPosA, me.p_vPosB)

    return me
end ThreeDLine

-----
-- ThreeDLine::mUpdate()
-----
-- a_vPointA    - vector new start position
-- a_vPointB    - vector new end position
-----
on mUpdate me, a_vPointA, a_vPointB

    -- make the connector position itself between the A
    -- and B parent nodes. This is a little but tricky.
    me.p_vPosA = a_vPointA
    me.p_vPosB = a_vPointB

    vPosC = vector( me.p_vPosA.x, \
                    me.p_vPosA.y - 2.000, \
                    me.p_vPosA.z - 2.000)

    me.p_mshMesh.vertexList = [me.p_vPosA, \
                                me.p_vPosB, \
                                vPosC]

```

Andrew Phelps is an assistant professor in the B. Thomas Golisano College of Computing and Information Sciences at the Rochester Institute of Technology. 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. Andy regularly teaches coursework in multi-media programming, game programming, and simulation/visualization. amp@it.rit.edu

```
end mUpdate
```

```
//-----
function ThreeDLine(a_sName, a_aColor,
                   a_vPoint1, a_vPoint2) {

    this.p_vPosA = a_vPoint1;
    this.p_vPosB = a_vPoint2;

    //create a mesh for this connector
    this.p_mshMesh = _global.D3D_WORLD.getProp(
        "g_3DWorld").newMesh(a_sName,1,3,3,3,0);

    //set color list
    this.p_mshMesh.colorList = list( a_aColor[1],
                                    a_aColor[2 ],
                                    a_aColor[2]);

    //set vertex list
    this.p_mshMesh.vertexList = list(vector(0,0,0),
                                     vector(0,100,0),
                                     vec-
tor(100,100,0));

    //set normal list
    //NOTE: this is a hack, normals here make ok
    //lines, but are not technically correct for 3D
    //lighting...
    this.p_mshMesh.normallist = list( vector(0,1,0),
                                     vector(0,1,0),
                                     vector(0,1,0));

    //set the vertices and colors into the mesh
    //NOTE: Need to use getPropRef to parse the lingo
    //arrays that are embedded in the S3D Xtra
    this.p_oFace = this.p_mshMesh.getPropRef("face",1)
    this.p_oFace.vertices = list(1,2,3);
    this.p_oFace.getPropRef("face", 1).colors =
        list(1,2,3);

    //build the triangle
    //this.p_mshMesh.generateNormals(symbol("flat"));
    this.p_mshMesh.build();
```

```
//create a model from our triangle
this.p_mModel = _global.D3D_WORLD.getProp(
    "g_3DWorld").newModel(this.p_sName, this.p_mshMesh);

    this.p_mModel.shader = _global.D3D_WORLD.getProp(
        "g_3DWorld").
getProp("shader",1);

    this.p_mModel.shader.shininess = 0;
    this.p_mModel.shader.blend = 10;
    this.p_mModel.shader.texture = 0;
    this.p_mModel.shader.emissive = color(50,50,50);

    this.mUpdate(this.p_vPosA, this.p_vPosB);

    return this;
}

/*-----*/
Connector::mUpdate

-----*/
ThreeDLine.prototype.mUpdate = function(a_vA, a_vB) {

    //make the connector position itself between
    //the A and B parent nodes.

    this.p_vPosA = a_vA;
    this.p_vPosB = a_vB;

    var vPosC = vector( vPosA.x,
                       vPosA.y - 2.000,
                       vPosA.z - 2.000);

    this.p_mshMesh.vertexList = list(vPosA,
                                     vPosB,
                                     vPosC);
}
//-----
```




getting started with

macromedia
captivate

by lisa heselton

B

y now you've probably heard about Captivate (formerly RoboDemo), Macromedia's tool for creating engaging software simulations, demonstrations, and tutorials. You've decided to take the plunge and create your first "captivating" demonstration, but how do you get started? This article will walk you through the basics of producing your very first Captivate project. If you don't own Captivate, you can download the 30-day trial from Macromedia's Web site. Captivate is currently Windows only.



figure 1



should break up or 'chunk' your work into smaller segments. This helps the viewer understand the information, and also helps keep your file sizes smaller. If you try to cram too much action into one movie, you may end up choking systems with older graphic cards. Instead, create a menu for each segment and create separate movies accordingly. This can be done using the MenuBuilder application that is a part of Captivate.

Once you have a good idea of what your movie will need and how it should flow, you can begin. You may need to do several "takes" to get some elements of your project exactly how you want them, but most actions that you record will be easy to adjust and edit after you have finished recording.

Starting a New Project

When you first open Captivate, you are presented with a dialog screen. This screen shows you what you can do within Captivate – open an existing movie, create a new movie, etc. Since this is your first Captivate project, you'll select the "Record or create a new movie" option (see Figure 1).

Once you've chosen to start a new movie, Captivate still needs more information from you – it needs to know whether you'll be recording an application, the full screen, or a custom portion of your screen. If you're recording an application, you can snap it into a Captivate window to make it smaller. This enables you to fit your demo onto another interface (such as a Web page), and it reduces the overall size of your file. If you're making a movie that will be distributed on a disc or presented full-screen (such as a kiosk), then you can choose to capture the entire screen, or even select a custom portion, such as if you wanted to eliminate certain areas (e.g., the task bar). We'll do our first movie from an application (see Figure 2).

Now you'll need to navigate to the application that you want to record. Captivate will give you a series of options for recording a specific size or a preset size. If you select the drop-down menu that says "Optionally, select a window you'd like to record", the application that you choose will resize to your specified dimensions (see Figure 3).

Before you start, it's a good idea to plan ahead. That starts with asking some very basic questions:

- What are your objectives?
- Who will your audience be?
- How will you distribute your work?

If you know the answers to these questions beforehand, it will be much easier to develop your project. Captivate calls projects *movies*; like a movie, they'll have a beginning, a plot in the middle, and a conclusion. Even for small movies, it's a good idea to sketch out a storyboard. It doesn't have to be fancy (doodles on napkins are pretty commonplace), but it should help you determine the flow of your content.

Having planned your project, you next need to identify what you will need to capture. This means you have to make sure that you have everything you need before you begin – additional graphics, audio files, scripts – anything you may want to include in your movie. You can add elements later, but the more prepared you are before you begin, the easier it will be to complete your work.

If you're not sure about what types of elements you might want to include, try to find some samples of a project similar to yours. Take notes about the elements that are used, and how others use them successfully. For some projects, an audio soundtrack might be too distracting, but for others it may enhance the visuals. Try to get feedback from potential end users during development to make sure you don't go too far with any added elements.

Tip: If your project covers several objectives or describes a lengthy process, you

figure 2



figure 3



figure 4



"Once you have a good idea of what your movie will need and how it should flow, you can begin"

Before we dive into our recording, we may want to adjust some of the recording options by choosing the Options button from the Recording dialog. You can select the language that you want to use, choose a style for your screen prompts and dialogs, and choose what kind of recording to make. Since we're just getting started, we'll create a demonstration, which is a movie that shows the viewer the steps needed to perform a function or task. For now, we won't be recording any narration or changing any of the settings in the other tabs. The options to hear keystroke and camera sounds are for your assistance only; these sounds will not appear in your final project (see Figure 4).

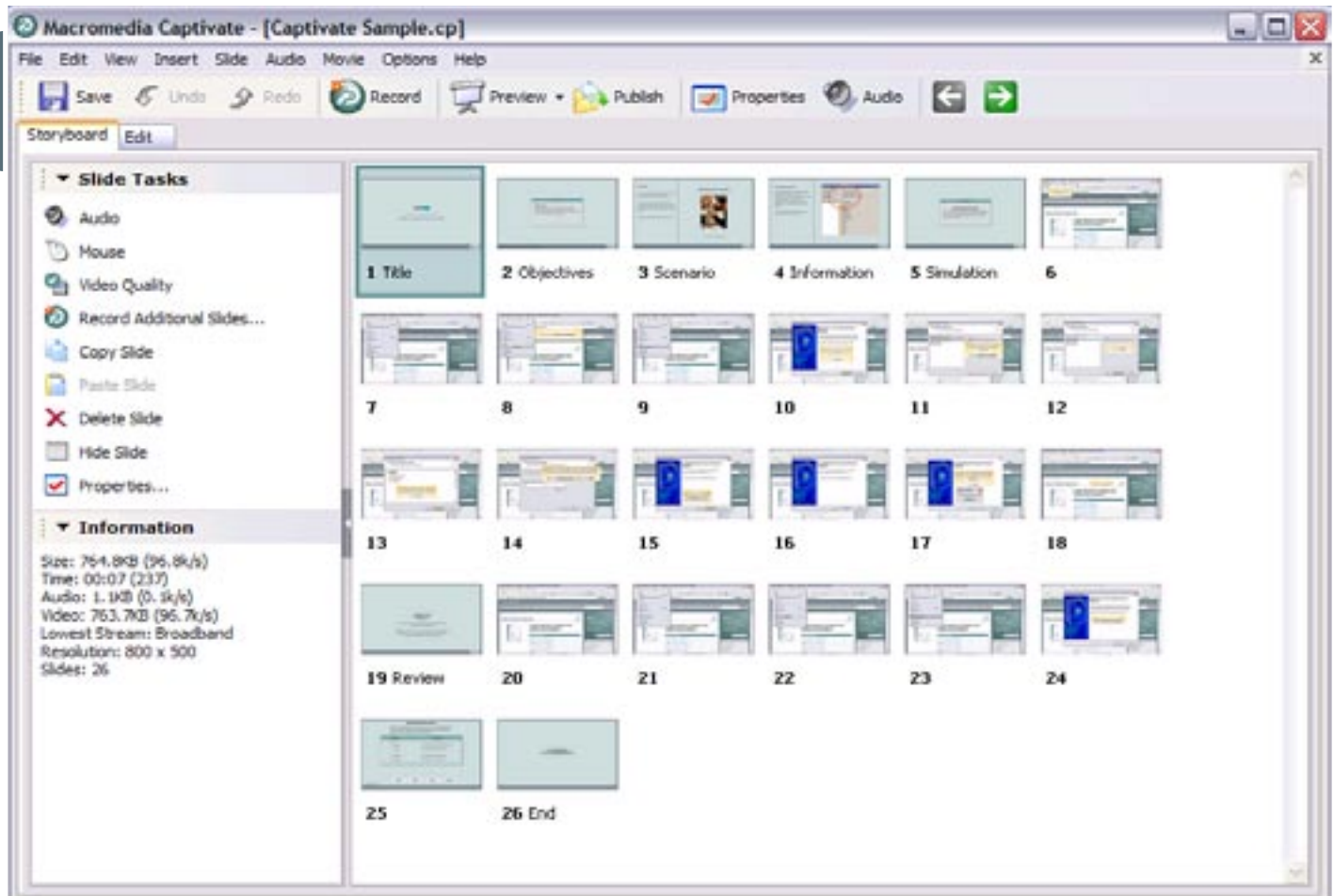
Lights, Camera, Action!

Now that your settings have been completed, press OK in the Recording Options dialog, and then the Record button to begin recording. Place the content so that the viewer can follow along with you comfortably. It is possible to capture narration while recording, but you will get much better results if you concentrate on recording the screen actions separately from the audio. You may want to read along with a script when you are recording so that your actions are in sync with any narration that will be added. You can adjust the timing of your actions later if needed.

Tip: You'll want to make a static page (or pages) to place at the beginning of your project to introduce your objective(s) and tell the audience what the project is about, what level of experience and knowledge they should have, and what they should be able to accomplish once they have viewed it.

When recording, any object that you select on the screen will trigger a screen capture – for example, if you press a Submit button, Captivate will take a snapshot of that action and record it. For more dynamic actions, such as typing a file name in a Save As... dialog, Captivate will automatically record the full process

figure 5



Advertiser	URL	Phone	Page
CFDynamics	www.cfdynamics.com	866-233-9626	5
Ektron	www.ektron.com/cfdj	866-4-EKTRON	24
EV1 Servers	www.ev1servers.net	800-504-SURF	9
HostMySite.com	www.hostmysite.com/mxdj	877-248-4678	35
Interakt	http://ktml.interaktonline.com/		6
Intermedia.net	www.intermedia.net	800-379-7729	23
ISSJ	www.ISSJournal.com	888-303-5282	37
IT Solutions Guide	www.sys-con.com	201-802-3021	45
Macromedia	www.macromedia.com/go/volvo	415-252-2000	Cover 2
Macromedia	www.webpublishingsystem.com	415-252-2000	Cover 4
Nidus Corp.	www.brainstormer.org	888-894-3840	31
PaperThin	www.paperthin.com	800-940-3087	17
Seapine Software	www.seapine.com/webdev	888-683-6456	13
ServerSide	www.serverside.net	888-682-2544	Cover 3
Web Services Edge 2005	www.sys-con.com/edge	201-802-3066	47-56

as a video segment.

When you have finished capturing your actions, you will need to press the End key to stop recording. Captivate will then assemble your work and open your movie in Storyboard view, which shows you your project in slides (see Figure 5).

You can now preview your project by selecting Preview from the top of the screen. To play the entire project from start to finish, select Movie from this menu, or press F4.

If your project is absolutely perfect, great! You can skip ahead to the section on Publishing next month. Chances are, though, that you'll need to make a few tweaks and adjustments to your work.

Editing Your Work

Once you've finished the recording process, there are two primary types of edits you may need to make: *changing what you recorded* (moving elements, deleting screens, adjusting timing, and so forth) and *adding elements to what you recorded* (sound, external files, additional Captivate recordings, etc.).

If you don't like the location of a slide or section of your project, you can make adjustments easily by selecting the desired slides and then dragging and

dropping them where you want them. If you're familiar with other slide-based applications, like PowerPoint, this is a familiar process. When you drop the slide where you want it, the slides will renumber for you. You can also remove a slide(s) by right-clicking on it and selecting Delete Slide. A dialog will ask if you really want to delete the slide. Click OK to delete it. If you need to make adjustments to a single slide, you can double-click on it to go into Edit mode for that slide.

Modifying Captions

Captivate automatically inserts text into each slide to describe the actions that were recorded. These are called *captions*. If you need to modify a caption, double-click on the slide you want to edit, then double-click on the caption itself. The Text Caption dialog will appear. From here, you can change the caption's style, font, font size, font color, and even the text itself. Keep in mind that you will want to select only fonts that will be installed on a user's system, just like you would for a Web page. Press Apply to

preview your modifications (see Figure 7).

If you select "Apply properties to all 'captions' in the movie," a global style will be created. All captions (existing or new) will take on that style within your movie. When you are finished making changes, press OK.

If you need to adjust the caption's size, drag one of the caption's selection handles to expand or contract the caption. Be careful not to obscure any text when resizing. To move the caption, click once inside it, and then drag it into place. For more precise placement, use the arrow keys on your keyboard.

Adjusting Timing

One of the greatest features in Captivate is its timeline. The timeline allows you to adjust when individual elements occur, as well as their duration. Elements can be any of the audio, text, or visual pieces that make up your slide.

This is especially useful for when you want to lengthen or shorten a segment after recording, or if you need to change the order in which elements appear. If you click on an element in the timeline to select



figure 6

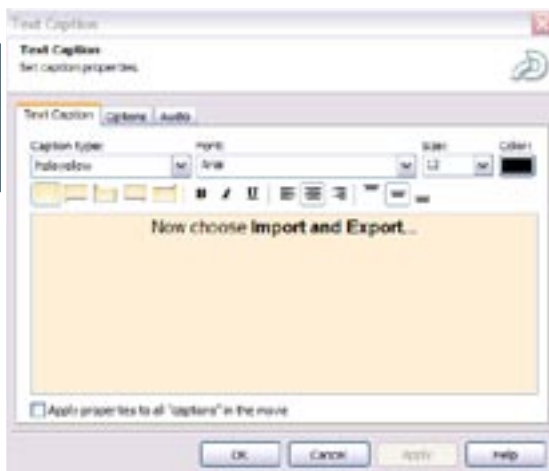


figure 7

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



"One of the greatest features in Captivate is its timeline"



it, you can drag it left (to place it earlier) or right (to have it come in later). If you need to lengthen or shorten an element's duration, select it on the right-hand side, where it ends, then move the double arrow cursor left to shorten it or right to lengthen it. You'll see the selection expand or shrink accordingly. You can preview just the slide you're editing by pressing F3.

Summary

Part Two of this article will go into detail on adding narration and audio to

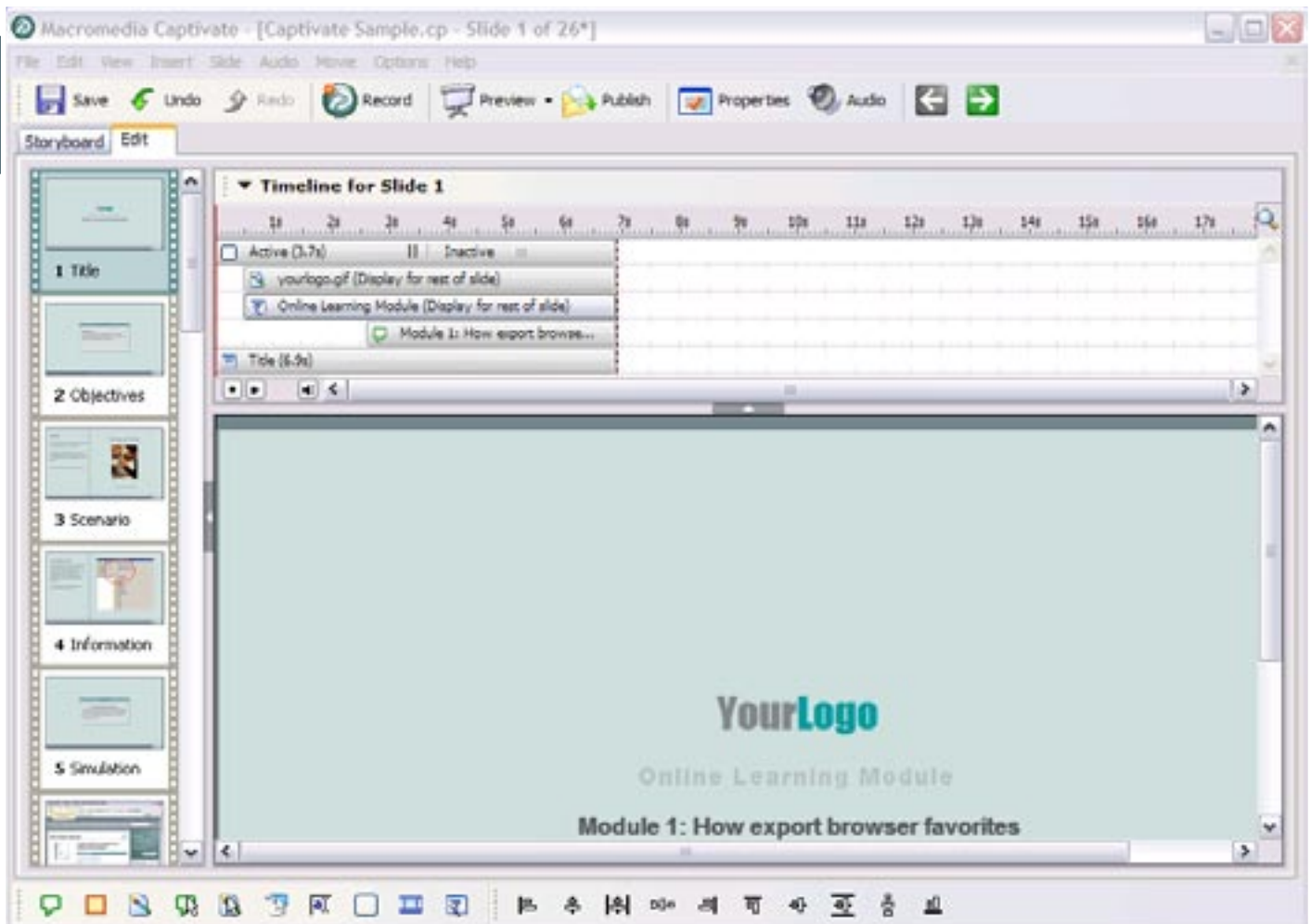
your project, importing other assets into your project, and adding a few dynamic elements to give it some life. Then we'll actually publish your masterpiece to share it with the rest of the world. See you then!

Additional Resources:

- *Captivate Demo Download:* www.macromedia.com/cfusion/tdrc/index.cfm?product=captivate
- *Captivate Developer Center:* www.macromedia.com/devnet/captivate/

Lisa Heselton is a contractor at U.S. Customs and Border Protection, developing instructor-led and Web-based learning products. She also manages the Department of Homeland Security Macromedia Users Group (www.dhsmug.us). Lisa's background includes marketing, video production, writing, business and artist management, Web design/development, and training. l.kavka@gmail.com

figure 8



events

10 pager

ever
10 p

ents ager

ever
10 p

ents ager

ever
10 p

ents ager

ever
10 p

ents ager

events 10 pager

Introducing the “Making Blackstone” Article Series

Blackstone heralds a new dawn for Macromedia ColdFusion

by **damon cooper & tim buntel**

It is hard to believe it's been nearly 10 years since Macromedia ColdFusion first began making Web application developers productive and helped them create dynamic Web sites easily. A lot has happened in the world and to application development, end-user expectations, the pervasiveness of technology in people's lives, and to the evolution of ColdFusion. However, with everything that has changed in the past decade, there are some fundamental truths in the ColdFusion mission that we have tried to ensure throughout that time and with each release. Among those principles is our belief that ColdFusion should:

- Make customers insanely productive in building applications of all sizes
- Make cutting-edge application development easy
- Make new types of solutions possible
- Make the lives of administrators and IT groups easy

The ColdFusion product went through a bit of a metamorphosis with the 6.0 [the ColdFusion MX] release several years ago. With that release, ColdFusion moved onto the Java platform and added support for CFCs (ColdFusion components) and Web services, among other features. Last year, the ColdFusion MX 6.1 release nailed bugs and stability issues, introduced dramatic new performance and scalability gains over 6.0., 5, and 4.5, and is a proven, rock-solid foundation on which we can engineer and innovate future versions of ColdFusion.

Earlier this year, we released an updater for the 6.1 release, which included hot-fix rollups and cleaned up some final issues. We feel very good about the current codebase, and we're hearing the same from customers as well.

By the way, if you're still on version 4 or version 5, you're missing out on some dramatic performance and scalability gains (ColdFusion MX 6.1 is up to 23 times faster than older versions). You can realize these benefits by simply upgrading your server.

Additionally, upgrading to ColdFusion MX 6.1 is the easiest way to upgrade to the ColdFusion Blackstone release. We're

working hard to ensure that the upgrade from ColdFusion MX 6.1 to ColdFusion Blackstone is absolutely painless, and as smooth and seamless as we can make it.

The Blackstone release will support upgrades from ColdFusion MX 6.0 and ColdFusion 5. Upgrading should be seamless, but ColdFusion MX 6.1 customers will have the smoothest transition. The reason we're so confident about this is that ColdFusion Blackstone really is the ColdFusion MX 6.1 Updater under the hood, with many additional fixed issues and all of the new and incredible Blackstone features.

On that note, we thought you would like to hear from the lead Blackstone engineering team members for each of the major new feature areas. We thought that in the spirit of the openness and transparency we've tried to maintain around this landmark release, you might like to get inside the minds of the people who have lived, breathed, and labored for well over a year and a half to make Blackstone.

Over the next few months, ColdFusion engineering team members will give you the inside scoop so you can hear first-hand accounts of the process of researching, innovating, creating, refining, testing, and polishing the new features. In some cases you'll hear about some of the internal debates, the disappointments and setbacks, the challenges and triumphs, and the major breakthroughs that all played key roles in “Making Blackstone.” Most important, you'll hear all about the new Blackstone features, have a chance to try out some of the new features with code samples, and learn what the features mean for you.

Blackstone delivers something for everyone: application developers, application users, system administrators, and emerging developers who have a limited amount of time to become productive in building Web applications.

We are extremely excited about the new features in this release and think that you will be too. We don't think we've ever witnessed a more passionate and energized group of people working together towards the common goal of delivering for our customers than the

people on the ColdFusion team during this release. In fact, the only other group of people that comes to mind who might be more passionate about what they're doing are ColdFusion customers themselves – your success is what drives us.

Accordingly, let us take this opportunity to say thank you. Thank you for your support over this incredible past decade. Thank you for your feedback and efforts to make the product better, and thank you for doing what you do.

We know many of you are the unsung heroes in your organizations, and we aim to arm you and keep you armed with innovative, productive, useful, scalable, and efficient options in ColdFusion as you solve problems, innovate new solutions, streamline processes, build new businesses, train more kids, sell more products, become more efficient, communicate more effectively, and make the world a better place in thousands of other ways.

Upcoming “Making Blackstone” Article Series

In the upcoming “Making Blackstone” article series that we will publish over the next few issues of *MXDJ*, you will have the opportunity to learn about new or updated features in Blackstone including:

- “Blackstone Reporting” by Dean Harmon, Sherman Gong, Collin Tobin, and Bill Sahlas
- “Blackstone Printable Output” by Xu Chen and Hiroshi Okugawa
- “Blackstone Event Gateways” by Tom Jordahl and Jim Schley
- “Rich Forms” by Mike Nimer and Kumaran Nallore
- “Blackstone Searching and Web Services” by Tom Jordahl and Jim Schley
- “Blackstone Clustering, Deployment, and Multi-instance Support” by Geoff Greene, Erik Tierney, Brent Baker, and Jim Murphy

Where to Go from Here

If you haven't purchased upgrades to or copies of ColdFusion MX 6.1 Enterprise with subscription, which includes entitlement to two years of software upgrades, check it out now – it's a great and cost-effective step. ☺

Damon Cooper is director of engineering for ColdFusion at Macromedia, Inc. His team is currently focused on building and delivering the next major version of ColdFusion, code-named “Blackstone.” His Web site is at www.dcooper.org.

Tim Buntel is ColdFusion product manager for Macromedia. His Web site is at www.buntel.com.

Commanders of Ingenuity

Frankly, the decision was a tough one: computer genius circa 1982 or tribute to Roller Jam (go Mr. Mean!). While both concepts felt right on for the Mindflood (www.mindflood.com) portfolio site, Sinclair Spectrum's Proclamation, "our computers can handle it" with a big thumbs up felt more effective than a reenactment of the California Quakes' signature move – the Skyscraper. By using our state-of-the-art supercomputer technology in conjunction with Macromedia Studio Pro, we created a seamless blend of nostalgia, innovation, and personality where visitors can flip through our work or view our Flash Video Web demo reel, both of which are kept up-to-date with ColdFusion MX and our Synergy content management system.



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 and ServerSide will waive the set-up fee.

ENTER CODE: mxd2004

The grass is greener at
www.serverside.net
888.682.2544
hosting@serverside.net





When was the last time your Intranet was updated?

Announcing the Macromedia® Web Publishing System. You build and manage the site. Users keep content fresh.

With Macromedia® Studio MX 2004, Contribute™ 3 and Contribute Publishing Services, you can affordably build, manage, and publish enterprise sites. And unlike the average content management system, this works. You determine who can edit and who can publish. Then users simply point and click to update any page. In no time, dated content will be a thing of the past. Learn more at www.webpublishingsystem.com

macromedia®
**WEB PUBLISHING
SYSTEM**

