

MX

developer's journal

THE LEADING MAGAZINE
FOR MACROMEDIA MX
DEVELOPERS & DESIGNERS

volume 3 issue 2 2005 www.mxdj.com

FLASH VIDEO FOCUS ISSUE



FLASH

Flash Video from the Inside
Flash Video for the Masses
Flash & Web Services
The Maelstrom in Your Future
Taking Action
Outside the Box

\$5.99US \$6.99CAN 03>



0 71486 02978 6





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





12



Flash Video
For the masses
by tom green



34



Sparkling Text Effects
Freehand's extrude tool in action
by ron rockwell



48



Introducing ColdFusion MX 7
Primarily an architectural release
by ben forta

7

A Time for Change
No time like the present
by charles e. brown

58

vanguard
The Exus
by exclude



8

Flash Video from the Inside
Exclusive Q&A with Chris Hock, Director of Product Marketing, Macromedia
interview by jeremy geelan



18

Flash, Web Services, and Data Binding
Part 1 - Be prepared to bind data like there's no tomorrow
by darron schall



22

The Maelstrom in Your Future
The next version of the Flash client is code-name "Maelstrom"
by waleed anbar



24

Taking Action
Learning ActionScript version 2.0, part 2
by charles brown



28

The "Outside the Box" Technology
An introduction to Flash Communication Server and Flash Video
by jim phelan



42

Building Reports with ColdFusion MX7
How to generate your first ColdFusion report
by collin tobin & dean harmon



54

Deploying Applications with ColdFusion MX7
Introducing sourceless and EAR/WAR file deployment
by dave carabetta

ARE YOU READY FOR

MX7

WE ARE

CFDynamics

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

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

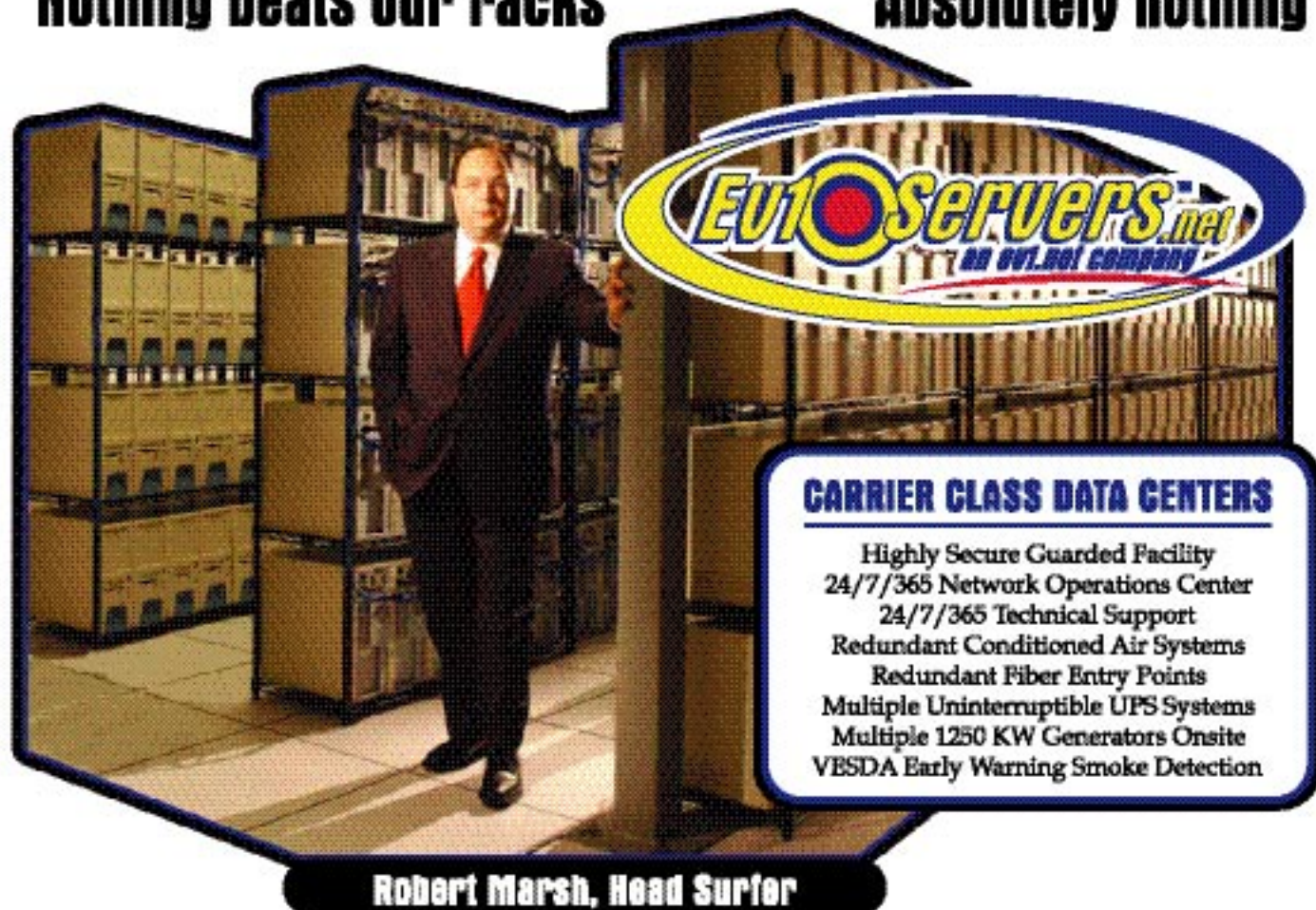
Contact CFDynamics at 1.866.233.9626 and **experience Coldfusion MX7.**

*Blackstone is the only provider for MX7. For Blackstone, call us to help you also because ColdFusion MX7 is now on hand.



Nothing beats our racks

Absolutely nothing



EVISERVERS.net
AN EVI.NET COMPANY

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 | eviservers.net

PLESK7
RELOADED
Preferred Control Panel

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

Group Publisher Jeremy Geelan
Art Director Louis F. Cuffari

EDITORIAL BOARD

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

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

EDITORIAL

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 Papazian, 201 802-3052
seta@sys-con.com

Technical Editors

Jesse Warden • Sarge Sargent

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

Subscriptions

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

Editorial and Advertising Offices

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

Worldwide Newsstand Distribution

Curtis Circulation Company, New Milford, NJ

List Rental Information

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

Promotional Reprints

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

A Time for Change

No time like the present

by charles e. brown

As we go to print, we are finally seeing the realization of what must have been the worst kept secret beta in history: "Blackstone." As a matter of fact, at the New Orleans conference in November, Macromedia finally threw in the towel and gave everyone a beta copy of it.

The beta was in remarkably final form and as of February 7 it is officially released as ColdFusion MX 7. All I can say is that it is an outstanding upgrade.

We will be devoting a significant number of pages in MXDJ to this latest release over the next few months. However, not to steal too much thunder from the two great articles in this issue, timed to coincide with the official launch – one by CF guru Ben Forta and the other by Collin Tobin and Dean Harmon – CFMX 7 has vastly expanded form and report capabilities. As a matter of fact, the form feature uses Flash components. The report-writing feature is on a par, in my opinion, with Crystal Reports. It even adds a number of extensions to Dreamweaver.

As with the present edition of ColdFusion, you can download a single-user copy from the Macromedia Web site. Please make it a point to follow this month's ColdFusion articles in this journal, and in the months to come: I strongly feel that it is going to become an even more integral part of an MX designer's life.

While it isn't new, Flash Video is also going to be a factor in our development careers. Beginning in this issue, you will see more tips and techniques for Flash Video, as well as a great Q & A with Macromedia's Chris Hock.

As developers, we cannot look at these changes as individual events. Instead, we must look at a much larger picture. We can't any longer look at each of the MX products as an entity

unto itself, but as evolving to a unified whole.

Let me paint just one possible scenario. Let's assume you want to develop an e-learning program. You would certainly start with Captivate. From there, you would export into Flash where you would add animations, and so on. Of course, you might want to introduce your topics, or explain concepts, using a combination of Flash Videos and various text effects (there are several examples of this on the Macromedia site). You might want to use Breeze as a delivery platform. It may also be necessary to have the e-learning simulation interact with data – which may occur through ColdFusion.

Get the picture? We must think in terms of the whole.

Last month, I alluded to changes in the website for this journal. In this editorial, I can give you a few more details.

It is my hope, by late Spring or early Summer, to have many of our articles interactive in our Web site. As an example, rather than just reading about Flash Video by Tom Green, we will have a live example on the Web site. Further, using Captivate, we will set up practice simulations to help you get some practical experience. In addition, we will be employing Flash Video to bring you conferences and discussions.

It's my goal to make the Web site a laboratory to implement MX products to their maximum. I am encouraging our many fine authors and editors to push the envelope even more than they have already. I want this to be more than just words on a page, I want it to be a living and organic experience for you, the reader.

It's my goal to help you walk away with as many new ideas each month as possible.

A time for change? In our profession isn't it always a time for change? ☺



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.



Flash Video from the Inside

Exclusive Q&A with Chris Hock, director of product marketing, Macromedia
interview by jeremy geelan

MXDJ: Chris thanks so much for agreeing to talk exclusively to *MX Developer's Journal* about Flash video. First off, how would you situate Flash video in the overall spectrum of Macromedia's steadily widening roster of offerings for developers and consumers alike? Is this "The Big One" or is it just another rich-application goodie among your host of offerings?

Chris Hock: I'd say that for Macromedia, "The Big One" is our overall strategy of delivering products that enable our customers to deliver fantastic digital experiences across platforms and devices. Whether you're using Flex to power the presentation-tier for your Web app, Breeze to deliver enterprise training and communications, or the Flash video platform to deliver compelling, interactive video experiences, they all enable developers and content creators to create the best possible experiences that work cross platform.

This is of particular importance for video just because it has traditionally been handled very poorly on the Web. I think everyone is familiar with clicking on a video link only to be taken to a Web page asking you if you want "56" or "220" and asking if you want this format or that format. These are the types of poor user experiences we think can be improved

with Flash video.

MXDJ: How about the different delivery options for Flash video – what are the options?

CH: There are really two delivery options that make sense when delivering video via the Flash Player: progressive download and streaming.

Both progressive download and streaming enable the Flash video file (FLV file) to be kept external to the SWF file. The FLV contains the video playback controls and UI thereby shortening publishing time, enables content to be changed without having to republish, and avoids any possible length of clip or audio-video synch issues.

Progressive download requires Flash Player 7 or higher and is served off any Web server. Streaming works with Flash Player 6 or higher and is served from the Flash Communication Server. The Video Developer Center on Macromedia.com has a great whitepaper (http://www.macromedia.com/devnet/mx/flash/articles/flv_download.html) on these delivery methods for people who want more info.

MXDJ: Which of those options are you expecting to have most traction?

CH: The traction varies depending on the

use of video. Streaming is good option for Web sites interested in using video delivery features such as bandwidth detection, quality of service monitoring, reporting, advanced seek and navigation, live video, video chat and messaging, as well as Web sites that want back-end efficiencies.

These types of companies typically serve lots of video. Therefore, streaming probably has more traction in terms of total amount of video delivered.

Progressive download, on the other hand, is a relatively simple method of delivering video and therefore it probably has more traction in terms of number of sites deployed on.

MXDJ: Is that traction evident in certain verticals in particular, or right across the board?

CH: Because Flash video provides publishers with complete creative control over the look and feel of the playback experience, verticals in which branding is important are moving towards Flash video. We're seeing a lot of traction in media and entertainment, advertising and agency work, retail and consumer product marketing.

MXDJ: Is there any way, or will there be in the future, of sending a 'deep link' that can take a friend or colleague to

“Verticals in which branding is important are moving towards Flash video. We're seeing a lot of traction in media and entertainment, advertising and agency work, retail and consumer product marketing”

the EXACT place on a Flash Video presentation that you want to share – or bring customers to, in a business context? Say, to the amazing “Director’s Cut” at <http://www.redbullcopilot.com/> but bypassing other parts of the overall presentation.

CH: Sure, this can be done today when you stream video with Flash Communication Server. Streaming doesn’t require the video before the start point to be downloaded before it gets played. Couple this with the ‘zero buffer, instant start’ feature of the Communication Server and you can make applications such as video blogging, video searching, video messaging, video presentations which have multiple camera angles, and similar in which you can send a person directly to a specific spot of the video and play it instantly.

MXDJ: We’ve talked about streaming video, but for those who don’t have Flash Communication Server, let’s go back and consider again both progressive download and embedding video within SWF files. Just how pervasive right now is Flash Player 7? In other words, at what point does embedding using Flash Player 6 just not make sense – what’s the tipping point do you think that developers should be keeping an eye out for?

CH: Embedding video in a SWF file is not the best way to deliver video via Flash Player for a number of reasons that I outlined above. Further, with the penetration of Flash Player 7 in the lower 80 percent range as of September 2004, our last date of measure, many customers are now satisfied delivering Flash Player 7 content. For those customers that need to reach the upper 90 percent range though – for example, companies delivering Flash video advertising – streaming is still a great option to enable you to reach those using Flash Player 6.

We publish the Flash Player version penetration statistics on our Web site (http://www.macromedia.com/software/player_census/flashplayer/version_penetration.html).

MXDJ: Who are the geniuses behind the FLV format?

CH: The FLV file format was developed in conjunction with the development of the Flash Communication Server. Jonathan

Gay, the original creator of Macromedia Flash, spearheaded both projects along with a lot of talented people on the Communication Server, Flash authoring and Flash Player development teams.

MXDJ: How large is the overall team that’s brought Flash video to be the reality it is today?

CH: It’s pretty tough to answer that question since Flash video is becoming ingrained in many products at Macromedia. It certainly started with the Flash Communication Server, Flash authoring and Flash Player teams back in the MX timeframe, but you have to also include the Dreamweaver team who contributed the Macromedia Video Kit and the Breeze team who continue to enable Flash video capabilities in the Breeze products.

In addition, while not officially on the Macromedia Flash development team, there have been a number of partners who played very important roles in bringing Flash video to the point it’s at today. VitalStream, our original Flash Video Streaming Service partner, really started the ball rolling by enabling large-scale Flash video deployments. Sorenson continues to help us on a number of encoding tool-related projects. The New Toronto Group were out there very early with custom development work and training on all things Flash Communication Server and Flash video related. Last but not least, the talented developers out there such as Creative Bubble, Fantasy Interactive, IQTV, Odopod, Productorials, Vimation, and others who are actually out there creating fantastic Flash video experiences deserve mention. These guys are the real team!

MXDJ: The Red Bull Copilot example we talked about before even contains client logic, so that users can decide for themselves what camera view they’d prefer in order to experience the demo flight. Is that, would you say, the “killer app” here, the ability not just to have video on Web sites, but to have interactive experiences from the get-go?

CH: I think interactive video experiences are very important when it comes to video on the Web and devices. Unlike TV, video on the Web is often much more of a lean-forward experience. Users want

to be able to interact with the content to get the information they’re looking for quickly and efficiently.

This doesn’t necessarily mean that all video has to have the complexity of multiple camera angles such as the Red Bull example. Interactivity could be as simple and straight forward as navigational elements that enable a viewer to jump to a particular point of a long video. It could be clickable items in the video that display related or associated information. Amazon.com used this method very effectively recently while highlighting five short films on their Web site. If viewers liked the blouse that Minnie Driver wore in the film short, they could click on a credit and be taken to the page on Amazon.com where that blouse was available for sale.

It should also be noted that interactivity alone is not enough for great video experiences. Enabling publishers to have full creative control over their video experience so it’s seamlessly integrated into the look and feel of their site is a very important element. Last, playback via a ubiquitous client is extremely important as well. After all, what good is it if people can’t see your great creative, interactive video?

MXDJ: If so, then what are the considerations from a developer’s point of view? How difficult is it to do that kind of client logic?

CH: Developers who are familiar with

Profile: Chris Hock



Chris Hock is director of product marketing at Macromedia. In this role, he’s in charge

of Flash video business and led the creation and launch of various video related products at Macromedia including the Flash Communication Server and the Flash Video Streaming Service. chock@macromedia.com



ActionScript and with programming Flash should have a relatively straightforward time programming interactive video applications. Rather than controlling a piece of text data however you're controlling an audio or video stream.

MXDJ: Can video playlists be controlled externally – I mean, can Web site owners update their video content dynamically?

CH: Absolutely. XML is a great way to do this. We've recently published a tutorial on the Macromedia Developer Center (http://www.macromedia.com/devnet/mx/flash/articles/video_player.html/) that walks people through how to create a Flash video player that reads an XML file containing a list of available "clips"

CH: The ability to skin, and reskin, your video player is an area where Flash video has a very strong advantage over other formats. It's the only format where the look and feel of the media player can be changed as easily as the content. Consider how useful this can be: large recording companies can develop a Flash app for streaming MP3 files and then quickly reskin this as needed for all of their associated recording labels, artists, and partners. A Flash developer can create a basic video player out of Flash and then quickly skin that player for various clients and projects as the need arises.

MXDJ: Is it possible to create an on-demand playlist, so that, say, an ad could be inserted before each clip plays?

Video Streaming Service.

MXDJ: What of the future? What does Macromedia see as being 'The Final Destination' for Flash Video? Are you seeing anything that's blowing you away simply because no one on the team ever thought of it being used like that?

CH: A rather simplistic, but significantly better vision of the future is when we've reached a point where you click "play" on a Web video and the thing just plays. No bandwidth questions, no buffering, no installing stuff. Just play the video.

To answer the ingenuity part of your question, I have seen some pretty cool things done with Flash video. The set of interactive applications that Kevin Towes and the New Toronto Group created for "ZeD UnCut" - a live, interactive national TV broadcast and Web simulcast that was aired across throughout Canada on the CBC network - completely blew me away. They were nominated for a 2004 Emmy Award by the National Academy of Television Arts & Sciences so I guess I wasn't the only one who was impressed.

I thought the Subservient Chicken site was pretty clever as was Ohayo Players, a game developed by Japanese Web development company Bascule that used motion detection in Web cams to move players.

In regards to the future of the Flash platform, we're hard at work on enhancing the expressiveness of the experience, making better performance and expanding its already powerful capabilities as a development platform. All three of these areas will directly involve Flash video.

MXDJ: Last, do you think that Flash video articles, such as MXDJ is aiming to publish in the near future, is going to be a growth area too?

CH: Take a look at all the topics that we've covered in this discussion here and I think it's pretty clear that we've just started to scratch the surface of what you can do with Flash video. I anticipate seeing many more articles on Flash video in MXDJ the future.

MXDJ: Sounds good to us! Thanks Chris and best of luck for an amazing 2005.

CH: Thanks, my pleasure speaking with you. ☺

"I think interactive video experiences are very important when it comes to video on the Web and devices"

and a playlist for each "clip." This relatively simple setup gives you an amazing amount of flexibility to deliver video content without ever needing to go back into Flash authoring and republishing your SWF file.

Just by changing the XML file, you can alter which clips are available to the user when they connect. You can have the clip items point to various sections (e.g., chapters) of one long video, or point to an individual video clip, or point to a playlist of multiple clips. All of this just by changing the contents of the XML file.

If you use a digital media management system such as WorldNow or the Platform, the creation of this XML file can be generated automatically creating a completely automated Flash video delivery system.

MXDJ: How about reuse and reskinning?

CH: Yes. This could be accomplished via an XML playlist file such as I described above. One of the items in your playlist could be an ad. You can place that ad in front or, behind, or even right smack dab in the middle of your content just by tweaking a couple of lines of your XML file.

This question was specifically about on-demand playlists, but it's also worth mentioning that everything that we're talking about here can also be done with server-side streams as well. In other words, you can deliver live video streams or simulated live video streams (e.g., like TV - streamed live to everyone at the same time, but using recorded video) as well. You can have your own Web TV channel, complete with a mixture of live and recorded programs and inserted with ads, station IDs, and public service announcements at set intervals, all streamed from Flash Communication Server or the Flash

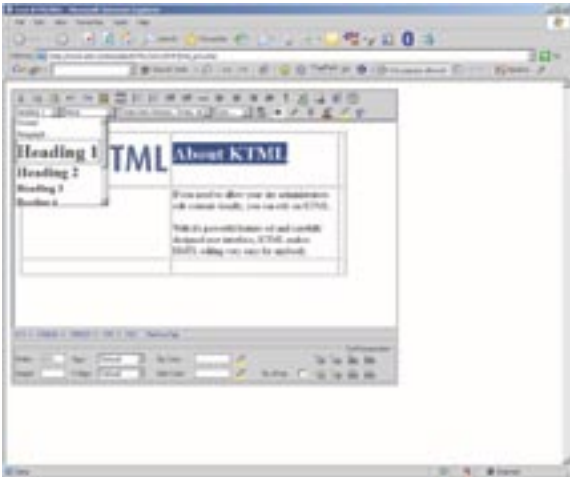
Web (r)evolution:



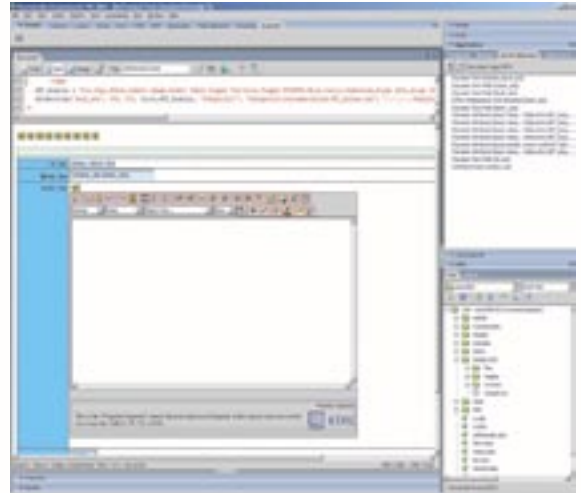
KTML

Let your clients edit their Web sites content through the browser

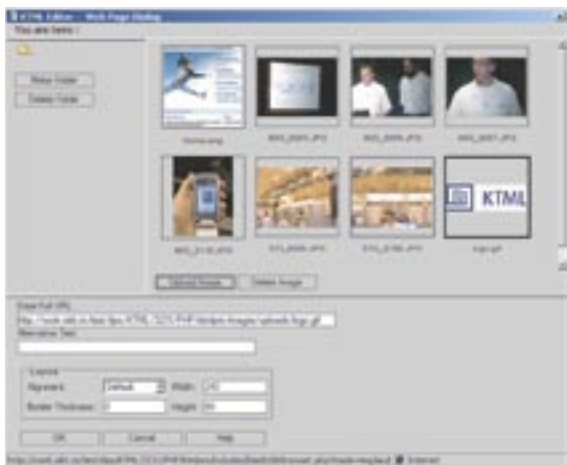
Visually use CSS styles (Word like)



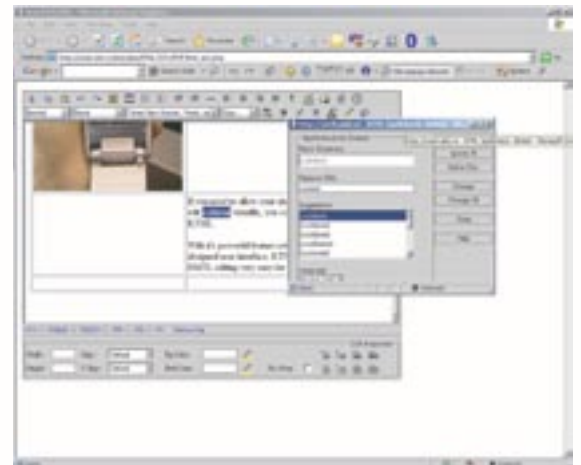
Native Dreamweaver MX integration



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/>



FLASH VIDEO FOR THE MASSES

When I first published my observation that “QuickTime is dead” in this publication and at Community MX, I never expected the response that article would garner. It ranged from, “Dude, you are sooo wrong!” to “Finally, we are free.” Earlier this year, Macromedia, as is so typical of the company, quietly dropped the “Macromedia Video Kit” on the developer community and suddenly, video was available to practically everybody who designed Web sites. by tom green





Since the introduction of the Video Kit, I have had the rather interesting experience of demonstrating it at a number of major events such as Digital Design World, Web Design World, MXimize, and to my local Dreamweaver user group. Presenting in these forums is a unique experience because these are the very users to whom the product is targeted and their reaction has been a unanimous, "Come on Tom, it can't be that simple." Not only is it that simple, it is code free. The implication is quite powerful: Macromedia has put the power of Flash video in the hands of everybody who designs sites using Dreamweaver and has a "Not for me, dude" attitude toward Flash MX and Flash MX Professional. In fact, my first reaction to seeing it and using it as one of the "Flash Video Voices in The Wilderness" was, "Great, they just put me out of a job." In this article I will walk you through the creation and addition of a Flash video to a Dreamweaver page using the "Kit" which is a steal at \$99.00 and is available in Mac and PC versions through the Macromedia store.

Once you download and unzip the

"Kit" you will discover it is composed of two major elements: A Sorenson Squeeze 4 Lite installer and a Dreamweaver Extension. Install the extension and Sorenson Squeeze. With those two tasks out of the way you are ready to get into the "Flash Video Game."

"Squeezing" Video

The version of Squeeze you will be using is the "Lite" version of the application. The full version, available for \$449 through Sorenson Media, is an invaluable tool if you are doing a lot of video. The difference between the "Lite" version and the "Professional" version is an expanded number of output options. For example, video output in Lite is limited to .swf, FLV, .mov and MP3. The pro version adds output to MPEG4, MPEG 1/2, Realmedia, and Windows media.

To "squeeze" a video, follow these steps:

1. Open Squeeze Lite. The interface that opens is really simple to follow. In the upper left corner you can import a file, a bunch of files, or even connect a camera to your computer and capture that. The filters allow you to

"play" with video and the "Format and Compression" settings allow you to choose a preset for the output. On the right side of the screen is where the work is done. The preview is at the top and the compression is done below it (see Figure 1).

2. Click the "Import File" button and navigate to the location of the video to be used in the Web page. When you click "Open" the file will appear in the Preview area of the interface. There is another way of importing a file. If you drag the video file from the desktop or a folder onto the interface, the file will import and appear in the Preview area.
3. Open the "Macromedia Flash Video (.FLV)" section. A number of presets will appear. Select the 56K_Dial_Up_Stream and click the "Apply" button. The stream will appear in the Compression area – Sorenson calls this area a "Batch Tree" – under the preview. You can also drag the preset across to the Compression area. When you release the mouse it will drop into place. The file will be named using the preset. You can change this later.
4. Click the arrow beside the Preset to

figure 1



open the video and audio presets. Double click either the MP3 or the SparkPro item to open the Audio/Video Compression Settings dialog box (see Figure 2).

- Click and hold the "Method" pop down and select "Sorenson 2-pass VBR." This compression option, in very simple terms, compresses the video twice resulting in a smaller file size.
- If you want to match the video playback to the Flash default playback of 12 frames per second, click and hold on the "Frame Rate" pop down and select 12.
- Click OK. The dialog box closes and the changes will appear in the compression area.

figure 2

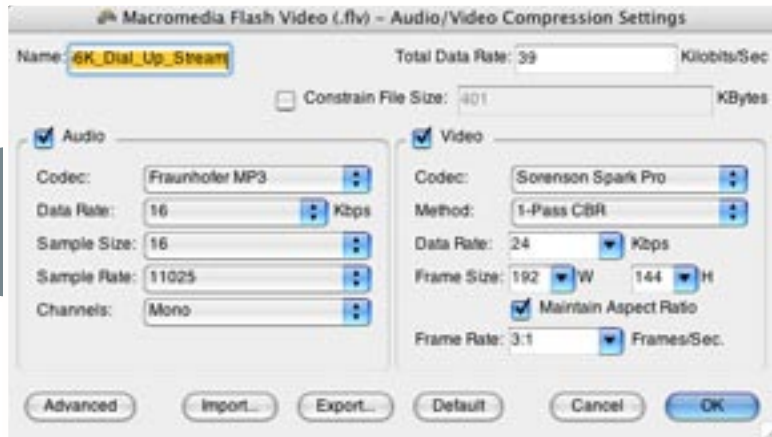
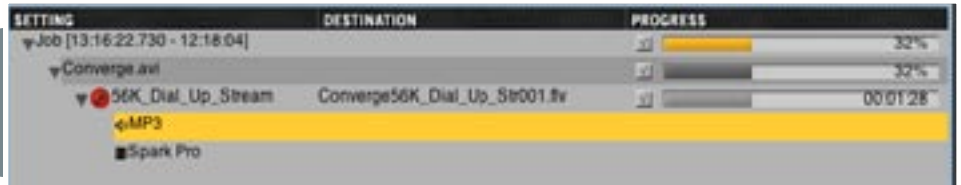


figure 3



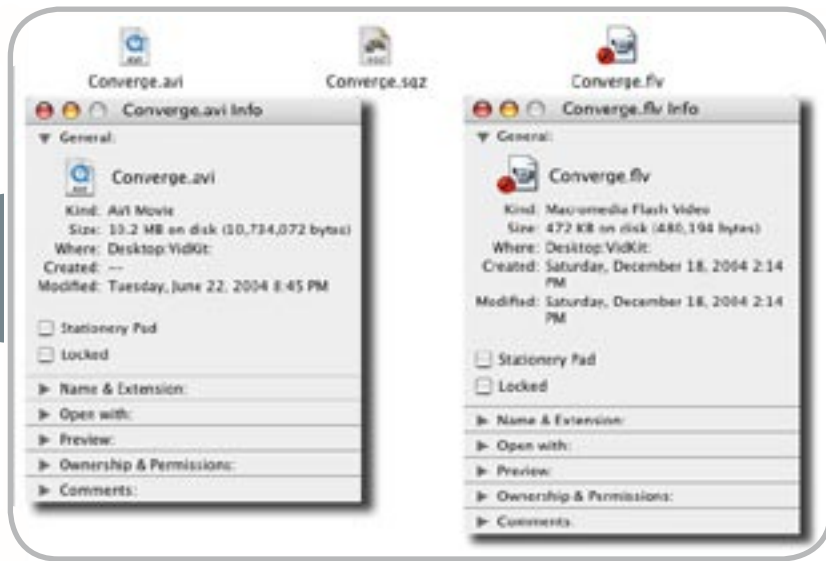
You can also double click a preset in the "Format and Compression Settings" and change its values. This is a very dangerous action because the change will apply globally and will change the application preset. Knowing that you can change the setting for the movie means that you should never need to change the preset values. Also keep in mind that if you do change the preset, this action cannot be undone. Consider yourself warned.

- Click the "Squeeze It" button. The video compression starts as indicated by the progress bars in the compression area (see Figure 3). If you feel you have made a mistake, click the "Stop It" button to halt the compression and return to your settings.

figure 4



figure 5



When the compression is completed you will be notified. One feature you will notice is the addition of a "Play" button beside the file name in the Compression area. Don't bother clicking it. If you do, you will launch Flash and then be told the file can't be opened. FLV files are imported into Flash and can only be played back from the Flash timeline through a Flash .swf file. This is why there was an error message.

- Quit Squeeze. When you quit, Squeeze asks you to save the file. This is a really neat new feature in the application. You aren't saving the .FLV, you are saving an .sqz file.

figure 6

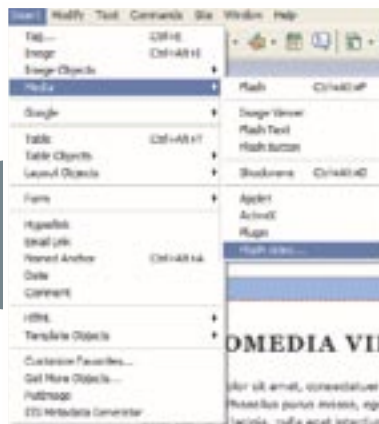


figure 7

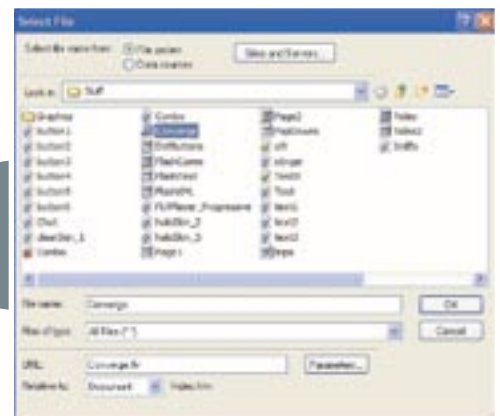


figure 8



figure 9

MACROMEDIA VIDEO KIT

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum vitae nulla a lectus cursus mattis. Phasellus purus massa, egetas sed, fringilla a, facilisis in, massa. Donec sit amet est. Nunc lacina, nulla eget interdum luctus, tunc est molestie dolor, et consectetur quam ipsum id nunc. Nunc eu lectus id erat sodales volutpat. Mauris venus enim a enim. Fusce tincidunt. Etiam laetitia mi vitae pede. Suspendisse feugiat, ipsum nec pretium malesuada, urna du rhonus laetia, eu tincidunt lectus tortor vel ante. Ut vel erat. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Pellentesque a urna. Sed quis nibh id pede venus dignissim. Praesent commodo eros eget ligula. Fusce vitae enim nec ante varius dignissim. Phasellus accumsan sapien non eu. In vitae metus ornare odio lobortis vulputate. Mauris volutpatin ultrices nibh. Aliquam vehicula turpis id fringilla lobortis. Vivamus accumsan ante id lectus. Nunc tunc vitae nibh. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Maximas libero du, portitor sed, partitor id, suscipit sit, ornis. Nunc cursus, tortor commodo tincidunt feugiat.



Flash Video in Dreamweaver rocks.

figure 10

MACROMEDIA VIDEO KIT

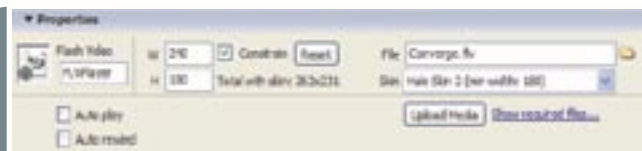
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum vitae nulla a lectus cursus mattis. Phasellus purus massa, egetas sed, fringilla a, facilisis in, massa. Donec sit amet est. Nunc lacina, nulla eget interdum luctus, tunc est molestie dolor, et consectetur quam ipsum id nunc. Nunc eu lectus id erat sodales volutpat. Mauris venus enim a enim. Fusce tincidunt. Etiam laetitia mi vitae pede. Suspendisse feugiat, ipsum nec pretium malesuada, urna du rhonus laetia, eu tincidunt lectus tortor vel ante. Ut vel erat. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Pellentesque a urna. Sed quis nibh id pede venus dignissim. Praesent commodo eros eget ligula. Fusce vitae enim nec ante varius dignissim. Phasellus accumsan sapien non eu. In vitae metus ornare odio lobortis vulputate. Mauris volutpatin ultrices nibh. Aliquam vehicula turpis id fringilla lobortis. Vivamus accumsan ante id lectus. Nunc tunc vitae nibh. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Maximas libero du, portitor sed, partitor id, suscipit sit, ornis. Nunc cursus, tortor commodo tincidunt feugiat.



Flash Video in Dreamweaver rocks.

figure 11



This file is a "life saver." If you double click it, Squeeze will open and display the Compression settings used to create the .FLV. The beauty of this feature is the ability to make changes without having to remember or note the settings used to create the file (see Figure 4). If the client complains

about the "quality" of the FLV you can simply open the .sqz and go back to work.

If you now compare the resulting file sizes, you will most likely have the same reaction as many people when I demo this process. The original movie that I

compressed weighed in at 10.2MB. And the FLV comes in at 472K. The reaction? For that massive a reduction there simply has to be a corresponding reduction in quality. Not really.

The AVI, for example, contained a sound track that output to 44Khz, 16-bit stereo. The compressor knocked that back to 11Khz mono which, in itself, has a profound impact on file size (see Figure 5). The physical size of the file changed from 320 X 240 to 192 X 144, which also has a huge impact on the final file size. A low data rate and the 2 Pass VBR also has an impact on file size. Again, play with the settings. For example, I have found a 300 kbps data rate with mono sound, 2 Pass VBR compression, a 320 X 240 frame size, and 12 frames per second output works very well for my purposes.

Adding the Video to a Dreamweaver page

Once the FLV is created, adding it to a Dreamweaver page is a painless process. You simply:

- Indicate where the video is to go
- Import the video into the page.
- Determine the playback controls.
- Set the physical size of the video.
- Upload and play.

Here's how:

1. Open the Dreamweaver document where the file is to be placed and click once on the page or in the <div> where the video is to appear.
2. Select Insert>Media>Flash Video... to open the Insert Flash Video dialog box (see Figure 6). This is the extension that was installed into Dreamweaver MX 2004.

The first thing you will notice about the dialog box is that it has been divided into two sections. At the top you choose the type of video you will be using. Your choices are Progressive Download, Streaming, and Flash Video Streaming Service Lite. If the FLV is to sit on the web server, choose Progressive Download. If you are using a Flash Communication Server, select Streaming; if you have taken advantage of the third-party Streaming services offered by Macromedia, such as VitalStream, you would select the third option. Just

keep in mind, selecting the streaming option will require that you to have an "rtmp server address" used by the Flash Communication Server and that selecting the third option will result in a dialog box asking you to log into your Streaming Service account and for you to paste the Embed Code for the account into the dialog box.

If you have access to neither the Flash Communication Server nor VitalStream, you will be selecting Progressive Download. This option uploads the FLV file to your Web server and the resulting .swf file will essentially make a "call" to the server that will "feed" the FLV into the .swf.

Having made that determination, the next step is to tell Dreamweaver where the video is located, what player to use, and the physical dimensions of the video.

1. Click the Browse button and navigate to the folder containing the video. Select the video and whether it will use a relative or absolute address (see Figure 7). Click OK.

2. Select a "Skin" from the pop-down menu. You have six choices that will be shown to you when you make a selec-

tion. Pay attention to the minimum widths.

3. Click the "Detect Size" button. The dimensions of the FLV file will appear in the height and width areas of the dialog box. These dimensions are important because they contain the metadata necessary to play the video in the Flash Player 7.

The next two choices- Auto play and Auto rewind - are optional. I prefer to let the user control when to play the video, which means I always leave Auto play deselected (see Figure 8).

4. Click OK and the video will appear as a gray box with an FLV icon on the page (see Figure 9).

If you use CSS-p to drive your page design, you are in for a very pleasant surprise. Adding a video to a <div> allows you to actually run text around the video as shown in Figure 9 or place it anywhere on the page. This means you now control the "Web turf" and the video plays exactly where you want it to play, not where Apple, Microsoft, or Real decide the video should appear. In fact, video can now be

treated almost like an image with a caption.

5. Save the file and press the F12 key to preview the file in your browser (see Figure 10).

If you need to change any of the settings select the file on the Dreamweaver page. The Property inspector will change to reflect your selection (see Figure 11). If you click the Upload Media button, the FLV and the files required for the skin will be uploaded to your Web server.

Conclusion:

As you have seen from this article, a \$99 investment gets you easily and painlessly into the Web video game. Squeeze 4 is simple to use and the addition of a .sqz file allows you to rapidly make changes without having to note the compression settings.

In Dreamweaver, the addition of video to your Web page has been simplified to a menu item and a dialog box. Hopefully, if we bump into each other when I am "demoing" the macromedia Video Kit you won't have to ask, "Come on, Tom, it can't be that simple?" It is. ☺

Teacher, author, chief cook and bottle washer. Instructor at Humber College's School of Media Studies in Toronto, Tom is also the author of Building Web Sites with Macromedia Studio MX and Building Dynamic Web Sites with Macromedia Studio MX 2004. Both are published by New Riders. tgreen17@cogeco.ca

Learn FreeHand!

Now you can ease any learning curve with nearly 200 movies and hands-on tutorials created by an artist who has used the program since it was introduced. Every menu item, feature, and tool is explained, and you're given a reason to use each of them.

...I learned that I've been working way too hard to accomplish various tasks since I was not aware of many of the time- and effort-saving tools.

The course looks great!

It has been very helpful.

I received the Casual FreeHand MX Course and love it

I am very impressed with your product.

Gain control over vector artwork, create layouts for brochures, booklets, instruction manuals, Web sites, Flash movies — and do it all at your own pace.

45

Training from the trenches!

www.brainstormer.org

Includes FREE FedEx domestic shipping and Global Priority Mail International.



Flash, Web Services, and Data Binding

Part 1 – Be prepared to bind data like there's no tomorrow!
by darron schall

This is not Web Services 101. This is not an introduction to Flash. This is not art. This is not a love song. Now that that's out the way, be prepared to bind data like there's no tomorrow.

In this article I'm going to show you three different ways to consume a Web service and some of the pros and cons of each approach. When you're done reading you should have an understanding of what data binding is, how to use it in your applications, and how to achieve similar results without using it.

Unfortunately for some, Flash Professional is required to follow along with the examples. If you haven't upgraded Flash to the Professional version yet, taking advantage of the Web service integration and data binding might just be a good enough reason to. Don't forget about the trial version available as well..

If you're ready, dust of your computer and let's go.

What Are Web Services?

A Web service is, more or less, a remote procedure call wrapped in an

XML package and typically transmitted over HTTP. You make a request to the URI of the Web service, passing in any appropriate parameters, and are greeted with an XML packet containing the result of the function call. The standard XML format for Web service messages is SOAP – Simple Object Access Protocol.

Web services can be inspected by requesting the WSDL – Web Services Description Language – of the service. The WSDL explains how to interact with the service, what methods are available, what their parameters are, etc. The WSDL format isn't the easiest to read. Thankfully, Flash will help us out with WSDL interpretation, which we'll see a little later.

There are many advantages to using Web services which can be found in the further reading links following this article. We still have a long way to go, so I'll just assume that you've decided to leverage a Web service or two in your next Flash project and now you're itching to know how.

What Is Data Binding?

Now that we have an idea what Web

services are, data binding is next on the explanation hit-list. Data binding is... well, magic. Through data binding you can "glue" two separate pieces of a program together. Whenever one piece changes the other piece will automatically notice that change and update itself without any coding necessary. As an added bonus, data binding can be a one- or two-way street.

Data binding is remarkably simple to use and can be done without writing a single line of code. Because of this, it's a blessing and a curse at the same time. It eliminates the need for you to program what could be complex interactions, but the "magical" aspects of data binding can make it a maintenance nightmare.

In the coming examples, I'll show you how to use it, how to avoid using it, and finally make some recommendations for when data binding is appropriate.

The Morse Code Web Service

For the three examples that we'll be going through, I've found a Web service at xmethods.com to use. The Web service, located at www.regomnet.de/morse.aspx, allows us to easily convert a string to Morse code and back again. We'll leverage this service to create a simple client that can encode and decode Morse code messages.

Let's get started - fire up the Flash IDE and meet me back here when you're ready.

The Web Services Panel

Understanding the WSDL of a Web service can be difficult. The Web service description for our Morse code Web service can be found at web221.area-18.server-home.net/Morse.asmx?WSDL – what a mess! Although the WSDL contains all of the methods available, it can be tricky finding them, let alone figuring out what parameters you need to pass. Fortunately the folks at Macromedia have

figure 1



greatly simplified the process.

In the Flash IDE, we're going to open up the Web services panel. You can find it Window -> Development Panels -> Web services. Once opened, we can load the WSDL of a Web service into it and Flash will parse it for us.

Click the globe icon in the top left corner of the panel to bring up the "Define Web Services" dialog. Click the plus icon to add a Web service, and type in the location of the WSDL for the Morse code Web service - <http://web221.area-18.server-home.net/Morse.asmx?WSDL> - as previously mentioned. Click OK, wait for the panel to complete processing, and then you'll see the newly added Web service.

Expand the service to see the methods available, and expand the methods to see their parameters and return types. Your panel should look like figure 1.

Want to see something cool? Make sure you have a blank document open, right click (control click on a Mac) on the MsgtoMorse method and select "Add Method Call." This creates a WebServiceConnector on the stage all set up for the operation. All we need to do now is give it an instance name and invoke it. Welcome to example 1!

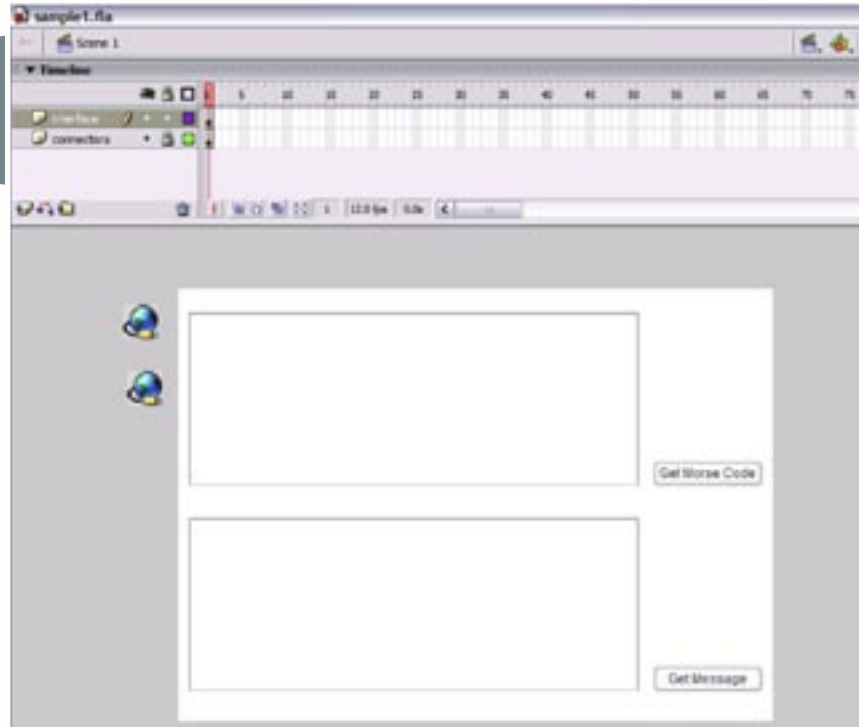
Select the WebServiceConnector that was just placed on the stage for us by adding a method call from the Web Services panel, and give it an instance name of "msg2morse_wsc." Add another method call for MorsetoMsg and give the newly placed WebServiceConnector component an instance name of "morse2msg_wsc." Drag both instances to the top left, outside of the stage area, and then change "Layer 1" to be "connectors."

Now that the connectors are set up, we'll create the interface. Lock the connectors layer so we don't accidentally place anything on it, and create a new "interface" layer.

From the components panel, drag a TextArea component to the stage, name it "msg_ta" and position and size it towards the top half of the stage. Drag another TextArea to the stage, name it "morse_ta" and position and size it on the bottom half of the stage.

Drag a Button component to the stage, name it "msg2morse_btn" and

figure 2



position and size it next to the msg_ta component. Change the "label" property to "Get Morse Code." Repeat the process for a "morse2msg_btn," using the label "Get Message."

With everything set up, your movie should look like figure 2.

Now that the interface is set up, we'll hook everything together via data binding. To access the data binding feature of Flash MX Professional 2004 we need to open the "Component Inspector" panel from the "Window" -> "Development Panels" menu item, and select the "Binding" tab. It looks like figure 3.

All of the magic happens through this panel. To integrate binding with our interface, unlock the "connectors"

layer and click on the msg2morse_wsc.

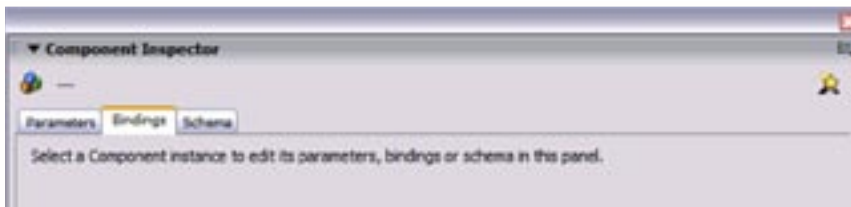
Remember, binding allows us to glue two pieces of data together. The Msg2Morse Web Service method needs to take in a string message, and it will return the Morse code result as a string. First we'll bind the message parameter to the text area - click on the plus icon in the top left of the panel to add data binding. Select "msg : string" under "params" in the Add Binding dialog, as pictured in figure 4 and 5.

After selecting what we want to bind, we need to pick what we want it to be bound to. In the Component Inspector panel, double click the "bound to" value. We're binding the message parameter, so we'll select the "msg_ta" from the list

"Data binding is remarkably simple to use and can be done without writing a single line of code"



figure 3



on the left, and then the “text” property from the list on the right. This will allow the user to type in the message that will be sent to the Web service for conversion.

After pressing the “OK” button, the binding is in place. The process is the same for the other components. With the message parameter bound, now we need to bind the “result” of the “msg2morse_wsc” to the “text” property “morse_ta.” Repeat the process above to add the binding.

Repeat the process again to bind the

“Morse” parameter of “morse2msg_wsc” to the “text” property of the “morse_ta” component. One last time, repeat the process to bind the “result” of the “morse2msg_wsc” instance to the “text” property of the “msg_ta” component.

The only thing left to do is wire the buttons. Add an “actions” layer at the top, and put the following code on the key-frame:

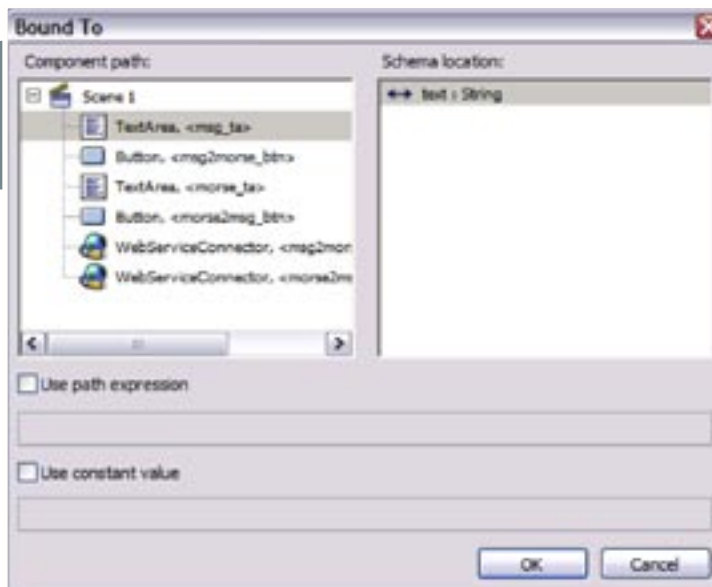
```
// onRelease fires when the user presses and release the button
msg2morse_btn.onRelease = function() {
    // trigger invokes the Web Service
    msg2morse_wsc.trigger();
}
morse2msg_btn.onRelease = function() {
    morse2msg_wsc.trigger();
}
```

Test the movie (CTRL+Enter on Windows) to see your creation. Type a message into the text area at the top, press the “Get Morse Code” button, and watch as the text area at the bottom is populated when the result is received. Enter “... / --- / ...”, minus the quotes, in the text area at the bottom then press the “Get Message” button next to it. Watch

figure 4



figure 5



the text at the top update when the results come back from the Web service call.

As you can see, data binding is a powerful concept. With a very minimal amount of effort we’ve created an application that relies on a Web service to handle business logic and uses data binding to glue the interface together. Hold your applause until the end, please.

While the Component Inspector panel makes data binding a cinch to use, there is a dark side to this approach. Looking at your creation there is no visual indication to show components with active data binding. If you want to look for binding, you need to click on every component and check out the Binding properties to see if any exists. If you didn’t know that data binding was used, maintaining the interface is almost impossible because of the “it’s magic” effect that the Binding panel has.

This reminds me of the un-maintainable nightmare of “jungle code” reminiscent of the Flash 5 days where code was placed “on” movie clips and took an expedition to find what you were looking for. The cost of this simplicity is a headache waiting to happen in the future. Believe me, I’ve been there.

Luckily for us there is a way to consolidate all of the data binding configuration in one location. We can leverage the power of binding, but greatly increase maintainability by making the bindings easy to find and modify. It turns out that we can code bindings “by hand” through ActionScript and place them in one central location.

In Part Two we’ll look at two other ways of building this same application, each with their pros and cons. Each example can be the “right” way to code something, depending on the application requirements and the person doing the coding. ☁

Darron Schall is an application developer interested in all things programming, from ActionScript to XML and everything in between. He has a BS in Computer Science from Lehigh University, and maintains a Flash-related weblog at www.darronschall.com. darron@darronschall.com

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

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

Surround SCM lets you...

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

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

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

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

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

Remotely access your source code repository from Dreamweaver MX.

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

Features:

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

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

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

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

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

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

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

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





The Maelstrom in Your Future

The next version of the Flash client is code-named "Maelstrom"
by waleed anbar

Over the past year, we have rededicated ourselves to getting broad community feedback on our future development plans and efforts. Some of that is done privately through beta lists, customer advisory boards, and customer visits. Increasingly, some of it will be done publicly. To this end, we have recently begun showing what we are working on for the next version of the Flash client, code-named "Maelstrom." The effort on Maelstrom roughly falls into three main themes: performance, expressiveness, and standardization.

Performance

Flash has evolved beyond its animation roots to become the leading rich client for complex business applications, an expressive medium for content, and a versatile communications endpoint. This evolution has increasingly put new types of performance demands on the Flash client.

To address this situation, a big part of our development focus for Maelstrom is aimed at improving performance. Accelerated graphics and code execution are all areas that show a lot of promise. For example, with enhanced bitmap image support, you can animate graphics as bitmaps instead of rendering them as vectors in each frame. This dramatically increases the number of objects that can be animated smoothly and preserves processing cycles for other tasks.

Expressiveness

Developers have told us that they would like more expressiveness – a richer palette of options to create powerful effects. We've taken that to heart and are working on a number of new features:

- Video: We plan to include an exceptional, new video codec that's comparable to some of the best codecs on the market today. This new codec allows you to deliver exceptional video quality at reasonable broadband rates

and at larger sizes. We also plan to support a real-time transparent alpha channel that will enable live composition of non-rectangular video, allowing for the layering of video with text, graphics, and even other video.

- Text: We plan to deliver a new text rendering solution that produces text as clear and readable as we have seen. It offers sharp, anti-aliased text and will support precision typographic features such as hinting and kerning. Furthermore, Flash text becomes much more readable, regardless of monitor type, resolution, or character set.
- Real-time graphic effects: We're working on a set of killer real-time graphics effects like glows, blurs, drop-shadows, color matrix, and others. You can apply them live to any objects in Flash, including graphics, text, and even video. Because these are applied live, they can react interactively. Because they are dynamic, they reduce author time and file size by eliminating several post-processing steps in Adobe Photoshop or Macromedia Fireworks that are required to achieve these effects today.
- Blend modes and gradients: Professional designers can now create high-quality effects by composing graphics using a variety of blending modes – overlays, lighten and darken, multiply, screen, and more. In addition to more precise control over radial gradients and gradient stops, the possibilities for runtime creative expression with graphic elements are endless.
- Powerful new APIs: Lower-level runtime access to graphic objects provides new options for visually tweaking your content, allowing developers to get high-performance, low-level control over graphics objects. By taking advantage of the performance improvements I mentioned previously, developers can create never-before-

seen graphics effects and visual experiences in real time. This fine-grained access also enables dramatic performance optimizations and control over runtime performance.

Standardization

As you probably already know, ActionScript is based on ECMAScript, the open standard scripting language. As an ECMA member, Macromedia continues to participate in the evolution of ECMAScript through the standards process. In Maelstrom, we plan to evolve ActionScript toward full compliance with the latest ECMAScript version, called Edition 4. This will bring new power to ActionScript.

Those of you who have seen the demos at Macromedia MAX and other conferences will likely agree with me that Maelstrom has many exciting features to offer and is consistent with where developers and designers using Flash need us to take it.

Showing technology while it is still under development is risky, however, so please keep these key thoughts in mind as you think about Maelstrom or any other technology we show before it is shipping:

- When we show you stuff that is still in the lab, that means it's still cooking. There is no guarantee that any of the features we show you or talk about will actually make it into the final shipping product.
- The whole point of showing things early is to get your feedback and have that feedback impact our future direction. We maintain a feature request form where you can send us suggestions and thoughts about features you like and don't like. Even though we typically do not respond directly to individual requests, we do read, catalog, and prioritize your suggestions as part of our development cycle.
- Please avoid the tendency to jump to dramatic conclusions about how the

technology will impact your present efforts. Again, this stuff is still in the lab and could change course by 180 degrees before it ships.

- Just because we are starting to show things, no implication should be drawn on the timing of the shipping products.

Maelstrom will be integrated into all of our products that rely on the Flash client. As has always been the case, existing content – regardless of when it was created or for which client version – will continue to be rendered exactly the same as the author intended.

There are many features in Maelstrom that are completely new and will require new code to be written if you want to take advantage of them in existing projects. We will offer more information on how to take advantage of these new features when we're closer to release. We plan to document extensively all the ways customers can take full advantage of what Maelstrom has to offer. We are committed to making this as easy as possible.

Finally, we have conducted download testing for different-sized clients

“Developers have told us that they would like more expressiveness – a richer palette of options to create powerful effects. We’ve taken that to heart”

and have concluded that the expected footprint of Maelstrom will not materially impact the historical adoption rate of new Flash client versions. We attribute this to many factors, including the rapid adoption of broadband worldwide. We think that Maelstrom will be the most significant release of the Flash client ever. Much of it is credited to all the great feedback and commentary that you have given us. Keep that feedback coming! ☺

Waleed Anbar is product manager, Flash Player and Central, at Macromedia. His life revolved around Macromedia Flash for four years before he finally came to Macromedia to help build the future of Flash and forge innovations in the world of interface design, graphic art, and information visualization. Ironically, he devotes his time indoors to making computers more effective so that future generations might be able to get outside once in a while.
wanbar@macromedia.com



Looking to Stay Ahead of the i-Technology Curve?

Subscribe to these FREE Newsletters >

Get the latest information on the most innovative products, new releases, interviews, industry developments, and i-technology news

Targeted to meet your professional needs, each newsletter is informative, insightful, and to the point. **And best of all – they're FREE!**

Your subscription is just a mouse-click away at www.sys-con.com

IT solutions	NET JOURNAL
JDJ	WebServices
Information STORAGE+ SECURITY JOURNAL	THE LINUX WORLD JOURNAL
LINUX BUSINESS WEEK	wireless
MX developer's journal	XML
wldj	WebSphere JOURNAL
	COLOFUSION

SYSCON MEDIA
The World's Leading i-Technology Publisher



Taking Action

Learning ActionScript version 2.0, part 2
by charles brown

In Part 1 I introduced you to very basic concepts of ActionScript 2.0.

We looked at some simple programs such as setting a variable, basic object-oriented concepts, and some basic interaction with the Flash MX 2004 Interface.

It is important to understand one fundamental concept: for a computer program to be a computer program, it must have three basic capabilities:

1. It must be able to perform a sequence of instructions.
2. It must be able to have a decision structure.
3. It must have a repeating structure.

The first one we certainly saw in Part 1 (Vol. 2, issue 12) of this series. Unless told to do otherwise, ActionScript 2 (as well as any program) will execute a line of code. Then, when it is finished executing that line of code, it will go on to the next one. Here, in Part 2, we will look at the second capability of making decisions. Before we begin, we must understand some programming basics.

Pseudocode

When I conduct my classes, the most common comment I get from beginning programmers is, "I have learned all of these great tools. Now, how do I use them for projects I work on?"

The key is in one word: Pseudocode!

All any computer does, even the most sophisticated, is carry out a predefined set of instructions. As sophisticated as they may seem, computers cannot vary from this set of instructions.

To understand pseudocode, let's try a little experiment. Let's pretend you don't have a computer; instead, you have to write the instructions out and tell someone how to do something by hand. As a very simple example, say you're teaching someone how to acknowledge a customer's payment of a bill. The pseudocode may go something like this: if the customer paid his bill in its entirety, send him a letter that says "your balance is 0." If the customer still has a balance, send him a letter telling him he still has a balance and write in what that balance is. Finally, if the customer overpays his bill,

send him a letter showing that he has a credit and write in what that credit is.

Believe it or not, this is the first step to writing a computer program. We analyze the steps needed to accomplish the task and write those steps out informally. Finally, when we have all the steps laid out, we translate them into whatever computer code we are using (AS2, Java, Visual Basic, etc.). For really sophisticated and complex programs, you may even want to develop flowcharts (I will discuss flowcharts next month).

Now we can start to translate it to Flash ActionScript.

For starters, we'll need to build the user interface. We will need a field for the present balance, amount paid, message, and a button to trigger to action. The user interface should look something like Figure 1.

While this is not essential, notice that I set the dynamic text, static text, and UI components on separate layers. Also, I created an action layer to place some AS code. Additionally I made the balance due a dynamic text field and the amount paid as an input text field. They are called, respectively, `balanceDue_txt` and `amountPaid_txt` (in Part 1 of this series I discussed naming conventions). I called the button `paid_btn` and, for the output, I used a TextArea component called `message_txt`.

Normally you would get the balance due from an outside data source; that is outside of the scope of this article, so we'll write a line of code to set it for us.

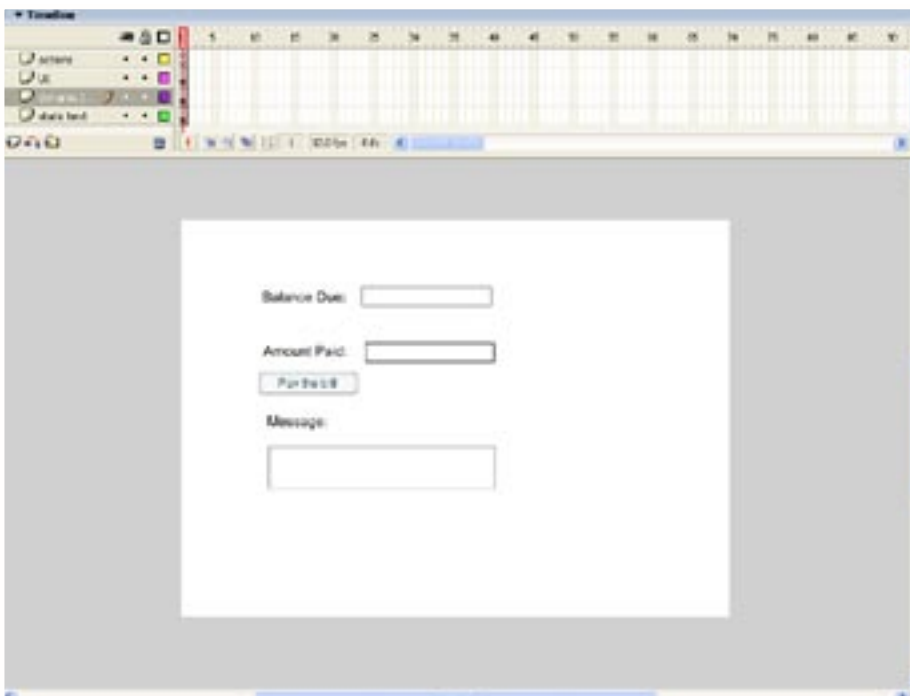
Go to the actions layer, open the Actions Panel, and enter the following line of code:

```
balanceDue_txt.text = "235.00";
```

Remember, the text property sets the text for the `balanceDue_txt` object. If you test it, you should see something similar to Figure 2.

The rest of the code will be tied to the button, so click on the button compo-

figure 1



nent and go to the Actions Panel.

You need to build the code within an event. For most buttons, the event is press. Your opening shell code should look as follows:

```
on(press)
{
}
}
```

Now, here is a little quirk that is true of most programs today: TextFields are just that, text. All they can work with is text. However, we need to perform some calculations to determine a balance due or credit. Happily, Flash has a Class file all ready to help us out (Class files were discussed in Part 1). This file is called Number. We can convert the text in balanceDue_txt to a number by setting the following variable:

```
on(press)
{
    var due:Number = Number(this._parent.
    balanceDue_txt.text);
}
}
```

As we discussed in Part 1, we set the variable "due" to be of data type Number. Next, we told AS2 to convert the text property of balanceDue_txt to a number. (Note: this._parent has to do with the relationship of the components to the time line. We will be discussing this up the road a bit. For the time being, when selecting components, use the Insert a Target Path tool located in the toolbar of the Actions Panel.)

We now need to do the same for the amountPaid_txt field.

```
on(press)
{
    var due:Number = Number(this._parent.
    balanceDue_txt.text);
    var paid:Number = Number(this._parent.
    amountPaid_txt.text);
}
}
```

Finally, we will need to calculate the balance left after the payment is made. We can do this with a variable that will hold the calculation as follows:

```
on(press)
{
```

```
    var due:Number = Number(this._parent.
    balanceDue_txt.text);
    var paid:Number = Number(this._parent.
    amountPaid_txt.text);
    var newBalance:Number = due - paid;
}
}
```

Again, this variable is of data type Number. As we discussed in Part 1, strict data typing can prevent a lot of possible errors up the road.

We can give our little program a quick test by adding the following code:

```
on(press)
{
    var due:Number = Number(this._parent.
    balanceDue_txt.text);
    var paid:Number = Number(this._parent.
    amountPaid_txt.text);
    var newBalance:Number = due - paid;
    this._parent.message_txt.text = "The
    balance due is " + newBalance;
}
}
```

The + connects the literal text (enclosed in quotes) with a variable. This is called concatenation. Believe it or not, when you concatenate, everything is automatically converted back to text.

If you give the program a test, type in a payment amount and click the button; you should see something like Figure 3. Do not use a dollar sign when you type the number. We will not be concerned about formatting the numbers in this article. (Note: if you don't type a number or use the dollar sign, you might get a NaN in the TextField box. We will be fixing that shortly.)

It's now time to get to the subject at hand. If you set a number less than \$235, you'll see the number. If you set a payment amount greater than \$235, you end up with a negative number. If you pay exactly \$235, it will show 0 as the balance. In all situations, the message is exactly the same with only the number changing.

We can give the message a bit more flexibility by using a decision structure. We have four possible scenarios:

1. The user types an amount less than the balance.
2. The user types an amount greater than the balance.
3. The user types an amount equal to the balance.

4. The user does not type any number.

We use an if/else structure. The generic syntax is:

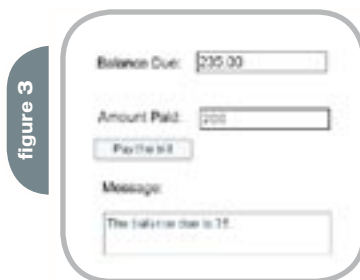
```
if (condition to test for)
{
}
}
```

The condition to test for returns either true or false (in computer terms, we call this a Boolean expression). Let's begin by taking out the line of code that sets the message_txt.text property. You can do this by either deleting it or commenting it out with the // characters (a discussion of commenting occurred in Part 1).

Let's set the following if statement.

```
on(press)
{
    var due:Number = Number(this._parent.
    balanceDue_txt.text);
    var paid:Number = Number(this._parent.
    amountPaid_txt.text);
    var newBalance:Number = due - paid;
    if (newBalance > 0)
    {
    }
}
}
```

This code contained within the {} will run only if the conditional statement is true (in this case, if newBalance is greater than 0). We will tell it to set the mes-





“We can give the message a bit more flexibility by using a decision structure”

sage as follows:

```
if (newBalance > 0)
{
    this._parent.message_txt.text = "You
    have a balance of " + newBalance;
}
```

If you test the movie and type an amount less than the balance, say 200, the message box displays the message. However, if you type an amount that is greater, there is no change in the message. This is because an amount that is greater will return a condition that is false and the code within that if statement will not run.

We now need to handle the second possibility where the user types an amount greater than the balance. In Listing 1 we append to an “if” statement with “else if.”

If you now test the movie and type an amount greater than the balance, say 300, a message will be displayed. You may notice that the amount displayed is a negative number. This is easily fixed by a tiny algebraic adjustment to the code.

```
this._parent.message_txt.text = "You
have a credit of " + -newBalance
```

That negative sign before the variable negates the negative sign (back to high school algebra).

You should be starting to get the idea, but we will now see a couple of fine

points. Let’s address the third scenario where the user types an amount equal to the balance due. Of course, since we are appending another if statement, we will use an else if (see Listing 2).

Notice that in the conditional statement I used an == as opposed to a single =. The double equal (==) is called a comparison statement because it is comparing what is on the left of the equal signs to what is on the right. A single equal sign is called an assignment statement because it assigns what is on the right side of the equal signs to what is on the left. Please note that there cannot be a space between the equal signs.

Finally, at the end of a decision structure you can place an else statement. The else is the code to execute if none of the if statements returns a value of true. Since it executes automatically if none of the if statements returns as true, it has no conditional statement of its own. In our simple example, we are going to inform the user that he did not type a number (see Listing 3).

Remember, the else statement is optional. If you don’t use it, control of the code will return to the first line of code after the decision structure.

While each of our statements has only one line of code within it, it is not unusual to have many lines of code contained within the braces. As a matter of fact, in some cases there could be an additional if statement within an if statement. This

is called an embedded statement. It might look something like this (this is not a working piece of code; it’s just an example):

```
if (newBalance > 0)
{
    this._parent.message_txt.text = "You
    have a balance of " + newBalance;
    if (newBalance > 50)
    {
        this._parent.message_txt.text =
        "..."/>

```

In one situation, I actually had five levels of if statement.

In our small example, we have no way of predicting what number the user will type. However, let’s assume we have a program where the user is restricted to only certain selections. For example, let’s say that the user can only type either 1, 2, or 3. Rather than build a series of if statements, you may find it a bit easier to use a variation of a decision structure called a Switch Statement. Our code might look something like Listing 4 (this is not meant to work in place of the above example).

There are a few things that need to be noted here. The beginning of the switch statement accepts the variable (in this case, the variable called selection) and assigns it to a matching case. Notice that at the end of each case there is a colon. When the code is completed for that case, the keyword break is used. This is the equivalent of a closing brace in an if statement. If you do not use the break keyword, the program will continue to the code for the remaining cases.

Finally, if something is entered that does not match any of the cases, it’s a good idea to use a default case. This is a general catch-all.

Next month we will talk about looping structures.

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

listing 1

```

on(press)
{
var due:Number = Number(this._parent.balanceDue_txt.text);
var paid:Number = Number(this._parent.amountPaid_txt.text);
var newBalance:Number = due - paid;
if (newBalance > 0)
{
this._parent.message_txt.text = "You have a balance of " + newBalance;
}
else if(newBalance < 0)
{
this._parent.message_txt.text = "You have a credit of " + newBalance;
}
}
}

```

listing 2

```

on(press)
{
var due:Number = Number(this._parent.balanceDue_txt.text);
var paid:Number = Number(this._parent.amountPaid_txt.text);
var newBalance:Number = due - paid;
if (newBalance > 0)
{
this._parent.message_txt.text = "You have a balance of " + newBalance;
}
else if(newBalance < 0)
{
this._parent.message_txt.text = "You have a credit of " + -newBalance;
}
else if(newBalance == 0)
{
this._parent.message_txt.text = "You are paid in full";
}
}
}

```

listing 3

```

on(press)
{
var due:Number = Number(this._parent.balanceDue_txt.text);
var paid:Number = Number(this._parent.amountPaid_txt.text);
var newBalance:Number = due - paid;
if (newBalance > 0)
{
this._parent.message_txt.text = "You have a balance of " + newBalance;
}
else if(newBalance < 0)
{
this._parent.message_txt.text = "You have a credit of " + -newBalance;
}
else if(newBalance == 0)
{
this._parent.message_txt.text = "You are paid in full";
}
else
{
this._parent.message_txt.text = "You did not enter a number";
}
}
}

```

listing 4

```

var selection:Number = Number(selection_txt.text);

switch (selection)
{
case 1:
message_txt.text = "You have selected 1";
break;

case 2:
message_txt.text = "You have selected 2";
break;

case 3:
message_txt.text = "You have selected 3";
break;

default:
message_txt.text = "You have not selected a number";
}
}

```

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



The “Outside the Box” Technology

*An Introduction to Flash Communication Server and Flash Video
and a tutorial on creating a video player*
by jim phelan

anyone who regularly visits Macromedia’s Web site has inevitably seen the Red Bull Copilot and Volvo V50 Web sites. These elegantly crafted applications are perfect examples to make the case for Flash video: they combine the rich user experience of a Flash application seamlessly with high quality streaming video.

Macromedia purports Flash video to be a technology that is “outside the box,” implying that it can be integrated gracefully with a Web site or Flash application without a stale, uninspired video player surrounding it. Aside from the other major benefits of video in Flash, such as Flash’s massive install base and cross-platform compatibility, this point should be well taken. I’ve used the phrase “Outside the Box” prominently in the title of this article because I believe that this development represents a sea change in the way we deliver media to our users: dynamic, interactive multimedia applications are now possible in a major way. As developers, we get to think in new and exciting ways about how we use Flash and how we deliver media.

On the other hand, although Flash video is definitively “outside the box,” it is not necessarily “out of the box,” so this article’s goal is to familiarize you with the tools and skills you’ll need to start developing applications to deliver video with Flash.

About this Article

If you’re an intermediate to advanced

Flash developer who wants to get his or her feet wet in Flash Communication Server technology, this article is for you. We’ll create a simple video player that highlights the key concepts in client-server applications with FlashComm. For the sake of clarity and brevity, I’ve kept the code minimal. In a more real-world video player, we would want to add features such as dynamic parameters, a resizing video window, a bandwidth tester and multiple bitrate FLVs, and skinning. As we develop the video player, we will create all of the code and assets from scratch and we won’t use pre-built components. This will enhance our understanding of how this technology works while also affording us a truly custom application.

Flash Communication Server: Making It All Work

Flash Communication Server MX is Macromedia’s application and streaming server, and it’s the technology that makes streaming video to a Flash application possible. Much like a Web or FTP server, FlashComm lives on a Windows or Linux box and hosts server applications which communicate with client applications. FlashComm applications are written in Actionscript for Communications, a language which is syntactically equivalent to Actionscript 1.0.

FlashComm and Flash Player communicate through the Real Time Messaging Protocol (RTMP), Macromedia’s proprietary protocol for communications. RTMP

is used to send audio, video, and data information between the server and client. RTMP, which transmits packets using TCP/IP, is very effective for transferring data as well as streaming audio and video. By default, FlashComm operates on port 1935, but can be configured to run on other ports as well. To skirt the issue of firewalls, the client and server can also communicate using HTTP-tunneling (known as RTMPT), which means that data is cleverly disguised as regular web traffic and sent through port 80, making FlashComm a viable technology for enterprise deployments.

FlashComm isn’t just for on-demand video. The rich server side scripting language allows you to develop extremely advanced multi-user applications. Flash Player can encode live video from a Web camera and stream it to the server, which opens up a world of possibilities. Video chat programs, live events, and multi-user games are just a few examples of applications that can be developed with FlashComm. At Stream57, for example, we develop applications for e-learning, live events, video conferencing, and marketing.

Installing Flash Communication Server

To get the full impact of this article, you should have access to a Flash Communication Server to develop on. Although I wouldn’t recommend purchasing it straight off (a professional

"Flash Communication Server MX is Macromedia’s application and streaming server, and it’s the technology that makes streaming video to a Flash application possible"

license costs almost \$5,000), I do recommend downloading the developer edition for free from Macromedia's Web site. You can download either a Windows or Linux version for your workstation or development machine; make sure to install the updater as well. I run FlashComm on my laptop alongside an IIS Web server without any issues. After installing, take note of two important directories:

The script files for your applications will always go in the "applications" directory, which is usually located at:

C:\Program Files\Macromedia\Flex Communication Server MX\applications

FlashComm's XML configuration files go in the "conf" directory, which is usually located at: C:\Program Files\Macromedia\Flex Communication Server MX\conf.

There will also be a Start menu folder for FlashComm, which contains the documentation and administration applications.

The Low-Down on Flash Video

Flash video is a unique format which uses a video codec from Sorenson called "Spark." Flash video lives in FLV files, which can be created using a tool such as Macromedia Video Exporter (in conjunction with a video editing application) or Sorenson Squeeze. To get started with Flash video, you may want to install one of these tools and create some FLVs to stream from your server.

Writing the Server Side Code

First, create a new folder in the FlashComm applications directory called "videoplayer." Within this folder, create a document in a text editor called "main.asc." This will be the main script that makes the videoplayer application run. See Listing 1 for the main.asc code.

Every time a client connects to this application, application.onConnect is called with a reference to the client object passed as a parameter. This basic script for our server side application accepts new connections from client applications and affords us a small amount of security by requiring that the connecting application supply a password.

The application we've created is

named "videoplayer" because that is the name of the directory we placed it in. We can also have multiple instances of applications running at the same time that use the same code but do not share properties or streams.

Start up the Communication App Inspector, which can be found in your Start menu. Normally, an application loads on the server only when a client requests it, so we'll start this one manually to make sure it's working correctly. After logging in with the credentials you entered during installation, type "videoplayer/myvideos" into the App/Inst text box and press "load." The "myvideos" instance of the application is loaded. Click on "videoplayer/myvideos" in the Active Application Instances window and press "View Detail." Press "Reload App" to see the output when the application starts. Among other things, you should see "videoplayer application started," which is our trace output from line 2.

Now, let's put some video on the server. You can create an FLV from another video using your editing application and Flash Video Exporter or Sorenson Squeeze, or you can download the sample FLV for this tutorial from www.flashalchemy.com/mxdj. Name your video "myvideo.flv" and place it in the "applications/videoplayer/streams/myvideos" directory (you'll have to create the "streams" and "myvideos" directories yourself.) All video files are housed in the "streams" folder and we place ours in the "myvideo" folder so it's available to the

"myvideo" instance of the application.

Writing the Client Side Code

First, we'll create a simple video player that connects to the server and plays the video stream. Create a new document in Flash and size it to 240x180. Select "New Video" from the Library panel options menu (see Figure 1). Drag the new video object from your library to the stage and size it to 240x180. Name this video instance "videoPlayback."

Add the code in Listing 2 to the actions on the first frame of your document. Publish the movie, and you should see the video play within a second or two. The first line of this code creates a new NetConnection object, and the following line connects to the server. The first parameter of the connect call is the URI of the server, which is broken up as protocol://host/appname/appinstance. In our case we're connecting to localhost since we're working on a development machine, but normally you would put the hostname of your production server here.

The NetStream object is what makes the video play, and is instantiated with a reference to the NetConnection object as the only parameter. After we create the NetStream, we call the play method with the name of our FLV as a parameter (the .flv extension is always omitted when playing streams.)

This all seems simple enough, so let's move on and take a look at how a slightly more advanced video player is constructed. You should download the source files for the VideoPlayer applica-

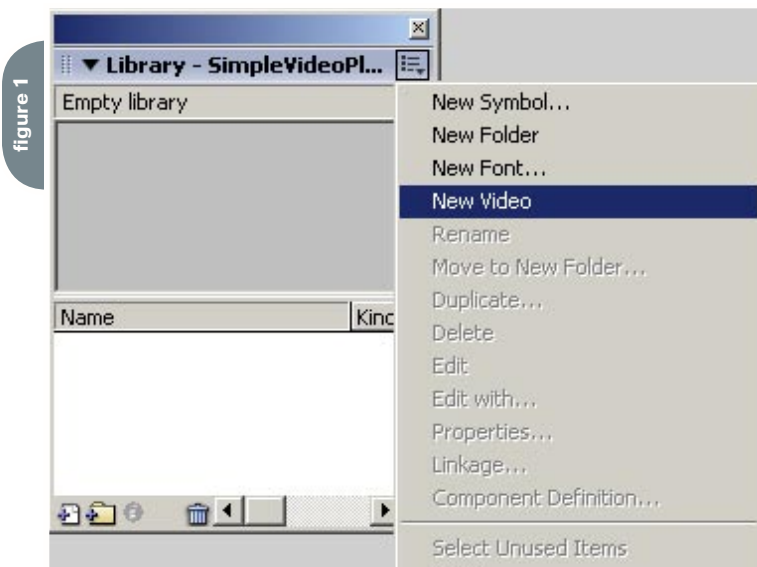


figure 2



tion from www.flashalchemy.com/mxdj before continuing.

Examining the VideoPlayer Application

The client side code for the VideoPlayer application is written in ActionScript 2.0, so you'll need Flash MX 2004 or Flash MX 2004 Professional to work with it.

The VideoPlayer application comprises several files. The main VideoPlayer.fla file contains the graphical assets and embedded video object, as well as a line of code that instantiates the VideoPlayer class. The VideoPlayer.as file holds the VideoPlayer class, which does most of the work for the application. Additionally, there is a file called VideoController.as, which controls the user interface elements of the application (such as the slider and play button.) For the purpose of this tutorial, we'll treat the VideoController class as a black box, but you can take a look at the class file if you're curious to see how it works.

Listing 3 shows the connectToServer() method of the VideoPlayer class, which is called from the class constructor. You'll notice that this is a bit more complex than our simple video player; this is because we need to know exactly what's happening as we connect to the server. By overriding the onStatus method of the NetConnection object, we can get valuable information about our connection. The code property of the onStatus information object supplies us with a textual code that represents the status of the connection. In most cases, we hope to get "NetConnection.Connect.Success," which indicates that our connection was successful, but a real-world player should handle errors as well. For a full list of status codes, see the "Client-Side Information Objects" section of the

FlashComm documentation.

Listing 4 shows a snippet of the initNetStream() method, which is called after the NetConnection succeeds. You'll see here that we are relying heavily on information object dispatched from the server to control our application. Also take note of the setBufferTime call: this method allows us to control how much media is buffered by the Flash Player before the video playback begins. In our simplified application we set the buffer time to 1 second and hope for the best, but in a production ready application we might fine tune buffering in consideration of users with slow or erratic Internet connections.

More Client-Server Interaction

The bulk of the code in the VideoPlayer application has to do with controlling playback with the VideoController, which allows seeking as well as pausing. Seeking, however, is not as easy as it may seem. To seek to specific points in the video using the slider bar, we'll need to know how long the video we're playing is. We'll also need this information to move the slider handle along as the video playback progresses. However, there is no reliable way to determine the length of the media we're playing without getting some information from the server. To make this work, we'll need to ask the server how long the stream is and wait for a response. This is called a "remote method," and is one of the most important concepts in creating rich client server Flash applications.

For this, we'll need to revisit our main.asc file. Listing 5 shows the new code we'll be adding to main.asc to allow clients to request the length of a stream. Since we're not working in ActionScript 2.0 on the server side, we need to regress to the old days of prototyped classes. Here, we're extending the Client class to have an additional method called "getStreamLength." We get the stream's length by employing the length method of the Stream class, and then return this to the client. The getStreamLength method is now available on any client object that exists in this application, and can be called remotely.

Back on the client side there's still a little more work to do. Listing 6 shows another snippet of the initNetStream() method of our VideoPlayer class, which calls get-

StreamLength by using the call method of the NetConnection class. The first parameter of the call is the name of the method we want to use. The second parameter is our result object; since this call is asynchronous (doesn't return immediately), Flash needs to know what to do with that result when it is received. The onResult method of our result object will be called with our result as a parameter. In our case, we take the result (which is the length of the stream in seconds) and store it for later use.

Using the VideoPlayer


Now that we've added the appropriate server side code, we can take a look at our VideoPlayer application. Because we changed the main.asc file, you may need to restart the server side application using the Communication App Inspector. If you run the player in the Flash authoring environment, you'll see some useful information in the output window.

I highly recommend examining the rest of the code that comprises the VideoPlayer application to see how it works. If you're feeling daring, you might add some extra features; making the server URI and media name dynamic would be a good start.

Resources

All of the files referenced in the article can be downloaded from: www.flashalchemy.com/mxdj.

Listed below are several resources you may find helpful if you want to learn more about Flash Communication Server. If you need a pre-built, customizable video player, I recommend Giacomo "Peldi" Guilizzoni's open source FLVPlayer, which is also listed below.

- *Macromedia Flash Video Developer Center:* www.macromedia.com/devnet/mx/flash/video.html
- *Macromedia Flash Communication Server Developer Center:* www.macromedia.com/devnet/mx/flashcom/
- *FLVPlayer:* www.peldi.com/blog/archives/flvplayer/index.html 

Jim Phelan is VP of Development at Stream57, a New York City firm specializing in creating Rich Internet Applications for live video events, videoconferencing, and e-learning. Jim is an avid enthusiast of Flash and develops primarily in ActionScript.
jim@flashalchemy.com



listing 1

```

application.onAppStart = function() {
    trace("VideoPlayer application started.");
}
application.onConnect = function(clientObj, authObj) {
    if (authObj.password == "hackme") {
        trace("Client accepted from "+clientObj.ip);
        application.acceptConnection(clientObj);
    } else {
        application.rejectConnection(clientObj);
    }
}

```

listing 2

```

var nc = new NetConnection();
nc.connect("rtmp://localhost/videooplayer/
myvideos",{password: "hackme"});
var ns = new NetStream(nc);
videoPlayback.attachVideo(ns);
ns.play("myvideo");

```

listing 3

```

private function connectToServer():Void {
    trace("Connecting to the server...");
    var videoPlayerRef:VideoPlayer = this; // Refers
back to this instance
    _netConn = new NetConnection();
    _netConn.onStatus = function(statusObj:Object):Void {
        trace("Got status from _netConn: "+statusObj.code);
        switch (statusObj.code) {
            case "NetConnection.Connect.Success":
                videoPlayerRef.initNetStream();
                break;
            case "NetConnection.Connect.Failed":
                trace("Connection to the server failed!");
                break;
            // Other status handling should be done here
        }
    }
    _netConn.connect("rtmp://localhost/videooplayer/
myvideos",{password: "hackme"});
}

```

listing 4

```

var videoPlayerRef:VideoPlayer = this;
_netStream = new NetStream(_netConn);
_netStream.setBufferTime(1); // Buffer 1 second of
video before playback
_netStream.onStatus = function(statusObj:Object):Void {

```

```

    trace("Got status from _netStream: "+statusObj.
code);
    switch (statusObj.code) {
        case "NetStream.Play.Start":
            if (videoPlayerRef._isSeeking) {
                videoPlayerRef._isSeeking = false;
                if (!videoPlayerRef._isPaused) {
                    videoPlayerRef.watchProgress(true);
                }
            } else {
                videoPlayerRef._isPaused = false;
                videoPlayerRef._videoController.showPauseBut-
ton();
                videoPlayerRef.watchProgress(true);
            }
            break;
        case "NetStream.Pause.Notify":
            videoPlayerRef._isPaused = true;
            videoPlayerRef._videoController.showPlayButton();
            videoPlayerRef.watchProgress(false);
            break;
        case "NetStream.Play.Stop":
            videoPlayerRef._isStopping = true;
            break;
        case "NetStream.Buffer.Empty":
            if (videoPlayerRef._isStopping) { // If the
stream has stopped and the buffer is empty, playback
is complete.
                videoPlayerRef.onMediaStop();
            }
            break;
    }
}

```

```

Client.prototype.getStreamLength = function(streamName)
{
    return Stream.length(streamName);
}

```

```

var streamLengthResult:Object = {};
streamLengthResult.onResult = function(streamLength:
Number):Void {
    trace("Got stream length result: "+streamLength);
    videoPlayerRef._streamLength = streamLength;
}
_netConn.call("getStreamLength",streamLengthResult,_
mediaName);

```

listing 5

listing 6

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



sparkling text effects

Sometimes you just need to have a little sparkle on your pages. If we're talking about Web pages, then those sparkles better shine and twinkle! You can use FreeHand MX to create the right amount of "bling" in a few simple steps.

by ron rockwell

freehand's extrude
tool in action



When you want a smooth, slick extrusion, FreeHand's Extrude tool will certainly do the trick. The problem is, however, that you end up with more or less flat coloring on the extruded sides. The 3D effect is correct, but it lacks realism. In this article we'll deal with that situation and kick up the action by creating an animation for the Web. Along the way, we'll use a few tools that are largely ignored – well, by me anyway. The article deals with a text object, but the final artwork could be just about any type of logo or flat graphic. If you'd like to see this in action, go to www.brainstormer.org/bling.html.

This extrusion will have a rounded outline, leaving the text inset such as you might see on an electric sign above a store. That allows us to stretch the limits of the Extrude tool just a bit.

To begin with, set the text in a strong font with a lot of surface area. It would be safe to use any font that's got "Black" or "Extra Bold" in the name. Anything less will have a much-reduced impact, but still be effective. Use a little extra positive kerning between letters to keep the extrusion's bevels from intersecting each other. Create a golden color in the Mixer panel, add it to the Swatches panel, and apply that color to the text. Convert the text to paths (Text > Convert to Paths). I constantly convert text to paths for one reason or another, and to save time scrolling through menus, I've added its icon to my main toolbar. To make it even more convenient, I created a custom keyboard shortcut that I can press with only one hand, freeing my other hand for my coffee cup.

In preparation for the extrusion, we need a bevel shape for the inset outline. Using the Pen tool, create a shape similar to the one in Figure 1. The distance between the bottom and top lines will be the thickness added around the actual letterforms. The distance from the starting point (1) to the left edge of the curve (2) will be the depth of the inset. The length of the bevel (3) isn't quite as important, but experimentation will give you various degrees of success for any given project.

Wherever you place the first point in the profile will be the location of the object that is going to be extruded. In this example, if you were to start draw-

ing at the right end (3) and work back to the inset point (1), your extrusion would be very complicated, but you would not see anything other than a straight, flat extrusion from the letterform's face. The curved part would end up being hidden at the back end of the extrusion. If you see something like that, it's a simple matter to change the direction of the path (Modify > Alter Path > Reverse Direction). You need to apply this action often if you've started with a regular shape such as an ellipse or rectangle and have cut pieces away from the shape.

Keep in mind that the profile you draw will be expanded or compressed to the length (or depth) of your extrusion. If you want a tight representation of the profile, then your extrusion must not be deeper or shallower than the profile you've drawn. The line weight (stroke width) is not important, nor is the color of the stroke. All you're doing is creating a profile. When it's done, copy it to the Clipboard.

Select the Extrude tool from the Tools panel, and click it in the middle of the text. You'll see the vanishing point, a rectangular block in perspective, and three icons in the middle of the text. For now, you can ignore them – the extrusion itself is done, and now you can modify its position in space.

Keep the Extrude tool active and double-click the extruded text to bring up a circle that may or may not encompass the entire text. If you click and drag the cursor inside the circle, you will move the text around its center of gravity – along all three axes at once. If you want to rotate the text further, you can click and drag the cursor outside the circle and rotate the text around a single axis. Try several positions until you see something you like. Remember to Save often.

The extrusion is rarely the depth you want. To change it, click outside the rotation circle to deselect it, and click the extrusion again. Then click and drag the small circle at the opposite end of the dotted line ending with an "X." The "X" marks the face of the object, the "O" denotes the far end of the extrusion, and the four diamonds indicate the vanishing point of the object's extrusion before you rotated and shifted it. Until you commit the extrusion to a simple graphic (by choosing "Release Extrude" from the

Modify menu), all these factors are editable.

Now look in the Object panel, specifically the three un-named buttons on the left side of the panel. The top one is for Extrude, and shows the various positions of elements in the extrusion you've made and manipulated. You can enter numbers here or change values to further fine tune your extrusion and its position in space. The middle button is Surface, and controls lighting and rendering options. The default surface settings will work fine for now, but change the lights as you wish. Finally, at the bottom, is Profile, where we can place the line drawing we made earlier. Just choose Bevel from the dropdown menu at the top of the panel (your choices are None, Bevel, or Static. Bevel will change the shape of the extrusion as it heads back to the vanishing point; Static acts as a spine for the extrusion – we definitely don't want that here! Click the Paste-In button and you'll see the path in the preview window, then the extrusion will adapt to its new bevel.

Sometimes, such as in this example, you get some surprise elements. What you see here are parts of the bevel that somehow get placed in front (or on top of) the front face. It doesn't happen with every letter, or with every adjustment in the extrusion's attitude. Sometimes moving the extrusion just a degree or two in any given direction will make these artifacts go away. Sometimes, nothing you do will negate them and you have to resort to a trick or two (we'll get into them a little later). Progress so far should lead you to something similar to Figure 2.

Release Extrusion

An extrusion is a group of many, many triangles arranged into facets that build the 3D object. If you select Remove Extrusion from the Modify menu, the extrusion will go away, and you'll have the original object on the page – where you usually go to start over. However, if you select Release Extrusion, the opportunity to further manipulate the position, lighting, profile and so on are released, and the object becomes that mass of triangles mentioned above. As well, the original object will be deformed or converted to the 3D perspective shape, and the entire extrusion will be grouped. This allows you to choose the Subselect



figure 1

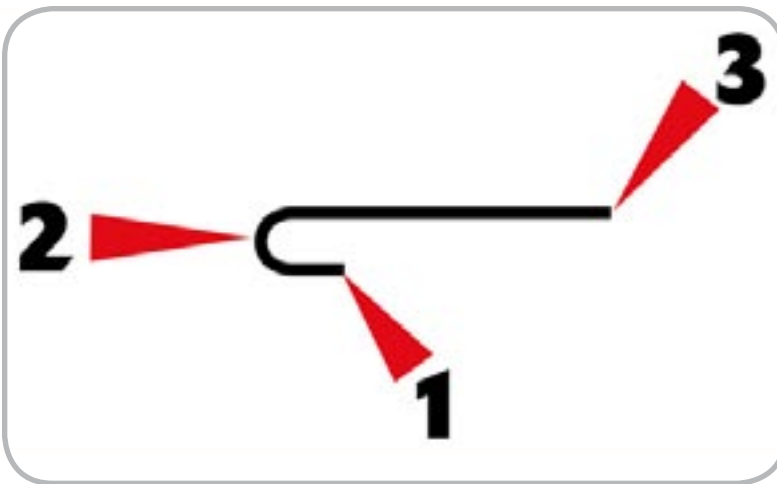


figure 2



figure 3



tool and delete each triangle artifact that you don't want. More important, you can select the front face of the extrusion and manipulate it.

Combine Faces

I don't want to have simple flat shading on the extrusion, so some extra work is required. We'll use the Lasso tool to select triangles on a single face at a time. Double-click its icon in the Tools panel and check the Contact Sensitive option (Figure 3) so you don't have to encompass entire objects. Then begin dragging within a given face. In Figure 4, the bottom of the "N" is being selected.

Use the Pointer tool with the Shift key held down to deselect extra triangles. When you have an entire face selected, go to **Modify > Combine > Union**. Apply a gradient filter such as seen in Figure 5, and delete the extraneous points in the path. Do the entire side or vertical plane in this manner. Then do the top (or bottom if your extrusion is seen from below) faces, but after using **Union** on each face, move to the next one, and when you have them all done, select them and choose **Modify > Join** so you can apply a gradient fill across the entire plane. The finished product should be similar to Figure 6.

Adjust the Front Letterforms

Making the inset bevel fit the profile is a little more demanding than what you've done so far, but it doesn't take long to get the hang of it. Start by Shift-

"This extrusion will have a rounded outline, leaving the text inset such as you might see on an electric sign above a store"

selecting the letter forms – if you accidentally snag a triangle, just click it again to deselect it. When all the letters are selected, Clone them. I usually clone an object like this in case I have to start from scratch later on – the original objects are still there, and I can delete them later.

The clones will move to the top of the stacking order, and you'll see right away that they cover up the near edges of the bevel. Change the color of the letters to what you want in the final result. Then work each near edge, one at a time. It's pretty easy to select all the points on the vertical face and shift them to comply with the bevel, but remember, this is in perspective; you must readjust every point to make it fit. Inner points such as found in the "N" and "W" have an additional element to deal with in that you must add a new point or two to create the depth adjustment around the bevels. It's easy enough to add a point, move it to the edge of the bevel, and drag out a control handle to create the correct fit and effect. If your letters have curves in them, your work is

a bit more complicated.

I try to get as much realism in a drawing as possible, and edge and corner treatments are very important. To create an edge highlight, draw a small circle and ungroup it. Drag the right side point of the circle far to the right (about ten to fifteen times the radius of the circle), and tuck in both control handles by clicking the Handles buttons in the object panel. This will give you a dart as shown in Figure 7. Choose Modify > Brush > Create Brush, and Copy the dart (instead of Convert). Name the brush White Dart, Edge Highlight, or something else memorable, because you won't be able to see it in the brushes menu. Use the Line tool to draw lines along leading edges of the letterforms, and apply the brush so the fat end of the dart is near the viewer, and it tapers toward the vanishing point. These highlights will add quite a bit of "bling" to the object.

Commit

At this point, the menial tasks are done, and it's time to work on the animation. But please don't even think about using an extrusion in an animation! The hundreds or thousands of triangles in the extrusion take many seconds to draw, and the resulting animation becomes extremely, excruciatingly boring to watch.

FreeHand offers you the choice of using pages and/or layers in your animated movie, but I like to keep everything simple by creating an animation created of pages, or layers on a single page. Call me obsessive-compulsive (I am...) but it's more effective for me to work by simplifying, thereby creating fewer opportunities for errors. So at this point do a Save As and change the name of the file to add something about "animation."

In the new file, select the letterforms – the ones that you changed color and shape to fit the bevels. Choose View > Hide Selection. Then use the Pointer tool to drag a selection box around the entire graphic. Go to Modify > Convert to Image. This is the main object of the new document, because this action will create an RGB TIFF bitmap of the extrusion, and that will negate any chance of you editing the vectors that created it. Hiding the letterforms protects them from the

figure 4

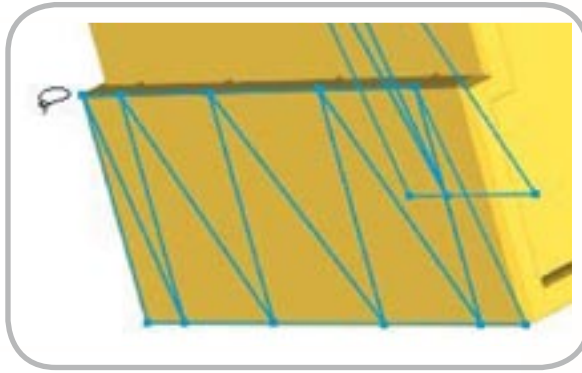


figure 5

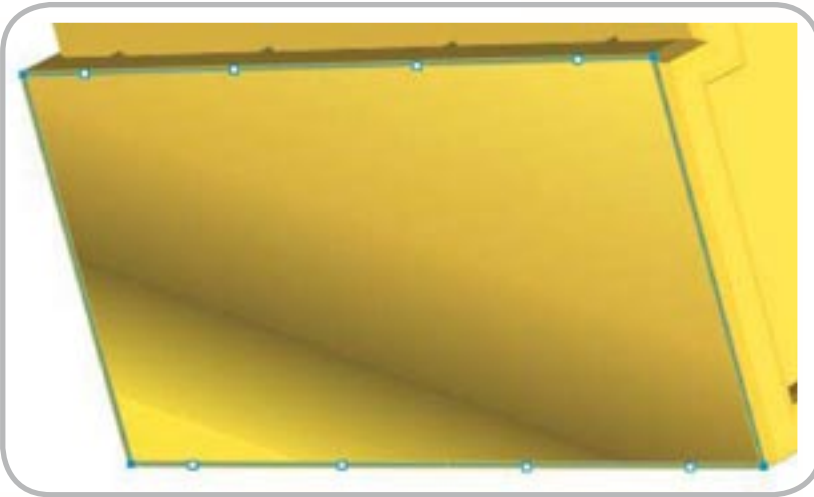


figure 6



conversion. A dialog box appears (Figure 8) that allows you to choose the resolution of the graphic, and the degree of anti-alias. I want to use this graphic at my working size, so I'll stick with 72-ppi (pixels per inch).

It is extremely important to note that if you wish to have a colored background for the objects, you must create a back-

ground first and select it at the same time as the graphic when you Convert to Image. Make it the final size you want it

figure 7





figure 8



wrong with FreeHand or your computer – a color shift will almost always occur when you convert color models, and some colors are more evident than others.

Select the original white dart shape, double-click the Mirror tool and set it as shown in Figure 10. When you click the

figure 9



figure 10

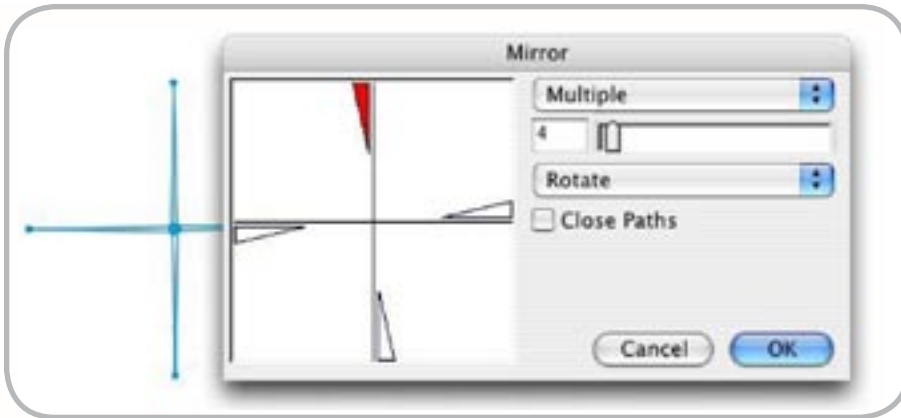
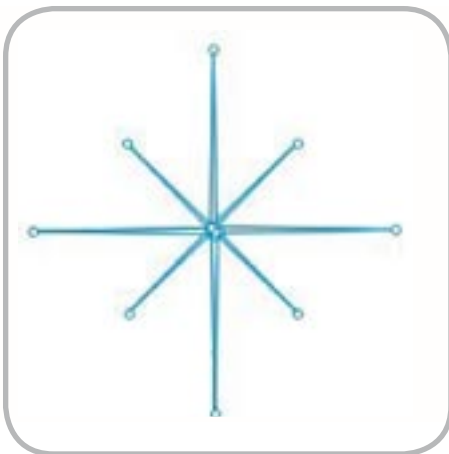


figure 11



to have, and for good measure, use RGB colors for the color make-up. If you've used CMYK colors to build the artwork so far, they will be converted to RGB as soon as Convert to Image is done. There will be a noticeable color shift! See Figure 9 to see the difference between the vector graphic on the left, and its conversion to an RGB TIFF on the right. There's nothing

cursor at the center of the circle end, you'll get the star shape in the figure.

Clone the four objects and rotate the clone 45°, then switch to the Scale tool and click in the intersection of the arms; drag the cursor down or to the left to shorten the arm length. Finally, use the pointer tool to drag a selection box around all eight darts (Figure 11) and choose Modify > Union to create a single star object. Delete its stroke, and give it a fill of white.

Now, create a small circle with a fill of white and no stroke. Double-click the Graphic Hose icon and Copy/Paste the circle and star into two new hoses. If you haven't done this before, just copy the star to the Clipboard, open the Graphic Hose (see Figure 12), and select New from the drop-down menu. The Name dialog box will appear, name your object, and click save. Then click the Paste In button. Since the star is white, you won't see anything, but it's there. Do the same procedure with the white dot, and name

it Specular Highlight. You won't be using the Graphic Hose items right now, but you're prepared for their application during the upcoming animation sequences.

Move the bitmap to the Background layer, either by selecting the bitmap and clicking on the Background layer in the Layers panel, or dragging the layer that contains the bitmap beneath the printing/non-printing line in the Layers panel. This will insure that the bitmap will appear in every frame of the movie.

Create the Spotlight Glow

Choose View > Show All to bring back the letters. They should all be selected, so, go to Modify > Join to create a compound path to work with. Create a linear gradient fill with darker and lighter tones in it to give the illusion of depth – and give the image a little punch at the same time.

Use the Rectangle tool to draw a tall, skinny rectangle just to the left of the letters that's as tall as the graphic. Use the Eyedropper tool to drag a selection of the color in the gradient fill onto the Tints panel. Give the rectangle a fill of somewhere between 40 and 80 percent tint of the color. Apply a stroke of none. Clone the rectangle and shift it to the right side of the graphic. Blend the two shapes, and adjust the number of steps so the edges of each rectangle barely touch each other – there should only be 8 – 10 steps in the blend.

The letters should be on the Foreground layer, and the extrusion should be on the Background layer. Create four or five new layers by going to the Options drop-down button at the top right corner of the Layers panel and choosing New (repeatedly). Create a clone of the letters and move them to that top layer and rename the layer "Animation."

With the letters still selected, choose Xtras > Animation > Release to layers. In the dialog box, choose Sequence, and be sure that the other options are deselected. This will place your blended shapes on as many new layers as you have steps in the blend.

The Boring Stuff

Now, go to the Layers panel Option drop-down menu and choose the All Off option. Click in the empty spot to the

be the pilot!

FREE SETUP on Shared Hosting Accounts With ColdFusion MX Support
Use Promo Code CF0J2005



INTERMEDIA.NET

WE DARE YOU TO TAKE A FREE TEST FLIGHT!

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

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

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

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

BE THE PILOT. Take a free test flight and see what our HostPilot™ Control Panel offers you beyond all others. Check out our SLA guarantees. To see more testimonials and to find out about our competitive advantages, visit our Web site at www.intermedia.net.

Managed Hosting • Shared Hosting • Microsoft Exchange Hosting

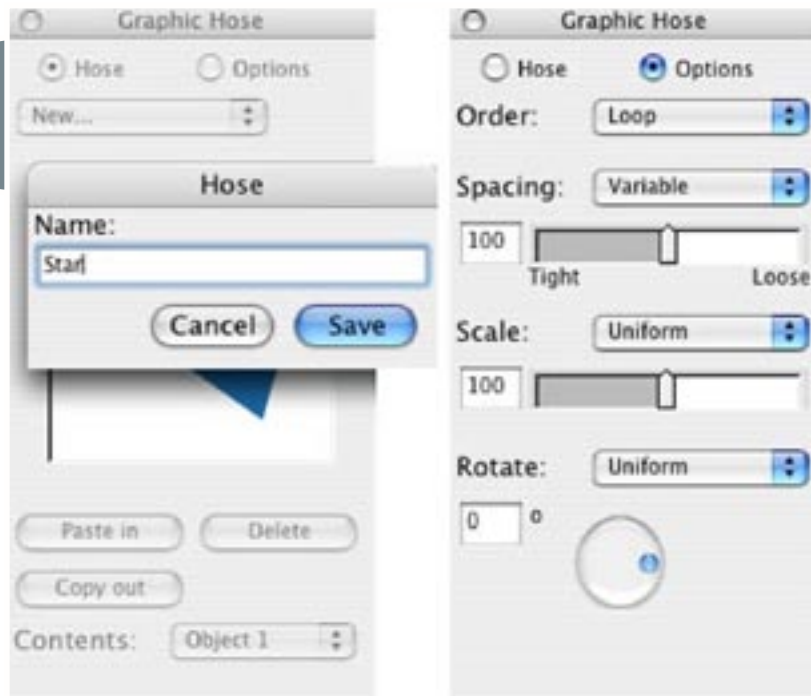
Call us at: 1.800.379.7729 • Visit us at: WWW.INTERMEDIA.NET



HostPilot™



figure 12



left of the Animation layer. This should make the letters and the first blend object appear. Select the letters and Clone them and move the clone to Animation-1 layer (the next layer up in the panel list). If the blend object overlaps the letters, select it and cut it, then select the letters and choose Edit > Paste Inside (Paste Contents). Hide this layer and make the next higher layer visible. Repeat the process (clone the letters and move them to the next higher layer; cut the blend shape; paste it inside the letters; hide the

layer; show the next higher layer, cuss me...) until all the blend steps are used up.

At this point, I like to preview the movie, so be sure to Save, then go to Window > Movie > Test (all layers must be visible in order to appear in the movie). It should look as if the blend swipes across the letters as if it were a spotlight from below. The problem is, it happens so quickly! To rectify the situation, with all the layers except the Foreground layer hidden, select the letters on the Foreground layer and clone

them. Drag a selection box around them and Blend them. I wouldn't go overboard with the number of steps in this blend, probably a dozen will be adequate. Use the Animate Xtra to create new non-animated layers. Create new layers either below the Animation layers, or above, or both. Each layer is a frame, and the default frame rate is 12-fps, so the amount of wait between spotlight flashes is up to you. When you have placed a number of layers, test the movie again.

When the movie runs long enough, and you have suitable delays between flashes, turn all layers off, and turn the Animate and Background layers back on. Then use the Graphic hose to apply sparkles and twinkles in appropriate areas of the moving spotlight (see Figure 13). If you want, you can place trailing twinkles behind the spotlight. Take one layer at a time, and be fairly consistent with the specular highlight placement so it doesn't look like there's a short-circuit inside the extrusion. Test the movie and make any adjustments that are necessary.

When you have everything working correctly, go to File > Export, and choose SWF from the drop-down file-type menu. The exported file can be placed in any Web page, just as you would a Flash file – because that's what it is. FreeHand isn't just for print – it does quite a decent job creating movies as well – give it a try! You can see the results of this exercise at www.brainstormer.org/bling.html.

figure 13



Reach Over 100,000

Enterprise Development Managers & Decision Makers with...



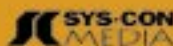
Offering leading software, services, and hardware vendors an opportunity to speak to over 100,000 purchasing decision makers about their products, the enterprise IT marketplace, and emerging trends critical to developers, programmers, and IT management

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



Building Reports with ColdFusion MX 7

How to generate your first ColdFusion MX 7 report (previous ColdFusion knowledge is helpful but not necessary)
by collin tobin & dean harmon

Reporting is one of the most common tasks developers have faced since the beginning of the IT revolution. In the past, Macromedia ColdFusion developers didn't have many choices for easily creating reports and integrating them with their applications. They either had to leave the comfort of using ColdFusion and use a third-party solution or write CFML to generate HTML layouts – a solution that involved writing a lot of CFML code to read data from database, slice and dice the data, and generate HTML output.

Doing that kind of work can be tedious, repetitive, and just plain boring.

“Reporting is one of the most common tasks developers have faced since the beginning of the IT revolution”

Even after all that hard work, the final result can be less than optimal. Because HTML has no notion of page breaks, page headers, and footers, the results seldom look very good when the browser sends a report to the printer.

In ColdFusion 7, the ColdFusion engineering team has created a reporting feature to help liberate you from such tedious tasks. Now you can spend more time developing dynamic applications and less time worrying about whether or not it will print well.

Requirements

To complete this tutorial you will need to install the software and files listed at the end of the article. Previous ColdFusion knowledge is helpful but not necessary.

Getting Familiar with the New CFREPORT Tag Syntax

The cfreport tag has undergone some slight revisions. Aside from still supporting its usual syntax that supports Crystal Reports integration, the new syntax is as follows:

```
<cfreport
    template = "report definition
filename"
    format = "PDF or FlashPaper" or
"excel"
    name = "cf variable"
    filename = "output filename"
```

```
    query = "query variable"
    overwrite = "yes" or "no"
    encryption = "128-bit" or "40-
bit" or "none"
    ownerpassword = "password"
    userpassword = "password"
    permissions = "permission list"
>
    cfreportparam tags [optional]
</cfreport>
```

The ColdFusion MX 7 Reporting Solution

The ColdFusion reporting solution contains two parts: a report engine and a report designer.

The report engine contains the following functionality: graphing capability, columnar layouts, support of report runtime input parameters, and the ability to output in three formats (Adobe PDF,

Macromedia FlashPaper, and Microsoft Excel). You can output these files directly to a browser, save them to disk, or send them through e-mail.

The report feature gives ColdFusion developers many advantages over third-party solutions:

- It understands CFML expressions
- It can use data sources defined in the ColdFusion Administrator
- It has access to shared scopes

The report builder contains the following wizards to help you create reports:

- Report Creation Wizard
- Chart Wizard
- Subreport Wizard
- Server Setup Wizard

Let's discuss the report builder some more.

Introducing the ColdFusion Report Builder

The ColdFusion Report Builder is a tool you can use to create ColdFusion report files (CFR files) that the report engine will consume. The Report Builder is only available to Windows users, even though the reports themselves can run on most platforms to which ColdFusion deploys (Windows, Linux, Solaris, Mac OS X, HP-UX, and AIX specifically – HP-UX and AIX are only supported through pure Java deployments).

When you run the Report Builder for the first time, the Server Setup Wizard helps you configure Report Builder to work with ColdFusion Server. You can modify all of the settings in the Server Setup Wizard from the preferences screen, or you can run the wizard again.

ColdFusion reporting uses a banded-report writing paradigm. In other words, each section of the report is separated into different bands; a section, or band, is a printable region of the report. Each

This article can also be read at the Macromedia Developer Center site (www.macromedia.com/devnet/)

band has its own rules that specify when and how to print. The Report Builder lists each of these bands in the work area from the top to the bottom in the following order:

- Report Header
- Page Header
- Column Header
- Detail, Column Footer
- Page Footer
- Report Footer

There is also a special band called “Watermark” that prints in the background of every page. You can get more information about the different bands by using the integrated help system from the Help menu.

The Report Builder layout has four main areas (see Figure 1):

- Workspace
- Properties panel
- Fields
- Parameters panel
- Text Styles panel

Use the Workspace in the middle of the screen to place your images, labels, or text – anything – that you want to print. When you select an object or text in the workspace, the Properties panel in the upper right displays the properties available for that object. The Fields and Parameters panel below the Properties panel displays the fields (either from your database or ones that the report calculates) and parameters that are available for your report. Last, use the Text Styles panel to set text attributes in your report quickly and easily.

Building a Sample Report

In this tutorial, you will build a sample report with the ColdFusion Report Builder to help you understand the following features of ColdFusion MX 7 reports:

- Layout of a banded report
- Database access
- Text formatting (styles)
- Text formatting (ColdFusion)
- Subreporting
- Group management
- Images and shapes
- Report and ColdFusion expressions
- Labels and fields
- Layout controls
- Fields and parameters

- Query Builder
- Output formats
- cfreport tag modifications

In the following section, you will create a new report that contains several elements: report header, page header, detail pages, and queries:

1. Open the Report Builder.
2. Create a new report document, selecting the As a Blank Report option.
3. Save your report to the webroot of your ColdFusion server, naming it FirstReport.cfr.

By default the Report Builder shows a select number of report bands: Report Header, Column Header, Detail, Column Footer, and Report Footer. The Report Builder doesn't show the following by default: Page Header, Page Footer, Watermark, and those bands created when using the “Group Management” option (explained later).

The bands displayed represent everything that your report can contain. Some bands, like the Report Header, appear only once – in this case at the top of the very first page of the report. Other bands, most importantly the Detail band, will be iterated over a number of times, depending on your query data, grouping, and number of pages.

Building a Query for Your Report in the Query Builder

To open the Query Builder, select Report > Report Query from the menu bar, or click the Report Query button on the menu bar. The Query Builder appears. Now select a data source:

1. Make sure that you have selected the

appropriate RDS server, choosing an RDS server from the pop-up list box above the tree control. The Report Builder displays a list of data sources defined for the associated ColdFusion server.

2. Select a data source by clicking it or by clicking the plus (+) sign next to the data source to expand the list.
3. Expand the data source tree.
4. To view the database schema, click the plus sign (+) next to Tables.
5. To view the column definitions, click the plus sign next to a table name.

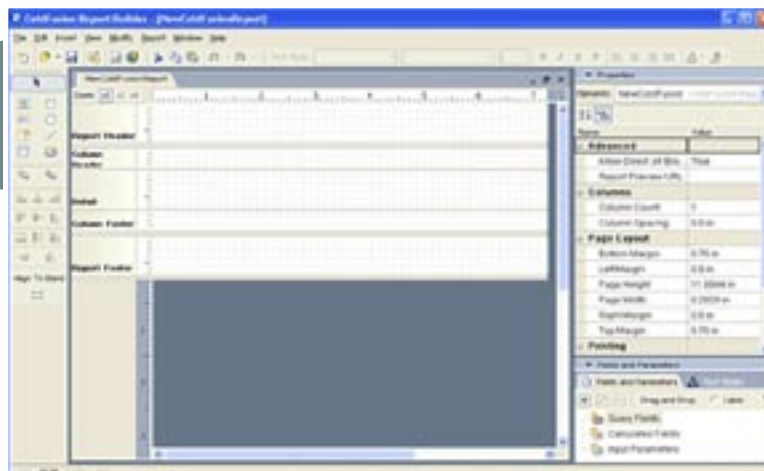
Build an SQL statement using the Table pane and the Properties panel:

1. Select an RDS server.
2. Expand a data source.
3. Drag one or more tables from the Table pane to the Data Source pane. Alternatively, right-click a table and select Add to Work Area. You can also select a table by double-clicking it or by right-clicking on the Table pane.
4. To remove a table from the Table pane, right-click it and select Remove Table.
5. (Optional) If you select more than one table, you must specify the column or columns used to join them by dragging a column from one table to the related column in the second table.
6. Double-click the columns for the SELECT statement.

As you select columns, the Query Builder creates the SELECT statement in the area in the bottom half of the pane:

1. (Optional) Specify the sort order and selection criteria.
2. Review the SELECT statement that displays in the SQL pane, and use the

figure 1





- Table and Properties panes to make adjustments, if necessary.
- (Optional) Click Test Query.
- Click Save.

The Query Builder saves the SQL statement as part of the report and defines query fields for columns returned by the result set. See the ColdFusion LiveDocs on Report Builder for more information. For the example in this tutorial, the Query Builder should look like Figure 2.

The report samples use “text styles” to format elements that display fonts. A text style is similar to a font style in Microsoft Word. Instead of explicitly associating a text element with font specifications, you associate the element with a text style.

Formatting Your Report

To begin formatting your report, take a look at Report Builder toolbox. The toolbox contains everything you need to draw, type, place, and embed everything you need in the report.

Formatting the Report Header

Use the following steps to create a Report Header from the Report Builder toolbox:

1. Click the lightning bolt button. This is the field control, which allows for the display of textual expressions to be rendered in your report. Expressions, as you’ll see, can contain any and all of the following: static text, dynamic text, calculated text, and concatenated text.

2. Use your cursor to resize the Report Header. A dialog box appears and asks you to select one of the following choices: pre-existing values in the report (such as data from the query you built in the Query Builder), or a manually entered expression. Select manually entered expression; and for the expression enter the following:

```
“Report Format is “ & param.outputFormatLabel
```

This expression concatenates static text with a predefined report parameter – named outputFormatLabel, specified in the cfreportparam tag, embedded in the cfreport tag – which ultimately invokes our report (covered later). So as we explained earlier, anything in the Report Header appears once on the first page of your report. In this instance, it will print out something like “Report Format is FlashPaper.”

3. Because you want the text to appear in the upper left of the Report Header while the text field is selected (denoted by the handles that appear on the rectangle), click the Align to Band button in the Align palette, click the Align Left Edges button, and then the Align Top Edges button. These steps move the text to the top-left band while respecting the outermost boundaries of the band.
4. To resize or move the band while you have selected it, click and drag the entire element, use your keyboard arrows, click and drag the element’s

handles, or alter the values for this element in the properties panel under “Layout” – values such as Height, Width, Left, Top, and so forth. For every new type of element you use in your report, note the related properties of that element in the Properties panel; this will help you be aware of every configurable property available to you.

5. Supply a static text report title. In the Toolbox, click the abc button (when you place your cursor over it, “Label” appears). Draw a rectangle the width of the entire Report Header band. Notice that when you release the cursor, the Report Builder doesn’t ask you to enter an expression, but to enter label text. Enter the following:

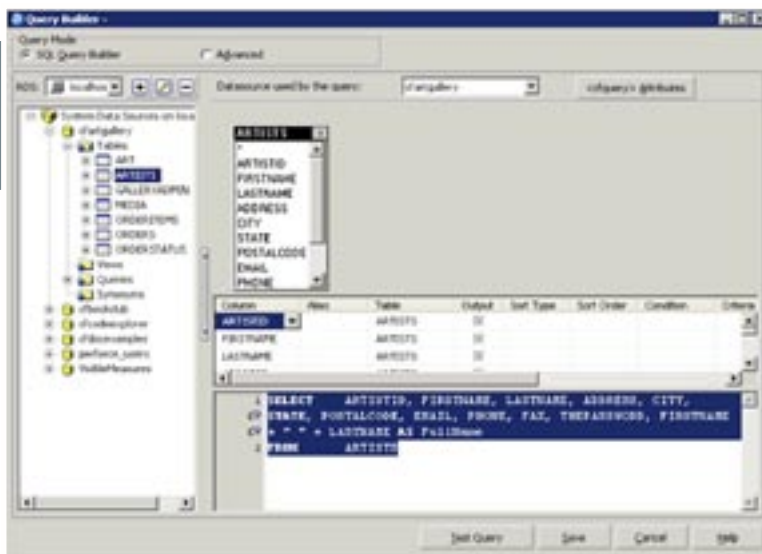
```
Works of Art on Consignment (grouped by Artist)
```

Notice that with any text, you can align the text box surrounding it and also align the text within the text box by selecting Properties Panel > Format > Alignment, or using the Text and Styles toolbar. Additionally, the dynamic and static fields appear differently in the report. Fields have a solid line on the text box perimeter. Labels only have bracket in its corners.

6. You’re almost done with the report header, but it has no color. Click the “Rectangle” button in the Toolbox and drag it much like a text control. Adjust the control points so that the rectangle occupies the entire width and height of the header band. While selecting the rectangle, set the background color to any color of your choice by selecting Properties Panel > Colors & Style > Background Color, or by using the Background Color paint bucket in the Text Styles panel.
7. Notice that the rectangle in the report covers the two elements you put in the report header – this is referred to as the z-order. You’ve essentially stacked your rectangle on top of the other elements. Right-click your rectangle and choose Send to Back. As with most every command in the Report Builder, there are at least one or two other ways to do the same (keyboard shortcuts, panels, toolboxes, right-click menus).

You’re done with the header!

figure 2



Subscribe Today!

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



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 i-Technology Publisher



Formatting the Page Header

Enter today's date in the Page Header.

Conveniently, this ColdFusion report processes much like a ColdFusion page itself, and has access to all variables accessible to the page that contains the cfreport tag invoking the report (that is, unless a user browses to the report definition directly), and also has access to all ColdFusion functions.

To explore functionality available to your report, drag a text field (dynamic – remember lightning bolt) onto the Page Header and enter the following expression:

```
DateFormat(Now(), "d mmmm yyyy")
```

You've just used a widely known ColdFusion function, Now(), and formatted it with another function, DateFormat(). Notice that as you entered the expression, the Expression Builder was insightful and prompted you with code completion options. Take advantage of this feature or simply continue typing.

Formatting the Detail Band

Now onto formatting the core of the report, the Detail Band. This band prints in your report for every row of data returned from your report query (the query that you specified in your report through the cfreport tag or that you embedded in the report itself, through the Query Builder).

See if you can mimic the layout described below in your detail band, knowing what you know about fields, labels, layout, and expressions. If you're not terribly knowledgeable with CFML, don't work on this until you're more familiar with the functions below. Note that items in parentheses are instructional; do not include them in the report. Give it a try:

```

----- Detail Band Start -----
(Label) Artist Contact Info
(Label) Name: (Field) query.FIRSTNAME
& " " & query.LASTNAME
(Label) Address: (Field) report.
FormatAddress(" ", query.ADDRESS,
" ", query.CITY, query.STATE, query.
POSTALCODE)
(Label) Email: (Field) query.EMAIL
(Label) Phone: (Field) report.
FormatPhone(query.PHONE, " ", " ")
(Label) Fax: (Field) report.
FormatPhone(query.FAX, " ", " ")
(Label) "See all artwork by " &

```

```

query.FIRSTNAME
----- Detail Band End -----

```

Note: To display the images in the reports you must specify the path for the image files dynamically. In this case, I have used Application.cfm to define an application-level variable called application.sImageLocation. In the report you named works-of-art-by-artist.cfr, the field element under the image label has the following expression: application.sImageLocation & "/" & query.LARGEIMAGE. Because the table stores the image filename, concatenating the path to the image helps your report dynamically find the image files.

Adding a Subreport

You have one more element to add to the Detail Band – another report! This other report serves as a subreport to the main report. Subreporting, conceptually, is much like how a ColdFusion developer would group query data, with nested cfoutput tags.

Use subreports to nest a report within another report. The data you display in a subreport is typically related to the data in the main report. You invoke a subreport by passing one or more subreport parameters from the main report to the subreport.

However, the data displayed in a subreport can also be unrelated to the data in the main report. Other common ways to refer to this relationship between these reports are master/detail or parent/child:

1. On the toolbar, select the Subreport tool and draw an area on the Detail Band.
2. When the Subreport Wizard prompts you, choose the From an Existing Report option.
3. Browse to the file called consignment-report.cfr.
4. The Subreport Wizard asks you to specify the bind parameters for the main report to the subreport. For this example choose #query.ARTISTID#.
5. Click Finished.

See documentation on subreports in the ColdFusion MX LiveDocs for more information.

Adding a Page Footer

In the Page Footer, place a Field control, entering within it the self-explanatory variable:

```
calc.PAGE_NUMBER
```

The calc.PAGE_NUMBER variable is one of a few variables available to any report. Other variables are:

- calc.PAGE_NUMBER
- calc.COLUMN_NUMBER
- calc.REPORT_COUNT
- calc.COLUMN_COUNT

Check out the Report Builder documentation in "Built-in Calculated Fields and Input Parameters" in ColdFusion MX 7.

LiveDocs

Browsing Your Report

The ColdFusion code needed to render this report is minimal because you've already embedded your query in the report definition. Save this ColdFusion code in your webroot (where your CFR file is located). Save it as FirstReport.cfm:

```

<cfreport template="FirstReport.cfr"
format="flashpaper">
    <cfreportparam
name="outputFormatLabel"
value="flashpaper">
    <cfreportparam name="reportTemplate" value="FirstReport">
</cfreport>

```

Browse to FirstReport.cfm and the FlashPaper version of your report appears. Go ahead and change the cfreport tag attribute value for format from flashpaper to pdf to excel.

Congratulations! You've generated your first ColdFusion MX 7 report. If you think that was fun, try the Report Wizard next!

Source Code and Files Necessary for This Article

ColdFusion MX 7

- Try
- Buy

Tutorials and sample files:

- reporting.zip (ZIP, 36K)

Note: This ZIP file contains the sample files to build your first report. If you installed ColdFusion MX 7 without the sample applications, run the installer again and select the Sample Applications option during the installation process. ☺

Collin Tobin joined Macromedia in 2001. He is currently a principal quality assurance engineer for ColdFusion. Previously he worked at Gtriad as a Web developer. ctobin@macromedia.com

Dean Harmon has been with Macromedia since 2001. Originally he was on the Visual Tools team developing HomeSite before transitioning to ColdFusion. He enjoys creating and playing video games and spending time with his wife and three children. dharmon@macromedia.com

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

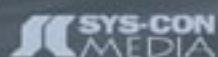


ADVERTISE

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

SUBSCRIBE

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



MX
developer's journal





INTRODUCING...

COLDFUSION MX 7

by ben forta

AFTER AN ENTIRE YEAR SPENT MEETING WITH AND SPEAKING TO THOUSANDS OF COLDFUSION DEVELOPERS, THE CF TEAM AT MACROMEDIA ARE UNLEASHING THIS MONTH THE FEATURE-RICH NEW RELEASE, CFMX 7.

IT'S BEEN THREE YEARS SINCE WE RELEASED MACROMEDIA COLDFUSION MX, THE MOST DRAMATIC AND AMBITIOUS COLDFUSION UPDATE EVER. COLDFUSION MX MARKED AN IMPORTANT MILESTONE IN THE COLDFUSION STORY. IT WAS A CHANCE FOR THE TEAM TO TAKE A BIG STEP BACK AND REBUILD COLDFUSION FROM THE GROUND UP, TAKING INTO ACCOUNT EVERYTHING WE HAD LEARNED ABOUT WEB APPLICATIONS AND HOW THEY ARE BUILT.

ColdFusion MX was primarily an architectural release. It featured things like the following:

- A brand-new, Java-based engine
- A true compiler
- Deployment on top of industry standard J2EE servers
- Better support for XML, SOAP, and other standards
- Access to the world of Java

Of course, ColdFusion MX (and ColdFusion MX 6.1) also boasted important new features, language enhancements, improved performance, as well as greater scalability and reliability. But at its core, ColdFusion MX was all about architecture, an incredible investment in the inner workings of ColdFusion so as to facilitate a world of new functionality. ColdFusion MX has been an incredibly successful product, and a large portion of the ColdFusion user base is already taking advantage of all it has to offer. And so with ColdFusion's new engine proving its mettle and developers busily exploring the opportunities it presents, the ColdFusion team was able to spend time building new features and functionality that were not possible in the past.

For over a year we met with and spoke to thousands of ColdFusion developers. We presented ideas and previews to hundreds of user groups worldwide, brainstormed with countless partners and customers, waded through mountains of wish-list feedback, and chatted with numerous users (both current and potential). When the dust settled, a series of goals emerged:

- Make new users far more successful. ColdFusion has always appealed to new developers. There is no other language or product as well suited to their needs as ColdFusion. New users (primarily those with a background in building Web pages and static sites) are an important part of the ColdFusion user base, and ColdFusion must remain dedicated to making successful development easier for these users. This involves the creation of Dreamweaver extensions and configuration screens, providing better out-of-the-box education, delivering more usable value, and more.
- Provide existing users with feature

and functionality that they can use immediately. Developers are never shy about what they want. We need to deliver the features and functionality they ask for.

- Help developers (our users) make their users happier. Consumers of ColdFusion applications have common requests – things they'd like to see in the applications created for them. Many of these requests revolve around how the application captures and presents data. You said that ColdFusion must provide powerful new capabilities for forms, reporting, and printing.
- Improve reliability and deployment options. ColdFusion's Java internals opened up all sorts of powerful and important deployment and reliability options. Now ColdFusion needs to make this more available to you and your applications, more than ever before.
- Innovate, innovate, innovate. ColdFusion pioneered rapid development on the Web. Indeed, there still is no quicker way to build Web-based applications. The ColdFusion development experience needs to be applied to new platforms and technologies, making them just as readily usable.

These are significant, even lofty, goals. We invested tens of thousands of development hours, launched the largest beta test program to date, and maintained regular contact with our customers so that they could keep us focused and on track.

The result is the most customer-driven ColdFusion version ever, a feature-rich release that solves real problems for real developers building real applications, a product that meets and exceeds the enumerated goals.

And so I'd like to take this opportunity to formally introduce you to ColdFusion MX 7.

Improved Form Field Validation Data entry forms have long been the Achilles heel of Web-based applications. Without bad-mouthing HTML forms (actually, there is little need to – you've all experienced the pain firsthand), ColdFusion MX 7 improves forms in several ways, starting with improved form field validation.

For starters, ColdFusion contains

additional validation types, including the oft-requested validations for e-mail and URLs. In addition, the JavaScript error message that appears when using client-side validation displays all of the validation errors at once, not just the first validation error.

Perhaps more importantly, it is now simpler to perform both client-side and server-side validation at once. The `cfinput` tag has a new attribute named `validateAt` that accepts three values:

- `onSubmit` (the default value) specifies client-side validation when the client submits the form, just like client-side validation in the current `cfinput` tag.
- `onBlur` specifies client-side validation as soon as a field loses focus (user tabs to the next field or clicks another field, for example).
- `onServer` specifies server-side validation, the same type of validation that hidden form fields perform. However, it does not require you actually to define those fields (the fields are still present, but are generated and embedded automatically).

Look at the following code:

```
<!-- Client-side validation on submit -->
<cfinput type="text"
         name="quantity"
         validate="integer"
         validateAt="onSubmit"
         required="yes"
         message="Numeric quantity is required!">

<!-- Client-side validation on loss of focus -->
<cfinput type="text"
         name="quantity"
         validate="integer"
         validateAt="onBlur"
         required="yes"
         message="Numeric quantity is required!">

<!-- Server-side validation -->
<cfinput type="text"
         name="quantity"
         validate="integer"
         validateAt="onServer"
         required="yes"
         message="Numeric quantity is required!">
```

This article can also be read at the Macromedia Developer Center site (www.macromedia.com/devnet/)



All three `cfinput` tags perform the same validation but at different points within the form submission process. The best part is that validation methods may be mixed. So to validate on the client and server you could do the following:

```
<!-- Client-side validation on submit -->
<cfinput type="text"
         name="quantity"
         validate="integer"
         validateAt="onSubmit,onServer"
         required="yes"
         message="Numeric quantity is required!">
```

Here the `validateAt` attribute specifies two values (`onSubmit` and `onServer`) so that ColdFusion generates client-side validation code and embeds hidden form fields for server-side validation.

Another validation enhancement is input masking. ColdFusion MX 7 introduces a new attribute to solve this problem: `mask` takes an input mask and uses it as a data input filter. A mask is a string comprised of special characters that are used to validate data entry: A question mark (?) allows any character, the letter A allows only alphabetical characters, the number 9 allows digits, and an X allows alphanumeric characters. Any other character is a literal and is itself embedded in the input. For example, to validate a three-digit age, you could do the following:

```
<cfinput type="text"
         name="age"
         maxLength="3"
         mask="999">
```

The mask filter "999" would only accept digits. If a user entered anything other than a digit, the tag would simply ignore that input. Similarly, to validate a US-style phone number in the format (123) 456-7890, you could use the following code:

```
<cfinput type="text"
         name="phone"
         maxLength="13"
         mask="(999) 999-9999">
```

Again, the mask attribute value allows

only digits, but inserts the other characters automatically.

For a Canadian postal code you could use the following:

```
<cfinput type="text"
         name="postalcode"
         maxLength="7"
         mask="A9A 9A9">
```

You get the idea. Input masking does not negate the need for input validation, but it does make for a far better user experience.

Flash Forms

Another important enhancement to forms is less of an enhancement and more of a drop in replacement. Macromedia Flash has long been a potential replacement for HTML forms, enabling developers to leverage the capable Flash Player to deliver a better user experience. Of course, that has meant learning Flash or deploying Macromedia Flex. For ColdFusion developers who simply want better forms, there needs to be a simpler solution. To make the creation of Flash-based forms easier for coders, ColdFusion MX 7 introduces a series of tags that make building powerful and sophisticated data-entry forms as simple as, well, ColdFusion. For example, if you need to prompt a user for a date (perhaps a date of birth), you can replace the HTML code:

```
<form action="" ...>
<input type="text" name="dob">
...
</form>
with the following:
<cfform format="flash" action="" ...>
<cfinput type="datefield" name="dob">
...
</cfform>
```

This creates a form with a text field, just like the HTML text field, except that this one displays a pop-up date chooser when a user selects the field. It's that simple. Using a combination of `cfinput` tags to create controls, and `cfform`-group tags to group them as needed, ColdFusion developers can generate Flash forms without knowing (or even owning) Flash.

A pop-up calendar is just the start of it. Other features include the following:

- Tree control
- Data grid
- Multipane forms (using tab or an accordion-style interface)
- Input masking
- Data binding between form controls
- Client-side events
- Integrated error checking and validation feedback

How does this all work? When ColdFusion processes a page containing these tags, it generates the Flash ActionScript needed to create the form and then compiles that code into a SWF file and embeds it in the page. All of that is hidden from ColdFusion developers who simply use CFML tags, just as it is already.

In other words, you can create forms that leverage the power of Flash while retaining the productivity and simplicity that has become the hallmark of ColdFusion. Stay posted to the ColdFusion Developer Center, as it will feature an article from the ColdFusion engineers who worked on Flash forms.

Printable Web Pages

Considering that most ColdFusion development involves applications that search and display data, it should come as no surprise that printing (or the inability to do so easily) has long been a source of aggravation for ColdFusion developers. After all, if you were to generate a Web page and then select File > Print in your browser, well, you'd never really know what to expect the printed output to look like.

Developers have had to resort to all sorts of tricks to control printed output generation, with varying degrees of success – until now.

ColdFusion MX 7 introduces a new tag (a family of tags, actually) that makes turning Web pages into printable content as painless as you'd expect from CFML. Look at the following code snippet:

```
<cfdocument format="pdf">
Here is some text.<br>

</cfdocument>
```

The `cfdocument` tag takes whatever

code you provide and generates printable documents in Adobe PDF and Macromedia FlashPaper formats. In this example, the PDF would contain a single line of text followed by an image on the next line. That's all it takes.

The cfdocument tag is designed to work with any Web pages. There is no need for XHTML or specific formatting. You can use inline formatting or CSS, you can embed images and links, you can use tables and <p> tags for alignment – it'll just work.

In addition, cfdocument tag supports the following, among other things:

- Page orientation of portrait or landscape
- Different (and custom) page sizes
- Headers and footers
- Different formatting options (page size, headers, and footers) for different sections of a document
- Saving generated output to disk

If you have Web pages that you'd like to print – just about any pages – one simple set of tags will solve the problem for you quickly and efficiently.

Reporting

In addition to free-form page printing, ColdFusion developers often have another printing-related need: structured reporting. Third-party reporting tools have long been difficult to integrate with ColdFusion applications. ColdFusion MX 7 introduces its own reporting solution, a Report Writer and a reporting engine.

ColdFusion MX 7 introduces a new file type, the CFR (ColdFusion Report) file. CFR files are report templates you create with the new ColdFusion Report Writer, which looks a lot like other report-building tools you may have used. Once you have created a report, you can embed it in your applications using a cfreport tag, like this:

```
<cfquery datasource="mysdn"
name="myQuery">
SELECT * FROM myTable
</cfquery>
<cfreport format="FlashPaper"
template="myReport.cfr"
query="myQuery">
```

As you can see, queries are passed to the cfreport tag at runtime. This

Advertiser	URL	Phone	Page
Blog-n-play.com	www.blog-n-play.com	888-303-5252	57
CFDynamics	www.cfdynamics.com	866-233-9626	5
EV1 Servers	www.ev1servers.net	800-504-SURF	6
FlashForward 2005	www.flashforwardconference.com	877-4FLASH4	31
HostMySite.com	www.hostmysite.com/mxdj	877-248-4678	59
Interakt	http://ktml.interaktonline.com/		11
Intermedia.net	www.intermedia.net	800-379-7729	39
ISSJ	www.ISSJournal.com	888-303-5282	45
IT Solutions Guide	www.sys-con.com	201-802-3021	41
Macromedia	www.macromedia.com/go/volvo	415-252-2000	2-3
Macromedia	www.webpublishingsystem.com	415-252-2000	60
MX Developer's Journal	www.sys-con.com/mx/subscription.cfm	888-303-5282	47
Nidus Corp.	www.brainstormer.org	888-894-3840	17
PaperThin	www.paperthin.com	800-940-3087	33
Seapine Software	www.seapine.com	888-683-6456	21
SYS-CON Newsletters	www.sys-con.com	888-303-5282	23
SYS-CON Reprints	www.sys-con.com	201-802-3026	53

means that a CFR file is actually more of a report template than an actual report, but you can use it to construct any SQL – dynamically if required. You have complete flexibility and control over how your ColdFusion application creates reports.

The ColdFusion reporting features the following:

- Reports that can be generated in PDF, FlashPaper, and Excel formats
- Intuitive report creation tool featuring report bands, data aggregation functions, embedding charting, and more
- Toolbars used to insert images, text, lines, labels, as well as to manage data alignment
- Integrated SQL query builder
- Support for subreports (reports embedded within other reports)
- Report creation wizard
- Charting wizard
- Full zoom and preview support

ColdFusion reporting is available in all editions of ColdFusion, and ColdFusion Enterprise provides greater control so as to manage report creation in high-usage

environments. Read more about reporting in the other CFMX 7 article in this issue, Collin Tobin & Dean Harmon's article, "Building Reports in ColdFusion MX 7."

Event Gateways

ColdFusion has long been an incredibly simple way to build Web applications. That's why we all use ColdFusion – for its simplicity (and therefore productivity). But ColdFusion is not exclusively tied to the Web. In fact, ColdFusion does not even talk to Web browsers; that is the Web server's job. ColdFusion simply executes scripts on the server in response to requests – requests which (thus far) have been originated typically through HTTP.

Could ColdFusion respond to other requests – for example, data sent to a specific port, or changes in a folder, or inbound SMS and IM messages, or database table changes, or...? The answer is yes: ColdFusion can respond to any and all of those; it just needs a way to know when those events occur. It needs gateways.

Gateways are interfaces to other



systems, ways for events to trigger ColdFusion processing. A gateway watching a folder on a server can trigger ColdFusion execution when folder contents change. A gateway connecting to an SMS provider can respond to inbound SMS messages (and send SMS messages as well). A database trigger can ping a gateway so that a database event forces ColdFusion processing (imagine being able to generate static HTML pages automatically and dynamically whenever back-end databases change).

Among the gateways included with ColdFusion MX 7 are the following:

- Asynchronous processing gateway
- Folder watcher gateway
- Socket gateway
- JMS gateway
- Lotus Sametime gateway
- XMPP protocol gateway
- SMS gateway

Third-party vendors are already hard at work creating ColdFusion MX 7 gateways. You could write your own gateways, too. The result is that ColdFusion is now able to interact with just about any back-end and technology available to you. You will be able to read more about event gateways and what you can do with them in Jim Schley and Tom Jordhal's article next issue, "Writing and Using Event Gateways in ColdFusion 7."

Improved Deployment

ColdFusion (as of ColdFusion MX) is a Sun-verified Java application, and is installed on top of J2EE servers like any other Java applications. Well, kind of. You can deploy ColdFusion MX (including ColdFusion MX 6.1) on top of a J2EE server but the complete process is one that does not excite J2EE administrators.

In J2EE-land, administrators are typically given an application to deploy, and they don't pay a whole lot of attention to what that application is and how it works. Nor should they. Developers should worry about applications and J2EE administrators should worry about servers staying up and running well.

How does this work? Applications to be deployed on a J2EE server are packaged as a single file, a Java archive file (usually with an EAR or WAR extension). The archive file contains everything

needed for an application to run: source code, configuration settings, supporting files – everything. Once an application has been tested and is ready for deployment, it is packaged—the package itself being test-deployed – and handed off to the J2EE administrator, who drops it onto the J2EE server. (I am simplifying things a bit, but the basic flow is accurate.) What J2EE administrators don't do, or don't like doing, is running through a post-installation to-do list containing things like creating a data source, setting up some mappings, installing these extensions, and so on.

Yet J2EE administrators deploying ColdFusion MX must do just that. You can deploy ColdFusion itself – core engine, compiler, and runtime services – like any other Java application, but that is just ColdFusion itself. Once you deploy ColdFusion, someone still needs to move all the CFML and CFC files over and use the ColdFusion Administrator to define data sources and mappings and more. In other words, while ColdFusion itself is deployed like any other J2EE application, the total experience of deploying a ColdFusion application is not.

ColdFusion MX 7 changes this by giving you the ability to build complete J2EE deployment packages. It comes with a packaging tool that creates a complete EAR or WAR file that can contain the ColdFusion runtime (with or without specific features), application code, data sources, and more. The tool can take some time to run because building a complete, deployable EAR or WAR file is not a quick process. When it's finished, however, you can give that package to a J2EE administrator to deploy just like any other Java application. This means you can deploy ColdFusion applications on a J2EE server that is not running ColdFusion because your Java package file will contain the ColdFusion engine.

This is an important, and much-needed enhancement. From a J2EE administrator's perspective, deploying ColdFusion MX 7 applications will be just like deploying any Java applications. Actually, J2EE administrators don't even have to know that it is a ColdFusion application. To them, it's Java pure and simple.

Multiple Instances Made Easier

ColdFusion MX Enterprise supports multiple instances of ColdFusion – although that is less a ColdFusion feature and more a J2EE-server feature that ColdFusion users can take advantage of. After all, ColdFusion MX is a Java application. I've discussed the benefits and importance of using multiple instances before but, simply put, using multiple ColdFusion instances provides greater security, stability, and scalability. It's almost like having ColdFusion installed on multiple physical servers, except it's all on one server.

If you have an existing J2EE server, you can create multiple EAR or WAR files and then deploy them as you would any other Java application. If you do not have an existing J2EE server, the ColdFusion installer can install JRun 4 for you. In doing so, it creates and deploys the first ColdFusion instance so that you can be up and running immediately. But when you want to deploy additional instances, things get a little tricky for users without experience in J2EE server administration. You need to use the J2EE server administration tools to create a new server, run the ColdFusion installer to create the EAR or WAR, expand the files (if using JRun), make tweaks to an XML file, and then copy the expanded files into the server folders. This is all feasible, but not exactly a trivial process. Unfortunately, this is why so many users have yet to deploy multiple instances.

ColdFusion MX 7 make this process much simpler. It has the same three installation options as ColdFusion MX 6.1, but selecting the JRun + ColdFusion option in ColdFusion MX 7 installs additional administration screens that make the deployment of instances (and even the creation of clusters of instances) as simple as any other ColdFusion administration processes. You'll be able simply to fill in a form and click a button to create a new instance – without needing to use the JRun management tools or the ColdFusion installer, without needing any XML tweaks, and without even knowing what an EAR or WAR file is.

How could you use this new functionality? Consider these use cases:

- You are about to deploy an update to your application code and you need

to maintain the existing application as a fallback, just in case something goes wrong. Simply create a new instance (you could even create a CAR file using the old instance to save data sources and any other needed configuration, launch the ColdFusion Administrator for the new instance, and import the CAR file to import those settings), copy the code, associate your web server to the new instance, stop the old instance (to prevent resources from being used unnecessarily), and you are done. If you then need to roll back, start the old instance and associate your web server to it again. Clean and simple.

- You have an existing application that spikes in load (holiday shoppers maybe), and you want an additional server running the same application so that you can handle a greater load, and also provide failover in the event that a server problem occurs. Simply create a new instance, point to the Java package containing the code and settings used for the first instance, and let ColdFusion do its thing. You'll have a second instance

created, configured like the first, and containing the same application as the first. You can then use a second screen to create a cluster – perhaps to enable session sharing between the instances.

You get the idea

Of course, for those who want more control, the installer will still install JRun with its own management software just as it does now – and you can deploy and manage applications just as you can now. But for those of you who simply want to leverage what is undoubtedly the most significant benefit of ColdFusion Enterprise over ColdFusion Standard, ColdFusion MX 7 makes life much simpler. Stay posted to the ColdFusion Developer Center; it will feature an article from the ColdFusion engineers who worked on the Enterprise Manager.

Lots of Other Goodies

This is just the start of it. Other ColdFusion MX 7 goodies include the following:

- XForms support
- Administration API Improved Web ser-

vices support

- Dreamweaver extensions
- Simplified authentication against NT domains and Active Directory

ColdFusion MX is the most eagerly anticipated version of ColdFusion in a long time. This is ColdFusion designed by developers, for developers; the ColdFusion created in response to vast user feedback; and thus the ColdFusion that solves real problems right out of the box.

If you are using an older version of ColdFusion, now is the time to upgrade. If you have yet to experience ColdFusion development, there has never been a better or more exciting time to start than now. Check out the upgrade options for ColdFusion MX 7. ☺

Ben Forta is the Macromedia senior product evangelist and the author of numerous books, including ColdFusion Web Application Construction Kit and its sequel Advanced ColdFusion Application Development, as well as books on SQL, JavaServer Pages, WAP, Windows development, and more. ben@forta.com

ONCE YOU'RE IN IT...

ONCE YOU'RE IN IT...



- WebServices
- Storage & Security
- IT Solutions Guide
- PowerBuilder
- ColdFusion
- WebSphere
- LinuxWorld
- Wireless
- WLDJ
- .NET
- JDJ
- MX

...REPRINT IT



...REPRINT IT

Contact Kristin Kuhnle
201.802.3026
kristin@sys-con.com



RePrints

Deploying Applications with ColdFusion MX7

Introducing sourceless and EAR/WAR file deployment
by dave carabetta

The release of ColdFusion MX 7 provides developers with several options for deploying their CFML applications. While the current option of using ColdFusion Archives (CAR files) has worked to this point, a fundamental problem still remains in that the source code needed to be included.

However, the two new options available in this release, sourceless and EAR/WAR file deployment, truly give developers the flexibility to deploy their code while optionally preserving their valuable intellectual property. (It is important to note that New Atlanta's BlueDragon Server line has supported sourceless deployment of CFML source

code since the introduction of their 6.1 product line in June 2004. All references to sourceless and EAR/WAR deployments as "new features" apply solely to Macromedia ColdFusion MX 7.) Coupled with the streamlined ability to add new ColdFusion instances to an environment via the ColdFusion MX Administrator (Enterprise edition only), much of the manual work associated with application deployment has been automated, allowing the developer to focus on development instead.

Precompiled and Sourceless Deployment

Before ColdFusion MX 7, the only

option for companies to protect their commercial CFML code distribution was the cfencode utility. However, anybody who has worked with ColdFusion for any reasonable amount of time knows that a simple Internet search will turn up a utility for decompiling the templates, which essentially rendered cfencode pointless. So for many years, all that developers could realistically do was release their code to the public and simply hope that the end user would abide by the license agreement. Further, many companies, not just those selling commercial applications, simply weren't comfortable running production applications without some sort of source code protection.

ColdFusion MX 7 ships with a new cfcompile utility that has two main uses. First, it can be used to simply compile your ColdFusion source code into the resulting Java class files so that the initial load time is quicker. Second, it allows for precompiling CFML code (CFM, CFC, and CFR templates) to Java byte code, which can then be deployed without the source templates. This utility, which can be found in the cfroot/bin (for the server configuration) and in the cf_webapp_root/WEB-INF/cfusion/bin (for the multi-server and J2EE configuration) directory, can be run on one template (i.e., a custom tag) or an entire directory structure containing a complete CFML application (i.e., a blogging application).

In both cases, its usage is incredibly simple. To precompile CFM, CFC, and CFR templates (note that this is not for sourceless deployment, which will be covered shortly), the general usage is

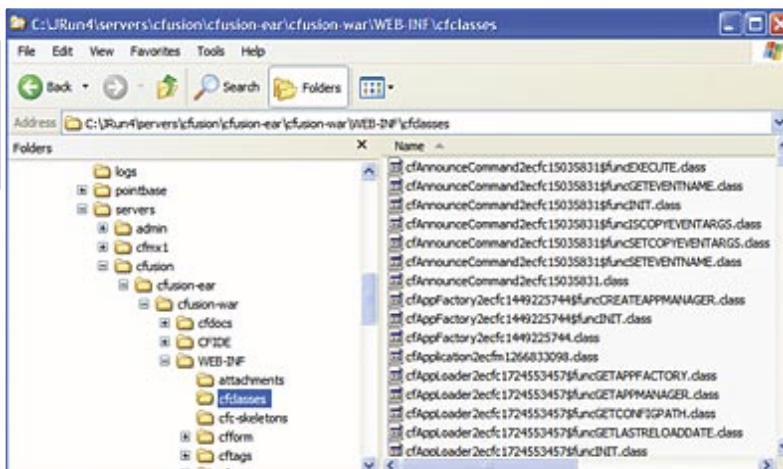
```
cfcompile webroot [directory-to-compile]
```

where webroot is the absolute path to

figure 1

```
C:\WINDOWS\system32\cmd.exe
C:\inetpub\wwwroot\MachII\framework\commands\EventMappingCommand.cfc...
C:\inetpub\wwwroot\MachII\framework\commands\FilterCommand.cfc...
C:\inetpub\wwwroot\MachII\framework\commands\NotifyCommand.cfc...
C:\inetpub\wwwroot\MachII\framework\event\EventCommand.cfc...
C:\inetpub\wwwroot\MachII\framework\event\EventContext.cfc...
C:\inetpub\wwwroot\MachII\framework\event\EventHandler.cfc...
C:\inetpub\wwwroot\MachII\framework\event\FilterManager.cfc...
C:\inetpub\wwwroot\MachII\framework\event\Invoker.cfc...
C:\inetpub\wwwroot\MachII\framework\event\InvokerEvent.cfc...
C:\inetpub\wwwroot\MachII\framework\event\Listener.cfc...
C:\inetpub\wwwroot\MachII\framework\event\ListenerInvoker.cfc...
C:\inetpub\wwwroot\MachII\framework\event\ListenerManager.cfc...
C:\inetpub\wwwroot\MachII\framework\event\Plugin.cfc...
C:\inetpub\wwwroot\MachII\framework\event\PluginManager.cfc...
C:\inetpub\wwwroot\MachII\framework\event\PropertyManager.cfc...
C:\inetpub\wwwroot\MachII\framework\event\RequestHandler.cfc...
C:\inetpub\wwwroot\MachII\framework\event\ViewContext.cfc...
C:\inetpub\wwwroot\MachII\framework\event\ViewManager.cfc...
C:\inetpub\wwwroot\MachII\framework\event\Index.cfm...
C:\inetpub\wwwroot\MachII\framework\event\ii.cfm...
C:\inetpub\wwwroot\MachII\framework\event\SimplePlugin.cfc...
C:\inetpub\wwwroot\MachII\framework\event\Util\BeanUtil.cfc...
C:\inetpub\wwwroot\MachII\framework\event\Util\Exception.cfc...
C:\inetpub\wwwroot\MachII\framework\event\Util\Queue.cfc...
total 40
elapsed 5.36 sec
C:\inetpub\wwwroot\MachII\framework\event\Util\Queue.cfc... successful 40
```

figure 2



the site's Web root and the directory-to-compile is an optional parameter specifying the directory to compile (Note: all subdirectories are automatically compiled as well). If no directory is specified, the utility will compile all CFM, CFC, and CFR templates from the Web root, recursively. This usage of cfcompile is the same process that the ColdFusion runtime uses when first loading a page: it parses the file, compiles it, and stores the appropriate class files in the cfclasses directory under your ColdFusion installation. For a small set of files, this might not seem very useful due to the massive compiler optimizations that the ColdFusion MX runtime has received since its first release. However, for large sites, this initial overhead savings could prove valuable.

As an example, the Mach-II framework files are used as the directory structure (the freely available code is located at www.mach-ii.com/code.cfm). Assuming that the Web root is C:\inetpub\wwwroot, the following command-line statement would precompile all the templates in the framework:

```
cfcompile C:/inetpub/wwwroot C:/inetpub/wwwroot/MachII
```

As seen in Figure 1, the forty templates comprising the base framework were compiled in 5.36 seconds. Figure 2 shows the resulting class files in Windows Explorer. A count of the number of resulting class files output shows that 345 files were generated for a total directory size of just over 1 MB (remember, the number of compiled class files is not one-to-one with the number of ColdFusion templates. ColdFusion Components, for example, have their methods broken out into individual class files). Running this utility has just saved the ColdFusion runtime from having to generate over 300 files when they are first loaded!

The second use of this utility lies in its ability to precompile templates for sourceless deployment. What this means is that applications can now be distributed without the underlying source code and without the fear that a vendor's intellectual property will be compromised.

The general usage is:

```
cfcompile -deploy webroot directory-to-compile output-directory
```

Once again, using the Mach-II core file distribution, the sourceless deployment syntax is:

```
cfcompile -deploy C:/inetpub/wwwroot C:/inetpub/wwwroot/MachII C:/inetpub/wwwroot/MachII_Sourceless
```

The major usage difference is the presence of the "-deploy" switch that tells the cfcompile utility to compile the CFML source code to Java byte code. Further, the output-directory is required since the resulting files will still have the same name and file extension as the source files. Once this process is done, it is simply a matter of backing up the original CFML source code and copying the generated byte code templates into the original directory. That's it! You now have a copy of the Mach-II framework that can be run without the original source code!

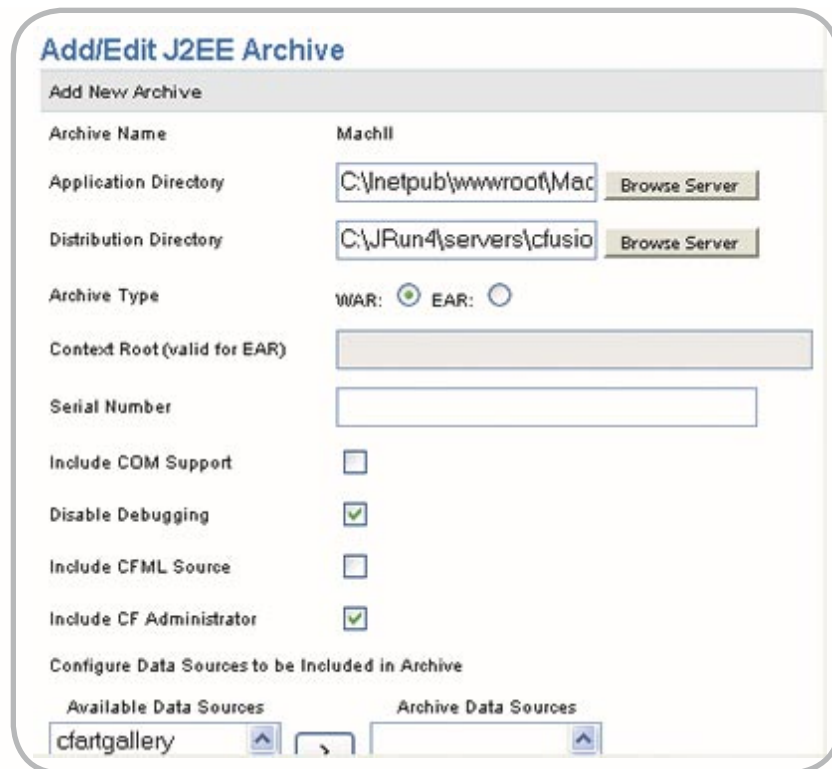
EAR/WAR Deployment

While the cfcompile utility satisfies the long-standing enhancement request for protection of source code, it does not answer the other request that applications be easier to deploy. ColdFusion MX has been a pure J2EE

application since its initial release. A traditional J2EE application is bundled as a EAR (Enterprise Application Archive) or a WAR (Web Application Archive) file. At companies that run a pure J2EE environment, system administrators simply take the EAR/WAR file and drop it into the J2EE instance. However, traditional ColdFusion deployments meant that a server first had to be installed (the ColdFusion runtime) and then the CFML source code installed separately. This did not sit well with system administrators, who felt this two-step process was awkward, and hence limited ColdFusion's adoption in those environments.

To solve this disconnect, ColdFusion MX 7 introduces the option of packaging CFML applications as a EAR or WAR file so that one file can be handed off to the system administrator for deployment. This option means that deployment and management is easier for the system administrator while subsequently allowing ColdFusion to be a viable development alternative when building J2EE applications. Found under the "Packaging & Deployment" section in the ColdFusion MX Administrator (Enterprise version only), the "J2EE Archives (.ear/.war)" option provides a variety of options such as whether or not to include

figure 3





the source code or whether or not to include the ColdFusion Administrator. Figure 3 shows an example of bundling the Mach-II framework into a WAR file (MachII.war).

Having created your EAR or WAR file, it is now literally as easy as sending the resulting file to your system administrator for deployment – no separate server installation is needed!

J2EE Instance Management

Running ColdFusion on multiple instances has many benefits such as application isolation, greater stability, and greater scalability (for more information on the benefits of multiple J2EE instances, see Ben Forta's July 2003 *CFDJ* article entitled "When One ColdFusion Is Not Enough" at <http://sys-con.com/story/?storyid=42050&de=1>). Before the release of ColdFusion MX 7, the creation of new J2EE instances in which ColdFusion server would be deployed (remember, ColdFusion MX is simply a standard Java application deployed into a J2EE server) left a lot of addi-

tional setup work for administrators. From a separate interface for actually creating the instance (i.e., the JRun Admin console) to the manual editing of the underlying XML configuration files (to get features such as session replication properly working), creating a new instance was too tedious of a process, and one that caused many administrators to forgo the feature.

Recognizing this barrier to multiple instance deployment adoption, Macromedia has included a new feature in the ColdFusion Administrator for creating new J2EE servers with no more than a couple of mouse clicks. (Note: this feature is only available if the integrated JRun + ColdFusion option was selected while running the installer. Macromedia did not back-port this option for existing JRun installations.) Found in the ColdFusion MX Administrator under the "Enterprise Manager" navigation bar heading, setting up a new J2EE instance really could not be any easier. From the main screen, simply click the "Add New Instance" button. As seen in Figure 4, only a few pieces of information are

needed for ColdFusion to create the new instance.

The "Server Name" field is required, as this is the name of the J2EE instance. The next two fields are optional. The "Server Directory" field (which is automatically filled in based on the "Server Name" value filled in) is where the instance will reside on the file system. The "Create From EAR/WAR" option is where the EAR/WAR packaging feature mentioned previously really shines. Remember the MachII.war file created earlier that had the ColdFusion runtime bundled with it? Well, if it needs to be deployed to a separate instance, point to the WAR file created and the Manager will deploy the ColdFusion runtime from the bundled archive. If the JRun installation is on a Windows server, there are also options to create the new instance as a Windows service and to define the service with an auto restart recovery option. Clicking the "Submit" button will start the process of installing the new instance, deploying the ColdFusion server within it, optionally deploying the application to run within it, and starting the instance. When everything is complete, the screen will look similar to Figure 5.

It is important to realize that the setup of a J2EE instance and the subsequent deployment of ColdFusion MX within it require a lot of files to be laid down. Therefore, it might be tempting to think that the process is not responding. Rest assured that the process is indeed running under the hood and simply takes several minutes to complete.

figure 4

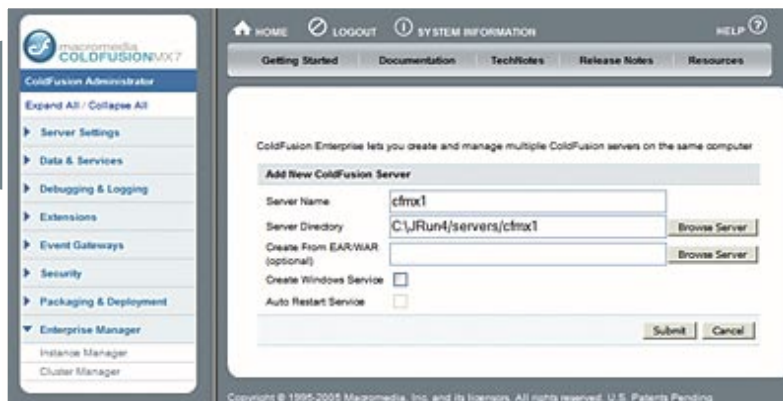
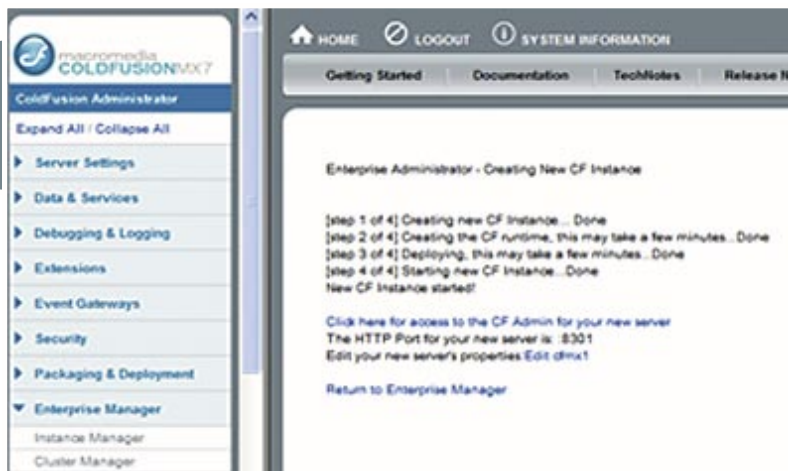


figure 5



Licensing and Other Restrictions

The precompile and sourceless deployment licensing and restrictions are pretty straightforward. The ability to run applications without the source code is only compatible with ColdFusion MX 7 installations – it is not backwards compatible with ColdFusion MX 6.1 and earlier. That being said, the ability to both compile and run applications without the source code is available across all editions of the ColdFusion MX 7 product line – Developer, Standard, and Enterprise. This means that you do not have to have an Enterprise license to

compile your application for Enterprise customers.

The EAR/WAR deployment option is a little trickier. First off, the ability to create the package is available in any edition of ColdFusion MX 7. However, because EAR/WAR files can only be deployed to J2EE servers, ColdFusion MX 7 Enterprise is the only edition that supports the deployment of these packages. (This deployment restriction is consistent with New Atlanta's BlueDragon product line, as seen at www.newatlanta.com/products/bluedragon/product_info/features.cfm#FCMatrix.) Further, deploying an EAR or WAR package requires the purchase of ColdFusion MX 7 Enterprise licenses to match the number of CPUs on the deployment server. For example, if the package is being deployed to a 4-CPU server, then a 4-CPU Enterprise license of ColdFusion MX is required. (Note that Macromedia has changed its server licensing policy with the release of ColdFusion MX 7 such that a 2-CPU license can no longer be split across two 1-CPU machines. Each physical server now requires a separate license. Further, the licensing is based on the number of physical processors in the server only.) If a serial number is not applied when the EAR/WAR package is first created, then either the ColdFusion Administrator must be included with the package, or the Administrator API must be used so that the user can apply their license upon deployment (See the Macromedia LiveDocs for more information on the Administrator API). If the Administrator is not included and no serial number is applied via the Administrator API, the application is deployed as a 30-day trial that reverts to the Developer version upon expiration (which is restricted to localhost access plus 2 client machines).

While seemingly a hassle on the surface, the EAR/WAR deployment option is still much easier than previous versions' deployment experience. As an example, if a company wanted to purchase a bug tracking system with ColdFusion, that company would have to do the following using pre-MX 7:

- 1) Buy the bug tracking system application itself
- 2) Purchase ColdFusion and the appropriate number of licenses

- 3) Install ColdFusion server
- 4) Deploy the bug tracking system to the new server

With ColdFusion MX 7's EAR/WAR packaging option, that same company would have to do the following:

- 1) Buy the bug tracking system with the ColdFusion runtime bundled as one cost
- 2) Purchase and apply the appropriate number of ColdFusion licenses

Much Simpler, Isn't It?

The Enterprise Manager feature for creating new J2EE instances is limited to the Enterprise edition. However, because the licensing is a "per CPU" agreement, there is no limit to the number of instances that can be created. Further, the Instance Manager can only create instances on the same server from which the ColdFusion MX Administrator is installed as opposed to across a network to another installation.

Summary

After years of developer demand, Macromedia has come through with an intelligent, straightforward way to confidently deploy CFML applications. The cfcompile utility improves initial template load time and gives a developer confidence that the innovation and hard work that they have put into a product will not be compromised, while the ability for ColdFusion to be packaged up into one clean installation step opens it up to an increased number of companies who previously would have shunned the idea because of the fragmented install process. The J2EE Instance Manager vastly simplifies this process because system administrators now simply have to click a few buttons as opposed to running several installers. ☺

Dave Carabetta has been working with ColdFusion for over eight years now. Currently, he is the lead developer for a commercial real estate financial services company in New York City, where he develops CF applications as well as managing the web site's cluster setup. He's currently working on ColdFusion/Flex integrated applications. cbetta@hotmail.com

THREE REASONS TO

blog-n-play.com

- 1 Get instantly published to 2 million+ readers per month!**

blog-n-play™ is the only FREE custom blog address you can own that comes with instant access to the entire i-technology community. Have your blog read alongside the world's leading authorities, makers and shakers of the industry, including well-known and highly respected i-technology writers and editors.
- 2 Own a most prestigious blog address!**

blog-n-play™ gives you the most prestigious blog address. There is no other blog community in the world that offers such a targeted address, and comes with an instant targeted readership.
- 3 Best blog engine in the world...**

blog-n-play™ is powered by *Blog-City™*, the most feature rich and bleeding-edge blog engine in the world, designed by Alan Williamson, the legendary editor of *JDJ*. Alan kept the i-technology community bloggers' demanding needs in mind and integrated your blog page to your favorite magazine's Web site.



www.TAMI.linuxworld.com
"Many blogs to choose from"

PICK YOUR MOST PRESTIGIOUS ADDRESS

IT Solutions Guide	MX Dev. Journal
Storage+Security Journal	ColdFusion Dev. Journal
JDJ: Java	XML-Journal
Web Services Journal	Wireless Business & Tech.
.NET Dev. Journal	WebSphere Journal
LinuxWorld Magazine	WLDJ: WebLogic
LinuxBusinessWeek	PowerBuilder Dev. Journal
Eclipse Dev. Journal	

3 MINUTE SETUP

Sign up for your FREE blog Today!



— This site will go beta February 13, 2005.



The Exus

t
 he images represent statements of the innovative navigation system (called "The Exus") used for the second version of the Excluse Studio's Website (www.excluse.com). While creating it, I tried to think of inventing a completely new navigation system that at the same time would not be too complicated for the customer. I came up with about four different navigation systems, but The Exus seemed to be the most appropriate for my idea. Even now, if you go to www.excluse.com, by default the navigation is a common navigation with buttons. If you want to see The Exus in action you will have to switch the navigation type by clicking on the "excluse navigation" button. On the 15th day of its launching, excluse.com got a Best Website award from The American Design Awards for the month of November 2004, as well as several other features/articles in several Web magazines. 





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)





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**

