

MX

developer's journal

THE LEADING MAGAZINE FOR
MACROMEDIA MX DEVELOPMENT,
DESIGN, & PRODUCTION TOOLS

volume 4 issue 1 2006 www.mxdj.com

JAVASCRIPT



DREAMWEAVER

Building a Forum



FIREWORKS

Creating Fireworks Panels



FREEHAND

Sketching a Site Navigation Scheme



COLDFUSION

The Real Estate Sample Application



DIRECTOR

Creating a Shockwave 3D Game Level

THE ABILITY TO SEAMLESSLY INTEGRATE
YOUR FLASH CONTENT WITHIN YOUR
HTML CONTENT IS ESSENTIAL



Presorted
Standard
US Postage
PAID
St. Croix Press



Rev Up Your Flash Video

Stay Ahead of the Competition With VitalStream and the Enhanced Video Features in Flash 8

With over two years of experience in delivering much of today's most popular media, VitalStream® is the first and most experienced Flash™ video streaming service provider.

Enhanced Flash 8 Video Features:

- New VP6 codec delivers higher quality video at the same bit rate
- 8-bit alpha channel transparency enables you to blend video with other elements
- Improved live video capabilities

VitalStream Complete Toolset for Flash:

- MediaConsole®
- MediaOps™ SDK
- Flash Authentication
- Reporting Dashboard



*Integrate Streaming Media
Into Your Flash Projects*

Take Advantage of the Enhanced Video Features in Macromedia Flash 8
Call (800) 254-7554 or Download Tutorials at www.vitalstream.com/go/mxdj



macromedia®
FLASH VIDEO
STREAMING SERVICE

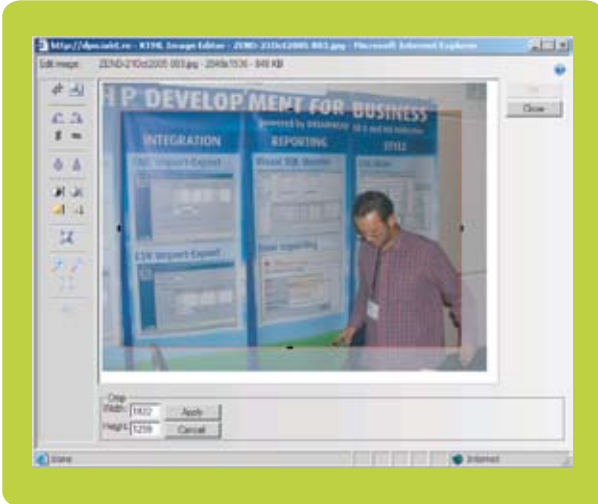
Call (800) 254-7554
Visit www.vitalstream.com



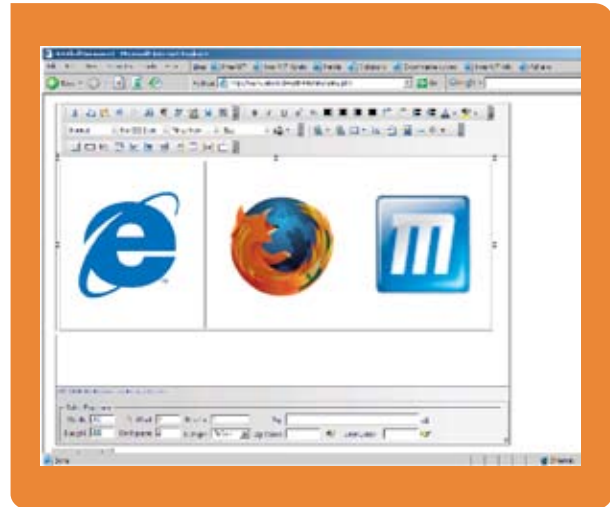
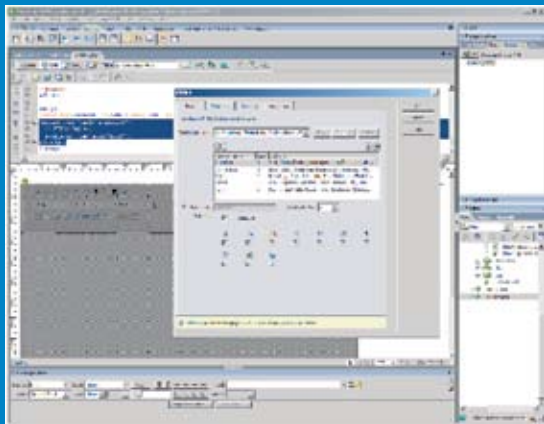
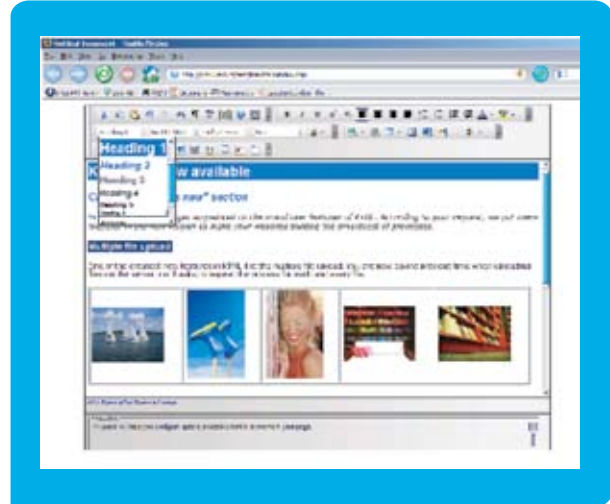
KTML4 Word editing in browser

Along with the most affordable **UNLIMITED** licence

Revolutionary Image Editor



Word-like visual CSS styles



Fast Dreamweaver integration

- Instant paste from Word
- Incredible speed
- Easy to use Word-like toolbars
- Improved CSS authoring
- Remote File Explorer
- XHTML 1.1 compliant

Wide browser compatibility

- Multiple file upload at once
- HTML Table Editor
- Support for multimedia (Flash, Avi)
- Documents management (.doc, .pdf)
- Page templates
- WAI compliant

Please visit www.interaktonline.com/ktml4/ for details



22 Building an Image Injector
Using ExternalInterface to update HTML
by david vogleer




7 Adobe Has Inherited Web History from Macromedia
...and will be making more Web history in 2006, with Flash
by jeremy geelan

 **8** Building a Forum with Dreamweaver
A forum can really help you build up a community around your website
by marius zaharia & cristian ivascu

 **28** Creating Fireworks Panels
Custom panels that unlock hidden functionality
by trevor mccauley

 **36** Sketching a Site Navigation Scheme
Create preliminary sketches of your site's navigation scheme
by jon varese

 **40** The Real Estate Sample Application
Create an application that allows users to retrieve records from a database
by nahuel foronda & laura arguello

 **46** Creating a Shockwave 3D Game Level in 30 Minutes
Export Shockwave 3D content using any of the major 3D applications
by mal duffin



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

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

When calling your web host for support you want answers, not an annoying song stuck in your head from spending all day on hold. At Edgewebhosting.net, we'll answer your call in two rings or less. There's no annoying on-hold music, no recorded messages or confusing menu merry-go-rounds. And when you call, one of our qualified experts will have the answers you're looking for. Not that you'll need to call us often since our self-healing servers virtually eliminate the potential for problems and automatically resolve most CF, ASP, .NET, SQL, IIS and Linux problems in 60 seconds or less with no human interaction.

Sleep soundly, take a vacation, and be confident knowing your server will be housed in one of the most redundant self-owned datacenters in the world alongside some of the largest sites on the Internet today and kept online and operational by one of the most advanced teams of skilled Gurus, DBAs, Network and Systems Engineers.

By the Numbers:

- 2 Rings or less, live support
- 100% Guarantee
- 99.999% Uptime
- 2.6 GBPS Redundant Internet Fiber Connectivity
- 1st Tier Carrier Neutral Facility
- 24 x 7 Emergency support
- 24 Hour Backup
- Type IV Redundant Datacenter



For a new kind of easy listening,
talk to EdgeWebHosting.net

<http://edgewebhosting.net>



2003 - 2006

- Shared Hosting
- Managed Dedicated Servers
- Managed Colocation
- Semi-Private Servers
- ColdFusion
- BlueDragon
- ASP
- .NET
- .Linux
- .Java
- SQL Server
- .MySQL
- Self-Healing Servers



STREAM57



Customized Flash-based media solutions

Software and services for collaboration, video conferencing, and e-learning

visit us
stream57.com

e-mail us
streamline@stream57.com

call us
212.909.2550 x1012

Group Publisher Jeremy Geelan
Art Director Louis F. Cuffari

EDITORIAL BOARD
Dreamweaver Editor
Dave McFarland
Flash Editor
Brian Eubanks
Fireworks Editor
Joyce J. Evans
FreeHand Editor
Louis F. Cuffari
Director Editor
Andrew Phelps
Captivate Editor
Tom Green

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

EDITORIAL
Editor
Nancy Valentine, 201 802-3044
nancy@sys-con.com

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

Technical Editors
Jesse Warden • Sarge Sargent

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

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

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

Worldwide Newsstand Distribution
Curtis Circulation Company, New Milford, NJ

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

Promotional Reprints
Megan Mussa
megan@sys-con.com

Copyright © 2006
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. Macromedia and Macromedia products are trademarks or registered trademarks of Macromedia, Inc. 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

Adobe Has Inherited Web History from Macromedia...

...and will be making more web history in 2006, with Flash
by jeremy geelan

One of the "inflexion points" of the development of the Web, when commentators and analysts draw breath for long enough to chronicle its history, is certain to be the day that Google, through first Gmail and then Google Maps, opened the eyes of millions to the fact that the Web can be smarter, more responsive, and interactive...above all, that it doesn't have to involve "click, wait, and refresh."

Google happened to use AJAX (Asynchronous JavaScript + XML) for those two bellwether apps. But those of us who have followed the Macromedia, now Adobe, story know of course that AJAX is very much a Johnny-come-lately to the Rich Internet Applications party.

Whereas "AJAX" is barely a year old (Jesse James Garrett having coined it Feb 18, 2005), "Rich Internet Applications" as a term was in circulation many years before. Already at the beginning of the new century, Macromedia's developer-innovators were using the term, so that by 2003 for example "Flash-enabled RIAs" was really quite a common term in developer mailing lists and Macromedia user groups.

Macromedia anticipated AJAX in every way, pioneering the RIA approach of which AJAX is merely a subset, and in 2006 its prescient approach is productized in Flash and Flex and FlexBuilder and Flash Player, products which over time Adobe is likely to help turn into the best-known suite of software on the planet.

From a personal perspective I shall never forget a session that Christophe Coenraets gave at one of the SYS-CON Events conferences I was fortunate enough to Chair, namely Web Services Edge Conference & Expo (East) 2004, in Boston. The session was called "Code-Based Rich Internet Applications with Macromedia Flex" and covered using components, layouts, and managers to build user interfaces as well as using Flex's XML-based language, MXML, to create and manipulate client-side data models.

What I remember above all was that the delegates in the completely packed room were spellbound, literally spellbound, by Christophe's agile demonstration of how much could be done with the Web vs. how little anyone was (then) doing. It was as if we'd lost our collective critical judgement, his presentation seemed to imply—as users, and therefore as developers, we'd begun to accept latency that ought never to have been tolerated, and had almost been self-brainwashed into thinking of the Web as "pages" when that in many circumstance was far from the best metaphor. For many in the room, it was an epiphany, you could sense it.

So in reality the inflexion point came long before AJAX, and even longer before Gmail and Google Maps. And it came from Macromedia as much as anyone.

SYS-CON Events—a wholly owned subsidiary of SYS-CON Media which brings you MXDJ each month—has been working with Adobe behind the scenes just recently to bring you the RIA story in a major educational event so that you don't fall into the trap of failing to see the wood for the trees. After all, Flash also interacts with JavaScript on a web page, and it's a highly suitable, lightweight tool for highly responsive web apps. Google Maps could just as easily have been done in Flash several years before it was done in AJAX. Why it wasn't, who knows? Perhaps this will be one of the enigmas discussed at our upcoming Rich Internet Apps Conference in August 2006 (<http://www.RIAConference.com>).

So, will Adobe try and crush AJAX, Microsoft-style? And/or openLaszlo? Most definitely not. Under Kevin Lynch's technologically nuanced and developer-centric leadership, it seems much more likely that Adobe's Platform Business Unit will concentrate on inspiring developers to use Flash and HTML "with an AJAX approach" to build Web 2.0-style applica-

— continued on page 20

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

d

f

fw

fb

fd

db

Building a Forum with Dreamweaver

A forum can really help you build up a community around your website
by marius zaharia & cristian ivascu

Forums (message boards) are a great way to share ideas, debate on current issues, or offer technical assistance to your clients. A forum can really help you build up a community around your website, increase traffic, and improve communication with your clients. Although you can find a wealth of free forum applications out there, which you can install and customize for your needs, this article teaches you how to build your own forum from scratch to make it fit your specific requirements. (You can view this article in its original www.macromedia.com/devnet.)

We will walk you from designing and setting up your database to building a basic front end, adding user authentication, allowing subscription to message threads, and sending topic reply notifications by e-mail. You can have a fully functional forum in no time, with minimum effort and coding.

Planning the Forum

In this tutorial, you will use Dreamweaver 8 and ImpAKT to build a forum application which will allow users to:

- View a list of discussion topics
- Read the messages associated to each topic
- Register an account with the forum, and upload a profile photo
- Login and logout
- Post a new message in any topic
- Reply to other messages
- Subscribe to a discussion thread (a chain of messages that are replies to an initial message)
- Receive e-mail notifications when a new message is posted in a thread

they have subscribed to

Before you start building the forum, please make sure that:

- You have a running database server and web server with PHP language support
- You have installed the required software: Dreamweaver and ImpAKT
- You have correctly defined a Dreamweaver site with a PHP_MySQL testing server
- You have set up the database provided with this tutorial and have created a connection to it

Setting Up the Development Framework

First you have to set up your application server environment and install Dreamweaver 8 and ImpAKT.

This article assumes that you have already installed and configured a web server with PHP and MySQL support. If this is not the case, take a look at the following articles from the Developer Center:

- [Setting Up Your PHP Server Environment Using Linux, Apache, MySQL and PHP](#) (by David Sklar) if you use the LAMP platform
- [Setting Up the PHP, MySQL, and Apache Server Platform on Macintosh OS X for Dreamweaver MX](#) (by Minh Huynh) if you're working in Mac OS
- [Setting up PHP for Microsoft IIS](#) (by Andrew Stopford) if you're a fan of IIS

If you're using Windows, also take a look at WAMP Server, a free tool that automatically installs Apache, PHP 5, MySQL server, phpMyAdmin, and SQLiteManager. For detailed instructions

on installing WAMP Server, see this technical note on the InterAKT site.

Before you install ImpAKT, you need to have installed Dreamweaver 8. You can install and manage the extensions using Extension Manager, an accessory to Dreamweaver.

What Is ImpAKT?

ImpAKT is a Dreamweaver extension that offers a collection of wizards and server behaviors that will help you set up a dynamic website from scratch, with no coding at all. ImpAKT automates most common programming tasks and patterns, and even helps inexperienced HTML designers perform complex tasks. For example, ImpAKT will help you set up your site for user authentication, validate your visitors' input, send e-mails after forms have been submitted, and upload files and images.

Extensions are software tools that enhance the standard functionality of Dreamweaver, much like plug-ins or add-ons. They can vary from simple code snippets to complex applications. Their purpose also varies: they can reduce manual coding, minimize repetitive tasks, fix certain bugs or errors, increase productivity, or build a specific object.

Extensions are usually delivered as executable MXP files. When you run them, the Extension Manager initiates the installation.

Start the installation by downloading the extension using the links at the beginning of this article. Then double-click the MXP file. After the ImpAKT installation is initiated, you'll see a disclaimer. If you agree to the terms, click

Accept to continue the installation. When the Extension Manager finishes installing (usually just a few seconds), you'll see a dialog box notifying you that the installation is complete.

You'll notice that ImpAKT is listed in the Extension Manager window.

If Dreamweaver is open, close it and then relaunch it so the changes caused by ImpAKT can take effect. Each installed extension modifies the Dreamweaver standard workspace by adding its own commands, buttons, or menus. The ImpAKT elements are added to the Insert bar and to the Server Behaviors menu (in the Application panel).

Notice that a new tab called MX Kollection appears in the Insert bar.

ImpAKT is part of a bundle of Dreamweaver extensions, which is why the tab is called MX Kollection. For more information, visit the MX Kollection home page.

From the MX Kollection tab, you can access some of the most common wizards and commands in ImpAKT that help you build database-driven web forms for inserting, updating, and deleting information, and set up professional user authentication for your website.

The second area where you can access ImpAKT commands is the Server Behaviors tab of the Application panel.

In the next section, you will define your site in Dreamweaver.

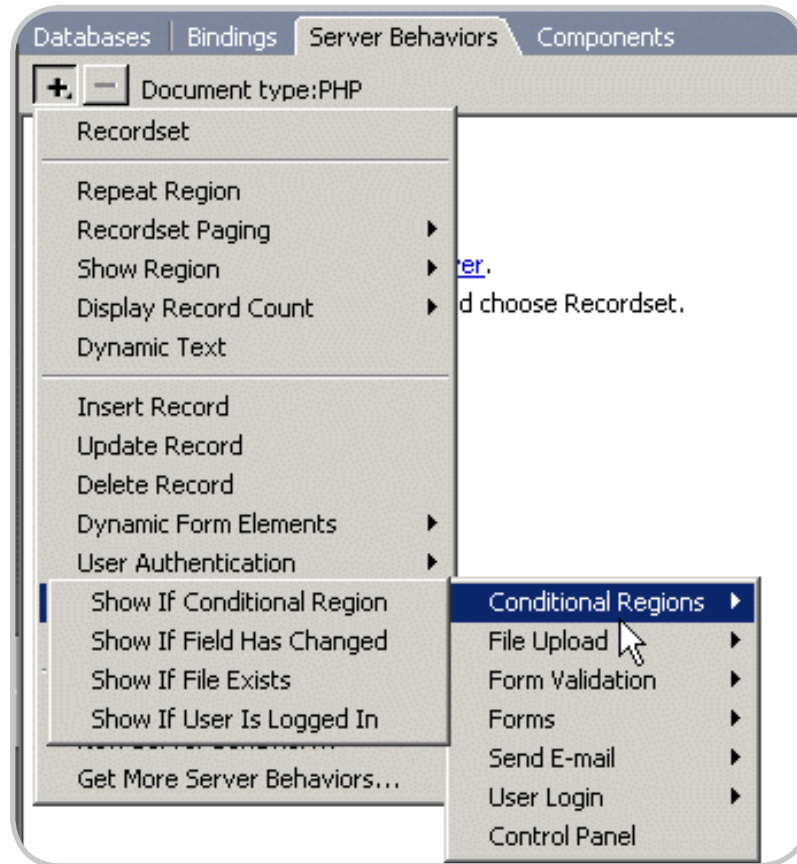
Before you build the forum, you need to set up your site in Dreamweaver.

For a quick site setup, take a look at this TechNote: How to Define a Site in Dreamweaver. A minimal setup requires you to configure at least these sections:

- Local Info, which identifies the site files in Dreamweaver and enables the site management features
- Remote Info, which defines the settings to access the remote web server where your site will be uploaded
- Testing Sever, which defines a private place to act like a public server to test the application and connections to the database

Follow the instructions in this TechNote and make sure you set the server model to PHP MySQL in the Testing Server section.

Download the tutorial sample files and unzip them to your site's root folder.



In the next section, you will set up and connect to your database.

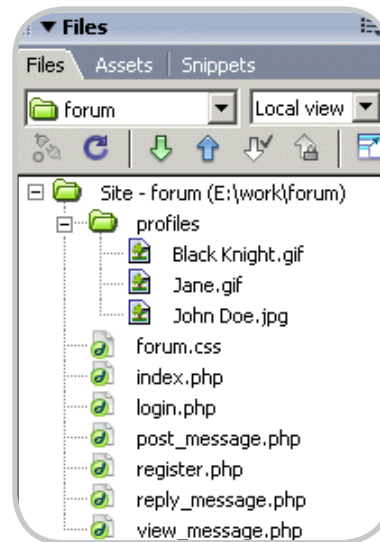
Setting Up Your Database

The forum is completely dynamic. It stores and extracts all information from a database. A typical forum is basically a collection of messages posted by users and organized by topics. Therefore you need at least these three tables for your forum database:

- one for user information: user_usr
- one for messages: message_msp
- one for topics: topic_top

The SQL code that creates the topic_top table is pretty straightforward. Notice that the topic title is defined as a unique key, meaning there shouldn't be any topics with the same title:

```
CREATE TABLE topic_top (
    id_top int(11) NOT NULL auto_increment,
    title_top varchar(100) NOT NULL
    default '',
    description_top varchar(200) default
    NULL,
    PRIMARY KEY (id_top),
    UNIQUE KEY title_top (title_top)
```



);

Here is the code that creates the message_msg table:

```
CREATE TABLE message_msg (
    id_msg int(11) NOT NULL auto_increment,
    idtop_msg int(11) NOT NULL default
    '0',
    idmsg_msg int(11) default NULL,
```

```

id_init_msg int(11) NOT NULL default
'0',
idusr_msg int(11) NOT NULL default
'0',
date_msg datetime NOT NULL default
'0000-00-00 00:00:00',
subject_msg varchar(100) default
NULL,
content_msg text NOT NULL,
subscribe_msg tinyint(4) NOT NULL
default '0',
PRIMARY KEY (id_msg)
);

```

The following snippet shows the SQL code for setting up the table:

```

CREATE TABLE user_usr (
id_usr int(11) NOT NULL auto_incre-
ment,
name_usr varchar(100) NOT NULL
default '',
email_usr varchar(150) NOT NULL
default '',
password_usr varchar(100) NOT NULL
default '',
active_usr tinyint(2) NOT NULL
default '0',
randomkey_usr varchar(100) NOT NULL
default '',
PRIMARY KEY (id_usr),
UNIQUE KEY name_usr (name_usr)
);

```

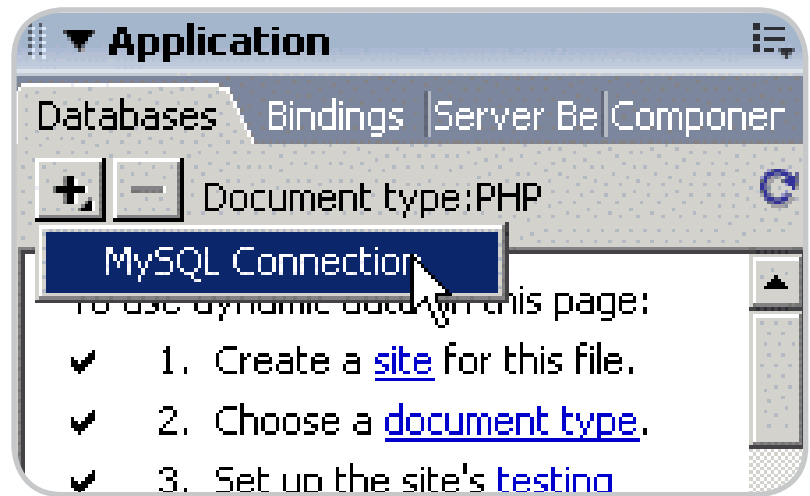
Notice that because the user name must be unique, it was defined as a unique key.

If you are unfamiliar with database terminology (primary keys, foreign keys, self foreign keys, and the like) and database architecture techniques, check out the Database Concepts article on the InterAKT site.

Now that you have an idea of how your database will look, fire up the sample SQL script in your MySQL console or in your favorite database management software (such as phpMyAdmin) and create the forum database. Include the following line at the beginning if you want to create a separate database:

```
create database mm_forum;
```

The sample SQL script also contains some dummy data, which you will use when building the forum front end. Now you're ready to connect to your newly created database.



Connecting to the Database

To connect to the database in Dreamweaver, follow these steps:

- Start Dreamweaver.
- Open the index.php page from the forum site.
- Choose Application panel > Databases tab, click Plus (+), and select MySQL Connection.
- Configure your connection to your newly created database.

If you are not sure how to connect to the database server, ask your system administrator for the proper access information. Click the Select button to locate the forum database. You can test the connection using the Test button.

In the next section you will learn how to build the forum home page, which displays a list of topic titles and descriptions.

Displaying the Forum Topics

The forum home page displays all topic titles and descriptions. Beside these values, which are extracted from the database, you will have to display links to enable users to set up an account or log in.

All the information will be displayed in the index.php page. The page layout is already designed with CSS, and has areas (<div> containers) for displaying various information from the database.

To see the <div> containers better, activate the CSS Layout Backgrounds and CSS Layout Outlines options by choosing View > Visual Aids and then selecting the option in the submenu.

Notice that the <div> tags now appear more visible. The CSS Layout Backgrounds option adds a different

color to every <div> tag to make them easier to distinguish from one another.

Turning on the CSS Layout Backgrounds option does not display the real color of the <div> containers but it can help you sort out which element belongs into which container.

Note: CSS stands for Cascading Style Sheets—a standard that enables you to separate content from presentation. You can create styles that define the font characteristics, colors, and even positioning of your content. You can then apply these styles to the desired elements. The CSS styles are already defined in the forum.css style sheet that comes with the tutorial sample files.

The topic is a link to the page displaying the corresponding messages. To create the topic list, you have to do the following:

- Open the index.php page in Dreamweaver.
- Create the recordset that retrieves topic information, the ID, title, and description.
- Display the recordset contents on the page.

To create the recordset that retrieves the forum topics, follow these steps:

- Click the Plus (+) button on the Bindings tab in the Application panel. From the pop-up menu select Recordset (Query).

Configure the recordset dialog box options:

- For the recordset name enter rsTopics.
- In the Connection pop-up menu, select connForum.
- Select the topic_top table in the Table

Intermedia.NET...

Now serving the hottest ColdFusion.

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

Intermedia.NET's premier hosting services include:

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

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

Call us at: 1.888.379.7729

e-mail us at: sales@intermedia.NET

Visit us at: www.intermedia.NET



INTERMEDIA.NET



pop-up menu.

- To sort the topics alphabetically, select the title_top column in the Sort pop-up menu; for the method (the next pop-up menu), select Ascending.
- Click OK to close the dialog box and add the recordset. The recordset appears in the Bindings tab.
- After having created the recordset, you must use its fields to build the home page:
- With the index.php page open, expand the rsTopics recordset in the Bindings tab.
- Clear all content from the second <div> container (the white one).
- Drag and drop the title_top field on the Bindings tab in the second <div> container.
- Place the cursor after the added topic title and press Enter (Return) to create a new paragraph.
- To distinguish the topic title from its description better, format it as Heading 3 by selecting the dynamic text and then choosing Heading 3 from the Format pop-up menu in the Property inspector.
- The topic title must be linked to the messages page. This page displays all messages from the selected topic. The messages are filtered by the topic ID, which is passed from the home page as a URL parameter. With the topic title selected, click the Folder icon in the Property inspector to define the link.
- In the dialog box that opens, select the view_message.php page, and then click the Parameters button to define the URL variable that will filter the messages:
- In the Parameters dialog box, click the Plus (+) button to add a new URL parameter.
- In the Name field enter id_top.
- For the Value, click the Dynamic Data icon (the yellow lightning bolt), and then select the id_top field of the rsTopics recordset.
- Click OK to define the parameter value.
- Click OK twice to create the link.
- Drag and drop the description_top recordset field in the Bindings tab in the paragraph below the link.

At this point, if you try to preview your page in the browser, you'll notice that only the first topic is displayed. You need to loop through all records in order



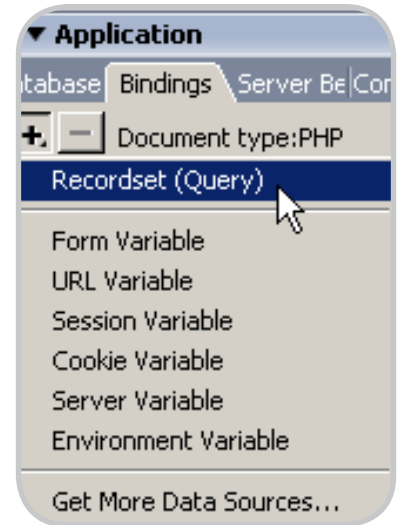
to display all the topics:

- Select the topic title and description and apply a Repeated Region server behavior. You can access it from the Application tab of the Insert bar.
- Note: Make sure you select just the dynamic text, not the <div> container.
- In the Repeat Region dialog box, configure the repeat region to use the rsTopics recordset and click All Records to display all records.
- Click the OK button to close the dialog box and apply the server behavior. You'll notice that a gray border labeled Repeat is displayed around the topic information.
- Save the page and upload it to the testing server. Press F12 (Windows) or Option+F12 (Macintosh) to load the page in your web browser.
- http://www.macromedia.com/devnet/dreamweaver/articles/forum_pt1/fig24_lg.html

In the next section, you'll change the way links look using CSS styles.

In this section you will use the unified CSS panel in Dreamweaver 8 to style the links in your forum. (Julie Hallstrom takes you through all the new CSS features in Dreamweaver 8 in her article, *An Overview of CSS in Dreamweaver 8*.) Here you will make the forum links orange and then apply a mouseover effect, so that they become underlined when users hover their mouse cursors over them:

- With the index.php page open in Dreamweaver, select the dynamic text that displays the topic title. Then, in the tag selector, click the <a> tag.
- The CSS panel displays the existing styles. If you cannot see the CSS panel, select Window > CSS Styles.
- Notice that the a:link and a:visited styles are already defined for you, but they have no properties. To define a new property, click the Add Property link.
- Select Text-Decoration from the first pop-up menu and None from the



second one. This removes the default underlining of links.

- Now make the links orange. Click the Add Property text to define a new rule. From the properties pop-up menu, select the Color option.
- Select the color by clicking the button displayed in the value field. Pick an orange shade from the color picker.
- Add a mouse-over effect for links. To do that, you need to define a new CSS rule by clicking the corresponding button from the CSS panel.
- In the displayed dialog box, select the Advanced option button and then choose the a:hover selector and make sure you define it in the forum.css style sheet.
- Click OK to create the rule. A new dialog box is displayed, where you can define the CSS properties for the new rule. The only property you need, for now, is to make the text underlined. Click the Underline option in the Decoration section and then click OK.

Notice how easy it is to edit the existing properties, add new properties, or define a new CSS rule from scratch in Dreamweaver 8—you can do it all from one panel. The topic title links have now changed their color.

Notice that the CSS file attached to the page is opened. Save the CSS file and the page and upload both of them to the

server. Preview the forum home page in the browser to see the changes.

In the next section, you will learn how to display messages in a particular topic.

The topic messages will be displayed in the `view_message.php` page. The page layout is already created using `<div>` tags. You can change the existing styles and layout on the CSS panel.

The page displays the following for each message:

- User name and avatar (profile photo) in the left section
- Message subject, body, and date of posting in the center `<div>` tags
- Link to reply to the message

To create the message list, do the following:

- Create an advanced recordset that retrieves information from two tables and filters it by the topic ID received as a URL parameter.
- Display the user name and a thumbnail of the profile photo.
- Display the message details.
- Use conditional regions to control the output.

First, you must create the advanced recordset. This recordset will have to perform a join between the `message_msg` and `user_usr` tables.

Note: The SQL join is an operation that enables you to retrieve information from two or more tables that are related through a foreign key. You can learn more about joins by reading the [Relations Between Tables tutorial on the InterAKT site](#).

To create the recordset, follow these steps:

- In the Bindings tab, click the Plus (+) button. Select Recordset (Query) from the pop-up menu.
- In the simple recordset configuration dialog box, enter `rsMessages` as the recordset name.
- From the Connection pop-up menu, select the `connForum` database connection.
- In the Table pop-up menu, select the `message_msg` table.
- To retrieve messages in a particular topic, you must add a filter condition. In the Filter pop-up menu, select the foreign key to the topic table: `idtop_`

`msg`. For the value to match, select URL Parameter from the pop-up menu. For the parameter name, enter `id_top` (the name specified when you created the topic link).

- Define the join between the `message_msg` and `user_usr` tables. Because you cannot define it in the simple recordset window, click the Advanced button in the Recordset dialog box.
- Enter the SQL code in the SQL text area. When switching from the simple dialog box to the advanced one, your settings will be preserved in the SQL code.
- You have to change this code in order to perform the join. Place the cursor after the `message_msg` table name and enter the following code:
 - `LEFT JOIN user_usr ON idusr_msg = id_usr`
 - This is all it takes to create the query.
 - Review the completed dialog box and click OK to close it.

The recordset is displayed in the Bindings tab. It contains fields from both tables. After having retrieved all necessary information from the database, it is time to put it to use and display it in the page.

Displaying Profile Photos

Here is how you want to display the user information: user name followed by a thumbnail of the user's photo. Place your cursor in the left `<div>` container and remove the placeholder text User information. Drag the `name_usr` field from the Bindings tab and drop it where the deleted text was.

Now make some room for the image. Place the cursor after the dynamic text you've entered and press Shift+Enter. This inserts a line break after the user name. To display a thumbnail of the user's photo, you will use an ImpAKT server behavior called Show Thumbnail. This server behavior generates thumbnails of the selected images from your server. Select the Server Behaviors tab > Plus (+) > MX Kollection > File Upload, and then select Show Thumbnail.

Configure the user interface as follows:

- In the File folder text box, enter the path to where the user photos are stored. Click Browse and select the Profiles folder.

SYS-CON MEDIA

President & CEO

Fuat Kircaali, 201 802-3001

fuat@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 Manager

Megan Mussa, 201 802-3023

megan@sys-con.com

Associate Sales Managers

Kerry Mealia, 201 802-3026

kerry@sys-con.com

PRODUCTION

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 Directors

Tami Beatty, 201 802-3038

tami@sys-con.com

Abraham Addo, 201 802-3037

abraham@sys-con.com

Assistant Art Directors

Andrea Boden, 201 802-3034

andrea@sys-con.com

Video Production

Ryan Palmieri, 201 802-3036

ryan@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

Wayne Uffleman, 201 802-3057

wayne@sys-con.com

ACCOUNTING

Financial Analyst

Joan LaRose, 201 802-3081

joan@sys-con.com

Accounts Payable

Betty White, 201 802-3002

betty@sys-con.com

Accounts Receivable

Gail Naples, 201 802-3062

gailn@sys-con.com

CUSTOMER RELATIONS

Circulation Service Coordinators

Edna Earle Russell, 201 802-3081

edna@sys-con.com

JDJ Store Manager

Brundila Staropoli, 201 802-3000

bruni@sys-con.com





Note: Make sure your database is populated with the sample data from the `mm_forum.sql` script and that the Profiles folder is inside your site root and contains the sample images that come with this tutorial.

- In the Recordset pop-up menu, select the `rsMessages` recordset you created earlier.
- In the Field pop-up menu, select the `photo_usr` recordset field that stores the photo names.
- In the Resize method pop-up menu, select Proportional - Fit to box. This way, the aspect ratio of the photos is preserved, while guaranteeing that images fit the specified space.
- For the box size, enter 80 for the Width and 80 for the Height, (these dimensions are specified in pixels).
- Click OK to apply the server behavior. An image placeholder is displayed in the Design view. You cannot see the actual photos, however, because they are retrieved dynamically from the database.
- Note: After you apply any of the ImpAKT server behaviors or wizards, a new folder will be created in your site root, called `includes`. This folder stores the files required for the server behaviors and wizards to work. You must upload the entire folder to your remote or testing server before previewing any pages in the browser.

Next, display the actual message information:

- Select and delete the message subject placeholder text. Drag and drop the `subject_msg` field from the Bindings tab in its place.
- Repeat the same procedure to replace the message body text with the `content_msg` recordset field.
- The last row displays the date and time when the message was posted to the forum. Remove the existing text and type Posted on (make sure to leave a trailing space so that the date will not appear right next to the text). From the Bindings tab, drag and drop the `date_msg` field after the entered text.
- At this point, all the necessary information is displayed on the page, but for one message only. To display all records, you must use a Repeated Region server behavior, just as you did for the topics list. From the tag selector, click the `<div.content>` tag to select it.
- Doing this selects the content within the white `<div>` tag.
- Go to the Application tab of the Insert bar and click the Repeated Region icon. Configure the dialog box that opens to use the `rsMessages` recordset and to show all records.

To test the page, save it and upload it to the testing server. Do not forget to upload the `includes` folder as well. Load the forum home page in the browser and click a topic title. If you use the sample data from the SQL script that came with this tutorial, select the topic called "Investment Opportunities"—this is the only topic that contains some sample messages.

In the next section, you will learn how to show or hide content using conditional regions.

Controlling Output with Conditional Regions

What happens if you click a topic that contains no messages?

To avoid having users see this type of page, you can use conditional regions to control what content is displayed and when. Dreamweaver 8 comes with a predefined set of conditional regions that help you hide or display content only when a condition is met—if a recordset is empty, if it is the first record in the recordset, and so on. In addition to this

set, ImpAKT offers a set of conditional regions—if a user is logged in, if a file exists or a field has changed.

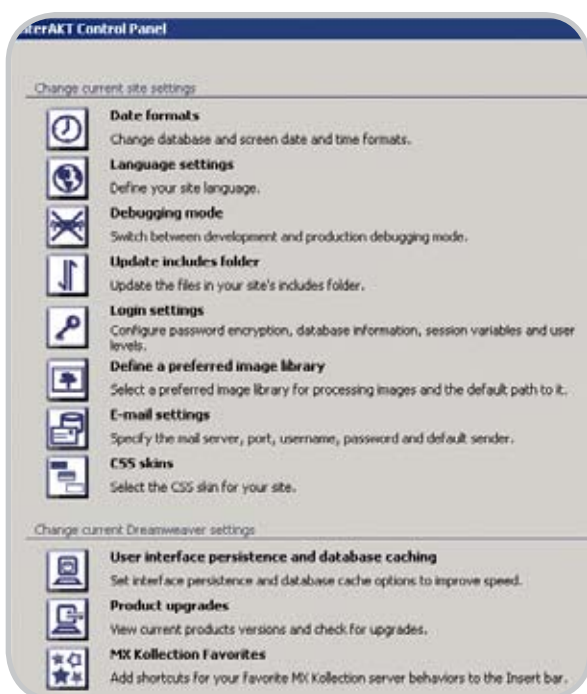
The message list elements must be displayed only when messages exist for the selected topic. To enforce this behavior, you will use the Show If Recordset Is Not Empty server behavior:

- Select the `<div.content>` tag from the tag selector.
- Apply the Show If Recordset Is Not Empty server behavior from the Server Behaviors tab > Plus (+) > Show Region.
- In the Show If Recordset Is Not Empty dialog box, configure the user interface to use the `rsMessages` recordset.
- Click OK to apply the server behavior. A gray border labeled "Show If..." appears around the message details.
- Now if a topic without messages is selected, the page will no longer look broken.
- Add a message that tells users the topic is empty. To do that, place the cursor inside the content `<div>` tag, on the very last blank space. Press Shift+Enter to create a new line and type the following text: There are no messages in this topic yet.
- Select the text you just entered and apply the Show If Recordset Is Empty conditional region (from the Application tab in the Insert bar).
- In the displayed dialog box, select the `rsMessages` recordset, as you did for the previous conditional region.
- Click OK to apply the conditional region.
- Now if you access a topic that has no messages, the text you entered above is displayed.

Finally, think of the following situation: Not all users might want to include a photo next to their name. If you display the messages from the Investment Opportunities topic, you'll notice that Angel has not uploaded a profile photo.

In this case, a default image is displayed. To display no image at all, you have to apply another conditional region:

- In Design view, select the image placeholder.
- Select the Server Behaviors tab from the Application panel and click Plus (+) > MX Kollection > Conditional Regions > Show If File Exists. This will



FLASHFORWARD

CONFERENCE & FILM FESTIVAL

Washington State Convention & Trade Center

SEATTLE
Feb 27 - Mar 2

'06



Join the 15th Flashforward Conference & Film Festival

- Keep your skills current
- Learn from experts
- Be inspired by the Film Festival
- Network at the job fair
- See the latest products
- Attend industry parties
- See Adobe/Macromedia keynote
- Visit an awesome new city
- Live, breath, eat, sleep Flash® for four days!

Confirmed Topics:

Workshops covering ActionScript, Flash Video, Flex 2, and Flash for Educators.
Sessions on Flash 8 and Animation, Games, Portfolios, ActionScript 3, Flex 2, Open Source, Mobile Devices, Audio, Art, Creative Commons, User Interfaces, Accessibility, Installations, Education, Security, JSAPI, Fuse, Physical Computing, Video Compositing, Color, Testing, and Effects.

Other sessions cover related technologies such as After Effects, Ajax, Breeze, Captivate, ColdFusion, FlashBlock, Flex, MySQL, PHP, and Photoshop.

As always we will deconstruct the best Flash websites, showcase inspirational designers, and more.

For more information, visit :
flashforwardconference.com

Save \$50
with this special discount code:
FFWDVMX6
For group discounts
(over four people)
and educational pricing,
please contact:
registration@flashforwardconference.com
or call
877-435-2744
toll free
(001-410-386-0646 outside the US)
during East coast business hours

Produced by:



Platinum Sponsor:



Silver Sponsor:



Bookstore Sponsor:



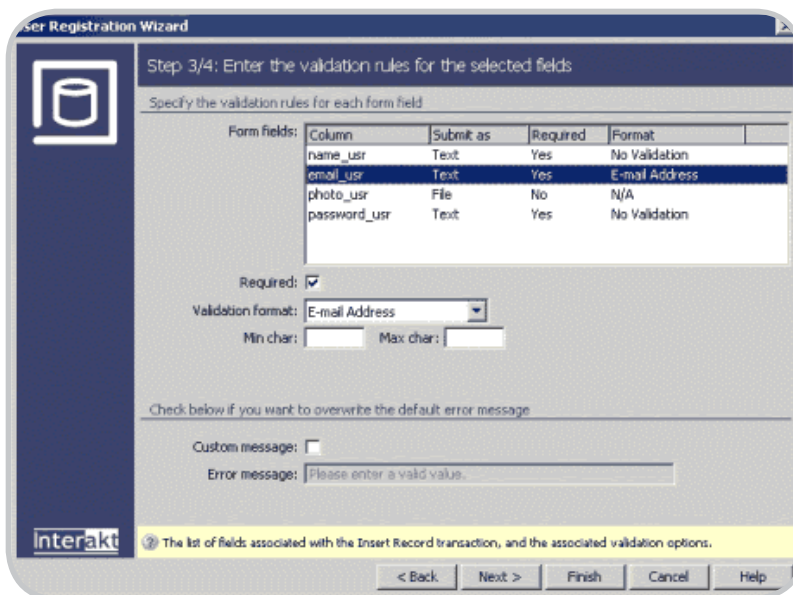
Bronze Sponsor:



Media Sponsors:



creativepro



hide the default image (“Image not found”) for users who haven’t uploaded an avatar.

- Configure the conditional region dialog box as follows:
- Select the folder that stores profile photos: profiles/
- Select the recordset that displays the message information: rsMessages
- In the Field pop-up menu, select the field that stores the filename: photo_usr
- Click OK when done.
- Save your page, upload it to the testing server, and preview it in the browser.

Congratulations! The pages for displaying forum topics and messages are now completed.

Where to Go from Here

In this tutorial, you have learned how to connect to a database, pull data from your tables, and style it with CSS using Dreamweaver 8 and ImpAKT.

In Part 2 you will learn how to use the ImpAKT wizards and server behaviors to implement a user authentication system, create forms to post new messages or reply to existing ones, and notify subscribers of replies by e-mail.

Setting Up User Authentication

In the following sections of this tutorial, you will use ImpAKT to build a user authentication system for your forum. You need authentication to prevent

pranksters from posting a load of garbage to your forum and getting away with it. If users want to post messages to the forum, they must register an account, activate it, and then log in. Only then will they be able to post messages to the forum.

To build the user authentication system, you will do the following:

- Configure user login settings from the InterAKT Control Panel.
- Create a user registration page.
- Create the user login page.
- Create a logout link.
- Use conditional regions to control which content is public and which is private.

Configuring Login Settings

The InterAKT Control Panel comes with your ImpAKT installation and is a central place for all configuration options for your site. The InterAKT Control Panel can be accessed from the MX Kollection tab of the Insert bar.

Alternatively, you can access the InterAKT Control Panel by clicking the Server Behaviors tab > MX Kollection submenu.

The InterAKT Control Panel offers a quick and convenient way to configure the most common settings for your development environment and for your site. Once configured, the settings will be remembered, thus saving you a lot of time.

Here are just a few of the settings you can configure with the InterAKT Control Panel:

- Date formats: Change the date and time formats for the pages built with ImpAKT to match your specific regional settings (American, German, and so on).
- Language settings: Change the language for form buttons, error messages, labels, and hints generated by ImpAKT. You can choose one of the predefined 11 languages.
- Debugging mode: Configure the level of detail of error messages or set the error reporting method (errors can be sent by e-mail to the webmaster or logged to a file on the server).
- Login settings: Configure the user authentication options.
- E-mail settings: Configure the outgoing e-mail server, port, user name, and password.
- CSS skins: Change the look of the entire site by selecting a different CSS skin.
- User interface persistence and database caching: Enable these settings to speed up your work. If user interface persistence is enabled, all ImpAKT interfaces will remember the settings you entered the last time.

All ImpAKT interfaces feature contextual help—short instructions that guide you through configuring each setting.

In addition to these settings, you can also configure an image processing library, perform product upgrades, or update your site.

The Login Settings section is the place where you configure all user authentication options, including database information, password encryption, user levels, and redirect pages. Follow these steps to configure the login settings:

- In the InterAKT Control Panel, click the Login Settings icon. This opens a dialog box showing four tabs: Options, Database, Session, and User Levels.
- Configure the Options tab: For increased security, passwords will be encrypted in the database. For this example, the forum will not use access levels, so set the restriction to use user name and password only.
- Leave the Auto Login Validity value at its default setting. Users will be able to automatically log in to the forum. They will be asked to authenticate again only if 30 days have passed since the

Marius Zaharia is the documentation manager at InterAKT Online, a developer of professional tools for dynamic web development. When he’s not writing articles and tutorials to guide web developers, he enjoys learning new things and exploring new technologies. His interests range from web development to politics and avantgarde electronic music.

Cristian Ivascu is a technical writer with InterAKT Online. He is a strong supporter of open-source software and a fan of Japanese culture and rock music.

last time they visited the forum.

- In the Database tab, define which table stores the user information and what columns to use for authentication:

Select the database connection (conForum) and the table that stores user information (user_usr). The primary key is automatically detected and set to id_usr.

From the Username pop-up menu, select the name_usr column.

From the Password pop-up menu, select the password_usr column.

From the E-mail pop-up menu, select the email_usr column.

From the Active pop-up menu, select the active_usr column. If you specify a column that stores the account status (active/inactive), account activation will be available. When a user registers an account, an e-mail message will be sent containing an activation link. Unless the user clicks that link, the account cannot be used.

In the Random key pop-up menu, select the randomkey_usr column.

A random key consists of a randomly generated set of alphanumeric characters used for improving website security. A random key is commonly used in user account activation, online purchases, or other activities that use URLs that ought to be difficult to guess. This prevents potentially malevolent users from accessing the URLs by directly entering them in the browser window in order to activate accounts for other users. ImpAKT automatically generates unique random keys for each user and stores them in the specified column.

- In the Session tab, you can define which session variables to create upon login and what information to store in them. By default, two variables will be created: kt_login_id (stores the user's unique ID) and kt_login_user (stores the user name). You don't need additional session variables for this tutorial, so leave this tab as is.
- Because the forum does not use access levels, you need to define on the User Levels tab only those pages to redirect the user to. You have to set the login page and the pages to be redirected to when authentication succeeds or fails. Use the login.php page from the site root for the Login Page and Default Redirect on Fail text boxes. When the login is successful, the user must be

redirected to the forum home page.

- Click OK to apply the changes and then click Close to exit the InterAKT Control Panel.

Because sending e-mail notifications for account activation is part of the user registration process, you will also need to configure the e-mail settings as explained in the following section.

Configuring E-mail Settings

Forum users will receive several automatic e-mail messages:

- Account activation e-mail
- Welcome message after activation
- Password reminder e-mail
- Topic reply notifications

This is why you need to set an outgoing e-mail server. You can configure e-mail settings from the InterAKT Control Panel by clicking the E-mail Settings section.

In the dialog box that opens, you can configure the following:

- Outgoing server's name or IP address; if you are not sure about this, consult your network administrator
- Port number (25 by default)
- Username and password used to log in to the mail server (if required)
- Default sender's e-mail address; this will appear in the From field in all message headers sent from the forum

Click OK to save the configuration, then click Close to exit the control panel. You have finished configuring user authentication for your site.

In the next section, you will build the user registration form.

Building the User Registration Page

To post messages, a user must have an active account. In this section, you will build the page that enables users to register to the forum by filling in a user name, password, and valid e-mail address, and by uploading a profile photo—all without writing a single line of code.

The page you will work on is register.php. The page layout is already created and is similar to the other forum pages.

You will build the registration form with the User Registration Wizard and then add a file upload feature with the Upload and Resize Image server behavior from ImpAKT.

Using the User Registration Wizard

To build the registration form, follow these steps:

- Delete the placeholder text ("Register form here") from the main <div> tag.
- Go to the MX Kollektion tab of the Insert bar and click the User Registration Wizard icon to open the wizard.
- Step 1 of the wizard displays the connection and table settings you made in the InterAKT Control Panel.
- Click Next to proceed to Step 2, where you can define the form fields to display. The Form fields grid displays all the table columns that can be displayed as form elements. You can add or remove fields from the grid by clicking the Plus (+) and Minus (–) buttons on top of the grid.
- Remove the active_usr and randomkey_usr fields from the grid. Select each of them and click the Minus (–) button.
- Now do the following:
 - The name_usr field is already configured. It will be displayed as a text box labeled "Name".
 - Select the email_usr field, and change its label from Email to E-mail.
 - From the Display As pop-up menu, select the photo_usr field and change it to a File Field control to allow users to upload their avatar image.
 - The password_usr field is already configured to be displayed as a password field, so you can move on to the next step of the wizard by clicking Next.
- In Step 3, you will define the validation rules for each form field. Because you configured the login settings in the InterAKT Control Panel, the validation rules for the user name, e-mail, and password fields are already in place. All three fields are required. The validity of the e-mail address is verified (which means users can't type bogus text that is not an e-mail address). You do not need to change any settings in this step.
- In Step 4, you can define extra features for the registration:
 - To send a welcome message to a new user, check the Send Welcome E-mail option.
 - To enable account activation, check the Use Account Activation option. The

wizard will handle all the necessary details such as generating random keys, creating the activation page, and sending the activation e-mail message.

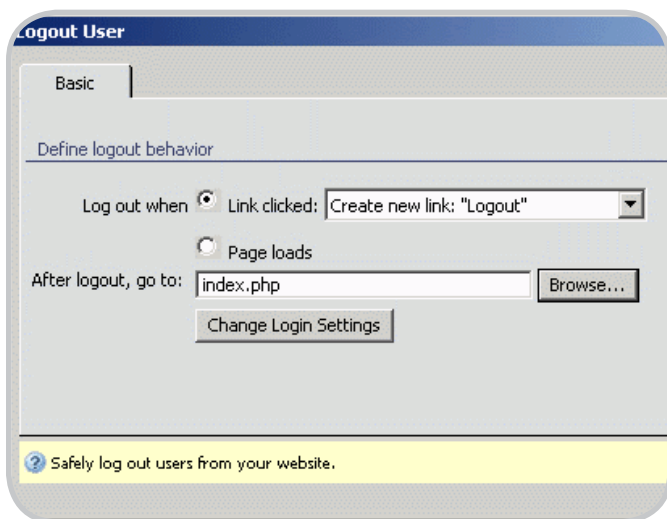
- Click Finish to complete the wizard and apply the changes.

The wizard generates the registration form with two password fields, creates the activate.php page that handles account activations, and adds the server behaviors that verify whether the user name is unique, the e-mail address is valid, and the two passwords match.

Save your page and upload all files to your testing server. If you preview the page in a browser.

Notice that required fields are marked with an asterisk (*) and the E-mail field has a hint indicating the correct format. Put your form to the test to see how it works. Try to register an account using an incorrect e-mail address. An error message will be displayed next to the e-mail field, but the data you entered is not lost.

You can also test the form to see what happens if you type two different passwords or if you try to register an account using a user name that already exists in the database.



Adding a File Upload Feature

Although you can choose a file in the Photo field, the image is not uploaded to the server. To build a functional file upload, you will use the Upload and Resize Image server behavior:

- Open the register.php page in Dreamweaver, go to Server Behaviors

tab, and select Plus (+) > MX Kollektion > File Upload > Upload and Resize Image.

- In the Basic tab, define where to get the images from and what to do with them:
- From the Form Field pop-up menu, select the form control for uploading files. You can only select form1.photo_usr because this is the only file field on your page.
- From the Table Column pop-up menu, select the column that will store the filename: photo_usr
- From the Upload Folder pop-up menu, enter the path where the uploaded images will be saved. Click Browse and select the profiles folder.
- In order to save space on your server, resize the photos while keeping their aspect ratios. Choose Proportional – Fit to Box and make them 80 pixels wide by 80 pixels high..
- In the File tab, you can specify various security settings: the maximum file size allowed and the list of permitted file extensions. Use the default values for these options. You can also specify the preferred overwriting conflict resolution. Leave the default Automatic Renaming method. (If a user uploads a file having the same name as another photo on the server, say angel.gif, it will be automatically renamed to angel_1.gif.)

You don't need to configure anything in the Advanced tab (this is where you can register the file upload action to more transactions and assign it a priority). Click OK to apply the changes.

The image upload feature now works. Try registering a new account and uploading a photo. The photo is displayed when the user posts a message.

After registration, you will also receive two e-mail messages—one asking to activate your account and a welcome message.

You can edit the contents of the two e-mail messages by editing the corresponding HTML files located in the includes\mailtemplates\ folder.

In the next section, you will create the login form.

Building the Login Page

The layout of the login page is already created. You only have to apply the Login Form Wizard:

- Open the login.php page in Dreamweaver.
- Select and remove the placeholder text ("Login form here").
- Go to the MX Kollektion tab of the Insert bar and click the Login Form Wizard icon.
- This wizard consists of two steps. The first one is purely informative, displaying the settings you made in the Control Panel. Click Next to continue.
- In Step 2 configure specific login options:
 - To enable users to remain logged in, select the Create "Remember Me" Checkbox option.
 - To build a page that automatically generates a new password for a user and sends it by e-mail, select the Create "Forgot Password" Page option.
- Click Finish to complete the wizard and apply the changes.

The wizard automatically generates the application logic to verify whether the user and the password are correct and whether the account is active. It also generates the forgot_password.php file and a link to it in the login form.

Save the page and upload it to the server. Make sure you also upload the forgot_password.php file. You can test the login form in your browser to see how it works. If you log in as an inactive user (user name: Angel, password: root), you will receive a warning message.

In the next section, you will create a logout link that safely destroys all session variables.

Creating the Logout Link

Although an automatic logout takes place when the session expires or the browser window is closed, you must add an explicit logout link which enables users to clear the session information safely. You must add this link to the following forum pages:

- reply_message.php
- view_message.php
- post_message.php
- index.php

The following steps explain how to create the link on the forum home page.

CFDynamics

Dedicated Server Packages Starting at \$189/mo.

All dedicated servers include:

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

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

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


Real Freedom.

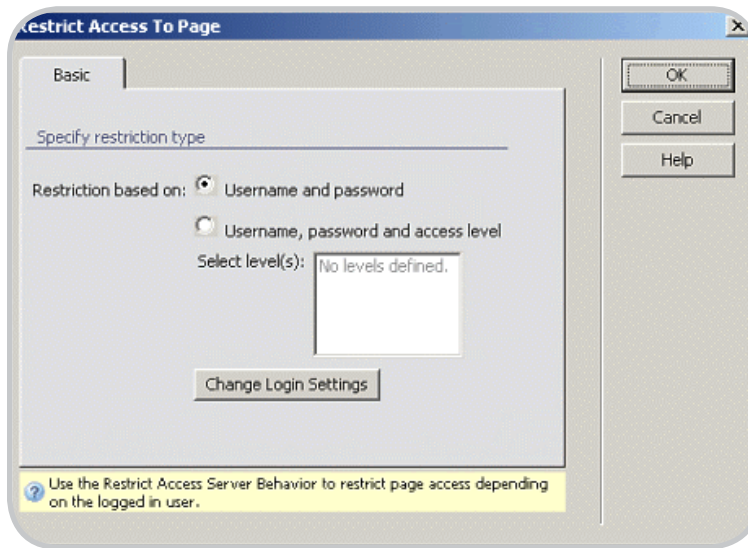


Visit us online or call to order!

 macromedia
ALLIANCE PARTNER

Paper|Thin Partner

 **BlueDragon Certified**
V.I.P. Hosting Partner



You will need to repeat the steps for the other pages as well:

- Open `index.php` in the Code view.
- Place the cursor after the Login link in the top-right `<div>` container (the one with the white links inside).
- Apply the Logout User server behavior from the Server Behaviors tab by selecting Plus (+) > MX Kollection > User Login.
- Configure the dialog box:
 - Select the first option button; the the logout operation will be triggered by a link. From the pop-up menu, select Create New Link: "Logout".
 - After users log out, they will be redirected to the forum home page. Click the Browse button and select the `index.php` page.
 - Click OK to apply the server behavior.

Notice that the Logout link is created.

Repeat these steps for the `reply_message.php`, `view_message.php`, and `post_message.php` pages.

Not all links must be visible all the time. Logged-in users have no need to see the Login and Register links, just as the Logout link means nothing for the user who hasn't logged in or doesn't have an account. In the next section, you will use conditional regions to control which links are displayed and when.

Hiding Content with Conditional Regions

To control what links are displayed when users are logged in and what to display when they are not, you will use a con-

ditional region added by ImpAKT: Show If User Is Logged In. You can access it from the Server Behaviors tab by selecting Plus (+) > MX Kollection > Conditional Regions.

Follow these steps to display the logout link for logged-in users and the others for the rest:

- Open the `index.php` page.
- Select the Logout link in the top-right corner.
- Apply the Show If User Is Logged In server behavior. For the restriction type, select the Username and Password option. Check the Has ELSE option. Text in the ELSE region will be displayed if the condition is not met (the user is not logged in).
- Click OK to apply the server behavior. In Dreamweaver, a gray border appears around the Logout link, along with some placeholder text (Else text: Replace this).
- Select the Register | Login text and cut it from the page.
- Select the placeholder text (Else text: Replace this) and remove it. Then paste the Register and Login links from the Clipboard.
- Save the page and preview it in a browser. Because you are not logged in.

Try to log in using one of the sample user accounts. Notice that only the Logout link is visible. Repeat the previous steps for the `view_message.php`, `post_message.php`, and `reply_message.php` pages.

That's it! You have successfully built a user authentication system for your forum. ☺☺

Adobe Has Inherited Web History from Macromedia...

— continued from page 7

tions. Especially since it is Kevin himself who oversees Adobe's developer relations program.

As he wrote just recently, in the last issue of MXDJ—in the article based on his industry-respected blog:

"There is clearly a resurgence in how HTML can be used to deliver application user interfaces and terrific progress has been made on that. In addition, Flash brings capabilities that HTML doesn't currently have, and they can be used together to great benefit -- in fact, Flash has already been architected to fit perfectly in the Web 2.0 model.

For example, Adaptive Path has been working on a great new application called MeasureMap that helps people track traffic on their blogs and is being built with a combination of HTML and Flash on the client. Another is how Flickr is using both HTML and Flash, for example implementing the organizer and slideshow with Flash and the photo index with HTML. The language in Flash is ActionScript which is the same as JavaScript, both ECMA standard languages, and it's very simple to call between code in HTML and Flash, enabling smooth integration with a free open-source integration kit."

In the February issue of MXDJ, I'm delighted to give you a heads-up that Adobe's Christian Cantrell—the author of numerous tutorials and white papers as well as being coauthor of *Flash Enabled: Flash Design & Development for Devices* and of the *JavaScript/Flash Integration Kit*—will be writing for us about AJAX to Flash integration, on which he's probably leading expert in the world.

The Adobe product roadmap, mark my words, will continue to keep you as developers far ahead of Microsofties and Ajaxians and all the rest. As it emerges later in the year, you'll be blown away. Meantime, enjoy this issue. And do please keep your 2006 feedback and suggestions coming, by e-mailing 2006@sys-con.com. "None of us is as smart as all of us," as they say! ☺☺

BALANCE

Designer/developer, front-end/back-end, clients/sanity. . .web development is a balance and we can help you maintain it. Join now and experience a wealth of training resources tailored to the tools you use every day.

www.communitymx.com



Visit www.communitymx.com/trial/ for your free 10 day trial.

COMMUNITY
MX



Building an Image Injector

Using ExternalInterface to update HTML
by david vogleer

flash has been communicating with JavaScript for a long time through `getURL` and `fscCommand`, but with Flash 8 it's easier than ever. With the `ExternalInterface` class, you cannot only call JavaScript functions, but also have JavaScript call Flash functions. And now that JavaScript is getting more and more publicity in the form of AJAX (Asynchronous JavaScript and XML), the ability to seamlessly integrate your Flash content within your HTML content is essential.

This article is going to cover some basic uses of `ExternalInterface` and move into some of the more undocumented ways to use it. Then we will finish up by building a small image injector that will take a selected image from one browser window and push it into another HTML page without the use of a `LocalConnection` object.

ExternalInterface Basics

The method being used to make calls to JavaScript is the static `call` method of the `ExternalInterface` class. Here is the basic syntax for calling this method:

```
ExternalInterface.call(function:
string, parameters:object);
```

The first parameter is the name of the function you're calling as a string. And following that parameter are all the parameters (separated by commas) for the function being called. The parameters being passed to the JavaScript function can be any valid `ActionScript` data type. Here is an example of calling a function in JavaScript:

```
//first, import the necessary class
import flash.external.
ExternalInterface;

//call the helloWorld function
ExternalInterface.call("helloWorld",
getTimer());
```

Not only are we calling the `helloWorld` function in JavaScript (which we will write next), but we are passing it the number of milliseconds that have passed before the function is called using `getTimer()`. The next step is to publish the SWF and the HTML file, then open the HTML file and place this code after the closing `</head>` tag.

```
<script>
    function helloWorld(ms){
        alert("It took "+ms+" millisec-
onds to say hello world");
    }
</script>
```

What this function will do is take the amount of milliseconds being passed to it and use it in a statement that will appear in an alert box.

Now you can save the HTML file and publish it along with the SWF file up to a web server to test it, where you should see an alert box similar to the following image.

The reason the files must be on a web server instead of being able to test it locally is because of the Flash 8 security sandbox. The `allowScriptAccess` parameter (found in both the object and the embed tag) in the HTML has a default value of "sameDomain", which means that

the file will not allow calls to JavaScript from Flash while running locally on your machine (or from Flash files running from different domains). You can of course fix that by changing the value of `allowScriptAccess` to "always" in both places.

Returning Values

With `ExternalInterface`, you can also get a return value if the JavaScript function you're calling has one. To get the return value, just set a variable directly to the `ExternalInterface.call()` you are making like this:

```
var squareNum = ExternalInterface.
call("square", 4);
```

Providing you have a JavaScript function that will do the work, the variable `squareNum` will now have a value of 16.

A more practical example might be when a user has forgotten his or her password. Most systems will send an email with either the password or a link to a page where the password can be reset, but some systems require you to be able to answer a simple question that was set by the user such as "What is the name of your pet?" or "What is your mother's maiden name?". These questions are usually presented on a separate page, but they could be easily presented right from the current page in a prompt box like the following example.

As you can see from the following image, the setup for the stage is your general User Name/Password style setup, but the "Forgot Password" Button component has an instance name of `forgot_btn`. And then the `ActionScript` in the first frame will look like this:

```

//import the class we need
import flash.external.
ExternalInterface;

//the question and answer
var question:String = "Who is your
favorite cartoon character?";
var answer:String = "Bender";

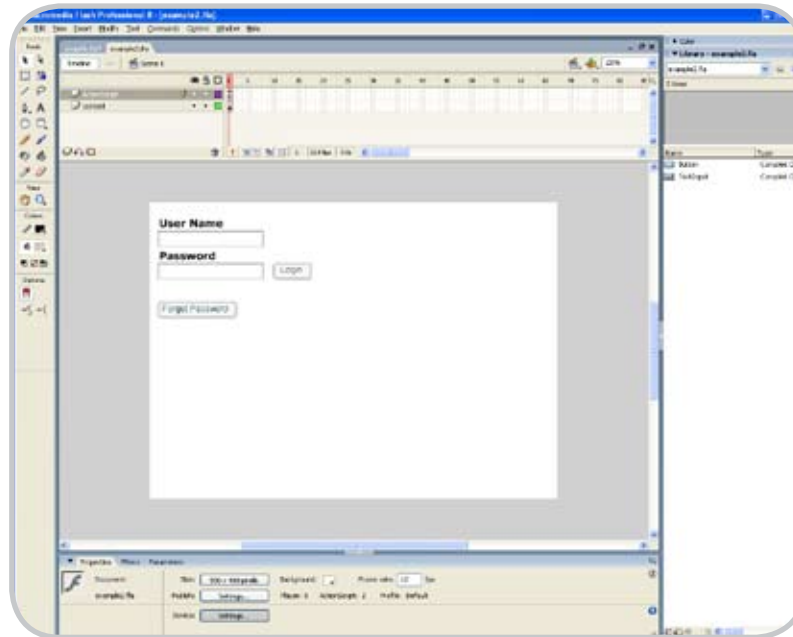
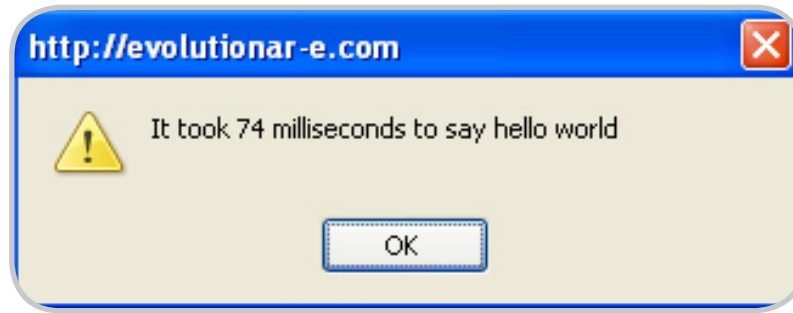
forgot_btn.clickHandler = function(){
    //call the prompt and wait for the
    answer
    var returnedAnswer =
ExternalInterface.call("prompt", ques-
tion);
    //make sure the user didn't hit
    cancel
    if(returnedAnswer){
        if(returnedAnswer == answer){
            var msg:String = "That is
correct! Your password has been
emailed to you.";
        }else{
            var msg:String = "That is
incorrect!";
        }
        //send the results to an alert
        box
        ExternalInterface.call("alert",
msg);
    }
}
}

```

The code above should look somewhat familiar. It initially imports the class we need to work with JavaScript. Then both the question and the answer variables are set. Next is the event for when a user clicks the "Forgot Password" button, which directly calls the prompt function without any need for a custom JavaScript function. Once a user has filled in the answer to the prompt question, the result is returned to returnedAnswer for comparison against the real answer. As long as the user does not press the cancel button in the prompt, they will receive an alert telling them whether or not the answer is correct.

You can publish this file and the HTML up the server, and then click "Forgot Password" to see a prompt similar to the following image. And of course this technique becomes even more powerful when hooking it into a backend database to make it dynamic.

As mentioned earlier, notice how the



code in this example has no need for custom functions in JavaScript. Instead, you can directly call all the available JavaScript functions directly from Flash, but not just functions.

Setting Properties

Although it is undocumented, you can in fact set properties in JavaScript right from Flash using ExternalInterface. The key to setting properties from the call method is using an equal sign (=) after the property name in the first parameter, and then putting the new value of that property as the second parameter. Here is an example that will set the background color of the HTML page to red:

```

ExternalInterface.call("document.
bgColor=", "#ff0000");

```

As a more practical example, you can also set properties of individual elements using first the getElementById method of the document object and then set the content of that element using the innerHTML property.

The first thing you will need in this new file is a TextArea component and a Button component. Give the TextArea an instance name of text_ta and the Button an instance name of submit_btn with the label property of "Submit" like the figure below. Then the ActionScript to make it work looks like this:

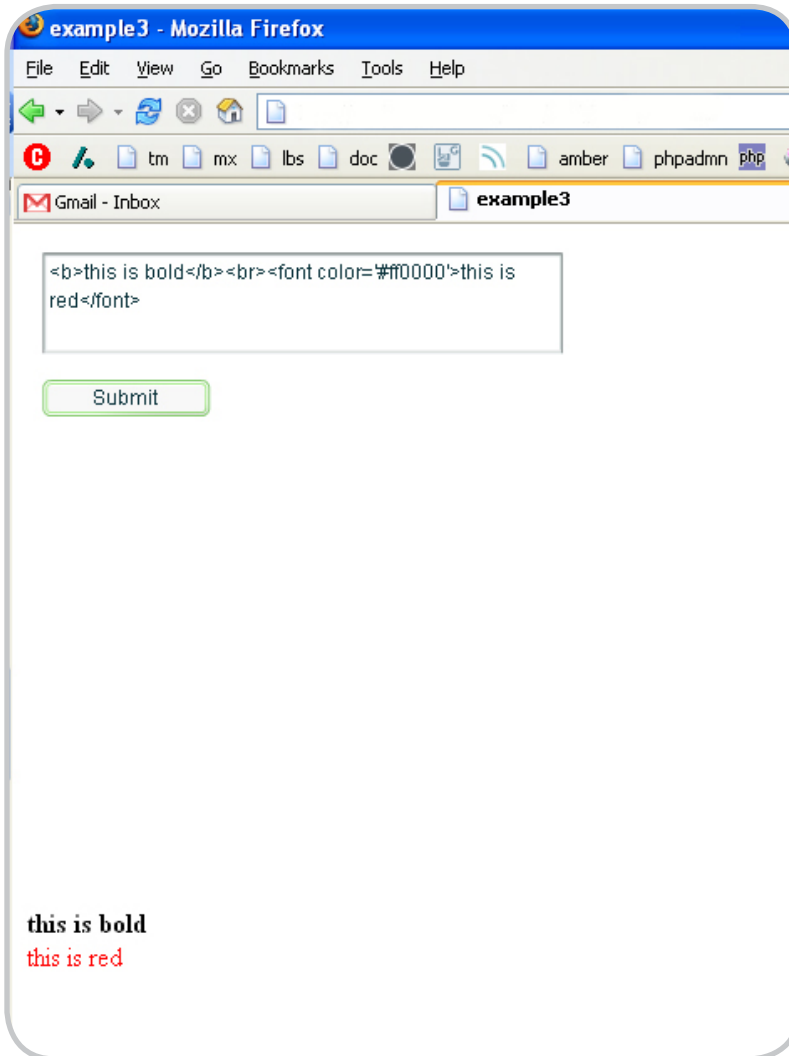
```

//import the class we need
import flash.external.
ExternalInterface;

//when the button is clicked, send
the text
submit_btn.clickHandler = function(){
    ExternalInterface.call("document.
getElementById('content').innerHTML=",
text_ta.text);
}

```

Notice we are looking for an element with the id attribute of "content". We have to put that into the HTML next, so publish both the SWF and the HTML. Then open the HTML doc and add this line after the closing </object> tag for the SWF:



```
<div id="content"></div>
```

This is where the content is going to appear. Save the file and push both the HTML and the SWF to a web server (or change the allowScriptAccess properties to "always" for local testing) and test it. You will see that when you click the submit button, the text in the TextArea component will appear under the SWF. And because your sending text that will render as HTML, you can use HTML tags like the following figure shows.

You can also change properties and call functions in another HTML window, as you will see in the final example.

Building the Image Injector

Now let's take what has been covered so far and apply it to a real world example by building an image injector. The idea for this example is that there can be a pop up window that will control an image in the opening window. This is useful for testing different looks of a page, or

in conjunction with a content management system (which is why the injector was originally created).

The first part of the injector example will be the page that will be receiving the new images. Open up Dreamweaver, start a new HTML page and place this code in it:

```
<html>
<head>
<title>My Favorite Picture</title>
</head>
<script>
function openInjector() {
    window.open('swapPic.html', 'changer',
        'toolbars=no,resizeable=no,scrollbars=
no,width=410,height=410');
}
</script>

<body>
<h3>This is my Favorite image:</h3>
<a href="javascript:
openInjector();"><div
id="imageLoader"><img src='imgs/water.
jpg' border="0" /></div></a>
<p>(click the image to change it)</p>
</body>
</html>
```

The above HTML does a few things that are important. First it creates the JavaScript function for when someone clicks on the image. That function will open the pop up, which we will create later ("swapPic.html"). We also create the <div> tag we need with a default image already in it. And the entire div is wrapped with an <a> tag that will make it so when the user clicks the image, the JavaScript function will be called, and the pop up will appear.

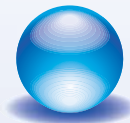
Before the actual injector is built, there are a couple of PHP pages that will make the injector easier to manage. The first is a page that will grab all the JPEG images from a given directory and return them to Flash. The second page will create thumbnails of those images for displaying in Flash. Because the full size images can be of different size, it's important to make the thumbnails consistent for being displayed in the injector.

```
fileList.php
```

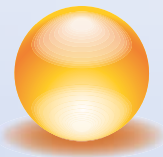
```
<?php
```


H  S T I N G . c o m

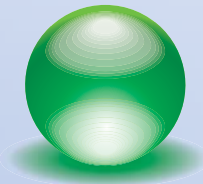
THE FIRST NAME IN HOSTING



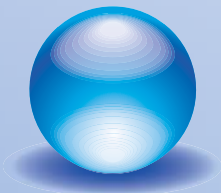
Developers!



Developers!



Developers!



We Love Them!

SERVE. SUPPORT. SECURE. STORE.

www.hosting.com

d

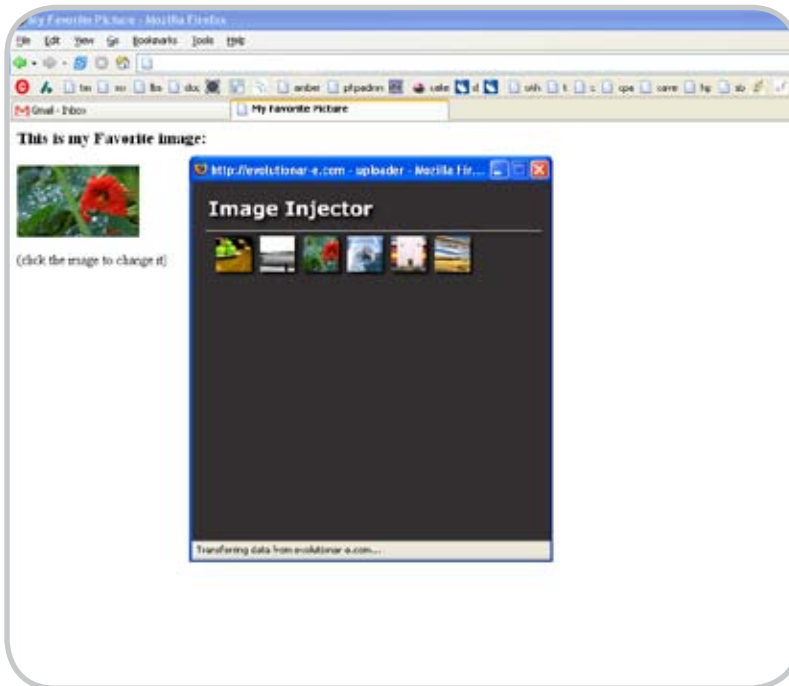
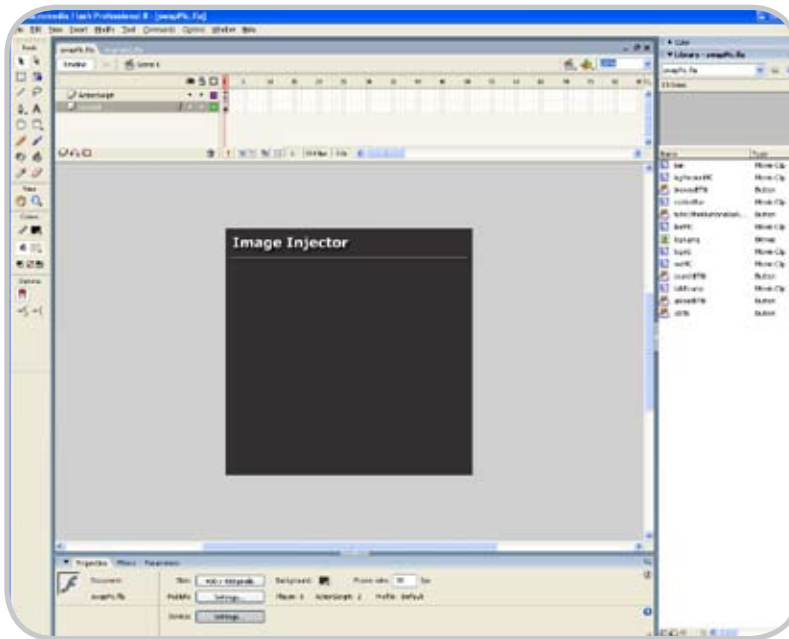
f

fu

fb

fd

db



```

$fileList;
$picDir = $_POST['sentDir'];//get the
directory
$dDir = opendir($picDir);//open the
directory
while($file = readdir($dDir)){
    //we only need the JPEGs for this
example
    if(substr_count($file, ".jpg") >
0){
        $fileList .= $file . "~";
    }
}
//close the directory
closedir($dDir);

```

```

//send the files back to Flash
echo "files=" . $fileList
?>

```

This file opens the directory passed to it from Flash, and returns a string of all the JPEG images separated by a tilde (~) character.

```
makeThumb.php
```

```

<?php
// The file, thumbnail size and zoom
level
$filename = $_GET['sentFile'];
$thumbSize = $_GET['sentSize'];

```

```

$imgZoom = $_GET['sentZoom'];

// Content type being returned
header('Content-type: image/jpeg');

// Get new dimensions
list($width, $height) =
getimagesize($filename);
if($width > $height){
    $new_width = $height * $imgZoom;
}else{
    $new_width = $width * $imgZoom;
}
//Set the starting point for the
thumbnail creation
$SX = ($width-$new_width)/2;
$SY = ($height-$new_width)/2;

// create the image from the original
$image_p = imagecreatetruecolor($thumb
Size, $thumbSize);
$image = imagecreatefromjpeg($filena
me);

```

```

imagecopyresampled($image_p, $image,
0, 0, $SX, $SY, $thumbSize, $thumb
Size, $new_width, $new_width);

```

```

// output JPEG
imagejpeg($image_p, null, 100);
?>

```

When this file is called from Flash, you pass it three parameters:

- **sentFile:** the path to the JPEG file to use.
- **sentSize:** the width and height the thumbnail needs to be returned in.
- **sentZoom:** a floating point number between 1-0. The smaller the number, the more it zooms in.

This file does a lot. First it gets the current dimensions of the image. Then it uses those dimensions to get the largest square portion of the center of the image it can. Finally, it creates the new image and returns it back to Flash as a JPEG.

Now the HTML and the PHP pages are done, the final piece is the injector itself. Before anything is created on the stage, save this file as "swapPic.fla". Then create a second layer and call the top one ActionScript and the bottom one content. Reset the stage to 400x400 to fit nicely in the pop up. There is only a small header and a hori-

zontal bar in the content layer because most of the interface will be built with ActionScript.

In the ActionScript layer, place this code:

```
import flash.external.  
ExternalInterface;  
import flash.filters.DropShadowFilter;  
  
//a filter to give the thumbnails a  
little depth  
var myDrop:DropShadowFilter = new  
DropShadowFilter(4,45,0x000000,100,4,  
4,1,3);  
  
//the directory to get images from  
var dir:String = "imgs";  
//main path to files  
var mPath:String = "http://www.yourS-  
erver.com/";  
  
//the function called when a thumbnail  
is selected  
function sendFile(fl){  
    ExternalInterface.call ("opener.  
document.getElementById('imageLoader  
').innerHTML=", "<img src='"+dir+"/  
"+fl+"' border='0' />");  
}  
  
//the LoadVars object for collecting  
the image names from fileList.php  
var imgs_lv:LoadVars = new LoadVars();  
imgs_lv.onLoad = function(success){  
    if(success){  
        //the initial setting for col-  
umns and rows  
        var c = r = 0;  
        var imgs_array = this.files.  
split("-");  
        var tLength = imgs_array.length  
- 1;  
        var i = -1;  
        while(++i < tLength){  
            var img:MovieClip = createEm  
ptyMovieClip("img"+i+"_mc", i+1);  
            img.createEmptyMovieClip("ta  
rget_mc", 1).loadMovie(mPath+"makeThum  
b.php?sentFile="+dir+"/"+imgs_array[i]  
+"&sentSize=40&sentZoom=1");  
            img._x = (c*50)+20;  
            img._y = (r*50)+55;//the  
plus 55 is to fit under the title at  
the top  
            c++;  
            if(c == 7){  
                c = 0;
```

```
                r++;  
            }  
            //when a thumbnail is  
selected, call the function  
            img.file = imgs_array[i];  
            img.onRelease = function(){  
                sendFile(this.file);  
            }  
            //set the dropshadow  
            img.filters = [myDrop];  
        }  
    }else{  
        ExternalInterface.call("alert",  
"An error occured with the images.");  
    }  
}  
//go get the images  
imgs_lv.sendDir = dir;  
imgs_lv.sendAndLoad(mPath+"fileList.  
php", imgs_lv, "POST");
```

Most of this code is for controlling the alignment of the thumbnails. The first part grabs the two classes we need, including the Flash 8 DropShadow class to add some depth to the thumbnails. After that, we create a variable to hold the directory we are going to pass to the PHP script as well as a variable to hold the main path to the PHP scripts. The next part is the function that will be called when a thumbnail is clicked. Notice that in the call method, we are not only passing the property we want to access (the innerHTML property of the 'imageLoader' element), but we are also using the keyword opener before we access it. What this keyword does is tell JavaScript we want to access the element in the HTML page that opened this HTML page. This means that not only can we access all the properties of the current HTML page we're in, but other HTML pages as well (provided they are on the same domain when setting allowScriptAccess to "sameDomain").

After that, a LoadVars object is created to handle the communication between the PHP and Flash. Within its onLoad event, a while loop is used to cycle through all the images and create thumbnails. This is where the makeThumb.php script comes in handy, because we know each image being returned will consistently be the same size. While we are creating

the thumbnails, we set the onRelease event for them so when they are clicked, the sendFile function will be called updating the image in the other HTML page. Finally, we set the sentDir property of the LoadVars object and fetch all the images in that directory.

The only thing left to do is to publish and test it. So after you load the SWF, HTML and PHP pages (or pages that do the same thing as the PHP ones) to your server, you should be able to load the initial page, click on the image, which will open the Image Injector pop up, and swap out images in real time to see which one fits the best.

What's next? As much as I hate to use a cliché, you are bound only by your imagination. From some of the techniques in this article, you can see that not only do you have total access to JavaScript from Flash, but you can call functions, methods and even set properties of different pages from a single pop up window. Imagine being able to control layout and colors of one window from another, it will save a lot time on trial and error designing. ∞

*David Vogeleer is the author of the newly release Flash 8 Professional Programming Unleashed from Sams publishing. He has been certified as both a Flash *Developer and Instructor and enjoys speaking and writing about Flash whenever possible. David currently works as a multimedia specialist at OSEC while still maintaining EMLlabs.com, a site dedicated to real-world Flash usage that he co-founded in 2004 and writing for FlashMagazine.*



Creating Fireworks Panels

Custom panels that unlock hidden functionality
by trevor mccauley

When it comes to designing and optimizing web graphics, it doesn't get much better than Macromedia Fireworks—given its available tools and workflow. However, Fireworks is not necessarily limited to the tools that come with it in the default installation. You can modify the interface to include new, custom panels that unlock hidden functionality, introduce new functionality, or simply improve your workflow.

As a developer and designer, I find such extensions to the Fireworks interface invaluable in helping me quickly and easily obtain my graphics goal. In Part 1 of this tutorial, I will show you how you too can make your own custom panels for Fireworks using Macromedia Flash. Not only will I show you how they are made but I will also cover workflow tips for making panels and techniques to help you successfully debug them.

Understanding SWF Panel Interactions

Custom panels in Fireworks are made with Flash. They consist simply of a Flash SWF file, which Fireworks plays using an internal Flash player embedded within a panel window that looks and operates like any other panel that's native to the application. The SWF provides the panel with its own interface and a means for interacting with Fireworks using ActionScript. If you've ever used the Align panel in Fireworks, you've seen a custom SWF panel at work.

A custom panel window is capable of receiving commands from the SWF it's playing through ActionScript. One ActionScript command in particular

instructs the window to relay information—or, more specifically, a JavaScript command—from the SWF to the main Fireworks application. This command is the `MMEExecute()` command:

```
MMEExecute( JavaScriptCommand:String )
: String;
```

`MMEExecute` has one parameter, a string representing a Fireworks JavaScript command that is to be sent to Fireworks. When Fireworks receives this command, it runs it just as though it were on the Commands menu. When the command has completed, `MMEExecute` also returns any result of that command back to Flash in the form of a string. This allows the SWF to react to any command sent by `MMEExecute` if needed.

In its simplest form, `MMEExecute` is about all you need to create a simple panel and use it to interact with Fireworks. You can see how simple it is with the first example, the Create Ellipse panel.

Example 1: The Create Ellipse Panel

This example represents your basic first panel example, similar to “Hello World” examples you might encounter when learning programming languages. Because Fireworks is a graphics editing program and not a new programming language, I will use an ellipse in place of “Hello World.”

The Create Ellipse panel creates an ellipse in the current Fireworks document. The panel interface consists of a single button that, when clicked, creates the ellipse. As you might have guessed,

`MMEExecute` is used to instruct Fireworks to draw the ellipse when the button is clicked. The code for actually creating the ellipse can be written by hand if you are familiar enough with Fireworks JavaScript programming. But it's easier to extract it from the History panel after you first draw one yourself in Fireworks.

The History panel in Fireworks is much like the history panels in other applications that record your every action, step by step, within the program. In Fireworks, however—as well as other Studio applications such as Flash and Dreamweaver—the History panel records not only your steps but also the JavaScript command(s) used to generate those steps. You can therefore copy the JavaScript associated with any step(s) in your history to the Clipboard and reuse it elsewhere, such as in your custom commands or, as you will see in this example, custom panels.

Creating the Code for the Panel

Here is the procedure to get the code you need to create this panel:

- Open a new Fireworks 8 document. Within it, draw an ellipse that you would like Create Ellipse to generate for you.
- In the History panel, find the Ellipse command, select it, and use the Copy icon to copy the script to the Clipboard.
- Launch Flash 8 and open the Create Ellipse template.
- Source Files/Templates/Create Ellipse fla
- It should consist of a Button component with the label “Create Ellipse.”

- In Flash, select the Create Ellipse button and open the Actions panel, adding the following script:

```
on(click){
MMExecute("");
}
```

In the quote string in the MMExecute call, paste the Fireworks JavaScript command you copied from the History panel in Fireworks. You should have something that looks like the following:

```
on(click){
MMExecute("fw.getDocumentDOM().
addNewOval({left:6, top:5, right:98,
bottom:61});");
}
```

Publish your movie to create the SWF. You now have a SWF containing a button that, when clicked, uses MMExecute to send a command to the Macromedia application running the SWF. Ideally this should be Fireworks but simply publishing the SWF does not make Fireworks recognize it as a panel.

Making Fireworks Recognize the SWF

Fireworks recognizes custom panels by scanning for SWF files located in the Configuration/Command Panels folder within the Fireworks installation folder whenever the application launches. Each SWF located there is interpreted as a panel and is listed along with those available in the Windows menu. The name given to the panel in Fireworks is simply the SWF's filename, not including the extension.

For your Create Ellipse SWF to be recognized as a panel, you must put it in the Command Panels folder:

- Locate your Create Ellipse panel's SWF. If you are using the default publish settings for your Create Ellipse Flash document, it should reside in the same folder as the FLA itself.
- Find the Configuration/Command Panels folder in your Fireworks 8 installation folder.
- Place a copy of your published Create Ellipse.swf file in the folder:
- (Windows) C:\Program Files\ Macromedia\Fireworks 8\ Configuration\Command Panels (Mac

OS) Macintosh Hard Drive:Applications: Macromedia:Fireworks 8:Configuration: Command Panels

- Restart Fireworks.
- Open the Windows menu and select Create Ellipse.
- You should now see Create Ellipse.swf within the Fireworks interface as a panel.
- Create a new Fireworks document.
- Click the Create Ellipse button.

You should see a new ellipse created in your Fireworks document. Its size and shape should match exactly the one you created previously in Fireworks in order to get the command code from the History panel.

Workflow Tips for Fireworks JavaScript

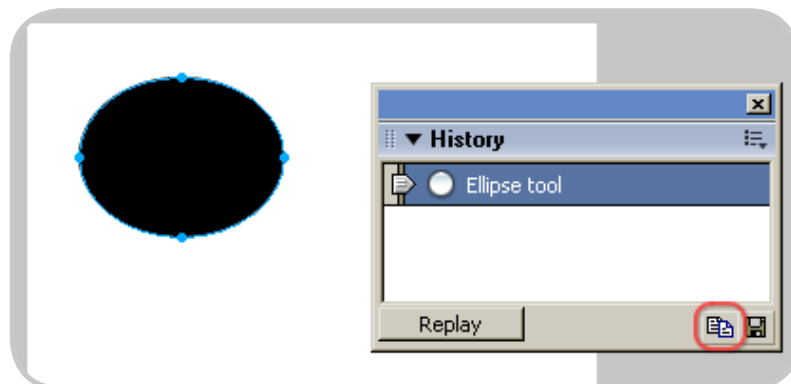
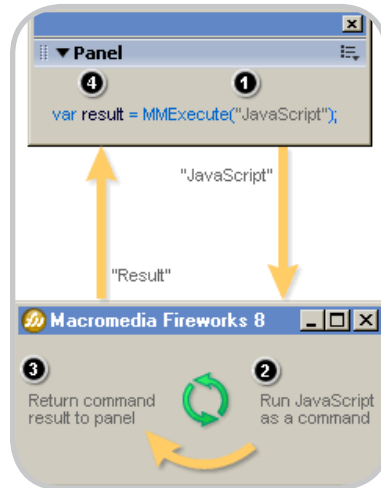
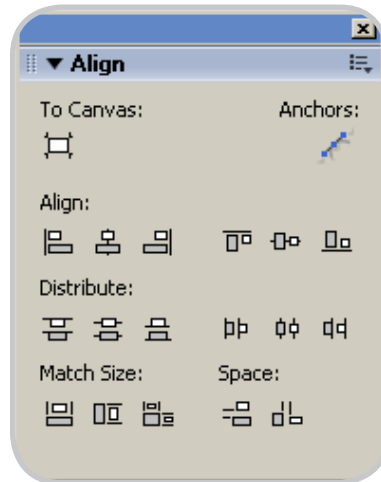
The Create Ellipse panel was pretty straightforward. As you begin to make more complex panels, however, it will become important to develop a process that makes creating and testing your panels as painless as possible. This is important because you're not only dealing with two applications but also two programming languages that must interact to make your custom panels work. Developing a good workflow can help make the process run smoothly.

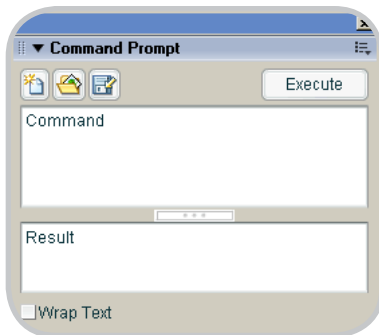
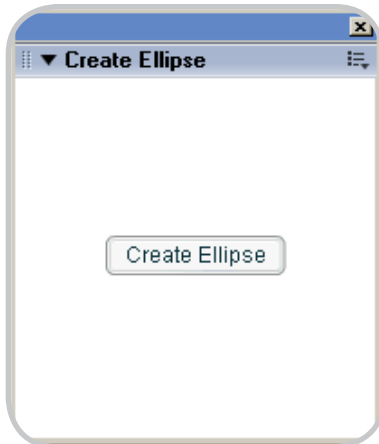
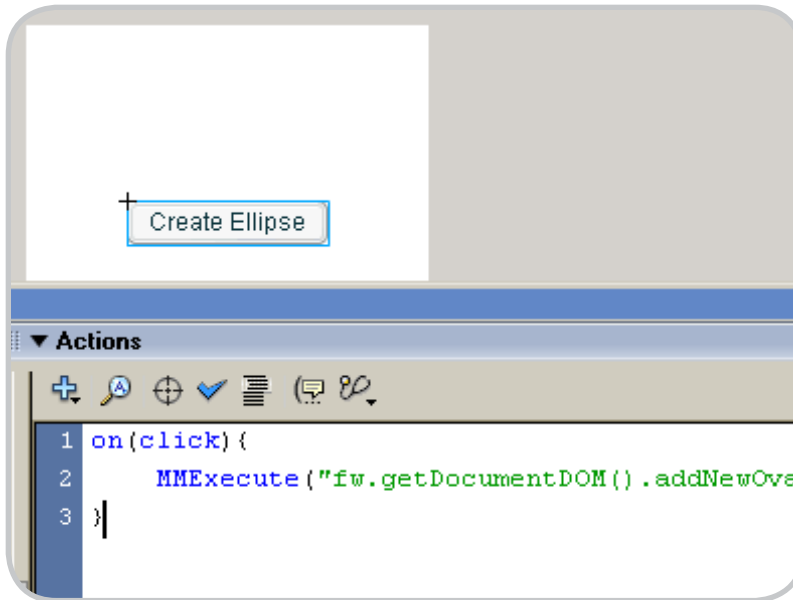
Test Fireworks Scripts in Fireworks First

Although custom Fireworks panels use Flash SWF files to operate, the real functionality of the panel lies within the Fireworks JavaScript code that MMExecute sends to Fireworks from the SWF. When you decide to make a panel, it is often a good idea to start by creating Fireworks JavaScript scripts separately and testing them directly within

Fireworks without a panel interface. That way, when you start working in Flash, you can at least know that you are working with functional Fireworks commands.

Dreamweaver 8 happens to be a great editor for creating and testing Fireworks JavaScript files. Although Dreamweaver 8 supports Fireworks JavaScript files, it does not inherently treat them like other JavaScript (.js) files. Included with this tutorial's source files is an extension which allows Dreamweaver to recognize JSF files as normal





JavaScript files, providing code coloring and code hints:

- Open DW_JSFsupport.mxp with the Macromedia Extension Manager.
- Restart Dreamweaver.

Also included with the source files is another extension, a Command Prompt panel for Fireworks. This custom panel provides a very simple text editor for writing and executing Fireworks JavaScript from directly within Fireworks.

Here's how you install the Command Prompt extension:

- Open FW_CommandPrompt.mxp with the Macromedia Extension Manager.
- Restart Fireworks.
- Access the Command Prompt panel from the Window menu in Fireworks.

Use the Fireworks History Panel to Help

As you saw when creating the Create Ellipse panel, having access to the commands created by Fireworks through its History panel is a great time-saver. Use it whenever you can to create commands quickly for your scripts rather than typing them by hand. You may even be able to learn something new by looking at the code generated by the History panel.

Keep a Fireworks JavaScript Reference Handy

Many Fireworks commands go beyond what you can retrieve from the History panel. When you create custom scripts, it is always a good idea to keep the Extending Fireworks documentation open and available for quick reference (available from the Help menu in Fireworks 8). Chances are good that you will need it—and need it often.

I suggest downloading the PDF version (ZIP, 1.2 MB) instead of the Help system. I find that searching within the PDF is more accurate than searching through Help. Plus you get to have every page open and easily accessible, unlike the segmented Help system. Be sure to keep up with the Fireworks 8 LiveDocs for

any corrections and suggestions made by other Fireworks users.

Reuse Your Scripts

When you work with Fireworks scripts, you may find that some scripts come in use over and over again. Keep a repository of these scripts so that you can easily reuse them if needed. If you are using Dreamweaver to write your scripts, make use of Dreamweaver's code snippets feature. For more information about the Snippets feature in Dreamweaver, see Working with Code Snippets in the Dreamweaver LiveDocs.

Publish Your Panel SWF Directly to the Command Panels Folder

By default Flash publishes a SWF in the same folder that the FLA is located. For Fireworks to recognize a SWF as a panel, however, it needs to exist within the Command Panels folder located in the Fireworks installation folder. To avoid the hassle of manually putting it there yourself, you can instruct Flash to publish the SWF directly to that folder. There are two ways to do this:

- Save your panel FLA within the Command Panels folder so that each time the SWF is published, it is automatically published to that folder as well.
- Recommended: Save your panel FLA wherever you want but adjust your FLA's publish settings (in Flash choose File > Publish Settings) by selecting the Command Panels folder as the destination folder for your published SWF.

Because Fireworks requires a panel SWF to reside in the Command Panels folder during startup, it's a good idea to publish your SWF immediately after creating your FLA, followed by a restart of Fireworks. That will make Fireworks recognize your SWF as a panel immediately. Then you can republish your SWF as needed to update that panel. Each time you publish, all you need to do is close and reopen the panel to reload the newest version of the SWF used by that panel.

Note: There are two ways to get rid of a panel in Fireworks: closing it and hiding



M E T A L I Q

MetalIQ Components (mCOM)

Creating data-driven interfaces for online applications can be complicated and time-consuming. Developers and designers worldwide need an extensible solution that saves time and headaches. There needs to be a stable, innovative approach to solving common UI problems, such as navigation, forms, filtering and sorting immense amounts of data.

Start your development with MetalIQ Components (**mCOM**). Completely new and developed from the ground up specifically for RIA (Rich Internet Applications) and online experiences, **mCOM** are easy to integrate, lightweight and built on the same industrial strength code deployed across some of the largest websites, portals and web applications in the world.

mCOM allow you to quickly create stable, lightweight data-driven interfaces that are extensible by designers and developers alike, both through extending classes and a simpler visual skinning process. **mCOM** include a fully implemented advanced focus manager that works seamlessly with all of our components, and allows for easy creation of tab groups. For anyone familiar with Flash components, there is no learning curve. In most cases, Flash components can be easily updated to **mCOM** with little or no modifications to your project.

Component List:

- Accordion
- Button
- Check Box
- Color Pickers
- Combo Box
- Data Grid
- Field Set
- List
- Menu
- Numeric Stepper
- Radio Button
- Scroll Bar
- Scrolling Pane
- Slider
- Tab Box
- Text Area
- Text Input
- And more...



mCOM
MetalIQ Components
in sample interface



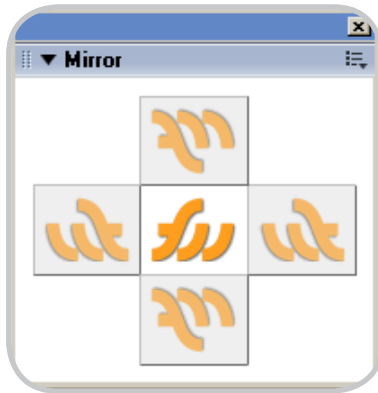
Key Features:

- Up to 75% smaller file size
- Faster download
- Lower CPU usage
- Simplified customization
- Flash Player 6, 7 and 8 compatible
- Reusable
- Easy to integrate

Integrated Technology:

- Effective, understandable code architecture
- Advanced tabbing management
- Improved event model
- Written in ActionScript 2.0
- Advanced focus manager

MetalIQ is a solutions company, specializing in industrial strength creative, code and consulting. Identifying, creating, and implementing innovative solutions that are usable, intuitive and appealing. MetalIQ offers a suite of solutions for video (**mVID**), dashboards (**mDAT**), applications (**mCAP**) and components (**mCOM**). Contact us by phone at 415.642.3332, email us at mCOM@metalIQ.com or visit us at metalIQ.com/mCOM.



it. When you reopen a panel, if it was hidden, it will not reload the SWF. The SWF is only reloaded if the panel is reopened after having been closed. To close a panel you must either right-click its title bar and select *Close Panel Group* or click the close button available to the panel when it's undocked. Hiding a panel happens when you reselect the panel name in the *Windows* menu when it is already opened. For undocked panels, hiding a panel in this manner makes it appear as though the panel were closed.

Test Incrementally

If you are dealing with a fairly complex panel—or one including many operations or options—it will be helpful to test each part of the panel separately. This lets you isolate problems as they occur and helps you deal with them more effectively. Attempting to debug a panel

when you don't know what's causing the problem is much more aggravating.

Debugging Fireworks Commands

Flash has quite a few tools to help developers debug panels specifically designed for it. Fireworks, on the other hand, has nothing of the sort. In fact, if you're lucky Fireworks will probably tell you simply that an error occurred by issuing an alert. This can make Fireworks scripts quite difficult to debug.

There's a two-step process involved when Fireworks receives a Fireworks JavaScript command. First it evaluates the script and then it executes the script. A problem can occur at either step—and either step can produce an error alert. Such alerts, however, will not always be invoked if the script is run from within a Flash panel—a reason why testing outside of the panels themselves can be beneficial.

When Fireworks encounters a problem with evaluation, no script will be executed and an error will be immediately issued. This usually means there's been a syntax error such as mismatched brackets ({}) or a misuse of operators. There will be no indication of where the error occurs. It will be left up to you to find out where that is. More advanced external script editors like Dreamweaver 8 should help you find such errors, thanks to features such as color coding. The following example will create an error in Fireworks due to an error in script evaluation:

```
function copy () {
    fw.getDocumentDOM().clipCopy();
}
copy();
```

The problem with the above script is that the copy function block was not closed with an ending bracket (}). As a result, Fireworks kills the command and issues an error before anything even gets to be executed.

Getting a script to evaluate correctly is half the battle. An error can still occur if there is a problem with execution. If that happens, Fireworks will issue an alert directly after reaching the problematic code. Problems in execution usually result from attempting to use a variable or function that does not exist or calling a native

Fireworks method with incorrect arguments. The following example will create an error in Fireworks due to an error in script execution:

```
function copy () {
    fw.getDocumentDOM().clipCopy();
}
copy();
```

Now the copy function is correctly defined here but when it is called at the bottom of the script, it's misspelled. Because the copie function doesn't actually exist, there is an error when Fireworks attempts to call it, which kills the script and issues the error alert.

Although the error message for both the previous examples is the same ("Could not run the script. An error occurred."), there is a slight advantage in dealing with errors resulting from execution. That advantage is the simple fact that code prior to the faulty code is correctly executed before Fireworks issues the error. This allows you to find out at least how far a script has gone before reaching bad code, or even if it manages to execute any code at all. The most common indicator is a simple JavaScript alert. Consider the following script:

```
alert("Defining copy function");
function copy () {
    fw.getDocumentDOM().clipCopy();
}
alert("Calling copy function");
copy();
alert("Script Complete");
```

When you run this script in Fireworks, you get three alert messages, "Defining copy function," "Calling copy function," and the inevitable error, "Could not run the script. An error occurred." The fact that "Defining copy function" is alerted tells you immediately that the script was evaluated correctly and that there are no errors with the syntax. Seeing "Calling copy function" tells you that copy was defined correctly and that it is about to be called within the script. Given that the error message is seen instead of the "Script Complete" alert indicates that there was something wrong with the call to the copy function. Upon closer inspection, the spelling error can be identified and corrected.

The previously mentioned Command Prompt extension is a great way to help test scripts in Fireworks. Keep in mind, however, that because the Command Prompt extension is a custom SWF panel, it may fail to provide some Fireworks error alerts when it encounters a problem in the script. Including your own alerts when using the Command Prompt is nonetheless beneficial in determining whether a script is correctly interpreted or whether or not it is executed completely and successfully by Fireworks.

Fireworks actually provides a means to help you debug SWF panels as they send the script to the application. It provides two Fireworks JavaScript methods—`fw.enableFlashDebugging()` and `fw.disableFlashDebugging()`—that let you turn debugging mode on and off for commands sent to Fireworks. Don't get too excited, though. All that this debugging includes is an alert of the script being sent to Fireworks from a call to `MMExecute`. Chances are that you, being the panel developer, already know what that script is so using these methods may only help so much. Calling `fw.enableFlashDebugging` turns on this alert feature, while `fw.disableFlashDebugging` turns it off.

Here are some things to consider when debugging:

- Check the spelling of functions and variable names. Capitalization counts.
- Remember that `MMExecute` uses a string for Fireworks JavaScript commands, so watch for quotes and character-escaping when sending a string from Flash to Fireworks. For example, this code will not work:
 - `MMExecute("alert("Hello");");`
- The open quote in the alert text actually closes the quote string used in `MMExecute`. Instead, you can use single quotes or escape nested double quotes with the backslash character:
 - `MMExecute("alert(\"Hello\");");`
- Try to use functions as much as possible. This helps you more easily find where a problem may exist and facilitates code reuse.

If you find something that isn't working right, and you think it should, check the Fireworks LiveDocs. You'll find corrections to the Help documentation there and possibly solutions to whatever

problem you may be encountering. Also feel free to ask questions in the Extending Fireworks Forum.

Now that you have a better understanding of debugging and improvements that can be made to your workflow, it's time to start with a new, slightly more complex example: the Mirror panel.

Example 2: The Mirror Panel

The Mirror Panel allows you to duplicate a selection in Fireworks by creating a mirrored copy of it. It consists of four buttons which lets the user mirror a selected object above, to the right, below, or to the left of its original location.

Before opening the Mirror panel source file in Flash, you should first develop some working Fireworks JavaScript that will perform the mirror action, thereby avoiding any additional complications introduced by Flash. For this example, I presume that you have installed the Command Prompt extension and are using it to test the script. If you do not, you can write your script in Dreamweaver and run it by choosing `Commands > Run Script` from within Fireworks.

Creating the Code for the Panel

Instead of attempting to figure out how to code a mirror from scratch, let Fireworks do much of the hard work for you by manually mirroring an object yourself and then copying the code found in the History panel:

- Open a new Fireworks document.
- Create or import a small test object to mirror. If you create your own, try not to make it symmetrical or you might not be able to tell if you correctly mirrored it.
- Hold down the `Alt` key in Windows (Option key on the Mac) and then drag your test object to create a copy to the right of the original.
- With the copy selected, choose `Modify > Transform > Flip horizontal` to flip it.
- A mirror of the original object has now been created. To see how it happened in terms of Fireworks JavaScript, take the relevant code from the History panel.
- Open the History panel.
- Select the steps that involve mirroring your test object.

- Copy the steps using the Copy icon.
- Open the Command Prompt panel by selecting `Window > Command Prompt`.
- Paste the History steps from the Clipboard to the Command Prompt editor.

You now have the steps required to mirror your test object. To see if they work, delete your test object, reselect the original, and run the steps in the Command Prompt using the `Execute` button. If you copied the steps correctly, your mirrored object should return just as it was before you deleted it.

Adding Variability to the Code

There still remains a small problem with the script as it exists in this form. The problem is that the position of the copy is currently hard-coded—based on where you dragged the copy when using your test object. Ideally the mirror script should work for any object of any size, and place the mirrored copy flush against the edge of the original. To make this happen, the script needs to be edited a little to take in account a dynamic offset based on the selection's width.

Assuming that one object is selected, you can get the selection's width using `fw.selection[0].width` where `fw.selection[0]` represents the first object in the current document's selection. Moving the copy by a distance of the selection's width will place it right next to the original.

With that in mind, edit the script so it looks like the following:

```
fw.getDocumentDOM().
moveSelectionBy({x:fw.selection[0].
width, y:0}, true, false);
fw.getDocumentDOM().
reflectSelection(true, false, "auto-
TrimImages transformAttributes");
```

That takes care of mirroring an object to the right, but this panel should also mirror it to the left, top, and bottom as well. Referencing the script above, you can make versions for each direction.

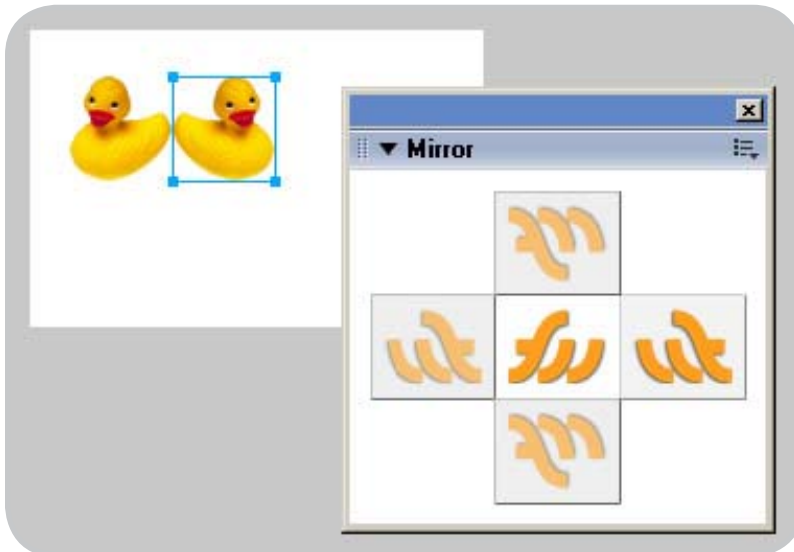
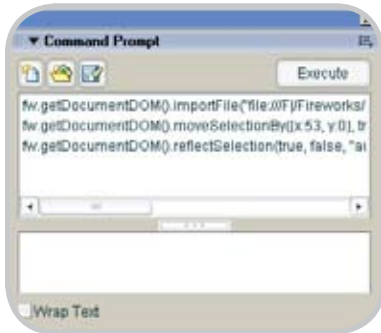
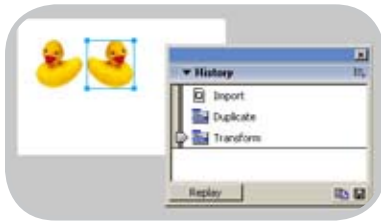
```
Left:
fw.getDocumentDOM().
moveSelectionBy({x:-fw.selection[0].
width, y:0}, true, false);
fw.getDocumentDOM().
```



```
reflectSelection(true, false, "auto-TrimImages transformAttributes");
Top:
fw.getDocumentDOM().
moveSelectionBy({x:0, y:-
fw.selection[0].height}, true, false);
fw.getDocumentDOM().
reflectSelection(false, true, "auto-TrimImages transformAttributes");
Bottom:
fw.getDocumentDOM().
moveSelectionBy({x:0, y:
fw.selection[0].height}, true, false);
fw.getDocumentDOM().
reflectSelection(false, true, "auto-TrimImages transformAttributes");
```

After making sure the scripts work, save them and move on to Flash:

- Open Mirror Template in Flash. It should consist of the interface used for the Mirror panel. You can find it at



Source Files/Templates/Mirror.fla.

- Invoke the publish settings by choosing File > Publish Settings.
- In the Formats tab, select Flash as the only format and set the publish destination to be your Configuration/Command Panels folder in your Fireworks installation folder.
- Publish your SWF.
- Restart Fireworks.

You have now established the Mirror panel in the Fireworks interface. You should be able to open it in Fireworks using Window > Mirror. It won't do anything at this point, but you at least have your foot in the door. To test your updates to the panel, simply close and reopen the panel window. The SWF will be updated with the last SWF you published.

Adding ActionScript to the Code

With all the groundwork set and the panel interface complete, you can start adding ActionScript. Like the Create Ellipse example earlier, the ActionScript for the Mirror panel is pretty basic. You're only dealing with button actions and using MMEExecute to send Fireworks JavaScript commands to Fireworks within them.

Given that there are four buttons, it would be best if the ActionScript was added to the Timeline (as opposed to directly on the buttons—something I did for the Create Ellipse panel for the sake of simplicity and convenience). In each button's onRelease event handler, you can use MMEExecute to relay the Fireworks JavaScript commands to

Fireworks. With the current scripts, however, you're now dealing with two lines of code instead of the one seen in Create Ellipse. That's no problem. Each line can go in its own MMEExecute function.

- In Flash, select the ActionScript frame on the main Timeline.
- Add the following script defining onRelease event handlers to each button: Notice that a single quote (') was used to define the Fireworks JavaScript string sent into MMEExecute. This is to prevent conflicts with the double quotes (") used in reflectSelection for "autoTrimImages transformAttributes". You could have also escaped the double quotes within the script using \". Using single quotes is often easier, however.

```
right_btn.onRelease = function(){
MMEExecute('fw.getDocumentDOM().
moveSelectionBy({x:fw.selection[0].
width, y:0}, true, false);');
MMEExecute('fw.getDocumentDOM().
reflectSelection(true, false, "auto-TrimImages transformAttributes");');
}
left_btn.onRelease = function(){
MMEExecute('fw.getDocumentDOM().
moveSelectionBy({x:-fw.selection[0].
width, y:0}, true, false);');
MMEExecute('fw.getDocumentDOM().
reflectSelection(true, false, "auto-TrimImages transformAttributes");');
}
top_btn.onRelease = function(){
MMEExecute('fw.getDocumentDOM().moveSelectionBy({x:0, y:-fw.selection[0].height}, true, false);');
MMEExecute('fw.getDocumentDOM().reflectSelection(false, true, "auto-TrimImages transformAttributes");');
}
bottom_btn.onRelease = function(){
MMEExecute('fw.getDocumentDOM().moveSelectionBy({x:0, y:fw.selection[0].height}, true, false);');
MMEExecute('fw.getDocumentDOM().reflectSelection(false, true, "auto-TrimImages transformAttributes");');
}
```

- Publish the SWF.
- Close and reopen the Mirror panel in Fireworks.

- Test it.

With any luck, you should have a fully functional Mirror panel in Fireworks.

Debugging the Mirror Panel

Though functionally the panel works fine, you may notice a small flaw if you pay close attention to it. Watch the History panel as you use the Mirror panel; you will see that each time an object is mirrored, two history steps are created instead of one. This is because MMEExecute is used twice each time a button in the Mirror panel is clicked. To keep History steps as condensed as possible, it would be best to reduce this to one.

This new script combines each set of MMEExecute commands into one. Replace it in Flash, republish, and retest:

```
right_btn.onRelease = function(){
    var command = 'fw.getDocumentDOM().
moveSelectionBy({x:fw.selection[0].
width, y:0}, true, false);'
    + 'fw.getDocumentDOM().
reflectSelection(true, false, "auto-
TrimImages transformAttributes");';
    MMEExecute(command);
}

left_btn.onRelease = function(){
    var command = 'fw.getDocumentDOM().
moveSelectionBy({x:-fw.selection[0].
width, y:0}, true, false);'
    + 'fw.getDocumentDOM().
reflectSelection(true, false, "auto-
TrimImages transformAttributes");';
    MMEExecute(command);
}

top_btn.onRelease = function(){
    var command = 'fw.getDocument-
DOM().moveSelectionBy({x:0, y:-
fw.selection[0].height}, true,
false);'
    + 'fw.getDocumentDOM().
reflectSelection(false, true, "auto-
TrimImages transformAttributes");';
    MMEExecute(command);
}

bottom_btn.onRelease = function(){
    var command = 'fw.getDocument-
DOM().moveSelectionBy({x:0, y:
fw.selection[0].height}, true,
false);'
    + 'fw.getDocumentDOM().
reflectSelection(false, true, "auto-
TrimImages transformAttributes");';
```

```
MMEExecute(command);
}
```

Now using the Mirror panel should result in only one History step.

With everything complete and working properly, there is one more concern that you may want to address: code reuse. Mirroring an object in Fireworks is a potentially useful action. If you should ever want to perform this action again, it would be helpful to have an easily reusable snippet of code that would let you do so. At this point, the code is pretty divided, having different implementations between four different buttons. Using a single function to represent a mirror in any direction would be useful and reusable. Such a function may look like the following:

```
function mirror(horiz, vert){
    var dom = fw.getDocumentDOM();
    dom.moveSelectionBy({x:horiz*fw.selec-
tion[0].width, y:vert*fw.selection[0].
height}, true, false);
    dom.reflectSelection(Boolean(horiz),
Boolean(vert), "autoTrimImages trans-
formAttributes");
}
```

Where *horiz* and *vert* represent directions of mirroring. Passing the value (-1, 0) to the mirror function mirrors to the left. Using (0, 1) mirrors to the bottom. Implemented in the Mirror panel, it looks like this:

```
var mirror = 'function mirror(horiz,
vert){'
+ ' var dom = fw.getDocumentDOM();'
+ ' dom.moveSelectionBy({x:horiz*fw.
selection[0].width, y:vert*fw.selec-
tion[0].height}, true, false);'
+ ' dom.reflectSelection(Boolean(ho-
riz), Boolean(vert), "autoTrimImages
transformAttributes");'
+ ' }';
right_btn.onRelease = function(){
    MMEExecute(mirror);
    MMEExecute('mirror(1, 0);');
}

left_btn.onRelease = function(){
    MMEExecute(mirror);
    MMEExecute('mirror(-1, 0);');
}


top_btn.onRelease = function(){
    MMEExecute(mirror);
    MMEExecute('mirror(0, -1);');
}
```

```
bottom_btn.onRelease = function(){
    MMEExecute(mirror);
    MMEExecute('mirror(0, 1);');
}
```

You may notice that this script is back to using MMEExecute twice again. It won't, however, create two steps in the History panel as the original version using two MMEExecute commands for each mirror did. This is because Fireworks registers History steps from JavaScript commands only if there was some reaction made as a result of the executed command. Defining a function causes no such reaction so no History step is recorded. Actually, calling the function does cause the history to be recorded, so only the second call to MMEExecute will result in a history step.

Where to Go from Here

The two panels I discussed in this tutorial provided very basic examples of custom panels in Fireworks. A lot of focus was directed towards workflow and debugging, helping you prepare for when you decide to develop more complicated panels.

In *Creating Fireworks Panels – Part 2: Advanced Custom Panel Development*, I cover a more complex example dealing with the advanced capabilities of custom panels in Fireworks. You will see how you can use the return result of MMEExecute to dictate actions in Flash as well as learn how to create some more advanced interfaces for panels using some new features available in Studio 8. 

Part 2 to this article can be found at www.macromedia.com/devnet.

Trevor McCauley first began working with Macromedia products while attending the University of Maryland, Baltimore County where he instructed classes and taught workshops covering Flash, Director, and Fireworks as an aside to earning a BA in Visual Arts. Currently Trevor works for a small production company developing content for websites and interactive CD/DVD ROMs. In his free time, he develops Flash and Fireworks content for his personal site, senocular.com, and moderates forums on popular Flash-related sites such as Kirupa.com, ActionScript.org, and UltraShock.com.



Sketching a Site Navigation Scheme

Create preliminary sketches of your site's navigation scheme
by jon varese

This tutorial illustrates how you can use Macromedia FreeHand MX to create preliminary sketches of your site's navigation scheme. The site navigation scheme is a map that shows how your web pages relate to one another. Specifically, it shows how users will travel through your site as they click links and interact with application interfaces. A site's navigation scheme is normally reflected in the navigation bars that are used on web pages.

This tutorial is the first in a series of tutorials that shows you how to build a dynamic website from start to finish with Macromedia Studio MX 2004. The complete set of tutorials is available in the Exploring Studio MX manual (www.macromedia.com/support/documentation/en/studio). It includes the following tutorials:

- Sketching a Site Navigation Scheme in FreeHand
- Creating a Table-based Page Layout in Dreamweaver
- Creating a Banner Graphic in Fireworks
- Building a Navigation Bar with Fireworks
- Building Rich Media with Fireworks and Flash
- Assembling and Editing Studio Assets
- Building a Database Search Feature

Planning the Site Navigation

After you determine your site's goals, target audience, and data requirements, you are ready to begin conceptualizing the site's navigation scheme. In this tutorial, you'll sketch the site navigation scheme for Trio Motors, a fictional company that sells energy-efficient automobiles. Trio Motors has requested an easy-

to-use navigation scheme that would allow their users to visit the company's showroom, see previews of upcoming releases, view a page for Trio automobile owners, and find information about news, events, and financial services. Trio Motors would also like the site to focus on a featured technology, and to provide a search application that lets users search for a Trio Motors dealer in their area of the country.

As the web developer for the Trio Motors website, you decide on a simple navigation scheme that will consist of five links, all centralized on a home page. At the top of the home page, you'll create a Macromedia Fireworks MX 2004 navigation bar that links to separate pages called Showroom, News and Events, Owners, Financial Services, and Previews. Elsewhere on the page, you'll create a link to a special page that provides current information about a featured technology. You'll also create a link to a results page that displays the results of the search application.

Customizing FreeHand for Multimedia/Web Output

FreeHand MX is a print-friendly and web-friendly, vector-based drawing application. Before you begin sketching your site navigation, you'll customize FreeHand for multimedia/web output. You'll also use RGB or Web Safe Colors throughout the tutorial by selecting colors from the Fill and Stroke Color pop-up menus. Web-safe color selection pop-up menus in FreeHand MX are identical to those in the other Macromedia Studio MX 2004 tools.

- In FreeHand, create a new file by selecting File > New.

- At the bottom of the Document window, select Pixels from the Units pop-up menu.

Note: The default unit of measurement for a new document is Points.

- In the Property inspector (Window > Document), click the Document tab.
- Select Web from the Page Size pop-up menu.
- This creates a page with a predefined size of 550 pixels by 400 pixels. You can also select Custom from the Page Size pop-up menu, and enter pixel dimensions of your own.
- Select Edit > Preferences.
- The Preferences dialog box appears.
- Click the Colors tab and select None from the Color Management pop-up menu.
- Turning off Color Management allows you to view colors in a generic RGB environment.
- Click OK.
- (Optional) Select Fit Page from the Magnification pop-up menu in the lower left corner of the Document window.
- This option sets the magnification level so that the active page fits inside the Document window. You can also set magnification levels manually. The magnification with which you choose to work depends on your monitor size and your own preferences.
- Save the new document (File > Save) as trio_navigation.FH11.
If you are completing this tutorial as part of the Exploring Studio MX manual series, you can save the new file in the

main directory of the Dreamweaver site you created in Chapter 6, "Setup for Sample ColdFusion Site." It is good practice to keep all site-related files in the same location. For more information, see the Exploring Studio MX manual.

Creating Rectangles for the Navigation Scheme

After you customize your document for the web, you are ready to sketch the site navigation scheme in FreeHand.

- Make sure that the file you just created, `trio_navigation.FH11`, is open in the Document window.
- In the Tools panel, click the Rectangle tool and drag a rectangle on the page.
- When you release the mouse button, the new rectangle is selected. Leave the rectangle selected so that you can alter its properties.
- Note: This tutorial uses the example of a rectangle that is 135 pixels wide by 52 pixels high. You can view and set the exact dimensions of a rectangle on the Object tab of the Property inspector (Window > Object).
- In the Tools panel, click the Fill color box and use the pop-up color palette to select a color for the inside of the rectangle. The example in this tutorial uses light blue (CCCCCCF) for the rectangle's fill.
- With the rectangle still selected, click the Stroke color box, and use the pop-up color palette to select a color for the outline of the rectangle. The Stroke color box is above the Fill color box in the Tools panel. The example uses a darker blue (000033) for the rectangle's stroke.
- Save your document (File > Save). Remember to save your work often as you make changes.
- In the Tools panel, click the Text tool and click once inside the rectangle.
- A text block appears inside the rectangle with a text ruler on top of it.
- On the Object tab of the Property inspector (Window > Object), do the following:
 - Select Arial from the Font pop-up menu.
 - Select Plain from the Style pop-up menu.
 - Select 18 from the Font Size pop-up menu.
 - Click the Align Center button
- In the Tools panel, click the Fill color

- box and select a fill color for the text. The example uses 000033, the same color as the outline of the rectangle.
- Type the word Home inside the rectangle.
- Click outside the text block to deselect it.
- Select both the rectangle and the text block by selecting Edit > Select > Select All.
- On the Align tab of the Align and Transform panel (Window > Align), select Align center from the Horizontal and Vertical pop-up menus, and click Apply.
- The text aligns vertically and horizontally in the center of the rectangle.
- With the rectangle and text block still selected, select Modify > Group.
- The rectangle and text block become a grouped object.
- Select Edit > Clone.
- A duplicate copy of the grouped object appears on top of the original rectangle.

Note: The Clone feature is also available in Fireworks MX 2004.

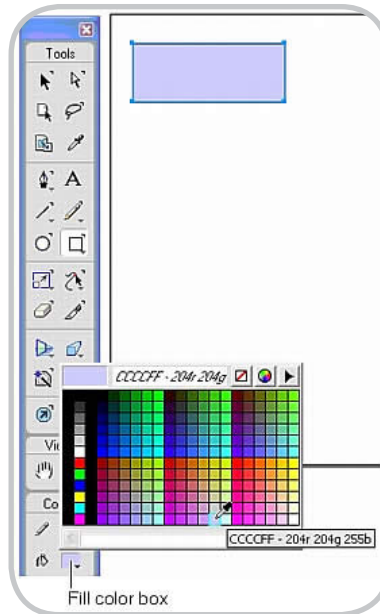
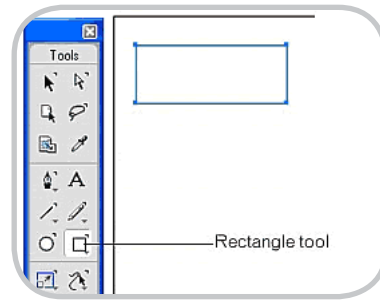
- Drag the duplicate copy of the object below the original rectangle, as shown.
- Continue to clone and drag the original grouped object until you have eight objects on the page. Use the Pointer tool to arrange your rectangles as follows:

You can also use the arrow keys on your computer keyboard to position the rectangles more precisely. The arrow keys move objects up, down, left, or right, one pixel at a time.

Changing Rectangle Text

Next, you'll change the text of your rectangles to describe the various pages of the website.

- In the Tools panel, click the Subselect tool.
- Click the word Home in the second rectangle (below the first rectangle on the page) to select the text block.
- In the Tools panel, click the Text tool.
- Select the text in the selected text block by holding down the mouse button as you drag over the word Home.
- Type the word Showroom.
- Click on a blank area of the document



- to deselect the text block.
- Repeat steps 1 through 6 for the remaining rectangles, using the following text: News and Events, Owners, Financial Services, Previews, Featured Technology, and Search.
- For longer text content, such as "Financial Services," press Enter (Windows) or Return (Macintosh) after the first word and type the next word on the second line. Shift-click the text block and the rectangle using the Subselect tool. Then realign the text in the center of the rectangle by selecting Align center from the Horizontal and Vertical pop-up menus in the Align and Transform panel (Window > Align).

Adding Connector Lines to the Document

Now you'll use the new Connector tool in FreeHand to add connector lines to the document. The connector lines show how the home page links to the individual files you specified.

- In the Tools panel, click the Connector tool.
- Move the pointer over the bottom line

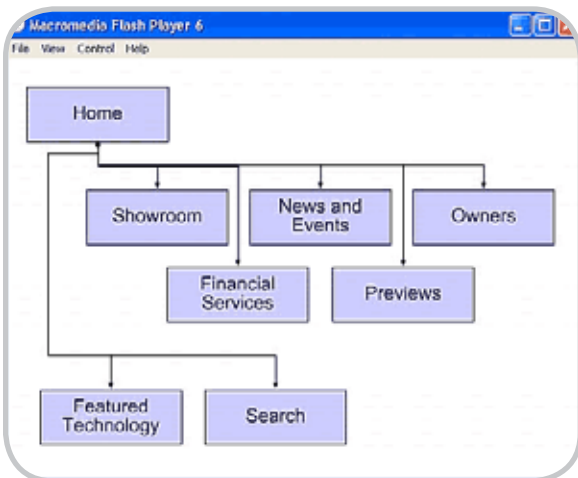


of the Home rectangle.

- Hold down the mouse button and drag a connector line down to the top line of the Showroom rectangle. When you begin dragging, the Home rectangle becomes selected, indicating that there is an object from which a connector line can start. When you reach the Showroom rectangle, it also becomes selected, indicating that there is an object where the connector line can end.
- Release the mouse button.
- A connector line appears between the Home rectangle and the Showroom rectangle.

Note: If there isn't enough space between rectangles to accommodate the default arrowheads of the connector line, the tool cannot draw the line properly. If you encounter this problem, try increasing the space between rectangles.

- Position the pointer near the starting point of the connector line you drew in the previous steps (the bottom line of the Home rectangle) and hold down the mouse button so that the Home rectangle is selected again.
- Drag another connector line to the top line of the News and Events rectangle. The Connector tool can be sensitive, so this may take some practice. Make sure that both rectangles are selected before you release the mouse button.
- Draw a third connector line from the bottom of the Home rectangle to the top of the Owners rectangle.
- Draw a fourth connector line from the bottom line of the Home rectangle to the top line of the Financial Services rectangle.



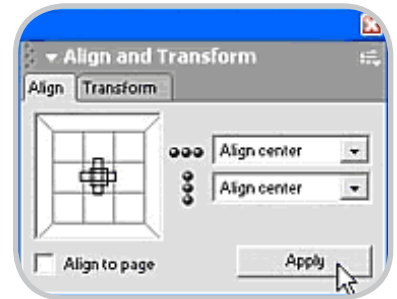
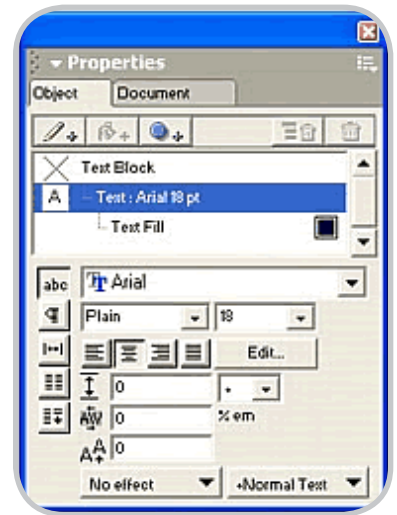
- Move the pointer over the handle at the center of the horizontal line you drew in step 8.
- The pointer changes to a resize pointer.
- Drag the horizontal line upward until it overlaps the other horizontal line on the page, as shown.
- Continue drawing connector lines until you have linked the Home rectangle to all the other rectangles on the page. Reshape connector lines as necessary by dragging the resize handles on the connector lines.

You now have a FreeHand document that shows the full navigation scheme. Your lines indicate that the Home page links to five separate pages called Showroom, News and Events, Owners, Financial Services, and Previews. It also links to a page about a featured technology, as well as to a page that provides the results of the search application that you'll build later.

Exporting the Plan as a SWF File

After sketching your site navigation, you can present the preliminary plan to your client or members of your group in a browser window. This is particularly convenient if you want to share the document with people who do not have FreeHand.

- With your navigation scheme open in FreeHand, select File > Export.
- The Export Document dialog box (Windows) or Export dialog box (Macintosh) appears.
- Select Macromedia Flash (SWF) from the Save As Type pop-up menu (Windows) or Format pop-up menu (Macintosh).
- (Optional) If you want the file to open with Macromedia Flash Player immediately upon export, select Open in External Application, select Macromedia Flash Player from the External Application dialog box, and click Open.
- Click Save (Windows) or Export (Macintosh).
- The file is exported as a Flash SWF file to the location specified in the Export Document dialog box (Windows) or Export dialog box (Macintosh). If you selected Open in External Application, the plan opens in Macromedia Flash Player.



- You can now present the plan on any computer running a web browser and Flash Player. To view the plan in a browser window, drag the trio_navigation.swf file to an open browser such as Internet Explorer or Netscape Navigator. You can also present the plan with Flash Player alone by double-clicking the file.

Note: When you open the trio_navigation.swf file in a browser, the plan is automatically resized to fit in the browser window.

Completed versions of the files you created in this tutorial are available in the trio_motors_complete directory on the Studio MX 2004 CD. To locate the directory, insert your Studio MX 2004 CD, click Browse CD contents, and select the Samples folder. ☁

Jon Michael Varese is a senior technical writer for Instructional Media Design at Macromedia. He has written web and print documentation for Dreamweaver, Fireworks, FreeHand, Flash, and ColdFusion. In the past he has held such illustrious positions as baker, clown, and funeral attendant. In addition to writing web tutorials, he is currently at work on his Ph.D. in 19th-century British Literature.

BY NOW THERE ISN'T A
SOFTWARE DEVELOPER ON EARTH
WHO ISN'T AWARE OF THE
COLLECTION OF PROGRAMMING
TECHNOLOGIES KNOWN AS AJAX!

REAL-WORLD
AJAX
ONE DAY SEMINAR
www.ajaxseminar.com

March 13, 2006

April 24, 2006



Marriott Marquis Times Square
New York, NY



DOUBLETREE
HOTEL
San Jose, CA

For more information
Call 201-802-3022 or
email events@sys-con.com



How, in concrete terms, can you take advantage in your own projects of this newly popular way of delivering online content to users without reloading an entire page?

How soon can you be monetizing AJAX?

This "Real-World AJAX" one-day seminar aims to answer these exact questions...

Led by "The Father of AJAX" himself, the charismatic Jesse James Garrett of Adaptive Path, "Real-World AJAX" has one overriding organizing principle: its aim is to make sure that delegates fortunate enough to secure themselves a place – before all seats are sold out – will leave the seminar with a sufficient grasp of Asynchronous JavaScript and XML to enable them to build their first AJAX application on their own when they get back to their offices.



Jesse James Garrett
Father of AJAX



Scott Dietzen
Creator of WebLogic, Ph.D., President and CTO, Zimbra



Bill Scott
AJAX Evangelist of Yahoo!



David Heinemeier Hansson
Creator of Ruby on Rails



Satish Dharmaraj
Father of JSP, Co-Founder & CEO, Zimbra



Rob Gonda
Best-Selling AJAX Author, CTO, iChameleon Group



Dion Hinchcliffe
Co-founder & CTO, Sphere of Influence Inc.



Ross Dargahi
Well-known AJAX Evangelist & Co-founder and VP of Engineering, Zimbra

Early Bird	\$995
<small>(Before January 31, 2006)</small>	
Discounted Price	\$1,195
<small>(Before February 28, 2006)</small>	
Seminar Price	\$1,295
<small>(After February 28, 2006, and if any seat is available)</small>	

HURRY!
LIMITED SEATING
THIS SEMINAR WILL
SELL-OUT
CALL 201-802-3022
TO REGISTER!

Jeremy Geelan
Conference Chair, Real-World AJAX
jeremy@sys-con.com

MEDIA SPONSOR



LIVE SIMULCAST!
AROUND THE WORLD ON SYS-CON.TV

PRODUCED BY
SYS-CON
EVENTS



The Real Estate Sample Application

Create an application that allows users to retrieve records from a database
by nahuel foronda & laura arguello

With the release of Macromedia ColdFusion 7 and the arrival of Flash

Forms, developers were presented with an alternative to HTML forms that offered them additional functionality, such as full-featured controls not available in HTML and built-in validation. That alone made Flash Forms appealing—and with the addition of pieces of ActionScript code, developers were able to create truly responsive forms. But because they were meant to be compatible with HTML forms, they still shared the same submit-refresh model. What if you could “submit” the form and, without a page refresh, get feedback from the server?

Enter Flash Remoting. In this series of tutorials you will learn how to create an application that allows users to search and retrieve records from a database, and then edit, add, and remove them from the database—all in one screen. Those functions will be presented in the context of a sample application, a real state management system that administers listings of properties for sale.

Overview of the Real Estate Management System Sample Application

This article focuses on the search functionality of the sample application.

When you open the application, a small panel on the left lets users search for a property by specifying search criteria, such as number of bedrooms or price range. The matching properties appear in a data grid in the upper right panel, and when a user selects a property, the details appear in the lower right panel.

In the following sections you will do the following:

- Create the search form.
- Write a component to make the search query.
- Create a Flash Remoting service that handles the search request.

Call the Flash Remoting service and show the results in a datagrid.

Creating the Search Form

The whole application user interface is one Flash Form. As such, its contents are within the cfform tag:

```
<cfform format="flash"
name="RealEstateAdmin">
<!-- form content -->
</cfform>
```

The only necessary attribute of the cfform tag is the format="flash" attribute to create a Flash Form. By assigning a name to the form, you will have a named scope that you can use later. You can also set the form's dimensions by using the width and height attributes.

The Search panel contains several controls, all of them enclosed in a cfformgroup tag with a type attribute of "panel":

```
<cfformgroup type="panel"
label="Search" height="440" >
<!-- controls -->
</cfformgroup>
```

These are the controls contained within the Search panel:

- The cfselect tags for Price range, Bedrooms, Bathrooms, and Minimum footage that create pop-up menus. You can populate cfselect tags from option

tags, queries, or a combination of the two, as shown in this example:

```
<cfselect name="search_priceRangeFrom"
query="priceRange"
display="label"
value="data"
queryposition="below">
<option value="0">No min.</option>
</cfselect>
```

- This cfselect tag uses a query called priceRange that contains two columns—data and label—which populate the search_priceRangeFrom select control. In addition, an extra option not present in the query (“No min.”) is manually added using an option tag. The name you give each control is important because you will later need to reference the control name when you submit the form.
- <cfinput type="text"> tag for “MLS number”, as follows:
- <cfinput name="search_mls_id" type="text" />
- <cfinput type="radio"> tags for “Status.” You need one cfinput tag for each type of listing status (Active, Sold, Back Up Offers, New Listing). Option button tags that belong to the same group must have the same name. See the following:

```
<cfinput type="radio" name="search_
status" value="active"
label="Active"/>
<cfinput type="radio" name="search_
status" value="backUp" label="Back Up
Offers"/>
<cfinput type="radio" name="search_
status" value="sold" label="Sold"/>
```

The sample files for this article can be downloaded at www.macromedia.com/devnet/coldfusion/articles/flashforms.html

<cfinput type="checkbox"> tags for "Amenities." You need one tag for each type of amenity (Pool, Laundry, Walk-in Closets, Fireplace). Unlike option buttons, each tag must have a different name, as follows:

```
<cfinput type="checkbox" name="search_hasPool" label="Pool"/>
<cfinput type="checkbox" name="search_hasLaundry" label="Laundry"/>
<cfinput type="checkbox" name="search_hasFireplace" label="Fireplace"/>
<cfformitem type="text"> tags for additional labels or titles, as follows:
```

```
<cfformitem type="text" style="fontWeight:bold;">Status:
</cfformitem>
```

<cfinput type="button"> tag for the Search button:

```
<cfinput type="button" name="search_submitBtn" value="Search" />
```

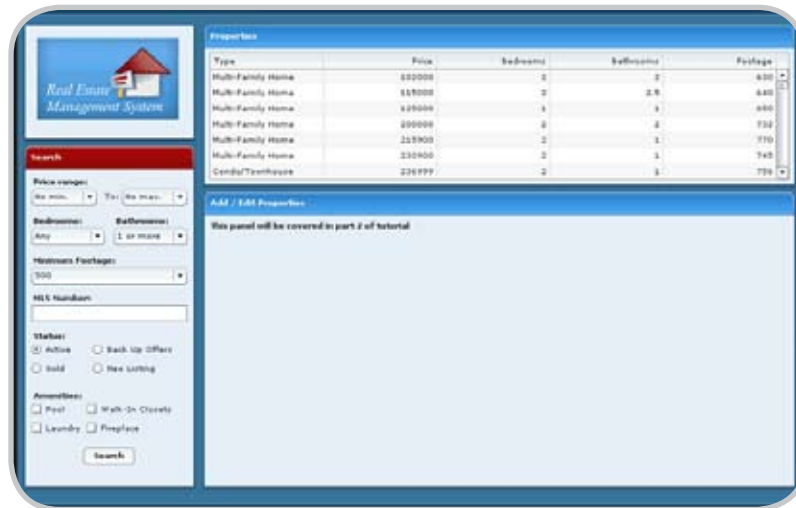
You must also use additional **cfform-group** tags to lay out the controls. Look at the `index.cfm` source file in the ZIP download to see the complete code.

This small section of the application does not do anything yet. It only accepts user input of standard form controls: text inputs, option buttons, check boxes, and selects. At this point, you would normally add a Submit button to submit the form for processing on the server. But don't do that yet—wait just a moment and complete the following sections.

Coding the Search Query

As you may expect, you must write ColdFusion code to search the database. The sample files contain a ColdFusion component (CFC), `components/ListingGateway.cfc`, that queries the database but you could use a custom tag as well.

The search in this component has several arguments that correspond to the different search criteria: status, price, pool and other amenities, and so forth. The contains a simple query using the `cfquery` tag. All the arguments have a default value and the SQL statements will vary each time a user specifies a different combination of arguments. For instance, if the user doesn't enter a status, the CFC ignores that criteria; therefore, the SQL statement will not contain that column in its WHERE clause. To exclude it, we used a simple `cfif` tag:



```
<cfcomponent name="ListingGateway">
    AND listing.hasPool = <cfqueryparam value="true" cfsqltype="CF_SQL_BIT">
    </cfif>
    <!-- additional sql removed -->
    ORDER BY price, listedOn
</cfquery>
    <cfreturn listingQuery />
</cffunction>
</cfcomponent>
```

```
<cfset var listingQuery = "" />
<cfquery name="listingQuery" datasource="realestate">
    SELECT *
    FROM listing
    WHERE
    listing.price >=
    <cfqueryparam value="#arguments.priceFrom#" cfsqltype="CF_SQL_MONEY"/>
    <cfif arguments.priceTo GT 0>
        AND listing.price <=
    <cfqueryparam value="#arguments.priceTo#" cfsqltype="CF_SQL_MONEY"/>
    </cfif>
    <cfif len(arguments.status)>
        AND listing.status =
    <cfqueryparam value="#arguments.status#" cfsqltype="CF_SQL_VARCHAR"/>
    </cfif>
    <cfif arguments.hasPool>
```

When the query finishes processing, the CFC returns the results to the caller through the `<cfreturn listingQuery />` tag.

So far, you should be familiar with the methods used. You created a form to request data from the user and wrapped a query in a CFC to retrieve data from a database. You may be wondering how to connect the form and the CFC. Read onward.

Creating the Flash Remoting Service

Flash Remoting is a Macromedia technology that lets Flash Player make asynchronous calls to web services and receive responses back from the call. During these calls, it can send and receive sets of data. It consists of a gateway that sits on the server and some components that are included in the SWF file. Fortunately, ColdFusion 7 already contains the Flash Remoting gateway, so you don't have to install any additional software. The SWF files generated by Flash Forms also contain



the Flash Remoting components because they are used internally.

Just as web services do, you must place Flash Remoting services in a web-accessible directory. To create such a service, you use a CFC, writing the functions that you want to provide to the service consumers. These functions must have their access attribute set to remote in order to be accessible to Flash consumers.

Here is a simple example of a service:

```
<cfcomponent>
<cffunction
name="search"
access="remote"
returntype="query"
output="false"
hint="Returns listings matching the
search criteria">

<cfargument name="status"
type="string" default="" />
<!-- code here that generates a query
called myQuery -->

<cfreturn myQuery/>
</cffunction>
</cfcomponent>
```

We could have included the code that queries the database in the service CFC but it is better to separate presentation

from data layers and keep well-defined tasks encapsulated in different components. In this way, the core components that access the database can be completely separate from service components that may access them. The service components, knowing that they are services, can offer additional functionality specifically targeted to Flash, such as formatting.

There is another advantage to this approach. It is common practice to keep components instantiated in a scope that spans several requests, such as the application or session scopes. When a Flash Remoting call is made to a component, however, this component is instantiated at that time—and every time a request is made, a new instance of the component is created.

This means that Flash Remoting cannot call an already instantiated component residing in memory in a shared scope. The only way to resolve this issue is to have a service component that differs from the component that does the actual work. When Flash Remoting calls a of the service component, this can reference the component residing in memory and make the appropriate invocations. For simplicity's sake, this example instantiates ListingGateway.cfc but the source files use the memory-resident CFC approach.

Because the Search panel sends simple data types that match the types expected by the ListingGateway component, the service can simply pass the same arguments along by using an argumentCollection:

```
<cffunction name="search"
access="remote"
returntype="query"
output="false"
hint="Returns listings matching the
search criteria">
<cfargument name="status"
type="string" default="" />
<cfargument name="priceFrom"
type="numeric" default="0" />
<cfargument name="priceTo"
type="numeric" default="0" />
<cfargument name="hasPool"
type="boolean" default="false" />
<!-- additional arguments removed -->
```

```
<!-- you could also use cfinvoke or
access a component in memory -->
```

```
<cfset var listingGateway =
createObject("component",
"RealEstate.components.
ListingGateway")/>

<cfreturn
listingGateway.search(argumentCollection=arguments) />
</cffunction>

Name this component ListingService.cfc
and save it in the services folders.
```

Calling the Service

Now that you have created a service, you are ready to call it. Provided that you are running ColdFusion on your local machine, the Flash Remoting gateway is typically located at `http://localhost/flashservices/gateway`. That is its default address for ColdFusion installations.

Please note that this address might be different if you are using the built-in web server option (if your web root is located at `http://localhost:8500`).

Adding ActionScript to the Flash Form

To call a service from a Flash Form, you must write some ActionScript. You can add any piece of ActionScript code to the form by writing inline statements in the controls' event handlers, such as `onClick`, or by writing code blocks in functions using the `<cfformitem type="script">` tag that has been added in the ColdFusion 7.01 Updater. For readability and easier maintenance, we recommend using functions.

Because you will use this Flash Remoting service several times, we recommend that you store it in a variable that can be used by other functions and controls. Wrap the Flash Remoting setup code that creates the service and stores this variable in a called `setUpRemoting()`:

```
<cfformitem type="script">
setUpRemoting():Void{
}
</cfformitem>
```

Inside the `cfformitem` tag, declare the connection and service proxy as local variables:

```
//note the gateway address
var connection:mx.remoting.Connection
```

ENGAGE AND EXPLORE...

The Technologies, Solutions and Applications that are Driving Today's Initiatives and Strategies...

JUNE 5-6, 2006 Roosevelt Hotel / New York, NY

RESISTER TODAY!

SOA 10th International WebServices Edge conference+expo



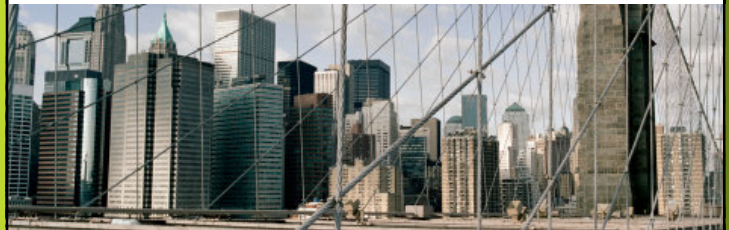
The Sixth Annual SOA Web Services Edge 2006 East - International Web Services Conference & Expo, to be held June 2006, announces that its Call for Papers is now open. Topics include all aspects of Web services and Service-Oriented Architecture

Suggested topics...

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

RESISTER TODAY!

2006 ENTERPRISE > OPENSOURCE CONFERENCE+EXPO



The first annual Enterprise Open Source Conference & Expo announces that its Call for Papers is now open. Topics include all aspects of Open Source technology. The Enterprise Open Source Conference & Expo is a software development and management conference addressing the emerging technologies, tools and strategies surrounding the development of open source software. We invite you to submit a proposal to present in the following topics. Case studies, tools, best practices, development, security, deployment, performance, challenges, application management, strategies and integration.

Suggested topics...

- > Open Source Licenses
- > Open Source & E-Mail
- > Databases
- > ROI Case Studies
- > Open Source ERP & CRM
- > Open-Source SIP
- > Testing
- > LAMP Technologies
- > Open Source on the Desktop
- > Open Source & Sarbanes-Oxley
- > IP Management

Submit Your Topic Today! events.sys-con.com

Sponsored by

WebServices
JOURNAL

XML
JOURNAL

NET
JOURNAL

eclipse
developer's journal

WebSphere
JOURNAL

Information
STORAGE+SECURITY
journal

wldj
DE LEADING
MAGAZINE
ON SERVICE
PERFORMANCE

JDJ

Linux
WORLD
MAGAZINE

MX
developer's journal

asp.net
PRO

SDTimes

CoDe

Software Test
& Performance

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



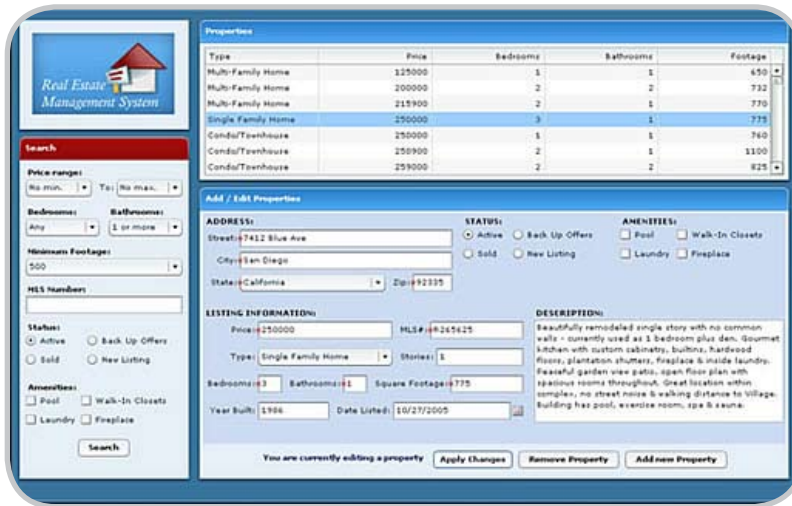
Attention Exhibitors:

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

For Exhibit and Sponsorship Information - **Call 201 802-3023**

Produced by **sys-con**
EVENTS

© 2005 WEB SERVICES EDGE. ALL RIGHTS RESERVED



```

=
mx.remoting.NetServices.createGatewayC
onnection("http://localhost/flashser-
vices/gateway/");

```

```

var myService:mx.remoting.
NetServiceProxy;
Declare an object that handles all the
responses:
var responseHandler:Object = {};

```

This object must implement the necessary functions that will be automatically called when the server sends a response from a CFC. The default functions called on this object are `onResult` and `onStatus`. The server calls `onResult` when the service sends a successful response and calls `onStatus` when the service throws an error, when or some other type of error occurs. That means that the `responseHandler` object must have those two functions if it is expected to act upon service responses:

```

responseHandler.onResult = ( results:
Object ):Void {
//handle service response
}
responseHandler.onStatus= (status:
Object ):Void {
//handle error
}

```

Once the `responseHandler` objects knows how to behave, the actual service can be instantiated and stored in a global variable (`RealEstateAdmin.myGlobalObjects`), as follows:

```

RealEstateAdmin.myGlobalObjects.
listingService = connection.

```

```

getService("RealEstate.services.
ListingService", responseHandler );

```

The `getService()` method of the connection class takes two parameters; the first is the path to the service. To get the path to your service, find the directory structure from your web root to your CFC, replace the slashes with dots, and remove the CFC extension.

For instance, if you saved the sample files in a folder called `RealEstate` in your web root and saved `ListingManager.cfc` in a folder called `services`, the path would be `/RealEstate/services/ListingManager.cfc` from a browser. Translating that to dot notation becomes `RealEstate.services.ListingManager`.

The second parameter is the object that handles the responses, which is the object you just created: `responseHandler`. Note that the service is not actually created until you call a method on the service. So if you enter an incorrect path, you will not know it until you try to call it.

Now you have a that sets up a Flash Remoting service, but if you never call this, the service will never be set up. You can use the `onload` attribute of the `cform` tag, which was introduced with the ColdFusion 7.01 Updater to call the, as follows:

```

<cform name="RealEstateAdmin"
format="flash" onload=" setUpRemot-
ing() ">

```

That is not enough, however, because the service is stored in a global variable that you must also declare as the form loads. You can create yet another to be called as the form loads that sets up this global variable and possibly other settings. While you are at it, call the `setUpRemoting()` right after that, as follows:

```

<cform name="RealEstateAdmin"
format="flash"
onload="formOnLoad();">

<cformitem type="script">

formOnLoad():Void{
//declare global variable
RealEstateAdmin.myGlobalObjects = {};

//call set up
setUpRemoting();

//set other properties as needed
}
//rest of the code

```

Assigning the Flash Remoting service to a global variable is only a suggestion. The most important code is what's inside the `setUpRemoting()`.

Finally, you are ready to call the service. Your Search panel readily accepts user input, you have set up the Flash Remoting service, and your CFC is eagerly waiting to be called. Remember how the Search panel had a Search button? Now you can make it functional by adding some actions to its `onClick` attribute:

```

<cfinput type="button"
name="searchSubmit" value="Search"
onclick="submitSearch()" />
But wait a minute. What is the submit-
Search() ? Because it is not a built-
in, you're going to create it next.
Creating the submitSearch()
Within the same <cformitem
type="script"> tag, add the new :
submitSearch():Void{

```

The purpose of the `submitSearch()` is to gather all the information entered in the Search panel and send it to the `ListingManager` service.

If you recall, the CFC has several arguments. When making the call, you could send each argument, in order, as parameters to the call. But that becomes error-prone if the has many arguments. An easy way to send several parameters is to use a structure with keys that have the same name as the arguments.

To create a structure, you instantiate an empty object (or structure) that will be the container for all the data entered in the Search panel form fields. Then, one by one, take the data entered in each field and assign it to a key in the empty object, as follows:

```
//get all the search criteria items
var searchArguments = {};

//in order to get data contained in
Text inputs,
//use myTextInputName.text
searchArguments.mls_id = search_mls_
id.text;

//to get data from radio buttons,
//use myRadioButtonName.selectedData
searchArguments.status = search_sta-
tus.selectedData;

//checkboxes store their true/false
value in their selected property,
//myCheckBoxName.selected
searchArguments.hasPool = search_
hasPool.selected;
searchArguments.hasWalkInClosets =
search_hasWalkInClosets.selected;
searchArguments.hasLaundry = search_
hasLaundry.selected;
searchArguments.hasFireplace = search_
hasFireplace.selected;

//to get the selected choice in a
dropdown,
//use mySelectName.value or mySelect-
Name.selectedItem.data
searchArguments.footage = search_foot-
age.value;
searchArguments.bedrooms = search_bed-
rooms.value;
searchArguments.bathrooms = search_
bathrooms.value;
searchArguments.priceFrom = search_
priceRangeFrom.value;
searchArguments.priceTo = search_
priceRangeTo.value;
```

Once you have gathered all the necessary data for the structure, you can make the call to your CFC, sending all the parameters as one structure. Recall that “search” was the name of the service in the ListingManager CFC:

```
RealEstateAdmin.myGlobalObjects.list-
ingService.search(searchArgs);
```

After you call the service and the matching records are retrieved, the server will send a query back with the results, even if it is an empty query. The responseHandler object handles the response in its onResult , but if you go back to the “Calling the Service” section, you will see

that the was empty, which doesn’t help much—when the results are returned, they will just be ignored.

Adding a Grid for the Results

The sample application shows the search results in a datagrid next to the Search panel. To implement this functionality, add a grid using cfgrid and name it listingGrid:

```
<cfgrid name="listingGrid"
rowheaders="false">
<cfgridcolumn name="price"
header="Price" />
<cfgridcolumn name="bedrooms"
header="Bedrooms" />
<cfgridcolumn name="bathrooms"
header="Bathrooms" />
<cfgridcolumn name="footage"
header="Footage" />
<cfgridcolumn name="type"
header="Type" />
</cfgrid>
```

Notice that the headers of visible columns are properly named and the name attribute of each cfgridcolumn tag corresponds to a query column name.

Now that you have a control, assign the results to the grid’s data provider:

```
var listingGrid = listingGrid;
responseHandler.onResult = (results:
Object):Void {
    listingGrid.dataProvider = results;
}
Because the CFC might throw an error,
you should also complete the onStatus
to handle error responses:
responseHandler.onStatus = ( stat:
Object ):Void {
    alert("Error: " + stat.description);
}
```

Dealing with Asynchronous Processes

If the service returns a query, you might be wondering why you can’t simply use the following code:

```
//wrong code
listingGrid.dataProvider = listingSer-
vice.search(searchArgs);
```

The above code is not correct because the call to search is done asynchronously. As

soon as the call is triggered, it does not wait for the server’s response; it continues running code written below it. For this reason, you must create an object (called responseHandler here) with functions that will be called when the server sends a response.

Another point to consider is that any communication across a network is unreliable. If the returned dataset is large, it might take some time to load. When using the page refresh model, the browser takes care of adding a progress bar to indicate that the next page is loading. Although this is not perfect, it does offer some feedback to the user. Now that your application does not use a page refresh model, how do you indicate to users that their request is being processed and they have to wait?

There are several approaches to this. A simple method is to show the default clock cursor that Flash Forms use when they load data. This is a simple solution that is consistent with the other functions of Flash Forms. To add that functionality, insert the following snippet before making the call to the remote service in index.cfm:

```
mx.managers.CursorManager.showBusyCur-
sor();
When the results come back, remove the
cursor:
mx.managers.CursorManager.removeBusy-
Cursor();
```

Add this snippet to both the onResult and onStatus functions because you don’t know which one will get called. ☁

Part 2 to this article can be found at www.macromedia.com/devnet.

Nahuel Foronda is the founder and lead Flash developer for Blue Instant, a web development firm specializing in Rich Internet Applications. During his five years of experience with Flash, he has created award-winning applications and websites. He also maintains a blog on Flash and ColdFusion called AS Fusion.

Laura Arguello is one of the founders of Blue Instant, where she has been creating web applications for the last five years. Apart from her company, she also maintains a blog, AS Fusion, where she and Nahuel Foronda write about ColdFusion, Flash, and other Macromedia and web technologies.



Creating a Shockwave 3D Game Level in 30 Minutes

Export Shockwave 3D content using any of the major 3D applications
by mal duffin

during the course of reading this article and carrying out the various exercises, you and your 3D artist colleagues will get a practical, hands-on idea of the workflow required to create and export Shockwave 3D content using any of the major 3D applications. You will create a 3D environment, export it as Shockwave 3D, and drive around it in a virtual vehicle using the keyboard's arrow keys.

You are not required to use Macromedia Director. However, I do hope that after you get an idea of the power of the real-3D features of Director, you or your programming colleagues will be encouraged to download the trial version of Director and start working on some new and exciting Shockwave 3D content.

It will take approximately 30 minutes to finish the exercises presented in this article.

Applications for Shockwave 3D Export

Make sure you have one of these supported applications installed. These applications all support Shockwave 3D export. The CanDo 3D Modeling Training Tool linked to above contains a sample 3D scene for each of these packages:

- 3ds Max
- Maya
- LightWave 3D
- trueSpace (GameSpace)
- Cinema 4D
- form-Z
- Eovia Carrara
- SoftImage|XSI
- 3D Canvas Pro

- EIAS (Electric Image Animation System)

The downloaded training application directly supports the Shockwave 3D export of all of these applications.

Why Create Shockwave 3D Content?

Shockwave 3D is extremely powerful and flexible technology. While there are a number of different technologies that allow for interactive 3D content to be deployed over the Internet—and they all support the same basic functionality required for many applications, such as displaying interactive products—only a few allow for more advanced applications, such as an interactive game, to be implemented.

As with any rich online media, the popularity and installed base of the plug-in is a very important factor. Since 1999, the Shockwave plug-in has incorporated interactive 3D technology (developed by some of Intel's top software engineers) as part of its standard installation. This has ensured that approximately two out of three computers connected to the Internet already have Shockwave installed on them.

Shockwave 3D supports many features, such as reflection mapping, multitexturing, particle systems, subdivisional surfaces, toon rendering, and progressive level of detail. Skinning and bones can be exported from certain 3D applications.

A key feature of Shockwave technology is that it allows for the integration of all forms of media, including the use of 3D content and 2D content (images, video, Flash, and so on) within the same interface. Some 3D-only solutions can be

quite restrictive in allowing developers to implement a user interface to their content.

Shockwave 3D currently does not support advanced features such as pixel and vertex shaders. These are available only on newer computer systems and are therefore not viewable on many computers. However, this is more than made up for by ensuring that content can be viewed on every single computer running Windows or Mac OS, regardless of whether or not an OpenGL or Direct 3D graphics card is even available (thanks to a built-in software renderer).

A recent update to the Shockwave plug-in has slimmed the size of the installer from over 4 MB to a very compact 2.5 MB. This makes it even more appealing to many computer users who want to view interactive content online and who still have dial-up modem connections.

Creating a Shockwave 3D Scene from Scratch

The application you downloaded with this article allows you to export your 3D scene into an interactive Shockwave 3D environment using a very simple export process. For many of the applications, this will be as straightforward as exporting a single file.

The process goes something like this:

- Create or load your 3D scene.
- Ensure that a model exists, called carStart.
- Note: This is the only requirement of the application. The model is used to position and size the vehicle within the scene.

- Export the scene as Shockwave 3D (overwrite the gamelevel.w3d scene).
- Run the application.

That's it! In most cases, the export process takes a matter of seconds. This allows 3D artists to try out a number of different ideas in a very short period of time.

To create a basic scene, follow these steps:

- Open your 3D application, create a box in the middle of the ground object, and call it carStart. The size of the box determines the size of the resulting vehicle within the Shockwave 3D environment.
- Create a ground area for the vehicle to start on. This can be as simple as a plane object, or as a textured object—for example, with a tiled grass texture.
- Export the scene as Shockwave 3D and save it over the gamelevel.w3d file.
- Note: If you read the Instructions.doc file contained in the sample files download, you'll see detailed instructions on how to export Shockwave 3D content from the various different 3D applications.
- Run the application. For Windows, run the file "Play Car Game Level.exe"; Mac users currently have to run index.htm.

You will now be able to drive the vehicle around the ground object, using the keyboard's arrow keys. In the next section, you will add new models to the scene.

Adding Models to the Scene

You are now going to add some objects to the environment:

- Add a second ground area with a concrete type texture applied to it.
- Add a jump ramp to allow access to the second ground area. In order to do this, first create a box.
- Drag two of the vertices of the box down to form a ramp.
- Apply different materials to the top and side of the ramp.
- Admire your new ramp.

Feel free to create and place additional models in the scene. In the next section, you will add some interactive elements to the scene.



Adding Interactive Elements to the Scene

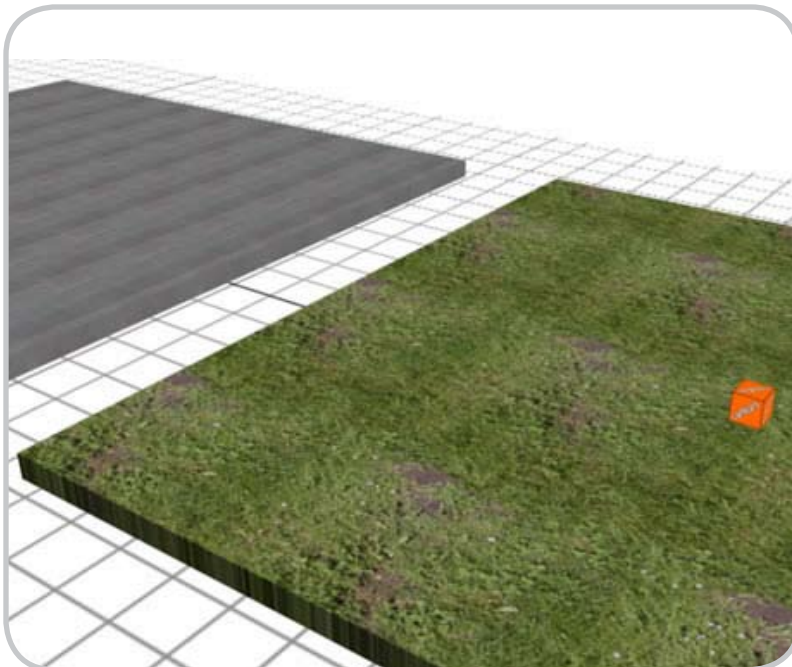
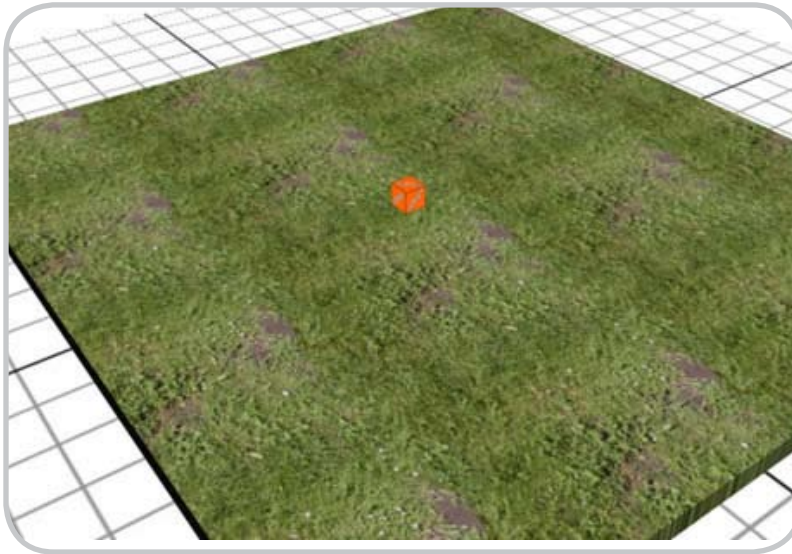
You are now going to add some interactive elements to the scene. Place special characters in the name of the object within the 3D modeling application to allow interactions to be built up.

You can find more information on the various tags in the included Instructions.doc file included in the sample files

download linked to at the beginning of the article.

I've added a water texture to a plane under the ground objects, and I've named this object gameLose. As soon as the vehicle touches this water model, the game ends. There is a gameWin box at the far side of the wall. As soon as the car connects with this object, the game is won.

In order to get to the gameWin box, however, the player must trigger off the



As one of the directors of CanDo Interactive, Mal Duffin has extensive experience creating interactive Shockwave 3D content. He previously worked on video games and virtual reality projects for companies like Sony, Shiny Entertainment, and IBM.

wall, called `triggerOffExit`, in the middle of the platform. This occurs when the player collects the two pickups, called `pickupExit01` and `pickupExit02`.

I have replaced the wall with a box with a tiled brick texture, and replaced the `pickupExit` objects with coins (cylinders).

In the next section, I will review changing the non-3D elements of the scene.

There are a number of external texture files, used by the game, that you can edit to help change the look and feel of the game. This helps to incorporate your Photoshop or Fireworks skills easily into the project.

Here are some of the main files, along with how they are used:

- `vanlogo.jpg`: Alters the logo on the back of the vehicle. The color of the very first (top left) pixel is used as the color of the rest of the vehicle. You can change the images on the sides and front of the vehicle by creating the images `vanlogoside.jpg`, `vanlogosideb.jpg`, and `vanlogofront.jpg`.
- `title.jpg`: Appears right at the start of the game. You can create `gamewin.jpg` and `gamelose.jpg`, which will appear when those various events appear. These can also be replaced with Flash SWF files. See `Instructions.doc` for more information.
- `logo.png`: Appears over the top of the main 3D game window. By using transparency within the PNG, you can overlay logos and instructions.
- `speedometer.png`: Displays the speedometer (position it with `gamelevel.xml`).
- `steering.png`: Shows the graphic for the steering wheel (position it with `gamelevel.xml`).

You can also alter the music and sound effects (all of these are optional):

- `music.mp3`: Plays and loops at the title sequence, and in-game.
- `voice.mp3`: Plays once in-game before the music plays and can be used to give vocal instructions.
- `titlemusic.mp3`: Overrides the music played at the title screen.
- `titlevoice.mp3`: Plays once at the title screen before the title music plays.
- `pickup.mp3`: Overrides the sound effect for collecting a pickup item.
- `pickupsfound.mp3`: Plays when all of the pickups in the scene are collected.
- `titlemusic.mp3`: Overrides the music played at the title screen.
- `gamewin.mp3`, `gamelose.mp3`, `gamewin-`

Welcome to the Future

REGISTER NOW!
www.iTVcon.com

CALL FOR PAPERS NOW OPEN!

LIVE SIMULCAST!
AROUND THE WORLD ON SYS-CON.TV

iTVCON.COM
INTERNET TV CONFERENCE & EXPO 2006

Coming in 2006 to New York City!

“Internet TV is wide open, it is global, and in true ‘Web 2.0’ spirit it is a direct-to-consumer opportunity!”



**For More Information, Call 201-802-3023
or Email itvcon@sys-con.com**

Welcome to the Future!

Did you already purchase your “.tv” domain name?
You can’t afford not to add Internet TV to your Website in 2006!

2005 was the year of streaming video and the birth of **Internet TV**, the long-awaited convergence of television and the Internet. Now that broadband is available to more than 100 million households worldwide, every corporate Website and every media company must now provide video content to remain competitive, not to mention live and interactive video Webinars and on-demand Webcasts.

20 years ago the advent of desktop publishing tools opened the doors for the creation of some of today’s well-known traditional print media companies as well as revolutionized corporate print communications. Today, with maturing digital video production, the advent of fully featured PVRs, and significant advances in streaming video technologies, **Internet TV** is here to stay and grow and will be a critical part of every Website and every business in the years to come.

It will also very rapidly become a huge challenge to network and cable television stations: **Internet TV** is about to change forever the \$300BN television industry, too.

The Internet killed most of print media (even though many publishers don’t realize it yet), Google killed traditional advertising models, and **Internet TV** will revolutionize television the way we watch it today. You need to be part of this change!

Jeremy Geelan
Conference Chair, iTVCon.com
jeremy@sys-con.com

PRODUCED BY
SYS-CON
EVENTS

List of Topics:

- > Advertising Models for Video-on-demand (VOD)
- > Internet TV Commercials
- > Mastering Adobe Flash Video
- > How to Harness Open Media Formats (DVB, etc)
- > Multicasting
- > Extending Internet TV to Windows CE-based Devices
- > Live Polling During Webcasts
- > Video Press Releases
- > Pay-Per-View
- > Screencasting
- > Video Search & Search Optimization
- > Syndication of Video Assets
- > V-Blogs & Videoblogging
- > Choosing Your PVR
- > Product Placement in Video Content
- > UK Perspective: BBC’s “Dirac Project”
- > Case Study: SuperSun, Hong Kong

- Track 1** Corporate marketing, advertising, product and brand managers
- Track 2** Software programmers, developers, Website owners and operators
- Track 3** Advertising agencies, advertisers and video content producers
- Track 4** Print and online content providers, representatives from traditional media companies, print and online magazine and newspaper publishers, network and cable television business managers



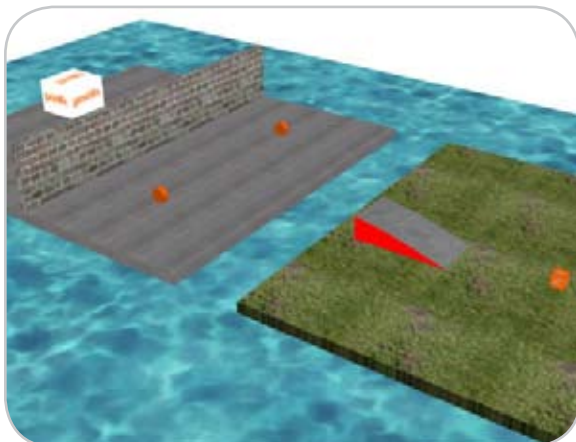
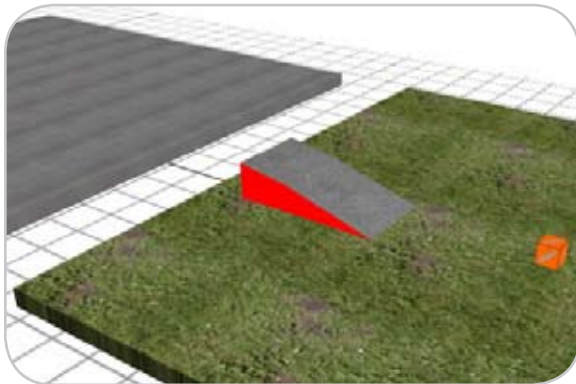
voice.mp3, gamelosevoice.mp3: Music and voice played at the game-win and game-lose screens.

Various elements of the game, such as the vehicle dynamics, can be altered using the gamelevel.xml file. See Instructions.doc for more information.

In the next section, you will export and load the sample scene.

Exporting and Loading the Sample Scene

When you download the application, you'll see a folder called "Sample 3D Scene." This contains a sample scene in the native format for all supported 3D applications. You can load



this scene into your application and get a feel for how some of the interactions can be set up. You'll see some examples of other game-related features, such as sky boxes and physical moving objects.

If you have an existing 3D scene that you would like to drive around, simply follow these steps:

- Load your 3D scene.
- Create a new object (for example, a box) in the scene:
Position it over some sort of ground model.
- Scale it to about the size of the required vehicle.
- Name this object carStart.
- Export the level, like before, as game-level.w3d.
- Run the executable file.

You should now be able to drive around your 3D scene.

Some Advanced Techniques

Consider undertaking these special projects to become more familiar with what I've covered in this article.

Creating a Sky Box

A sky box is a special game-related technique used to surround an environment with six prerendered images, mapped on to each side of a box object. By naming the object skybox, the game will automatically process the object and use it in the correct way.

Creating Physical, Movable Objects

When you create an object in the level, such as a crate or barrel, you can easily turn it into a movable object within the Havok physics system. To do this, place the tag moving before the object's name. The next time you export the scene, you will be able to collide with the object and it will behave in a physically correct manner.

You can discover more advanced techniques by reading Instructions.doc and Advanced Instructions.doc contained within the sample download file.

Viewing Existing Levels

You can see existing levels created using the training application in the CanDo Interactive Showcase (www.candointeractive.com/gamesdesign/car/levels).

Where to Go from Here

You now have experience creating and exporting Shockwave 3D content. You have seen how easy it is to export your environment, and you have an idea of the type of models that are suitable for use within that environment. Hopefully you have been inspired to create more Shockwave 3D content using the latest version of Director. ☺

Advertising Index

Advertiser	URL	Phone	Page
CFDynamics	www.cfdynamics.com	866-233-9626	19
Community MX	www.communitymx.com		21
EdgeWeb	http://edgewebhosting.com		5
Flashforward	www.flashforwardconference.com		15
InterAKT	www.interAKTonline.com/macromedia	4031 401.68.19	3
Intermedia	www.intermedia.net	888-379-7729	11
Macromedia Studio 8	www.macromedia.com/go/8_studio8	415-252-2000	52
MetalIQ	www.metalIQ.com	415-642-3332	31
NGASI	http://www.ngasi.com/dj.jsp	1-866-256-7973	51
Stream57	www.stream57.com	212-909-2550x1012	6
Vitalstream	www.vitalstream.com	800-254-7554	2

Advertiser index is provided as an additional service to our readers. Publisher does not assume any liability for omissions and/or misprints in this listing since this listing is not part of any insertion order.

**JAVA J2EE Hosting Automation Software
without costly support contracts or expensive
commercial software licenses.**



**AUTOMATES JAVA J2EE
(JSPs, Servlets, EJBs, Struts, and Spring)
HOSTING
INSTALLATION
CONFIGURATION
MANAGEMENT
DEPLOYMENT**

**for Apache JAKARTA Tomcat,
Geronimo, JBoss, Jetty,
and JOnAS
Application Servers.**

**FREE
DOWNLOAD**



<http://www.ngasi.com/jdj.jsp>

1.866.256.7973

Prices, license, and terms subject to change without notice. Please log on to our website for the latest price and terms. Copyright © 1999-2005 WebAppShowcase • All rights reserved • Various trademarks held by their respective owners.



WEB ALL-STAR seeks an Integrated Software Suite that's up for anything. Understand my need to create cool graphics one day, while tweaking CSS the next. Should be intuitive, multi-platform and not phased by my turbo-charged pace. High-maintenance is a no-no.



Different people. Different needs. One suite solution.

With the latest versions of Macromedia Dreamweaver®, Flash® Professional, Fireworks®, Contribute™, and FlashPaper™, the new Studio 8 is quite a catch. To meet Studio 8 and find all the tools you need to design, develop and maintain online experiences, visit www.macromedia.com/go/studio8_mxdj

macromedia®
STUDIO 8