

# MX

developer's journal

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

volume 4 issue 2 2006 www.mxdj.com



 **DREAMWEAVER**  
Building a Forum

 **FIREWORKS**  
The Low-Down on Pop-Up Menus

 **CENTRAL**  
Introducing COLA

 **BREEZE**  
Display a List of Recordings

 **CAPTIVATE**  
Right-Click Functionality

 **FLEX**  
Flash Comm

 **Using File Object for  
Video on Demand  
and MP3 Playback**



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



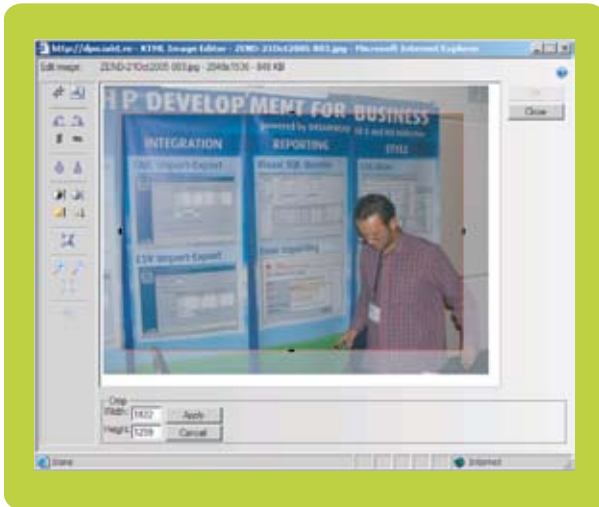
*Integrate Streaming Media  
Into Your Flash Projects*

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

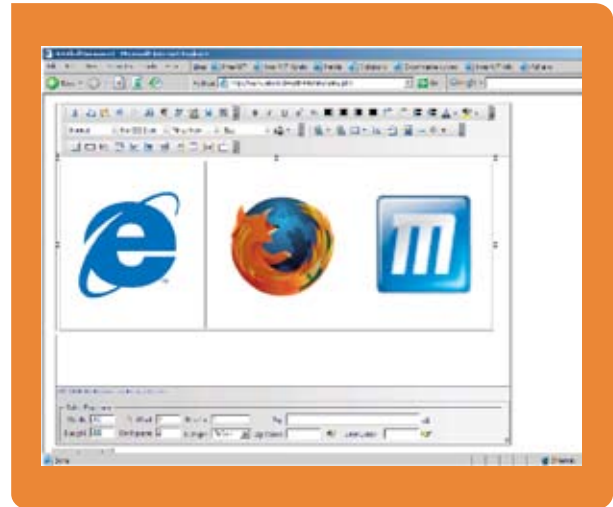
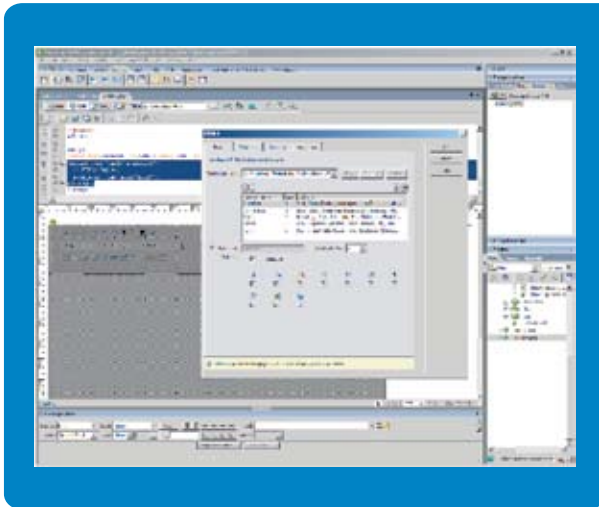
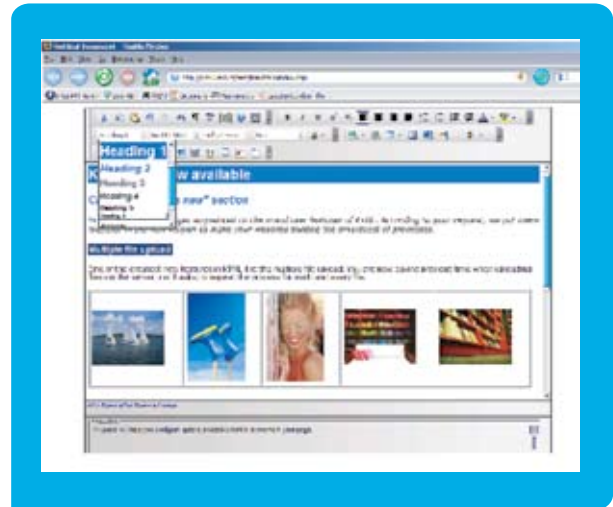
# KTML4 Word editing in browser

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/](http://www.interaktonline.com/ktml4/) for details




**24** Using File Object for Video on Demand and MP3 Playback  
*Flash Media Server 2*  
by robert sandie




 **7** Building a Forum with Dreamweaver  
*Part 2*  
by marius zaharia & cristian ivascu

 **18** Best Practices for Flash Player Detection  
*Can Flash detection be truly effective?*  
by robert hoekman

 **28** The Low-Down on Pop-Up Menus in Fireworks 8  
*The creation of a menu in various modes*  
by murray r. summers

 **32** Introducing COLA  
*COLA in a Nutshell*  
by ted patrick

 **36** Display a List of Recordings for a Meeting  
*Recording over the web*  
by joseph d. baarsch

 **40** Adding Right-Click Functionality  
*A solution that simulates right-click functionality*  
by steven shongrunden

 **44** Integrating Remote Sharded Objects  
*Flash Communication Server*  
by michael labriola



**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
- 1<sup>st</sup> 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](http://stream57.com)

e-mail us  
[streamline@stream57.com](mailto: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](mailto:subscribe@sys-con.com)  
U.S. Toll Free: 888 303-5282  
International: 201 802-3012  
Fax: 201 782-9600  
Cover Price U.S. \$5.99  
U.S. \$29.99 (12 issues/1 year)  
Canada/Mexico: \$49.99/year  
International: \$59.99/year  
Credit Card, U.S. Banks or Money Orders  
Back Issues: \$12/each

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

**Worldwide Newsstand Distribution**  
Curtis Circulation Company, New Milford, NJ

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

**Promotional Reprints**  
Megan Mussa  
[megan@sys-con.com](mailto: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

# Building a Forum with Dreamweaver

Part 2  
marius zaharia & cristian ivascu

tutorial

In Part 1 of this tutorial, we taught you how to pull information from a database in order to set up a forum front end and how to style it with CSS. In this article, you will see how easy it is to build a user authentication system for your forum.

## 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

This article originally appeared on [www.macromedia.com/devnet](http://www.macromedia.com/devnet). Reprinted with permission.



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 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.

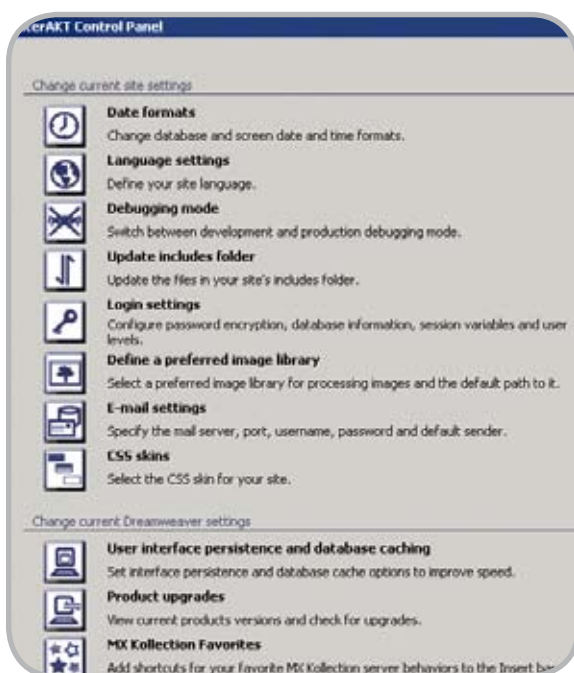
The page you will work on is register.php. The page layout is already created and is similar to the other forum pages

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 Kollection 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 click-





ing the Plus (+) and Minus (-) buttons on top of the grid.

- Remove the active\_usr and random\_key\_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 Kollection > 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 file-name: 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. Before moving to the next tab, verify that your interface.

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

### SYS-CON MEDIA

#### President & CEO

Fuat Kircaali, 201 802-3001

[fuat@sys-con.com](mailto:fuat@sys-con.com)

#### Group Publisher

Jeremy Geelan, 201 802-3040

[jeremy@sys-con.com](mailto:jeremy@sys-con.com)

### ADVERTISING

#### Senior Vice President, Sales & Marketing

Carmen Gonzalez, 201 802-3021

[carmen@sys-con.com](mailto:carmen@sys-con.com)

#### Vice President, Sales & Marketing

Miles Silverman, 201 802-3029

[miles@sys-con.com](mailto:miles@sys-con.com)

#### Advertising Sales Director

Robyn Forma, 201 802-3022

[robyn@sys-con.com](mailto:robyn@sys-con.com)

#### Advertising Sales Manager

Megan Mussa, 201 802-3023

[megan@sys-con.com](mailto:megan@sys-con.com)

#### Associate Sales Managers

Kerry Mealia, 201 802-3026

[kerry@sys-con.com](mailto:kerry@sys-con.com)

### PRODUCTION

#### Lead Designer

Louis F. Cuffari, 201 802-3035

[louis@sys-con.com](mailto:louis@sys-con.com)

#### Art Director

Alex Botero, 201 802-3031

[alex@sys-con.com](mailto:alex@sys-con.com)

#### Associate Art Directors

Tami Beatty, 201 802-3038

[tami@sys-con.com](mailto:tami@sys-con.com)

Abraham Addo, 201 802-3037

[abraham@sys-con.com](mailto:abraham@sys-con.com)

#### Assistant Art Directors

Andrea Boden, 201 802-3034

[andrea@sys-con.com](mailto:andrea@sys-con.com)

#### Video Production

Ryan Palmieri, 201 802-3036

[ryan@sys-con.com](mailto:ryan@sys-con.com)

### SYS-CON.COM

#### Consultant, Information Systems

Robert Diamond, 201 802-3051

[robert@sys-con.com](mailto:robert@sys-con.com)

#### Web Designers

Stephen Kilmurray, 201 802-3053

[stephen@sys-con.com](mailto:stephen@sys-con.com)

Wayne Uffleman, 201 802-3057

[wayne@sys-con.com](mailto:wayne@sys-con.com)

### ACCOUNTING

#### Financial Analyst

Joan LaRose, 201 802-3081

[joan@sys-con.com](mailto:joan@sys-con.com)

#### Accounts Payable

Betty White, 201 802-3002

[betty@sys-con.com](mailto:betty@sys-con.com)

#### Accounts Receivable

Gail Naples, 201 802-3062

[gain@sys-con.com](mailto:gain@sys-con.com)

### CUSTOMER RELATIONS

#### Circulation Service Coordinators

Edna Earle Russell, 201 802-3081

[edna@sys-con.com](mailto:edna@sys-con.com)

#### JDJ Store Manager

Brundila Staropoli, 201 802-3000

[bruni@sys-con.com](mailto:bruni@sys-con.com)





**Login Settings**

Options | Database | Session | User levels

Specify database information

Connection: connForum Define...

Table: user\_usr

Primary key: id\_usr  Numeric

Username: name\_usr

Password: password\_usr

E-mail: email\_usr

Active: active\_usr

Level: None  Numeric

Random key: randomkey\_usr

? Select the table column storing the random key.

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 Kollection 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. 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 conditional 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:

**Intermedia.NET...**

**Now serving the hottest ColdFusion.**

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

Intermedia.NET's premier hosting services include:

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

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



**INTERMEDIA.NET**

Call us at: **1.888.379.7729**

e-mail us at: **sales@intermedia.NET**

Visit us at: **www.intermedia.NET**

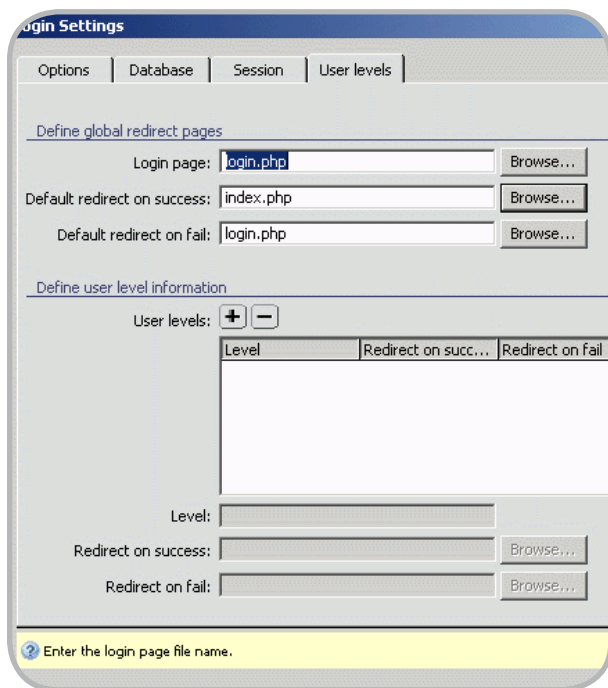


- Open the index.php page.
- Select the Logout link in the top-right corner.
- Apply the Show If User Is Logged In server behavior.

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, the register and login links will appear.

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.

## Creating the Link for Posting Messages

First, create the link to the page for posting messages. Follow these steps:

- Open the view\_message.php page in Dreamweaver.
- Place the cursor before the Logout link but inside the Show If User Is Logged In conditional region border.
- Tip: If you cannot place the cursor exactly, switch to Code view and place the cursor before the <a> tag corresponding to the Logout link. Then switch back to Design view.
- Enter the new link's text, Post message, and add a vertical bar separator after it.
- Select the text and click the Folder icon in the Property inspector.
- In the dialog box that opens, select the post\_message.php file from the site root.
- To define the URL parameters to pass, click the Parameters button. In the Parameters dialog box, configure a new parameter as follows:  
In the Name field enter id\_top.  
In the Value field, enter the following code:

```
<?php echo $_GET['id_top'];?>
```

This code retrieves the value of the URL parameter passed from the forum home page (when users select a topic).

Click OK twice to close the dialog boxes and add the link.

Because the link has been created before the logout link—but inside the conditional region—it is displayed only when the user is logged in. Preview the forum home page and log in with a valid user name and password, and then click a topic title. You'll see the link to post a message.

## Protecting the Page from Unauthorized Posts

Next, protect the page against posting messages without authorized access. Although users cannot see the post message link unless they are logged in, typing the page address in the Address bar of the browser will load the page even if users are not logged in. You must restrict access to the page, using the Restrict Access to Page server behavior from ImpAKT:

- Open the post\_message.php page.

- Apply the Restrict Access to Page server behavior from the Server Behaviors tab by selecting Plus (+) > MX Kollection > User Login.
- Select the Username and Password option to restrict access based on user name and password only.
- Click OK to apply the server behavior.

In the next section, you will create the form that enables users to post messages.

## Creating the Form That Enables Users to Post Messages

To allow users to post a message, you need to create an insert form that takes user input and adds it to the message\_msg database table. To build the page, follow these steps:

- Open the post\_message.php page in Dreamweaver.
- The layout is already created. In the main <div> container, select the placeholder text and remove it. This is where you will place the actual form for posting messages.
- Launch the Insert Record Form Wizard from the MX Kollection tab of the Insert bar.
- The wizard steps you through the process of generating the form and the insert transaction.
- In Step 1, select the database connection you've created in Part 1 of this tutorial (connForum) and the table to insert data into (message\_msg). After posting the message, users should be redirected to the view\_messages page, and the ID of the topic should be passed to it as a URL parameter. Otherwise, the page will be blank.
- To add the URL parameter to the page, without actually writing the code yourself, use the InterAKT Dynamic Data dialog box. Click the Browse button and select the view\_messages.php page in the site root. Type the following text after the page name in the After Inserting Go To text box:  
?id\_top=  
• Click the blue lightning bolt icon next to the text box. This opens the The InterAKT Dynamic Data dialog box.

Select the type of variable and enter its name:

In the Get Values From pop-up menu, select URL Parameter.

In the Variable text box, enter the name: id\_top.

- Click OK to return to Step 1 of the Insert Record Form Wizard.
- Click Next to proceed to Step 2, where you have to configure your form fields as follows:

Remove the idmsg\_msg and id\_init\_msg fields from the grid. These fields are not needed at this time. The idmsg\_msg column is needed only when replying (which is currently not the case), and the idinit\_msg field will receive its value after the transaction is completed.

Set the idtop\_msg field to be displayed as Text, and set Default Value to be equal to the URL parameter named id\_top. To do that, select the field and click the blue lightning bolt icon next to the Default Value text box. In the dialog box that opens, select URL Parameter from the first pop-up menu and enter id\_top in the Variable field.

Set the idusr\_msg field to be displayed as text, with Default Value equal to the session variable kt\_login\_id. Click the blue lightning bolt icon to access the InterAKT Dynamic Data dialog box, select the session variable kt\_login\_id, and then click OK.

- The date should also get its value automatically, in order to store the date and time when the message was posted. Select date\_message, set Display As to Text Field, and set Default Value to the following mark-up code: {NOW\_DT}
- ImpAKT uses a simple markup language that is replaced at runtime with the corresponding values. This markup language is easy to learn and is independent of the server model you are using. This means that {NOW\_DT} will always be replaced with the current date and time, whether you work on ASP, PHP, or ColdFusion.
- Keep the default settings of the subject\_msg field.
- Set the content\_msg field to be displayed as a text area.
- The subscribe\_msg column is stored as either 1 or 0 in your database. Therefore, you should display it as a check box; in the Submit As text box, enter Checkbox: 1,0.

- Click Next to proceed to Step 3 of the wizard, where you define any needed validation rules. The user will only see the subject and content fields. To stop users from posting empty comments, be sure to make the subject\_msg and content\_msg columns mandatory. Select each of them in the grid and select the Required option.
- To apply these settings and complete the wizard, click the Finish button.
- The web form is generated inside the center <div> container, and the necessary server behaviors are added to the Server Behaviors tab. If you need to edit any of these behaviors (the Insert Record transaction or the Validation options), simply double-click their names in the Server Behaviors tab and the corresponding dialog box will open.
- Since the Idtop, Idusr, and Date fields do not need to show up at runtime, select their rows and remove them. They should not be removed from the Insert Record transaction; remove them only from the HTML form. The idea is to have their values set automatically by the transaction—not to make them visible to users.
- In the Property inspector, enter Post in the Value text box to change the label of the Submit button.

In the next section, you will learn how to associate messages to a thread.

## Associating Messages to a Thread

At this point you can add messages in any topic you want by logging in and clicking the Post message link. However, records added through the insert form are not complete—they are missing the idmsg\_msg and idinit\_msg fields. (Remember when you removed them from the wizard in the previous section?)

For new messages (which are not replies to other messages), there is no parent message, so the idmsg\_msg column should be NULL. The idinit\_msg field stores the ID of the first message in a thread and will be used to notify subscribers when a reply is posted. All messages from the same thread have the same value of the idinit\_msg field.

The first message in any thread must have its idinit\_msg field value equal to its

ID (its primary key). However, the primary key of each message is only known when the message is inserted in the database; it cannot be known beforehand. In order to set the value of the idinit\_msg field after the message has been posted, you must execute some custom code that retrieves the ID of the message.

ImpAKT enables you to add custom actions and behaviors to any transaction using custom triggers, which are a simple way to extend the default server behaviors that come with ImpAKT. However, custom triggers may require good programming skills and a good understanding of how ImpAKT handles server-side code. That is why we will assist you in creating your first block of code for ImpAKT:

With the post\_message.php page open, select Server Behaviors tab > Plus (+) > MX Kollection > Forms > Custom Trigger. In the Custom Trigger dialog box, you have two tabs to configure:

Basic tab: Where you will write the code

Advanced tab: Where you will associate the custom action to the insert transaction and assign it a priority

- In the Basic tab, enter the following code:
- \$pkval = \$tNG->getColumnValue('id\_msg');
- \$query\_update\_initmsgid = "UPDATE message\_msg SET id\_init\_msg = '\$pkval' WHERE id\_msg = '\$pkval';"
- \$result\_update = \$tNG->connection->execute(\$query\_update\_initmsgid);
- The code performs an update on the last inserted message and sets the value of the idinit\_msg field equal to the message's primary key.

In the Advanced tab, you have to fill in the following options:

Trigger Name: Leave it to its default value

Trigger Type: This must be an AFTER trigger because it will be executed after the insert transaction; this way, the value of the primary key will be available

Trigger Priority: If you haven't added other triggers, you can leave it at its default value (50)

When you're done configuring the Custom Trigger, click the OK button to close the dialog box and return to Dreamweaver. You can edit the code at any time by double-clicking the Custom



Trigger server behavior in the Application panel.

Congratulations! You can now use the behavior to post messages to the forum. All the necessary columns will be filled in correctly by the transaction.

You can save and upload the page to the server. To view it, however, you will need to go to the login page and sign in. Otherwise, you will see a warning message similar to one.

After you log in, browse to the topic of interest and click the Post Message link.

You have now enabled users to post messages to the forum. In Part 4 of this tutorial, you will learn how to build a form for replying to messages and send automatic e-mail notifications when a reply is posted.

## Creating the Link to the Reply Form

The page used for replying to messages is called `reply_message.php`. Users can access it from the `view_message.php` page by clicking a Reply To Message link. To create this link, follow these steps:

- Open the `view_message.php` page in Dreamweaver.
- Look for the text “Reply to this message” and select it.
- The link will be displayed after each message. It must point to the `reply_message.php` page. The link must also pass the following parameters:  
Topic ID: `id_top`  
Message ID: `id_msg`  
Initial message ID: `idinit_msg`

- To create the link, right-click the selected text and choose Make Link from the menu.
- Select the `reply_message.php` page and then click the Parameters button to define the URL parameters mentioned above:

Click the Plus (+) button to add a new parameter.

Enter `id_top` for the name of the first parameter.

To specify a value, click the lightning bolt icon and then select the `idtop_msg` field of the `rsMessages` recordset.

- Click OK to close the user interface and add the URL parameter value.
- Repeat the steps above to add the other two parameters. They must be configured as follows:

For the message ID, enter `id_msg` for the parameter name and select the `id_msg` field as its value.

For the initial message ID, enter `id_init_msg` for the parameter name and select the `id_init_msg` field for its value.

- Click OK twice to close the dialog boxes and create the link.
- Select the link and apply a Show If User Is Logged In server behavior by selecting the Server Behaviors tab > Plus (+) > MX Kollection > Conditional Regions. Configure it to restrict the display based on user name and password only.
- Save the `view_message.php` page and close it.

Next you will build the reply form.

## Building the Reply Form

To build the reply form, follow these steps:

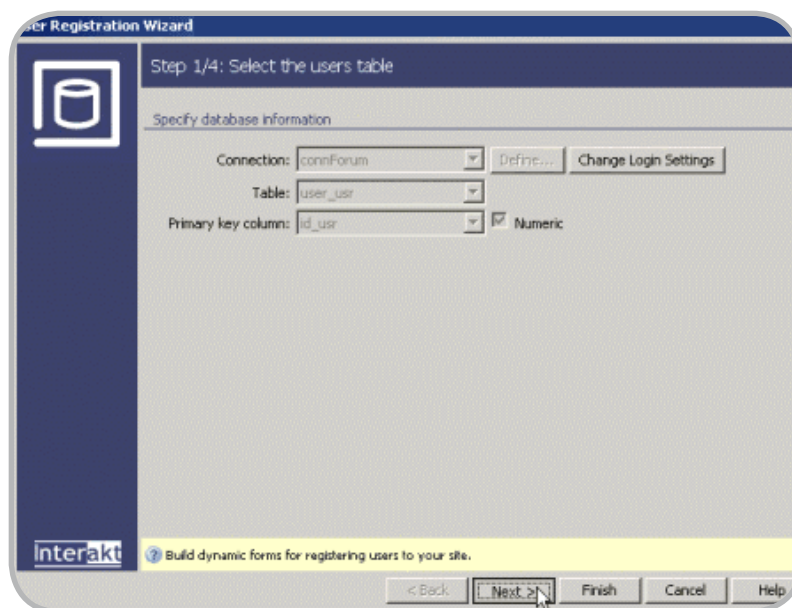
- Open the `reply_message.php` page in Dreamweaver. The page layout is already created.
- To allow only logged-in users to access this page, apply a Restrict Access to Page server behavior by selecting the Server Behaviors tab > Plus (+) > MX Kollection > User Login. Configure it as you did on the page for posting message in Part 3.
- Before building the form, you must create a filtered recordset that retrieves information about the message being replied to. The recordset is filtered by the message ID passed as a URL parameter from the `view_message.php` page. You will use this information to display the original message title and provide default values for the reply form.
- To create the recordset, click the Plus (+) button of the Bindings Tab and select Recordset (Query).

Enter `rsOldMessage` for the recordset name.

Set the database connection to `connForum` and select the `message_msg` table.

In the Filter pop-up menu, select the table’s primary key: `id_msg`. This must be equal to the `id_msg` URL parameter.

- Click OK to create the recordset.
- Remove the placeholder text from the main <div> container (the one with the white background) and type the following text:
- Reply to message:
- Next, drag the `subject_msg` field from the `rsOldMessage` recordset in the Bindings tab and drop it on the page, right after the text.
- Create a new paragraph after the dynamic text (press the Enter or Return key) and launch the Insert Record Form Wizard. You can access it from the MX Kollection tab of the Insert bar. Because you used the same wizard for the message-posting page, most of the interface fields should already be filled in, thanks to the user interface persistence (provided that this feature is



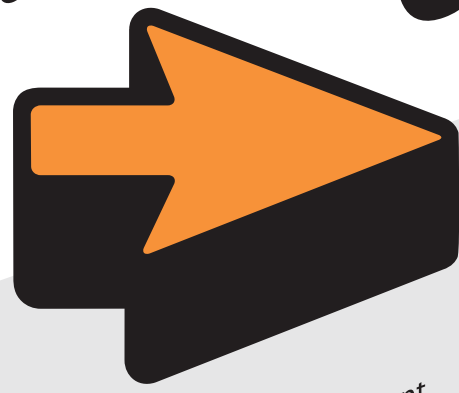
# FLASHFORWARD

## CONFERENCE & FILM FESTIVAL

Washington State Convention & Trade Center

SEATTLE  
Feb 27 - Mar 2

# 106



### Join the 15th Flashforward & Conference & Film Festival

- Keep your skills current
- Learn from experts
- Be inspired by the job fair
- 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](http://flashforwardconference.com)

Save \$50  
with this special discount code:  
**FFWDVMX6**  
For group discounts  
(over four people)  
and educational pricing,  
please contact:  
[registration@flashforwardconference.com](mailto: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:





- enabled in the InterAKT Control Panel).
- The first step is already filled in: the connection, table name, primary key, and redirect page are preserved from the last time you applied the wizard in the `post_message.php` page.
- Click Next without changing anything. In Step 2 of the wizard, each field still has its default value and field type configured as in the Post Message page. The following fields are already set up: `idtop_msg`, `idusr_msg`, `date_msg`, `subject_msg`, `content_msg`, and `subscribe_msg`.
- Because this is the reply page, you will have to change some settings and add the rest of the fields:

Set a default value for the message subject. It will display the original message subject, preceded by Re:, to indicate that the current message is a reply.

Click the InterAKT Dynamic Data icon (the blue lightning bolt next to the Default value text field) to select the `subject_msg` column from the `rsOldMessage` recordset.

After the dynamic value has been added to the Default Value text box, simply enter Re: before the automatically generated mark-up code.

You need to add two more fields to the insert transaction: `idmsg_msg` and `id_init_msg`. To add them, click the Plus (+) button on top of the grid and select the fields from the list.

For the `idmsg_msg` field, select Text in the Display As pop-up menu and use the `id_msg` field from the `rsOldMessage` recordset as its default value (again, click the blue lightning bolt icon to select the default value).

For the `id_init_msg` field, select Text in the Display As pop-up menu and use the

`id_init_msg` field from the `rsOldMessage` recordset as its default value (using the blue lightning bolt).

- In Step 3 of the wizard, apply validation rules on the transaction fields. Make all fields required, except the Subscribe check box. Also, all foreign keys (`idtop_msg`, `idusr_msg`, `idmsg_msg`, and `id_init_msg`) must be positive integers. Finally, the date field should have the Datetime format.
- Click Finish to close the wizard and apply the changes.
- After the form is added to the page, remove the rows containing the Date, `Idtop`, `Idusr`, `Idmsg`, and `Id_init` fields. These will remain in the insert transaction but should not be visible to the user.
- Rename the Insert record button; it should be labeled Reply.

The page for replying to messages is complete, and you can save it and upload it to the server. You cannot access this page directly unless you are logged in (because of the Restrict Access to Page server behavior).

With this page, the basic forum application is complete. Users can register accounts and view, post, and reply to messages.

In the next section you will enhance the forum application by keeping users informed about replies to their messages.

## Sending Reply Notifications by E-mail

In this section, you will enhance your forum by allowing users to receive an e-mail when a reply has been posted to one of the threads they subscribed to.

Subscribing to a thread is done when the Subscribe check box in the post or

reply message page is checked. When a user checks the Subscribe check box, the value of the `subscribe_msg` column from the `message_msg` table is set to 1. You will modify the `reply_message.php` page so that it sends an e-mail message to all users who have subscribed to the thread. Each user who checked the Subscribe check box when posting a message or a reply will receive an e-mail when another reply is posted by a different user.

The e-mail notification will be implemented using the Send E-mail To Recipients From Recordset server behavior. Before applying this server behavior, you must create the recordset that retrieves the e-mail addresses of all users subscribed to the thread:

- Open the `reply_message.php` page in Dreamweaver.
- The recordset that contains the e-mail addresses of all subscribed users will have to retrieve its data from two tables: `message_msg` and `user_usr`. To create this recordset, you also need to define these conditions.

The `subscribe_msg` column must be equal to 1 to prevent sending notifications to people who are not interested in receiving them.

The `id_init_msg` column must be equal to the `id_init_msg` URL parameter to make sure that only replies from the current thread are taken into account.

The `id_usr` column must be different from the `kt_login_id` session variable that stores the current user's ID. Otherwise, the user also receives a notification when he or she replies to one of the messages in the thread.

Go to the Bindings tab, click the Plus (+) button, and select the Recordset (Query) option from the pop-up menu that is displayed.

Configure the simple Recordset dialog box as follows:

In the Name text box, enter `rsSubscribers`.

In the Connection pop-up menu, select `connForum`.

In the Table pop-up menu, select `message_msg`.

Click the Advanced button to build the query conditions.

- In the Advanced Recordset dialog box you can enter your own SQL code and define variables. First, you must enter

### My Forum

<b>Name: *</b>	<input style="width: 90%;" type="text"/>
<b>E-mail: *</b>	<input style="width: 90%;" type="text"/> (E-mail)
<b>Photo:</b>	<input style="width: 80%;" type="text"/> <input type="button" value="Browse..."/>
<b>Password: *</b>	<input style="width: 90%;" type="password"/>
<b>Re-type Password:</b>	<input style="width: 90%;" type="password"/>
<input type="button" value="Register"/>	



the SQL code to perform the JOIN between the message\_msg and user\_usr tables. In the SQL text area, enter the following code after message\_msg:

```
LEFT JOIN user_usr ON idusr_msg=id_usr
```

Define the variables needed to build the conditions. Click the Plus (+) button next to the Variables grid and configure the Add parameter dialog box:

- In the Name text box enter `initid`.
- For the Default value enter `-1`.
- For the Runtime value enter `$_GET['id_init_msg']`.
- In a similar manner, add another parameter and configure it as follows:
- In the Name text box, enter `userid`.
- For the Default value, enter `-1`.
- In the Runtime value text box, enter the session variable storing the user ID:

```
$_SESSION['kt_login_id']
```

Write the SQL code that defines the conditions, using the variables you have defined. In the SQL text area, after the existing code, enter:

```
WHERE id_init_msg=initid AND id_usr!=userid AND subscribe_msg=1
```

Click OK to create the recordset.

When you have the e-mail addresses of the thread subscribers, you can apply the Send E-mail To Recipients From Recordset server behavior. This server behavior sends an automatic e-mail message to a list of e-mail addresses extracted from your database. You can access it from the Server Behaviors tab by selecting Plus (+) > MX Kollection > Send E-mail. The message will contain the entire reply contents, as well as a link to the thread.

Configure the server behavior as follows:

- In the Recordset pop-up menu, select the rsSubscribers recordset you created earlier.
- In the E-mail to menu, select the email\_usr field. This is the field that contains the e-mail addresses.
- In the From text box, enter an e-mail address or leave it at its default value. (This value can be changed by selecting E-mail Settings in the InterAKT Control Panel.)
- In the Subject text box, type the following text: A new reply has been posted to the Forum.

- The message body will contain both plain text (static values) and information from the database (dynamic values). With the InterAKT Dynamic Data, you can use fields from the current transaction, recordset fields, and other variables. You will include the reply subject and content in the e-mail message.
  - Place the cursor in the Content text area. On the first line you will address the recipient. Type Dear and then click the InterAKT Dynamic Data icon.
  - In the InterAKT Dynamic Data dialog box, select Recordset Field in the Get Value From pop-up menu. Choose the rsSubscribers recordset and the name\_usr field.
  - Click OK to insert the user's name. Notice the InterAKT mark-up code that is added in the message body.
  - To preserve formatting, you will send the e-mail in HTML format. You must use a basic set of tags to make sure the message is viewed as intended. For example, to create a new line, use the `<br />` tag.
  - After the mark-up code for the user name, enter the following:
    - `,<br />` A new reply has been posted to a thread you subscribed to:`<br />`
    - Tip: To make the text more clear, you can use the Enter (or Return) key to start a new line. However, new lines created like this will not be converted to line breaks in the actual message.
  - Add the message subject by using the InterAKT Dynamic Data dialog box again. This time you will retrieve the subject from the current insert transaction.
  - After the message subject, enter another `<br />` tag and then add the reply contents, as shown in the previous step. Add another `<br />` tag to separate the reply from the rest of the e-mail.
  - Finally, insert a link that points to the `view_message.php` page; this also passes the current topic's ID. After the last mark-up code, enter the following:
    - To read the entire thread, click `<a href='view_message.php?id_top=`
    - Select the `idtop_msg` field from the current transaction using the InterAKT Dynamic Data dialog box, as you did previously. This will add the corresponding mark-up for the URL parameter value.
  - Close the link tag. After the inserted mark-up code, enter the following:
    - `'>here</a>`.
  - The entire message body should look like this:
    - Dear {rsSubscribers.name\_usr},`<br />` A new reply has been posted to a thread you subscribed to:`<br />` {subject\_msg}`<br />` {content\_msg}`<br />` To read the entire thread, click `<a href='view_message.php?id_top={idtop_msg}'>here</a>`.
  - After you finish configuring the server behavior..
  - On the Options tab, verify that the HTML text option button is selected. This ensures that the message will be delivered in HTML format.
  - Click OK to close the dialog box and apply the server behavior.
- Now when users select the Subscribe option when posting a message or a reply.

Congratulations! You have finished building your very own forum application. Feel free to improve it in any way you want using any of the Dreamweaver 8 and ImpAKT server behaviors and features. Where to Go from Here

In this article, you have learned how to do the following:

- Set up a professional user authentication system for your web applications
- Build insert forms that submit information to a database
- Validate user input
- Send automatic e-mail messages

Upgrade to Dreamweaver 8 here and get your hands on a free ImpAKT license so you can start building dynamic web applications in no time. ☺

*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 avant-garde 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.*



# Best Practices for Flash Player Detection

*Can Flash detection be truly effective?*  
by robert hoekman

**t**he problem has riddled developers since the birth of Macromedia Flash: What happens when a site visitor doesn't have the version of Macromedia Flash Player needed for my content, or doesn't have one at all? Many answers have appeared over the years. And so far, the work has not been the result of developers trying to reinvent the wheel. Instead, it's been the result of developers trying to invent a wheel that actually works.

Can Flash detection be truly effective? Can it be seamless for site visitors and simple for you to implement? Can it make the player upgrade process friendlier?

In this article, I'll discuss the pros and cons of several approaches to Flash detection (and include a Flash Detection Experience Matrix). I also describe the new Flash 8 detection script, which, along with Flash Player Express Install, just might be the answer we've all been waiting for.

## A Brief History of Why We Need Flash Detection

With new versions of Macromedia Flash Player appearing on the web roughly every 18 months for the last several years, the task of ensuring your site visitors' player versions are compatible with your content can be downright frustrating. On the flip side, nothing is a bigger turn-off than landing on a web page that points a finger at you and says, "Macromedia Flash Player is required to view the content on this page," leaving you without a way to continue your experience without upgrading your player. After all, it's the developer's job to make sure you can do what you want to do on

his or her site. It's the developer's job to determine which version of Macromedia Flash Player is installed in your browser and to react accordingly.

User experience, as we all (should) know (by now), is vital to the success of a website or web application. Bad experiences are the equivalent of walking into a clothing store, asking a salesperson to see that fancy shirt hanging on the wall, and being told, flatly, "No." And telling me my system is somehow incompatible with your allegedly amazing website is like following up this experience by telling me to leave the store. Take a good look at the Flash detection experience implemented on your own website and try to see it the way your grandmother might see it. Does she have the plug-in she needs? If not, what does she see, if anything at all? What happens if she chooses not to upgrade? If she does upgrade, will she be able to quickly return to your page?

If you use Flash on your website, your only real option is to make sure you have an answer for all of the preceding questions, and implement a detection method that handles each situation.

So far, that's been much easier said than done.

## So Many Solutions, So Little Success

Over the years, I've met hundreds of Flash developers, and many of them have their own way of handling Flash detection, but several approaches have risen to the top of the stack to become at least somewhat standard. These approaches vary wildly in features, compatibility, effectiveness, and user experience, but one thing stands true regardless: none of

them works 100% of the time.

To make an informed decision about which approach is best for you and your site visitors, you need to know the facts. Lucky for you, I've done a lot of the legwork already and whipped up a fancy Flash Detection Experience Matrix for you, along with an overview of each of the tested approaches, and an introduction to the newest Flash detection script. Before you move on, however, it's important to understand what is needed to design a good Flash detection experience, for the simple act of displaying Flash content on your site presents several very key issues:

- Does the user have Macromedia Flash Player?
- If needed, does the user have JavaScript enabled?
- Does the user have the required version of Flash Player?

## Detecting the Existence of Macromedia Flash Player

To create a good detection experience, you need first to be able to specify what happens when Flash Player is not installed on the user's browser. If you offer no alternate content, is the user going to sit there staring at a blank screen, hoping that his or her web-savvy neighbor will come along and save the day? Most likely, the answer is no. Instead, your visitor will head off to your competitor's site, cursing your name the whole time. See, not every person that visits your site is going to want to perform a software installation just to view your content, regardless of how simple it is. So you need to offer your site visitor a compelling reason to perform the instal-

This article originally appeared on [www.macromedia.com/devnet](http://www.macromedia.com/devnet). Reprinted with permission.

lation. In other words, you need to tell your users what they will see or gain by installing the player, and make sure it's something that benefits them directly. Detecting Whether JavaScript Is Enabled If your Flash detection method is dependent on the use of JavaScript, you also need to plan what happens if the user has disabled the use of JavaScript in his or her browser. Ideally, you should be able to display alternate content, either asking the user (very nicely) to enable JavaScript and refresh the page, or offering HTML content in lieu of the Flash content.

## Detecting the Required Flash Player Version

Next, you must handle old versions of Flash Player. If your content is designed for Flash Player 8, for example, and your site visitor has Flash Player 6 installed, he or she should see alternate content. Again, the alternate content can either ask very nicely for the visitor to upgrade or offer HTML content in place of Flash content.

Finally, you probably want the solution to be easy to implement, and so effective that you don't have to think about it ever again.

Now that you have a list of required elements for a good Flash detection experience, let's take a quick look at some common options currently in existence.

## A History of Flash Detection Methods

Now, I'd like to give you a brief history of some of the methods developers have used to perform Flash detection to date.

### Flash Satay

Introduced in an article on A List Apart, Flash Satay was intended as a standards-compliant way to embed SWF files into HTML pages. The side-effect, however, is that it allows for the display of alternate content in the event Flash Player is not found, so it qualifies as a method of Flash detection. The idea is that, using valid XHTML, a browser with no Flash Player installed runs across an <object> tag trying to create a Flash object, realizes it can't do so, and simply skips to the next node, which can be a text node or an <img> tag specifying alternate content. In short, if the browser

can't create the object it needs, it displays the content in the next child node.

One issue with this solution is that it doesn't account for older plug-in versions. If the browser recognizes the object type, which it will if any version of Flash Player exists in the browser, it will create the Flash object. If your content uses Flash Player 7-specific features, and the user has Flash Player 6, she may see nothing but a white box on the HTML page.

This method also has the disadvantage of not streaming the SWF file when using Microsoft Internet Explorer. For large SWF files this can be reason enough not to use this method. Flash Satay is great for small movies, but once your content exceeds 100K you should look into alternative detection methods. They might not validate as standards-compliant XHTML, but they will provide a much better user experience, and that is what really counts in the end.

### The Redirect Method

Implemented in the Flash MX 2004 HTML templates and used as part of the old version of the Macromedia Flash Detection Kit, the redirect method involves running Flash detection on one page and redirecting the user to one of two other pages based on the result of the test. If the plug-in is installed and is the right version, the user lands on the page with the Flash content. If not, he or she lands on a page featuring alternate content. By default, this approach uses three HTML files and two SWF files (one is used as part of the detection process, and the other is for your content).

Using the approach is not only messy (requiring five files for one piece of content), but it also puts the user into a Back button loop. Clicking the Back button sends the user back to the detection script, which promptly returns him to a content page, rendering the Back button useless.

### The No-Detection Approach

This is really a non-approach. By using no Flash detection whatsoever, developers simply assume that every person visiting a site either already has the Flash plug-in installed, or the browser will attempt to handle the installation for the user. (Many modern browsers have a built-in prompt for starting the instal-

lation process, but not all of them.) And while this works fine in many cases, the browser prompt is a little scary. If you choose to install the plug-in, some browsers show you a dialog box titled "Security Warning."

Do you really want to put that in front of your users?

## The "Other" Approaches

Several common detection approaches were not even tested for the purposes of this article, but there's good reason for excluding each. Server-side detection, for example, is dependent on the server technology being used. Since this can change from project to project, it's not the ideal solution. And although Flash Player can handle version detection itself with a little bit of ActionScript, an inherent flaw with this approach, as you may realize, is that it can't possibly work if some version of Flash Player is not already installed. Users with no plug-in are left out in the cold.

Finally, the Mook FPI (created by renowned Flash developer Colin Mook) and used often by developers, is no longer even recommended by Mook himself. It also failed on at least one major Mac browser, so it didn't make the cut. As an alternative to detection, Mook recommends offering two entry points to your site, one for the HTML version and one for the Flash version. Of course, you'd have to do this for every major piece of Flash content on your site (and have an HTML version of it), and all along the way, you'd be assuming users know the difference between HTML and Flash. In many cases, they don't.

So what's a geek to do? If no solution is perfect, what can you do to create a good experience and ensure the visitor doesn't leave your site at the first sign of trouble?

## Using Flash 8 to Create Effective Detection Experiences

Well, folks, today is a good day: the search is over. The wheel has been invented. And tested. And taken on a nice, long road trip.

Say hello to the newest detection script, which you can implement easily using Macromedia Flash 8. Much like the Six Million Dollar Man, it's better, faster,



and stronger. And as an added bonus, you can actually rely on it.

Included as an HTML template option in Flash 8, this new detection script solution is implemented by setting your Publish Settings to publish an HTML page for your SWF file, and activating the Detect Flash Version check box prior to publishing. After you've published your files, the resulting HTML file contains all the code necessary to run effective detection, display alternate content to people with older versions of Macromedia Flash Player installed, and show different alternate content to those with JavaScript disabled.

In Macromedia Flash MX 2004, the solution was purely ActionScript-based and did not meet the needs of developers who wanted more fine-grained control over the user experience. The required browser redirect caused user confusion and in the end did not meet all the needs of the developer community. What is provided in Flash 8 is a huge improvement in that it takes the best ideas from scripts that have been in use for years and provides a solution that gets developers up and running quickly. Want to hear the best part? I personally tested the new script in several different browsers (the newest versions of Internet Explorer, Firefox, Netscape, Opera, and Safari) on Windows and Macintosh operating systems. Each method was tested with no plug-in installed, with an old version installed, and with JavaScript disabled. The new Flash 8 script is the only detection approach that performs correctly in every single case. Check out the Flash Detection Experience Matrix below:

Launch printable document (PDF, 16K)

You have to do a little bit of work to implement the Flash 8 detection script, as the detection template is not meant as the final version of a web page. However, it's not complicated, and doing so will help ensure you can keep all of your site

visitors happy. To make the code in the default HTML template useful in a real page, you need to do some copying and pasting, possibly modify the version number to detect, then go through the code and set what you'd like to display as alternate content when an old version of Macromedia Flash Player (or no player at all) is found, or JavaScript is disabled. The detection script has generic, default alternate content specified already, but you should really customize this to your (users') needs.

Here's a quick synopsis of the implementation process (demonstrated in the Captivate demo):

Play the demo: Implementing the Flash 8 Detection Script

- After you've exported HTML with Flash detection from Flash 8, move the JavaScript code found within the template's <head> tag to the same location in your page.
- In the Globals section of the JavaScript code, set the version number of the Flash version you'd like to detect.
- Next, move the second JavaScript instance from the template (the one found within the <body> tag, which triggers the first script) into the spot on your page where you'd like your Flash content to display.
- Replace the value of the alternateContent variable in the second JavaScript snippet with HTML for your own, custom content.
- Replace the HTML code found within the NOSCRIPT tag with appropriate alternate content.

As alternate content, consider what's actually needed for the user to continue his or her experience seamlessly. If the Flash content is nothing more than an ad banner, specify a GIF or JPEG image as alternate content and leave it at that. If the Flash content is required, consider using a teaser image (such as a screen shot of what the user would be seeing if the correct player was installed) in conjunction with asking the user nicely to upgrade and offering a link to the installer page. This way, you offer a compelling reason for the user to perform the installation.

If the user has JavaScript disabled (this is true in 10% of web browsers, on average), he or she will see the content

specified by the NOSCRIPT tag in your page. This could be a teaser image or plain text as well.

Most important is that what you show is helpful to the user experience. If the Flash content is not required, don't force (or even request) the upgrade. If it is required, offer a good reason to upgrade. If JavaScript is disabled, offer a good reason to enable it and refresh the page, or show appropriate alternate content (such as a GIF or JPEG image).

## Designing the Perfect Flash Detection Experience

Using the detection script discussed earlier, you will now be able to make sure that users have the correct player to view your content. If users do not have a version of Flash Player adequate to view your content, or they have browser scripting disabled, you will now have much greater control over the experience of upgrading users to the latest player. In the past the installation of the player differed between browsers and platforms, but now Macromedia has made a seamless process for developers to upgrade users. This new method is called Flash Player Express Install. Express Install is available for users who already have Flash Player 6.0r65 and above. This new installation process prompts users to upgrade to Flash Player 8 if they do not have it already and, when installation is complete, returns the user back to the page that started the process. If you really want your users to see the latest Flash 8 features in your Flash content, this is a great new way to upgrade them without a lot of pain.

Macromedia has enabled the new Flash Player Express Install mechanism with the release of Flash Player 8. This mechanism means that by sticking another small bit of HTML into your page, you can use the player itself to prompt the user to upgrade, and automatically return the user to your page upon successful completion of the installation. If the user chooses not to upgrade through the inline prompt, an event is fired that you can use to display alternate content. If you want to implement Flash Player Express Install, the full details and sample files are contained in the Flash Player Detection Kit.



# 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!

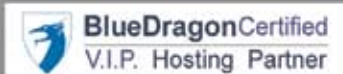
As one of the premier ColdFusion hosting community leaders, CFDynamics is constantly looking for ways to provide **better service** to ensure your satisfaction. Not only do we offer the finest in shared hosting plans, but we now offer the finest in 100% dedicated server plans! Now you can afford the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



Paper|Thin Partner



Visit us online or call to order!



So how can you implement the perfect Flash detection experience? Here's a quick synopsis, which combines the Flash 8 detection script and Flash Player Express Install.

- In your HTML page, detect the player version required for your page by implementing Flash 8 detection script, as described previously.
- Implement Flash Player Express Install, using the Flash Player Detection Kit and the tutorial in the next section.
- If version 6.0r65 or higher is found, but the player version is still not high enough for your content, the Flash Player Express Install will prompt the user to upgrade. If the user upgrades, the plug-in installs, the browser restarts, and the user is returned to your page automatically.
- If the user chooses not to upgrade, capture the event (with a bit of ActionScript, detailed in the next section), and either display alternate Flash content, redirect the user to another page (through a `getURL()` call), or perform an inline replacement of your Flash content through JavaScript. Either way, the user's experience continues seamlessly, and you earn yourself another happy customer.

For instructions on how to see each of the approaches described above in action, refer to the `READ_ME.txt` file included with the sample files you downloaded at the beginning of this article. Several versions of the same HTML page are included in the download, each of which uses a different Flash detection approach. To see the Flash 8 script solution in action on a real website, visit [www.rhjr.net/eReader](http://www.rhjr.net/eReader) (used for the sample page in the Captivate demo), which also includes examples of the new Flash Player 8 text rendering and drop shadow

“So how can you implement the perfect Flash detection experience?”

filters included with the Flash 8 authoring tool.

Finally, tell every web developer you know about this article. The sooner Flash Player Express Install becomes standard, the sooner we can stop frustrating users and start handling Flash experiences in an effective manner and improving user experiences on the web.

## Implementing Flash Player Express Install

The following tutorial describes how to implement the Flash Player Express Install mechanism.

*Note: Because users who have not installed the Flash 8 plug-in cannot see Flash 8 content, another SWF is used in place of your actual Flash content to handle the upgrade process. This SWF is named `playerProductInstall.swf` and is published for Flash plug-in version 6r65. Following these steps, you'll see how to modify the source file for this SWF and provide alternate content if the upgrade fails or is cancelled.*

This tutorial assumes that you have already added the Flash 8 detection scripts to a web page (as explained in the Captivate movie earlier in this article titled *Implementing the Flash 8 Detection Script*) and have already downloaded the Flash Detection Kit:

- Extract the Flash Detection Kit files from the ZIP file you downloaded and locate the “Express Installation” directory.
- Copy the following files from “Express Installation” into a new folder on your hard drive: `playerProductInstall.fla`, `playerProductInstall.html`, `playerProductInstall.as`, and `playerProductInstall-Callback.as`.
- Open `playerProductInstall.fla` in the Flash authoring tool. This file creates the SWF that users see if they do not have Flash Player 8 but do have Flash Player 6r65 or higher. Change the dimensions and background color to something that works with your web page. Generally, it's best to use the same dimensions as your own Flash content so that page rendering isn't affected as a result of displaying `playerProductInstall.swf`.
- Note: The dimensions of `playerProduct`

`Install.swf` must be at least 215 x 138 pixels, which is the minimum space required to display the Express Install prompt properly. Use the `AutoUpdater` symbol instance on the Stage as a guide.

- In `playerProductInstall.fla`, delete the instructions text from the Stage. If you like, change the color of the `AutoUpdater` symbol—for example, to blend in with the background color.
- Open `playerProductInstallCallback.as`. This file determines what the users see if the upgrade fails or is cancelled. The code in this ActionScript file looks like this:

```
function installStatus(statusValue) {
    if (statusValue == "Download.
Complete") {
        // Installation is complete. In most cases the browser window that this SWF
        // is hosted in will be closed by the installer or manually by the end user.
    } else if (statusValue == "Download.Cancelled") {
        // The end user chose "NO" when prompted to install the new player
        // by default no User Interface is presented in this case. It is left up to
        // the developer to provide an alternate experience in this case.
    } else if (statusValue == "Download.Failed") {
        // The end user failed to download the installer due to a network failure
        // by default no User Interface is presented in this case. It is left up to
        // the developer to provide an alternate experience in this case.
    }
}
```

- Replace the comments within each part of the `if/else` statement with your ActionScript that determines what happens in each event. For example, if `statusValue` (the variable that is tested in each part of the statement) is equal to `Download.Cancelled`, you might use

- a `getURL()` call to send the user to a different page on your site.
- Take a look at my eReader application ([www.rhjr.net/eReader](http://www.rhjr.net/eReader)) to see some examples of how this is used (of course, you'll have to uninstall your copy of Flash Player and install an older version first, then be sure to cancel the upgrade when the Flash plug-in prompt displays).
- Save your work when you're done.
- Republish `playerProductInstall.fla` to create your newer version of `playerProductInstall.swf`. Save `playerProductInstall.fla`, close both files, and then move `playerProductInstall.swf` into the same directory as your web page.
- Open `playerProductInstall.html` in a text editor (such as Macromedia Dreamweaver 8) and locate the JavaScript within the BODY element of the page.
- This script is similar to the one you added to your own web page earlier, but it also checks to see whether the user's machine has Flash Player version 6r65 or higher installed. If so, it displays `playerProductInstall.swf` in place of your Flash content.
- Copy the entire script (excluding the NOSCRIPT element, which does not need to change). In a moment, you'll paste this script into your own page.
- Open your own web page in a text editor (for the sake of the tutorial, we'll call it `index.html`). Highlight the script in the BODY element of `index.html` and paste the script you copied from `playerProductInstall.html` in its place.
- In the first part of the if statement in `index.html`, locate the `productInstallOETags` variable and replace the default height and width values with the height and width of your own, newer version of `playerProductInstall.swf`. The values you need to change are highlighted in the following code snippet:

```
if ( hasProductInstall && !hasRequestedVersion ) {
    var productInstallOETags =
'<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
+ 'width="550" height="200"'
+ 'codebase="http://download.
macromedia.com/pub/shockwave/cabs/
flash/swflash.cab">'
+ '<param name="movie" value="pl
```

```
ayerProductInstall.swf?MMredirectURL='
+MMredirectURL+'&MMPlayerType=ActiveX&
MMdoctitle='+MMdoctitle+' />'
+ '<param name="quality"
value="high" /><param name="bgcolor"
value="#3A6EA5" />'
+ '<embed src="playerProductIns
tall.swf?MMredirectURL='+MMredirectURL
+'&MMPlayerType=PlugIn" quality="high"
bgcolor="#3A6EA5" '
+ 'width="550" height="300"
name="detectiontest" align="middle"'
+ 'play="true"'
+ 'loop="false"'
+ 'quality="high"'
+ 'allowScriptAccess="sameDomain
"'
+ 'type="application/x-shock-
wave-flash"'
+ 'pluginspage="http://www.mac-
romedia.com/go/getflashplayer">'
+ '</embed>'
+ '</object>';
    document.write(productInstallOET
ags);
    Locate the first else/if part of the
statement and replace the values high-
lighted below:
} else if (hasRequestedVersion) { //
if we've detected an acceptable ver-
sion
    var oeTags = '<object
classid="clsid:D27CDB6E-AE6D-11cf-
96B8-444553540000"'
+ 'width="550" height="200"'
+ 'codebase="http://download.mac-
romedia.com/pub/shockwave/cabs/flash/
swflash.cab">'
+ '<param name="movie"
value="example.swf" /><param
name="quality" value="high" /><param
name="bgcolor" value="#FFFFFF" />'
+ '<embed src="example.swf"
quality="high" bgcolor="#FFFFFF" '
+ 'width="550" height="200"
name="detectiontest" align="middle"'
+ 'play="true"'
+ 'loop="false"'
+ 'quality="high"'
+ 'allowScriptAccess="sameDomain'
',
+ 'type="application/x-shockwave-
flash"'
+ 'pluginspage="http://www.macro-
media.com/go/getflashplayer">'
+ '</embed>'
+ '</object>';
    document.write(oeTags);
```

- This script runs if the correct Flash plug-in version is found, so replace the highlighted values with the dimensions and name of the Flash movie you'd like to display.
- Open `index.html` in a web browser and test it out. You'll need to uninstall Flash Player and install an older version to test it accurately.

If you followed the preceding steps, `index.html` should now show you the Express Install upgrade prompt when you come to the page using an older version of Flash Player.

## Where to Go from Here

To learn how to leverage Flash Player Express Install, refer to the Flash Detection Kit (FDK). The FDK also describes the other detection strategies mentioned in this article in more detail, including server-side and script-less approaches. You can see first-hand how the Flash 8 detection script works by creating a new Flash document in Flash 8, opening the Publish Settings, selecting the HTML tab, and selecting the Detect Flash Version check box. Click the Publish button and open the resulting HTML file in your favorite editor (such as Dreamweaver 8).

For a detailed description of the scripts used in the Flash 8 authoring tool, see *Future-Proofing Flash Player Detection Scripts* by Flash software engineer Michael Williams, the guru at Macromedia behind the Flash 8 detection script. Special thanks go to Michael for his incredible diligence in solving the continuing problem of effective Flash detection. He was a man on a mission, and users everywhere will benefit from his expertise. ☺

*Robert Hoekman, Jr., is a Macromedia Certified Professional and a user-experience advocate. He founded the Flash and Multimedia Users Group of Arizona (FMUG.az), currently works as a Flash developer for Interactive Alchemy, and is the author of Flash Out of the Box (O'Reilly, 2004) and the "10 Minutes with Flash" series of articles featured in InformIT.com's Flash Reference Guide. Learn more about Robert at [www.rhjr.net](http://www.rhjr.net).*



# Using File Object for Video on Demand and MP3 Playback

*One new feature of Flash Media Server 2 is the File Object feature*  
by robert sandie

**F**lash Media Server 2 offers a unique combination of traditional streaming media capabilities and a flexible development environment for creating and delivering media experiences to your audiences. These include traditional media like video on demand, live web-event broadcasts, and MP3 streaming, as well as rich media communication applications like video blogging, video messaging, and multimedia chat environments.

One new feature of Flash Media Server 2 is the File Object feature, which enables real-time read/write file access to a defined sandbox on your server, enabling content management systems and database applications without using Flash Remoting. This is a simple example that you can expand for use with File Object applications.

In this article, you will learn about a rather simple application that lets you display FLVs and MP3s on demand with predefined components you build using Flash Professional 8. This application automatically grabs FLVs and MP3s from a specific location on your server but is extensible for other uses. For example, to customize this application for a photo album using JPEGs, you would just redefine the filter so it calls JPEGs.

This application demonstrates new File Object properties on the server side and works in conjunction with the DataGrid, FLVPlayback, and MediaPlayer components on the client side. It demonstrates an alternative to XML-driven playback lists. If you wish to learn more about XML FLV lists, check out Lisa Larson's article, *Creating a Dynamic Playlist for Streaming Video*.

## Setting Up Your Environment

To follow this code walkthrough, you will need `on_demand_player.zip`, the sample file download that accompanies this article. Here's what you should do with the contents of the ZIP archive:

- Place the FileObj and MyCollection folders in your Flash Media Server applications directory (by default, this is `Flash Media Server 2\applications\`).
- Put the OnDemandPlayer folder anywhere on your development machine.
- Place the MP3s and FLVs that you want to stream into the `MyCollection/streams/_definst_` folder.

It's easy to change your default applications folder. Open `conf/fms.ini` in a text editor and change the variable `VHOST.APPSDIR`.

Flash Media Server 2 has two applications that control the logic, FileObj and MyCollection. When `OnDemandPlayer.swf` is instantiated, it accesses FileObj, which returns the content into a DataGrid. When the user clicks an object in the DataGrid, it calls the video based in MyCollection.

## Setting Up Flash Media Server

To make this application to work, you need to configure File Access and Stream Access.

### File Access

The File Object class allows access to a sandbox within your server file system. To protect against any misuse, Flash Media Server allows access to files within

a sandbox specified for the virtual host where the application instance is running. To define your server-side sandbox, go into your `application.xml` file in the FileObj folder. Define where you are storing the FLVs and MP3s that you wish the server to find:

```
<FileObject>
  <VirtualDirectory>/approval; C:\
Program Files\Macromedia\Flash Media
Server 2\applications\MyCollection\
streams\_definst_\
  </VirtualDirectory>
</FileObject>
```

You have just enabled secure file access to this defined sandbox. You can set up multiple file object directories by consecutively adding virtual directories.

### Stream Access

You will also need to set up your `Vhost.xml`, which you can find in your `conf/_defaultRoot/_defaultVhost/` subfolder in the Flash Media Server 2 folder because you must tell the server when you call "/approval" from the Vhost instance that it should look in the specified directory. You will need to define where your streams are located (should be the same as the setting you defined earlier for FileObject):

```
<VirtualDirectory>
  <Streams>/approval; C:\Program
Files\Macromedia\Flash Media Server
2\applications\MyCollection\streams\_
definst_\
  </Streams>
</Virtual Directory>
```

This article originally appeared on [www.macromedia.com/devnet](http://www.macromedia.com/devnet). Reprinted with permission.



## Examining the Server-Side ActionScript

There are two applications running on the Flash Media Server:

- FileObj supports file access and filter functions pointing to your stream sandbox.
- MyCollection supports FLVPlayback component functions.

Both run using the same default server-side ActionScript file, main.asc.

When you develop any Flash Media Server application, it is always easiest to start with the server-side ActionScript. Doing this allows you to debug and simulate the client-side actions using trace statements. This not only cuts down on your development time but prevents headaches later on.

You may notice that there are extra functions in main.asc in your FileObj folder. Notice there is more server-side ActionScript than is required for this application. This is what I use to handle all the file access commands in a familiar framework or API. I hope you will find this code useful and repurpose it for future file access functions.

Let's go over a few key functions in this file.

```
Constructor for File Class  
this.myFile = new File(name);
```

This construct command creates an instance of the file class as object myFile. The variable, name, can be either a file or directory. In this demonstration, you will be passing your alias directory into here to gather information about it (by default, approot).

```
File.List for File Class  
var dirList = this.myFile.  
list(filter);
```

This is the list function from which the DataGrid on the client side takes information. The list method returns an array with an element for each file in the directory. You can filter this further with the filter function:

```
function filter(name)  
{  
    if ( name.lastIndexOf(  
        ".flv") != -1 || name.lastIndexOf(".  
        mp3") != -1){  
        return true;  
    }  
    return false;  
}
```

```
mp3") != -1){  
    return true;  
}  
    return false;  
}
```

For instance, if you wanted to build from this and display only JPEG images, you could use this filter and list command to control what is being transferred to your client.

Here is a brief summary of other functions in this main.asc file in case you want to repurpose them for other applications using File Object. To find more specifics on the details of these function, read the Flash Media Server LiveDocs, which are a great resource for more information on each of these functions.

Here is a brief summary of the other File Object functions:

- newClient.mkdir creates a directory defined by dirName
- newClient.remove removes the given file object
- newClient.rename renames the given function
- newClient.copy copies the object to a given variable
- newClient.closeFile closes access to the file object

## FLV Playback Server-Side ActionScript

One of the most common errors that occurs when streaming with the FLVPlayback component in Flash Professional 8 is when you forget to add the main.asc file into the streaming Flash Media Server video directory. I have

included it as a part of this package so you will not encounter this problem yourself.

## Examining the Client-Side Setup

This client-side setup is rather simple because it is built with pre-existing components using basic OOP principles. The client-side framework consists of the following files located in the OnDemandPlayer folder:

- OnDemandPlayer.fla
- OnDemandPlayer.as

## FLA Components

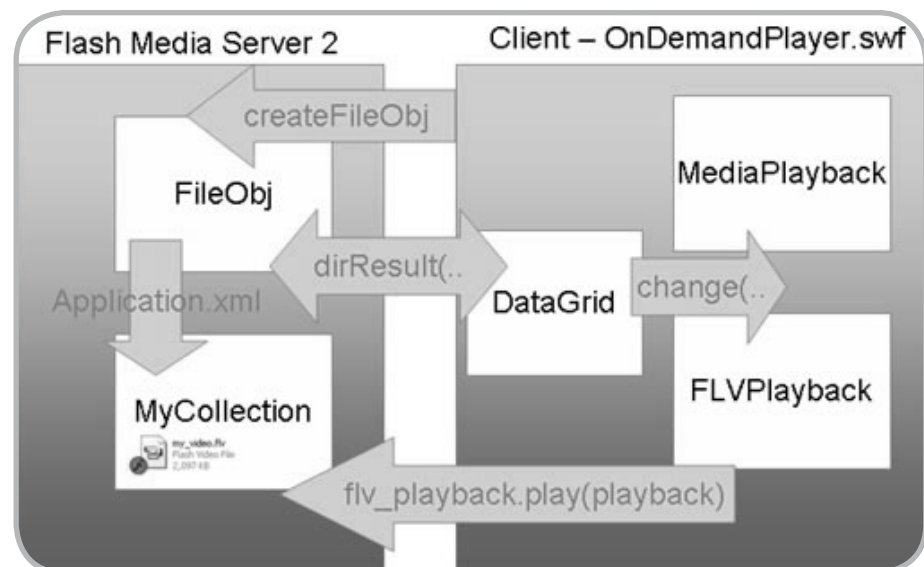
The FLA consists of three basic controls:

- FLVPlayback: Used for streaming FLVs (new in Flash Professional 8)
- MediaPlayer: The MP3 controller
- DataGrid: Used for connecting FLVs and MP3s with the underlying architecture
- OnDemandPlayer: A movie clip brought in pointing to ActionScript on the first frame (check the Library)

Connecting these client-side components is rather simple. Let's go over a few key points.

## ActionScript 2.0 and MovieClip Classes

A common best practice using OOP principles for scalable applications involves attaching movie clips as classes. This is an excellent method for organizing and attaching ActionScript classes to the Stage. Peldi, one of the architects behind Breeze,





describes this as a best practice. So if you are wondering how OnDemandPlayer.as is attached to OnDemandPlayer.fla, check out the Library.

This is comparable to using an #include file but is more manageable by simply attaching a movie clip from the Library:

```
attachMovie("OnDemandPlayer","mc",0);
Connecting with the Server
```

The file access handshake with the server is set up through a basic makeConnection function:

```
public function makeConnection():
Void {
    nc = new NetConnection();
    nc.connect("rtmp://" + serverName + "/"
    + appName);
    nc.owner = this;
    nc.onStatus = function(info) {
        if (info.code ==
        "NetConnection.Connect.Success") {
            owner.dir();
        }
    };
}
```

### Publishing Your Folder Directory

This connects to and sends Flash Media Server the folder name from which you wish to obtain your FLVs and MP3s:

```
nc.call("createFileObj", null, folder-
name);
```

### Getting the Stream Length

This connects with Flash Media Server to obtain the length for the FLV and MP3 lengths:

```
var streamLength = owner.nc.call("getS
treamLength", null, stream_name);
```

The FLVPlayback component has stream lengths built in. To set the MP3 length in MediaPlayer and display the length in the DataGrid, it must grab the information from the FileObj main.asc file.

### Obtaining the List of File Objects

On the client side, you must gather the stream names as well as other information regarding File Object in an array. This is done with the dirResult function:

```
private function dirResult(folderName,
owner):Void {
    this.onResult = function(retVal) {
        for (var i = 0; i < retVal.length;
i++) {
            var flv_name = (retVal[i].name).
substr((retVal[i].name).lastIn-
dexOf("/") + 1);
            var index = flv_name.lastIn-
dexOf(".");
            var stream_name = flv_name.
substring(index + 1, flv_name.length)
+ ":" + owner.folderName + "/" + flv_
name.substring(0, index);
            var streamLength = owner.nc.call("g
etStreamLength", null, stream_name);
            owner.myDP_array.addItem({Name:flv_
name, Length:retVal[i].streamLength,
CreationTime:retVal[i].creationTime,
LastModified:retVal[i].lastModified,
Size:retVal[i].length});
        }
    };
}
```

Notice that the object retVal contains more than simply the stream name. It

contains also creationTime, lastModified, and length (size of file).

These are just a few of the details you can gather using the List command on the server side. For a more complete list of File Objects check out Property Summary for File Class in the Flash Media Server LiveDocs.

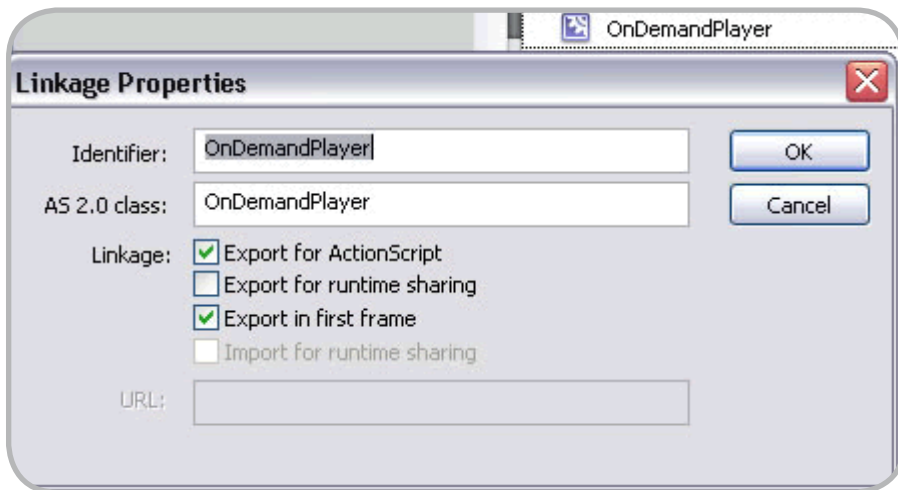
### Where to Go from Here

Using the file access and File Object features in Flash Media Server is an easy way to display content as an alternative to XML-driven data lists. The example shown in this article is a simple use case for just showing whatever media assets you have on your server. For other applications, you may want to consider combining the extensibility and organization of XML and ease of use with File Access as the perfect method for Flash Media Server on-demand applications. With this application you can easily drop your entire Flash video and MP3 media collection onto your server to stream your collection.

As learn about and come up with questions regarding Flash Media Server development, check out the Chattyfig list. This is the premier list for Flash Media Server developers and always contains informative development discussions. As a developer, I rely on this valuable resource whenever I face a tough problem. Just be careful what you ask, however, because if the answer is a simple Google's search away, they will hound you for it.

I hope this has helped you understand the new file access feature built into Flash Media Server. I wish you the best of luck with your future Flash Media Server adventures. ∞

*Robert Sandie interned at Macromedia as a product manager for the Flash Media Server team. He recently earned his degree in Computer Science and Business from Lehigh University in Bethlehem, Pennsylvania, where he also played football for the Mountain Hawks. Robert has a wealth of experiencing leading Flash projects and developing Flash video applications. To view his latest endeavors, catch up with him at robert-sandie.com.*



# 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](http://www.communitymx.com)



Visit [www.communitymx.com/trial/](http://www.communitymx.com/trial/) for your free 10 day trial.

**COMMUNITY**  
MX



# The Low-Down on Pop-Up Menus in Fireworks 8

*A brief description of the creation of a menu in various modes*  
by murray r. summers

Those of you who follow the Fireworks forum postings will probably recognize my name. I suppose I may have gotten a reputation as a crusader who opposes the use of the original Fireworks Pop-Up Menu as implemented in Macromedia Fireworks MX 2004 and earlier. I'm here to tell you that you are absolutely right. There are more than a few reasons not to use those pop-up menus, in my opinion, and I have spent a fair amount of time posting to this effect.

So why am I writing this article, then? Well, I have an agenda, as follows:

- I want to tell the Fireworks team that I appreciate the effort that they have devoted to making this new menu system.
- I want to tell you that I will no longer bore you with my usual posts about not using this menu system.
- I am very hopeful that the Dreamweaver development team will take these comments to heart and focus on the pop-up menu feature in future releases of that program.

To cut to the chase, I am happy to tell you that this feature is greatly improved in Fireworks 8.

In this article I briefly describe the creation of a menu in various modes. I examine and discuss the markup created by Fireworks ready for import into Dreamweaver, and describe what I think might be useful improvements to consider in the resulting markup.

## Building a Pop-Up Menu

Those of you who have used the pop-up menu feature in earlier versions

of Fireworks will be right at home in the current one. The setup and dialog panel are identical in Fireworks MX 2004 and Fireworks 8, offering exactly the same options and choices. Using this feature in Fireworks 8, you can build visually pleasing menu sets. It's easy to see its appeal.

To examine the pop-up menus that Fireworks 8 creates, you will need to create several extremely simple files. For the first one, open Fireworks and select File > New to create a new page—say, 640 x 400 pixels. Select the text tool and enter Link1 anywhere on the page.

Exit the text tool and then draw a slice around Link1 (I made mine 41 pixels wide and 23 pixels tall). Select it, name it link-1button, use the selected slice's bull's-eye to drop the context menu, and select Add Pop-up Menu from the menu choices. A panel will open, and you will make this slice show a simple, single-option submenu. In this panel enter Sub1-1 under the Text column head. There is no need to enter a link or a target.

Now click the Next button to advance to the Appearance tab (or just click the tab), where you can select that this is a vertical menu. With no further adjustment, the submenu will be drawn on top of the top menu choice, so you should advance to the Position tab and click the second position icon from the left. Leave the rest of the settings in this and the other tabs at their default values, and conclude this menu by clicking Done. Save this file as singlebutton.png.

Preview the menu by selecting File > Preview in Browser and then choosing your browser of choice. You should see a single-button menu which, upon rollover, drops a single-option submenu.

With singlebutton.png still open, you can now add more buttons to the menu. For the purposes of this tutorial, you will add three more menu buttons and give each of them a single pop-up submenu. To do this, deselect the visible slices option in the Web Layer tools, double-click Link1, and append Link2 Link3 Link4 to it (I have put two spaces between each link name).

Now exit text mode, enable the visible slices option, and select the slice tool to add a slice of similar dimension to link-1button to each of the three new menu options. Beginning with Link2, name each new slice using a nomenclature similar to that used previously: link2button, link3button, and link4button. One by one, select each new slice and use the bull's-eye to add a pop-up menu with a single submenu: Sub2-1, Sub3-1, and Sub4-1. Save this new menu arrangement as fourbutton.png.

A preview in browser should once again reveal what good work you do.

## Importing the Pop-Up Menu into Dreamweaver

Now that you have two menus created—one with a single button and a single pop-up menu, and one with four buttons, each with a single pop-up menu—it's time to get these into Dreamweaver and examine the markup.

Open singlebutton.png and select File > Export. Choose a landing site for the files from this (I created a folder called single and placed these files into that folder). Looking closely at the Export panel, make sure that the Export field is set to HTML and Images, the HTML field is set to Export HTML File, and the Slices field is set to Export Slices, with the

Include Areas Without Slices check box also enabled.

Click the Options button and choose your favorite HTML style and extension. Of the four check boxes on that HTML Setup panel, you need to have at least the bottom two selected (Use CSS for Popup Menus and Write CSS to an External File). Click OK to get out of this HTML Setup panel and the previous Export panel to save the files.

Do the same for fourbutton.png (I created a folder called fourbutton and placed these files there). You have now created all the files you need for the remainder of this tutorial. For me, this process gets the following relevant files in the single folder (and analogous ones in the fourbutton folder): singlebutton.html, singlebutton.css, and mm\_css\_menu.js. The singlebutton files will be used to describe the details of the menu and its construction. The fourbutton files are there to examine the CSS needed for this slightly more “real-world” example.

The process of inserting these menus into Dreamweaver is exactly the same as it has been for several releases now. Open your Dreamweaver page, place the insertion point where you want the menu inserted, and select Insert > Image Objects > Fireworks HTML. For the remainder of this tutorial, you will examine the code written by Fireworks to understand how these menus are built and how they work. I will use the original files written by Fireworks 8 for this explanation.

## Looking at the Code

A glance at the HTML immediately reveals that these menus are very different from those built in earlier versions of Fireworks. The most significant difference is that the links are not written into plain HTML markup on the page.

In Dreamweaver, I have defined a site especially for this tutorial. The root of this site contains both the single and the fourbutton folders. To see how Fireworks creates these menus, you should start simple. Use the Code view in Dreamweaver to take a look at the contents of singlebutton.html in the simple folder. You should see the following (the markup follows):

- The page is linked to mm\_css\_menu.js, a mere wisp of an 80-line JavaScript file

(compared to the original mm\_menu.js) containing the functions required to show and hide the submenus and to time their onscreen appearance.

- The CSS style sheet is specified with an embedded @import rule, which hides the styles from Netscape 4 and above.
- The CSS is linked with a “screen” media type, which hides it from printers or other devices.
- The menu and associated submenus are placed within a single outer div container, which will be called the “menu div.”
- The menu itself is a standard HTML anchor, placed in a table cell. This table will be called the “menu table.”
- The submenu is actually a nest of div tags placed in a div container, located at the bottom of the menu div’s markup below the table’s ending tag and above the closing </div> tag. This extra nesting is done presumably to accommodate complex, multilevel submenu systems.
- The submenu itself is nested within a container div that’s specific for that submenu.
- In the setup for the pop-up menus (the Appearance tab), you can elect to use either HTML or images for the submenu links. In this tutorial, you left the default values (HTML). Thus, the submenu is created from a plain text link, in this case, “Sub1-1.”
- There is a definite nomenclature used to identify each of these elements: the menu div, the submenu container div, and the specific submenu div. This nomenclature is a bit arcane, as you will see.

The salient HTML markup is shown here (along with my additions and showing the naming convention adopted for the divs):

```
<div id="FWTableContainer165705776"><
!-- This is the outer menu container
div -->
<table border="0" cellpadding="0"
cellspacing="0" width="640"><!-- This
is the menu table -->
<!-- fwtable fwsrc="singlebutton.
png" fwbase="singlebutton.gif"
fwstyle="Dreamweaver" fwdocid =
"165705776" fwnested="0" -->
<tr>
```

```
<!-- Shim row, height 1. -->
...snip...
<tr><!-- row 1 -->
...snip...
</tr>
<tr><!-- row 2 -->
<td rowspan="2">...snip...</td>
<td><a href="javascript:;"
onMouseOut="MM_menuStartTime-
out(1000);" onMouseOver="MM_menuSho
wMenu('MMMenuContainer0817151145_0',
'MMMenu0817151145_0',0,23,'link1b
utton');"></a></td><!-- This is the top
menu link image (link1button) -->
<td rowspan="2">...snip...</td>
<td>...snip...</td>
</tr>
<tr><!-- row 3 -->
<td>...snip...</td>
<td>...snip...</td>
</tr>
<!-- This table was automatically
created with Macromedia Fireworks
-->
<!-- http://www.macromedia.com -->
</table>
<div id="MMMenuContainer0817151145_
0"><!-- This is the submenu div -->
<div id="MMMenu0817151145_0"
onMouseOut="MM_menuStartTime-
out(1000);" onMouseOver="MM_menuReset-
Timeout();">
<a href="#"
id="MMMenu0817151145_0_Item_0" cla
ss="MMIFVStyleMMMenu0817151145_0"
onMouseOver="MM_menuOverMenuItem('MMMe
nu0817151145_0');">
Sub1-1
</a>
</div>
</div>
</div>
```

The menu function is controlled by event handlers applied to the various </div> tags. The logic is remarkably similar to the logic we used some years ago when such menus were created with DHTML timelines! Moving the mouse over the top menu link shows the submenu with a call to MM\_menuShowMenu, while moving the mouse away starts an internal timer. When this timer fires, the submenu is closed. Moving the mouse over the submenu div stops the



“I am heartened to see that it is possible to open an ‘old-style’ menu and resave it as a CSS menu”

timer started above, with a call to `MM_menuResetTimeout`, while moving the mouse away from this div starts it again.

The CSS for this menu is written as follows:

- An initial rule forcing images in table cells to `display:block`. This is a workaround for Netscape’s strict interpretation of inline display of images with a space underneath for text descenders on pages without a valid and complete doctype.
- A cascade of rules controlling the styles of the various menu and submenu divs and their contents, as you would expect.
- A `position:relative` for the menu div. This allows the menus to exist inside a table cell if necessary, even though they use absolute positioning.
- A box-model hack for Internet Explorer prior to IE6 on the submenu.
- Absolute positioning on the submenu div to place it, as desired, under the top menu link.

The details of the CSS file’s rules can be obtained by examining the respective CSS files produced by the export.

For `singlebutton.html`, the CSS file is 81 lines long. For `fourbutton.html`, the CSS file is 282 lines long; the additional lines are required for the individual styling allowed for each submenu. This means that more complex menus require longer and longer CSS files.

I would be remiss if I didn’t say more about this particular naming scheme. The nomenclature for the div tag IDs (hence their style rule selectors) is quite obscure, not to mention voluminous. Perhaps it would be possible to look to the slice names as a way of simplifying these IDs, thereby reducing the space requirements for them to be listed—not only on the tag selector in Dreamweaver but also in the code and CSS. As a user, you might want to edit these manually to something a bit more readable.

In addition, providing an option to style each submenu independently of the others (this is really imposed by the user interface for the menus because each submenu is a separate entity) appears to offer great flexibility, but it comes at a heavy price. The markup on the page as well as the selector nomenclature is unnecessarily complicated by this flexibil-

ity. My personal opinion is that this extra styling for each submenu is unnecessary. The menu system would profit greatly from this simplification.

In summary, I think the design of these pop-up menus is relatively standard, if not leading-edge. Most menu systems now being shown, discussed, and used are built on a nested list model rather than isolated anchor tags with show-hide div actions. The real benefit of the former approach is simplicity in presentation when JavaScript is not available and in semantic markup—lists having an inherent meaning, not shared by a series of adjacent anchor tags. I would personally like to see such an approach as a future enhancement in Fireworks.

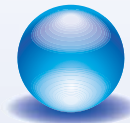
The menu should work well with Dreamweaver templates and should also be easily modified or debugged by someone with modest HTML/CSS skills. The menus themselves are readable by search engine spiders, although—like many other menu systems—they will not function without JavaScript enabled.

A final word of encouragement is appropriate, I think. I am heartened to see that it is possible to open an “old-style” menu and resave it as a CSS menu. This is a needed lifeline, in my opinion, for those who may be struggling with the previous all-JavaScript menus. I welcome its inclusion. ☺

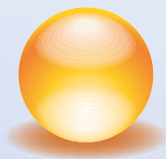
*Murray R. Summers is a Macromedia Certified Web Site Developer and Dreamweaver Developer and has contributed chapters to Dreamweaver 4 Magic (New Riders Press, 2001), Dreamweaver MX 2004 Magic (New Riders Press, 2003), and Dreamweaver 4: The Missing Manual by David Sawyer McFarland (O’Reilly, 2001). He was technical editor on the latter publication and on Roadmap to Macromedia Contribute by Joseph Lowery (Macromedia Press, 2003). Along with Brad Halstead, he cowrote the definitive study of the use of Dreamweaver MX snippets, library items, and templates in Dreamweaver MX Templates (New Riders Press, 2002). He has served as an invited speaker at TODCON (2001, Orlando), TODCON II (2002, Chicago), TODCON North (2003, Toronto), TODCON MX (2003, Las Vegas), TODCON IV (2004), and TODCON V (2005), and will hopefully continue to do so!*

H  S T I N G . c o m

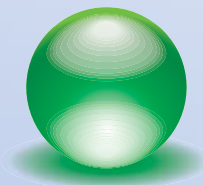
THE FIRST NAME IN HOSTING



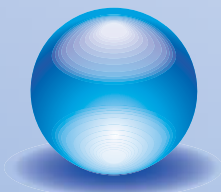
Developers!



Developers!



Developers!



We Love Them!

SERVE. SUPPORT. SECURE. STORE.

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



# Introducing COLA

*COLA in a Nutshell*  
by ted patrick

Applications for Macromedia Central can be licensed in any number of ways, thanks to the Open Distribution license introduced with Central 1.5. Getting started with managing licenses for your users can be tricky. Central Open Licensing Architecture (COLA) solves this problem by solving the issues of license management and providing a starting point for developers to implement their own license models. Version 1 is a reference implementation for PHP or Macromedia ColdFusion.

## COLA in a Nutshell

COLA provides a method for processing and tracking licenses for commercial Macromedia Central applications. It is server software in Macromedia ColdFusion and PHP that provides secure, automated payment and software licensing management, and enables you to sell an application to an end user. COLA gives developers a wide range of licensing options and can be modified as needed. Since COLA integrates with PayPal, software licensing and delivery is fully automated. The software ships as a reference implementation providing a solid foundation from which to customize a licensing solution. COLA is released as an open model—you can use it for free and modify the source code to suit your needs.

COLA manages software licenses on the server side through the product.xml file for a Central application. For every licensed user in COLA, there is a unique

URL to the product.xml file. As such, COLA has the ability to manage a single user allowing for simple license creation and expiration. COLA also provides the ability to detect fraudulent installations and prevent them from occurring. Unlike the try/buy model, you can easily revoke a license server-side that has been provided to an end user.

COLA is a gift to the Central developer community. However, it is not the final word on licensing Central applications. Thanks to COLA, developers can construct their own licensing schemes, and are encouraged to use COLA as a starting point. COLA is provided as is; it is supported neither by Macromedia nor by PowerSDK Software Corp.

## COLA Security

COLA security is based on the installation services provided by Central. When any application is installed, the URL to the product.xml file is encrypted and stored within Central. To make COLA simple, we reused this feature to store a license securely in the same way. In COLA, the individual license identifier is encoded in the URL to the product.xml file. This provides a first line of defense, since Central stores only the encrypted form of the URL. The second line of defense is that COLA can detect multiple installations for a single license. Should a user obtain a product.xml URL, the server can detect unauthorized installations and licenses can be easily revoked.

COLA passes values using the MD5 hash format. MD5 is designed to produce

a unique hash from an initial value. It is theoretically impossible to calculate the value from a hash so these are typically described as one-way hash functions. Within a COLA secured application, the License ID (LID) only occurs within the product.xml URL. All other files are secured under the hash of the LID. If you look at the Central path for a COLA application you can see the hash(LID) value but not the LID. This makes it easy for any part of an application (pod, agent, or shell) to test the license server side by simply passing the hash of the LID.

Because COLA does not require any modifications of an application's SWF file, it is essentially transparent to the developer. The developer need only create a licensed version of a Central application and modify the product.xml template. You also have the option to use COLA's server-side logic to integrate security more tightly. COLA provides a means of testing the license of a running application through an XML file and provides a callback to disable secondary installation.

As I mentioned earlier, you are free to extend and modify COLA to suit your needs. COLA provides a solid foundation on which to build custom commercial Central applications. It is simple enough to secure a wide range of applications and can be customized easily.

## COLA Example

It is helpful to understand the flow of licensing a COLA application. The following steps walk you through the entire process of a user purchasing an application:

This article originally appeared on [www.macromedia.com/devnet](http://www.macromedia.com/devnet). Reprinted with permission.



- The user installs a demo version of Icon Builder from the Central Application Finder.
- Note: Demo and trial versions are normal Central applications with limited functionality or timeouts.
- The user clicks a purchase button within the demo version and is taken to the PayPal site.
- The user completes purchase at PayPal.
- PayPal IPN notification calls the COLA IPN script. IPN notification is a private server-to-server HTTP/HTTPS POST request. The IPN request contains the purchase information and confirms that the user has purchased an application.
- The COLA IPN script creates a new License ID (LID) and an Install ID (IID) for the user. The script then e-mails the user an installation URL that contains the IID value along with a friendly message confirming the purchase.
- The user opens an e-mail and clicks the installation URL. The installation badge for the licensed application is displayed to the user.
- The user clicks the installation badge. Central is passed the licensed version of the product.xml URL, and the

licensed version of the application is installed. The product.xml file contains a <file> tag that invalidates the installation URL and installation badge by contacting the server. Note that the LID persists on the server; only the IID is single-use.

- The user has a COLA application installed. The installation URL is no longer valid preventing subsequent installations.

There are additional ways to use COLA beyond providing e-commerce functionality. The following list provides a few alternative ways of how you might use COLA:

- Enhanced security: Since each COLA installation contains a unique ID, you can augment security for network services with COLA. Instead of requiring only a user name and password, users must also access services from a Central application licensed to them.
- Private applications: COLA can be used to distribute private versions of an application. Say you have a customer service application that you do not want to expose publicly. Adding this application under COLA allows the

application to reside privately on the Internet and enables you to manage access closely through licenses.

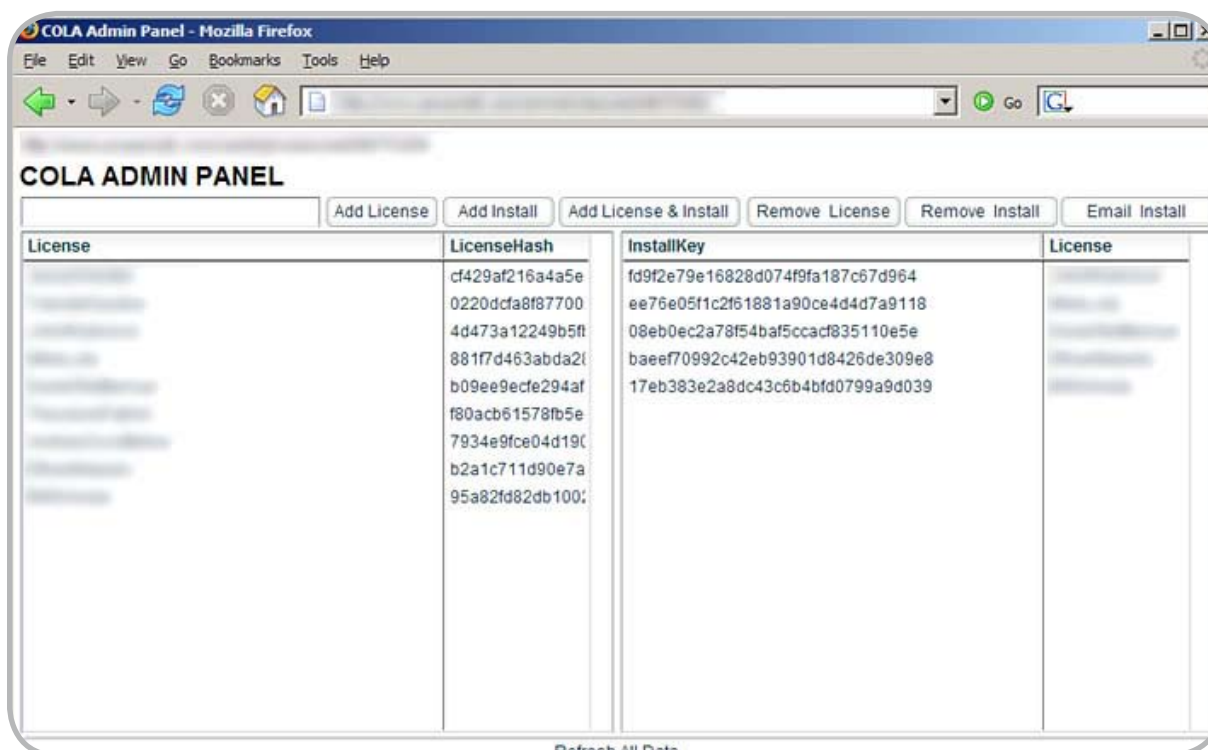
- Subscriptions: COLA can be used to secure subscription services. As each user is granted a unique license, you can allocate or remove content based on a user's license. Also, since licenses reside server side, you can upgrade a user from one type of access level to another with no installation changes.
- Personalized applications: With Macromedia Flex you can generate applications for end users based on their LID values. The contents of the SWF files installed within Central differ on a per-user basis.

COLA doesn't solve every problem for you. In providing an open solution, COLA enables you to add functionality to suit your own needs. Should you not find what you need in the base COLA features, just add it.

### COLA Benefits

COLA offers the following benefits:

- A wide range of features for developing Central applications.
- A Rich Internet Application (RIA) for





“COLA  
is free,  
server-side  
software  
that assists  
developers in  
creating secure  
commercial  
Central  
applications”

managing COLA licenses on the server side. The RIA unifies all the server features into a single location. This enables developers to view, create, or delete licenses and installations from the server. The RIA also enables you to create a license without requiring payment for testing, promotional, or other needs.

- Integration with third-party payment providers. The reference application ships with support for PayPal through Instant Payment Notification (IPN). When a payment is processed, IPN calls a URL. COLA supplies a working example of an IPN application that can create a license and installation, and then e-mail customers their licenses.

Basic support for monitoring installation activity and detecting fraud. This can be easily integrated with reporting to provide a view of an installation activity.

### COLA License and Installation

COLA is an unsupported product (see disclaimer). It is provided in source format for your own use within development projects. COLA is free and you can extend it to suit your needs. In general, COLA facilitates the creation of commercial applications in Central while providing a rich set of features.

### Installing COLA

To try COLA for yourself, follow these steps:

- Download cola1.0.zip if you haven't done so already. This source distribution contains implementations for ColdFusion and PHP in the COLA\_CF and COLA\_PHP subdirectories, respectively.
- Copy the appropriate implementation to your hosting server. The directory tree should be posted to the directory from which you will deliver your application.
- In the “try” directory, place the trial version of your application. Modify the product.xml file to install your application.
- In the “buy” directory, place the licensed version of your application. Modify the product.xml file to install correctly.

- If you are installing COLA\_CF implementation, create the “cfcola” data source on your ColdFusion server, using the cola.mdb Access database provided (or create an equivalent data source for your database of choice).
- Edit cola.cfm or manager.php to reflect your install location, and to change the default value for the admin key (used to view the admin panel and manage licenses and installations).
- Consult readme.html for further instructions.
- Navigate to the admin panel in a browser. If you installed COLA for an application hosted at <http://mydomain.com/myapp>, and your admin key is admin (not recommended!), the proper URL would be as follows:

```
COLA_PHP: http://mydomain.com/myapp/  
admin/  
COLA_CF: http://mydomain.com/myapp/  
cola.cfm?admin=admin
```

You should now be able to install trial applications, create licenses and installation IDs using the COLA admin panel, and install licensed applications. Refer to [interface.html](#) to determine the proper URLs for testing each operation.

### Summary

COLA is free, server-side software that assists developers in creating secure commercial Central applications. It offers a solid foundation on which to start development. After having written several commercial Central applications myself, I can confirm that COLA provides a balance of features to support most commercial development. Because it is distributed as an example of open-source software, it allows for deeper integration and tighter security. ☁

*Ted Patrick is the founder and principal developer at PowerSDK Software Corp. As a consultant, Ted has contributed to the technical side of many leading Macromedia Flash development projects and delivered presentations at conferences on Flash technology. You can check out his blog for more information about Flash-related technology.*



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.

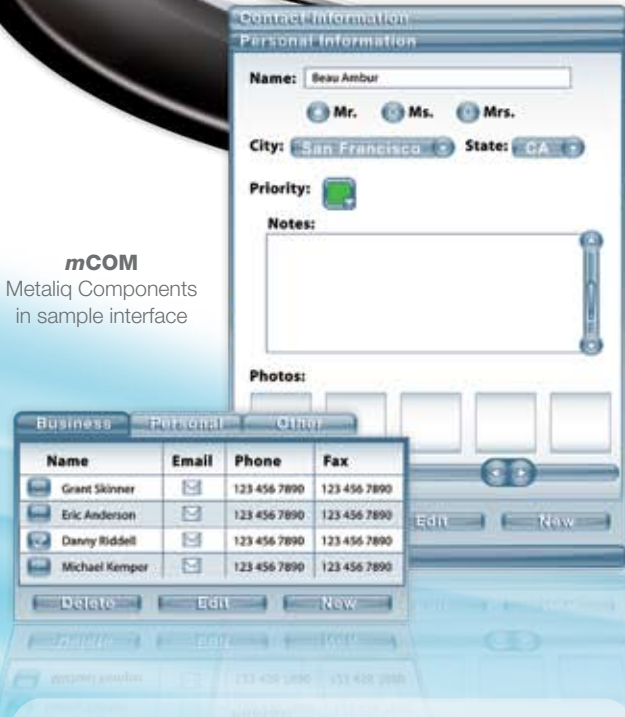
**mCOM** allow you to quickly create stable, lightweight data-driven interfaces that are extensible by designers and developers alike, both through extending classes and a simpler visual skinning process. **mCOM** include a fully implemented advanced focus manager that works seamlessly with all of our 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](mailto:mCOM@metalIQ.com) or visit us at [metalIQ.com/mCOM](http://metalIQ.com/mCOM).



# Using Breeze Web Services to Display a List of Recordings for a Meeting

*The Breeze 5 Meeting module creates SWF files to allow for easy delivery of the recording over the web*  
by joseph d. baarsch

One of the unique features of the Macromedia Breeze 5 Meeting module is the ability to capture or record the meetings as they take place for future replay. After a recording session finishes, the Breeze 5 Meeting module creates SWF files to allow for easy delivery of the recording over the web.

Another useful feature of Macromedia Breeze 5 is the ability to integrate Breeze in existing systems through the use of the Breeze Web Services API. Through web services, an exciting amount of opportunities become available to Breeze implementers—opportunities such as creating custom event listings and event and meeting detail pages. Another opportunity includes building reporting dashboards for graphically representing specific data sets on meeting usage, course reports, and event effectiveness. The beauty of the data being available as web services is in how it allows you to manipulate and integrate the data in ways that fit your organization best.

In this article, I focus on the Breeze Web Services API that allows you to retrieve a list of all the recordings associated with a single meeting. You can imagine the different contexts in which this information may be useful. For example, Colleges and Universities that want to integrate Macromedia Breeze into a course management system can quickly allow their clients to see a list of recordings for a meeting without leaving the course management system. Instructors who record every lecture with the Breeze 5 Live Meeting Module will find this list embedded in the context of their course extremely useful. After you have finished

this tutorial, you will be able to visualize how your organization can take advantage of the Breeze Web Services API.

## Getting Started

To get started, I will review how applications interact with Macromedia Breeze 5 through web services.

In this tutorial I will review an application that does the following activities:

- Create a Breeze session
- Retrieve XML containing a recording list from Breeze 5
- Convert XML to a Coldfusion MX 7 query object
- Display the recordings query

Unzip the recording\_list.zip file you downloaded at the beginning of this article. In this archive you will find the following files:

MeetingManager.cfc

This file manages the retrieval and conversion of recordings from the Breeze server.

SessionManager.cfc

This file manages the creation of sessions and destroying of sessions. You can reuse this component for any other web service integration you might do in the future.

RecordingList.cfm

This file is the beginning page and calls the two CFCs previously mentioned to retrieve and then to display the list of recordings.

ReadMe.txt

This file contains the file descriptions listed previously and any additional configuration notes relative to getting the sample code to run.

Before I walk you through the code, I would like to examine the individual web services that you will use. Breeze web services operate with commands called actions. You will use the login, logout, user-accounts, sco-info, and sco-contents actions. You will need to use the user-accounts action after logging in to verify that the session ID is valid and also to acquire the principal ID associated with the session. You will use the sco-info and sco-contents actions to retrieve all of the records associated with the meeting and the specific details about each recording in the list. You can use these three actions to retrieve a great deal of other information, but an explanation of this is outside the scope of this tutorial.

## RecordingsList.cfm Explained

To begin I will first review recordingList.cfm. You use this file to display data to the user and to call the CFCs, which will communicate with the Breeze server to get the recording data. First you must set some variables, which are needed for the CFCs to work properly (accesskey, webServicesUrl, meetingID, username, password). The username and password credentials should have proper rights to the meeting in which you are retrieving recordings for. You can leave the accesskey blank for Breeze 5 users; if you are still using Breeze 4.1

You must supply the correct accesskey to execute web services on the Breeze server. Contact your Breeze

This article originally appeared on [www.macromedia.com/devnet](http://www.macromedia.com/devnet). Reprinted with permission.

administrator to locate the accesskey. The webServicesUrl should be the URL to the Breeze server ending with a "/" (for example, http://breeze.uwstout.edu/).

After setting these variables in recordingsList.cfm, you must handle the creation of the CFCs and retrieve the recordings for the meeting specified. When you create the CFCs, the first thing to do is call the init method inside of the CFC. This method is designed to be a constructor method, which is used in object-orientated languages, such as Java. Since Coldfusion components do not have a constructor that is called implicitly, I usually call the init method explicitly to achieve the same effect. Next, call the createSession() method of the SessionManager component, which takes the username and password and returns a valid session ID. Pass the session ID into the MeetingManager.cfc getRecordings() method. Then you end the session promptly by calling the terminateSession() method on the SessionManager. The process of contacting the Breeze server and manipulating the XML returned is encapsulated inside these aforementioned methods. This is done by design to remove unnecessary complexity from the display page and to encourage reuse of the components. The rest of the recordingsList.cfm is provided to display the meeting list.

## Handling Session Management in SessionManager.cfc

Now you can turn your attention to SessionManager.cfc. The SessionManager component has four methods: init(), createSession(), getPrincipleID(), and terminateSession(). The init() method simply sets a few variables and does not require much attention. The CreateSession() method is charged with calling the login web service action and pulling the session ID out of the results. I use the cfhttp tag to handle the request to the Breeze server as shown below:

```
<cfset params = "accesskey=#variables.accessKey#&action=login&login=#username#&password=#password#" />
<cfhttp url="#webServiceUrl#api/xml?#params#" method="get" />
```

Note: As I mentioned earlier, accesskey is only needed for Breeze 4.1. If

you are using Breeze 5, you don't need to use accesskey.

Look specifically at the cfhttp.responseHeader variable to find the session ID in the header. See the Breeze Web Services API for more details on this process.

The getPrincipleID() method is used to obtain the principleID associated with a username and password. You can do this by calling the user-accounts web service action and then use the xmlSearch() function to verify that the request was returned. If so, extract the principleID from the return XML string. When extracting data from an XML object, it is sometimes helpful to use the cfdump tag on the XML object to take a look at the structure. Otherwise you may run into problems trying to obtain the exact path of the node data you are looking for. In this case, you can locate the principleID using this syntax:

```
<cfset principalID = xml.results.users.user.XmlAttributes["user-id"]>
```

Once you have obtained the principleID from the return XML you can return it to the calling script with a cfreturn tag.

The terminateSession() method simply calls the logout web service action with the session ID to end the session on the Breeze server. You will notice by looking at this method that the same set of ColdFusion tags are used to call and interpret an action from the Breeze server. I will parameterize this common task and construct it as a reusable method in the next CFC to increase cohesiveness.

## Manipulating XML Data in MeetingManager.cfc

Look at the meetingManager.cfc to see how I acquire the list of recordings and convert them into a query object. The MeetingManager.cfc has two public methods and four private ones. The public methods are init() and getRecordings(). The private methods are scoArray2Query(), sco2Record(), ISO8601ToDate(), and callBreeze(). The getRecordings() method is the only method the outside world cares about. The private methods are used internally in this component and exist to achieve highly cohesive code fragments. If you where to extend the functionality of the MeetingManager component or use it in a different context, you may want to make these methods public. However, for the purposes of this tutorial they should remain private. The getRecordings() takes two parameters, meetingID and a session ID. From these two pieces of information you will be able to call a series of web services actions to find all the data you need on the recordings for the meeting in question. You can start out by calling the sco-contents web service to obtain all the sco objects for this meeting. In the MeetingManager.cfc, I have broken out the task of calling the Breeze server into one common method that takes in parameters and passes back an XML string. This common functionality is contained in the callBreeze() private method. You will use this method to obtain a list of sco's, at which point you then use the xmlSearch() ColdFusion function to filter out all sco's that are not of type archive.

*Joseph D. Baarsch is currently employed as a Systems Developer at the University of Wisconsin - Stout and also does software engineering consulting through Spectrum Interactive LLC. In addition to his full time work, Joseph is deepening his understanding of engineering and design principles at the University of Minnesota in their Masters of Science in Software Engineering program.*

“Another useful feature of Macromedia Breeze 5 is the ability to integrate Breeze in existing systems through the use of the Breeze Web Services API”



```
<cfset recordings =
xmlSearch(xmlOutput, "//sco[@
icon='archive']")>
```

This type is what Breeze 5 associates with all recording sco's. The ColdFusion function xmlSearch() returns a data structure of type array, which is then pass to the scoArray2Query() private method for conversion to data type query.

You will be converting the XML to a query because when an XML node is returned from Breeze 5, the data is fragmented inside the heavily nested structure of XML, which complicates data manipulation for display. In addition to this, you are going to need to get more details about each recording before producing the list to the user.

Inside the scoArray2Query() method, I take the array of recording nodes and loop through each node and call the sco-info web service action for each recording in the array. You need to do this step to get more detailed information about the recording that is not provided in the initial node returned by the sco-contents

web service action. Once this call is made, you will pass the XML to the sco2Record() method to convert the XML into a query record. This is done by using the query-AddRow() and querySetCell() ColdFusion functions. One way that I have found to extract data from an XML node is to use the cfswitch and cfcase tags inside, of a cfloop structure.

```
<cfloop from="1" to="#arrayLen(node.
xmlChildren)#" index="key">
<cfswitch expression="#node.
xmlChildren[key].xmlName#">
<cfcase value="date-created" >
... do something ...
</cfcase>
<cfcase value="name">
... do something ...
</cfcase>
...
</cfswitch>
</cfloop>
```

In this method I also call the ISO8601ToDate() method to convert Breeze dates to ColdFusion data types. This method could be a global function

for your application but for this tutorial I have made it part of the MeetingManager component. After the sco2Record() method has finished executing, control is returned to the calling method, scoArray2Query(). This method continues to loop through all of the elements in the array and then returns the resulting query data structure through its cfreturn tag. This returns control to the getRecordings() method, which returns the query to recordingList.cfm.

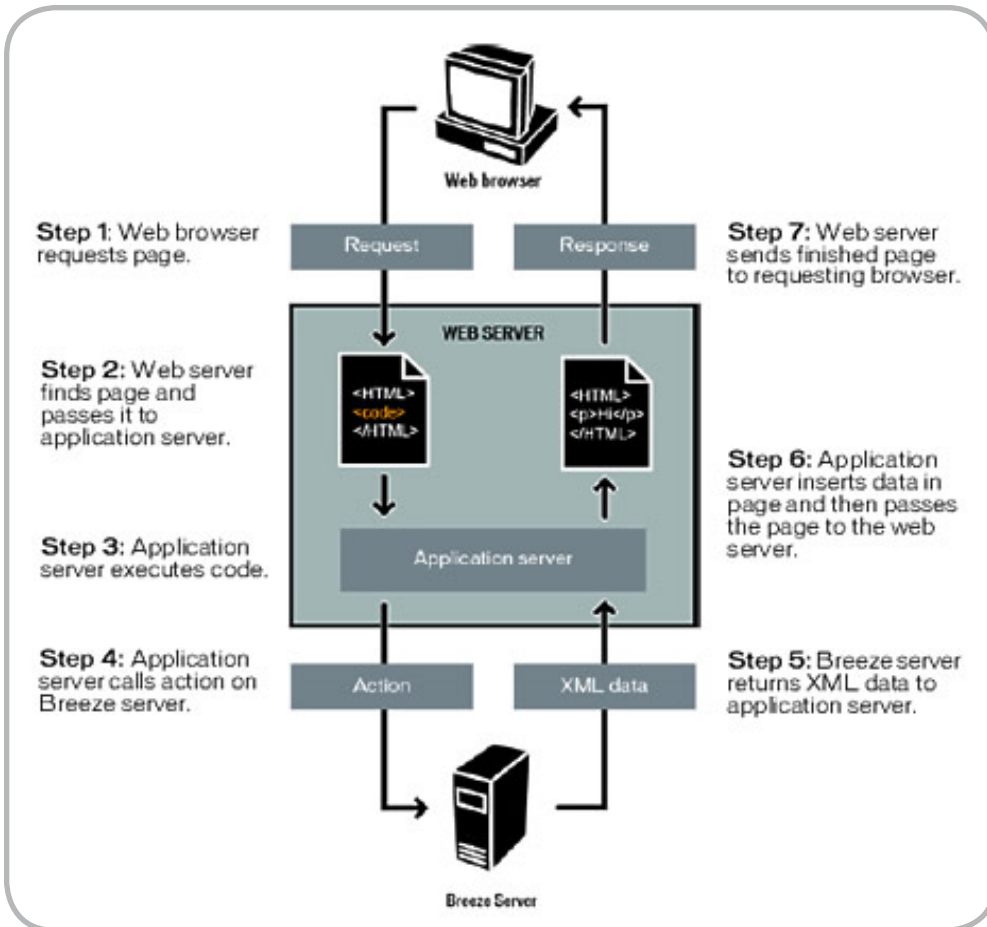
At this point you can run this code on your server by calling the recordingList.cfm in the browser. To verify that it is working, specify a meetingID to a meeting that has two or three recordings. Once run, your app will display a cfdump structure for the query object.

### Where to Go from Here

Now that you understand the pieces of this application, I encourage you to extend the application a bit. One way to extend the application would be to format the resulting query object in HTML and CSS. If you go this route, try to make the recording title a link to the actual recording. To facilitate single sign-on, remove the call to terminateSession() in recordingList.cfm and pass the session ID in the recording URL. Another extension could be to create a new method in MeetingManager.cfc called getPresentations() to return a query of presentations associated with a particular meeting. The getPresentations() method would be almost identical except you would need to filter on type presentation rather than archive.

I would also encourage you to try to integrate this code into your existing application. You may do this by populating the variables at the top of recordingList.cfm by using session, URL, or form-scoped variables. Perhaps you might set the Breeze URL in application.cfm along with the accessKey for Breeze 4.1 users.

Congratulations! You are well on your way to unlocking one of the best kept secrets in the Breeze Web Services API. Today you worked with some of the most important web services actions such as login, logout, sco-info, and user-info. From this point you should be able to experiment with the entire set of web services to best suit your application and integration needs. Happy coding! ☺



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  
**Marriott**  
Marriott Marquis Times Square  
New York, NY

April 24, 2006  
  
DOUBLETREE HOTEL  
San Jose, CA



For more information  
Call 201-802-3022 or  
email [events@sys-con.com](mailto: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.

**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](mailto:jeremy@sys-con.com)



**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

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

MEDIA SPONSOR



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

PRODUCED BY  
**sys-con**  
EVENTS



# Adding Right-Click Functionality to Captivate Content

*Instead of trying to avoid right-click events, I have come up with a solution that simulates right-click functionality*  
by **steven shongrunden**

If you have used Macromedia Captivate to create interactive software training simulations, you have probably come across a situation where you needed to simulate a right-click event. Unfortunately, the current version of Captivate cannot directly support this functionality because Macromedia Flash Player uses the right-click event to trigger the player menu, which enables you to change player settings or zoom in and out of the SWF file that is playing. This restriction can be frustrating if your demonstration or simulation needs to show that your software has functions only available by performing a right-click action.

Instead of trying to avoid right-click events, I have come up with a solution that simulates right-click functionality. This article explains my solution, which incorporates elements of HTML, CSS, and JavaScript.

## Configuring the HTML and CSS for a Shield

There is no way to stop the Flash Player context menu from appearing when it detects a right-click, but there are ways to stop the player from realizing that a right-click event has occurred. One way of doing this is to make the user think they clicked a Flash object when instead they clicked an invisible object placed in front of it. This way the invisible object acts as a shield that protects the Flash object from being clicked.

Before creating the “shield” layer, you must properly configure the Captivate object. Adjust the standard HTML output that Captivate generates with the following simple modifications:

- Set the object’s id attribute to “cp”.
- Add the wmode parameter and set it to “transparent”.
- Add the embed tag with the swliveconnect attribute set to “true”.

After making these changes, your HTML code for the Captivate object should look like the following (with your own width, height, and filename values):

```
<object classid="clsid:D27CDB6E-
AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.
com/pub/shockwave/cabs/flash/swflash.
cab#version=6,0,29,0" width="303"
height="320" id="cp">
  <param name="movie" value="right-
click-functionality.swf">
  <param name="quality"
value="high">
  <param name="menu" value="false">
  <param name="loop" value="0">
  <param name="wmode"
value="transparent">
  <param name="AllowScriptAccess"
value="always">
  <embed src="right-click-function-
ality.swf" AllowScriptAccess="always
" width="303" height="320" loop="0"
quality="high" pluginspage="http://
www.macromedia.com/go/getflashplayer"
type="application/x-shockwave-flash"
menu="false" swliveconnect="true"></
embed>
</object>
```

Now that you have set up the movie object correctly, add the shield layer on top of it. Start by inserting the following HTML just above the object tag:

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

The shield layer contains a transparent image that stops any mouse clicks from reaching the Flash application. Setting the galleryimg attribute to “no” prevents Internet Explorer from displaying the Image toolbar that appears when a user hovers over an image with the mouse.

The CSS for the shield image, shield layer, and Captivate content is quite simple:

```
div#shield {
  z-index: 2;
  position: absolute;
  display: none;
}

img.shield {
  width: 100%;
  height: 100%;
}

#cp {
  z-index: 1;
  position: relative;
}
```

You can add the CSS directly to the styles defined in the head of the HTML file, or place them in an externally linked style sheet.

This code sets the transparent GIF file to occupy 100% of the space of the shield layer. The shield layer is set so that it will not appear by default, and its position is set to “absolute”. The Z-index property specifies for the browser how to stack



components on the screen. Because the shield must be on top of the Captivate SWF file, you must set the shield's Z-index higher.

## Setting Up the Captivate Source File

I used JavaScript to create most of the functionality in this solution. The JavaScript code positions the shield, turns it on and off, and decides if the user right-clicked in the appropriate area.

Before you can use the JavaScript code, you need some information from the Captivate content. You need to specify when the JavaScript should show the shield and where the user must click to advance to the next slide.

I created a special SWF file to handle this responsibility (see the right-click-box.swf file in the sample file at the beginning of this article). Think of this SWF file as Captivate's built-in Click Box. When the animation appears in the Timeline, it specifies the JavaScript to enable right-click functionality. If a slide in your Captivate content requires a right-click to advance, use the following steps as you author your Captivate content:

- Start Demo: Adding Right Click Functionality to Captivate Content
- Add a Click Box over the target area.
- Click the Insert menu, select Animation, and select the right-click-box.swf file.
- Place the animation over the Click Box.
- Ensure that the animation does not touch the start or end of the slide in the Timeline.

The animation must not touch the start or end of the slide's Timeline because that could cause problems with enabling right-click functionality on slides before or after the current slide.

After following these steps and publishing your Captivate content as a SWF file, you are ready finally to implement the JavaScript that brings everything together.

*Note: Moving data from Captivate to JavaScript in this fashion is outside the scope of this article. Interested Flash developers can examine the right-click-box.fla sample file to see how I do this.*

## Implementing the JavaScript

The JavaScript can be broken down into three sections. One section acquires the necessary data from the Captivate demo or simulation, another section sets up the shield, and the final section decides what to do when the user right-clicks the shield.

The first section is called when the right-click-box.swf animation starts to play in the Timeline:

```
// Array to hold data sent by the
right-click box we inserted in
Captivate
var rightClickBoxData = new
Array(4);

// Receive the function call from the
movie clip
var InternetExplorer = navigator.
appName.indexOf("Microsoft") != -1;
function cp_DoFSCommand(command,
args) {
var myFlashObj =
InternetExplorer ? cp : document.cp;

// Set the dimensions of the
right click box
// rightClickBoxData[0] = x posi-
tion (relative to Captivate movie)
// rightClickBoxData[1] = y posi-
tion (relative to Captivate movie)
// rightClickBoxData[2] = width
// rightClickBoxData[3] = height
rightClickBoxData = args.
split(" ");

// Enable right click function-
ality
rightClickToAd-
vance();
}
if (navigator.appName
&& navigator.appName.
indexOf("Microsoft") != -
1 && navigator.userAgent.
indexOf("Windows") != -1
&& navigator.userAgent.
indexOf("Windows 3.1") ==
-1) {
document.write('<SCRIPT
LANGUAGE=VBScript> \n');
document.write('on error
resume next \n');
document.write('Sub
cp_FSCommand(ByVal command,
```

```
ByVal args)\n');
document.write(' call cp_
DoFSCommand(command, args)\n');
document.write('end sub\n');
document.write('</SCRIPT\' \n');
}
```

When the second section runs, it calls the rightClickToAdvance() function, which enables and positions the shield:

```
function rightClickToAdvance(){

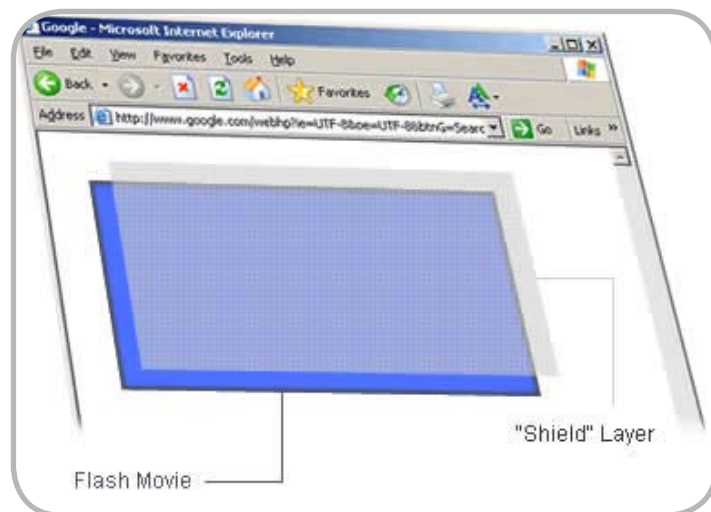
// Define the shield and
Captivate objects
var shield = document.getElementByI
d("shield");
var cp = document.
getElementById("cp");

// Enable the shield
shield.style.display = "inline";

// Place the shield over top of
only the Captivate SWF file
shield.style.top = cp.offsetTop;
shield.style.left = cp.offsetLeft;
shield.style.width = cp.width;
shield.style.height = cp.height;

// Perform the userRightClicked
function when the user press the right
mouse button
document.oncontextmenu = userRight-
Clicked;

// Stop the user from being able
to select the shield
document.onselectstart = new
Function("return false");
}
```





The last section of code indicates to the browser that right-clicks should call the `userRightClicked()` function.

The `userRightClicked()` function determines whether the user right-clicked a location inside the target area of the right-click box. If the user's right-click was not in the target area, nothing happens. If the user right-clicks the target area, the `userRightClicked()` function disables the shield, advances the Captivate simulation to the next slide, and returns functionality back to normal:

```
// Function to handle when the user
right clicks on the shield
function userRightClicked() {

    // Get the dimensions of the
user's click
    var clickX = event.clientX + docu-
```

```
ment.body.scrollLeft;
    var clickY = event.clientY + docu-
ment.body.scrollTop;
```

```
// Get the dimensions of the
right click box
    var dimX = eval(rightClickBoxData
[0]);
    var dimY = eval(rightClickBoxData
[1]);
    var width = eval(rightClickBoxData
[2]);
    var height = eval(rightClickBoxDat
a[3]);
```

```
// Get the position of the
Captivate content relative to the top
left corner of the page
    var cpX = document.
getElementById("cp").offsetLeft;
    var cpY = document.
getElementById("cp").offsetTop;
```

```
// Only advance the slide if the
user right-clicked in the correct area
    if ((clickX > cpX + dimX) &&
(clickX < cpX + dimX + width)){
        if ((clickY > cpY + dimY) &&
(clickY < cpY + dimY + height)){

            // Advance the slide
            window.cp.SetVariable("rdcmdNex
tSlide","1");
```

```
// Get rid of the shield
    document.
getElementById("shield").style.display
= "none";

    // Return to normal function-
ality
    document.oncontextmenu = null;
    document.onselectstart = null;
}

return false;
}
```

After combining all of these elements, your Captivate content now has fully functional, right-click simulation abilities. If you don't want to code this tutorial from scratch, edit the `right-click-functionality-example.htm` page and modify the object tag's attributes for filename, width, and height to reflect your Captivate SWF file.

*Note: Before running the follow- ing Captivate simulation on your local machine, you must make a change in Flash Player 8 settings. New security features in Flash Player 8 stop Captivate from communicating with JavaScript when you view the file on your local hard drive. Follow the instructions in the Breeze Presenter TechNote, Security Warning with Flash Player 8, to change these set- tings in Flash Player 8 so that you can run these files properly on your local hard drive. This issue does not apply to Captivate content hosted on a web server. The right-click functionality only runs on Microsoft Internet Explorer.*

Start simulation of right-click func- tionality

In the simulation, the prompt asks you to click "Close". Right-click My Computer to see the simulated right-click functionality explained in this tutorial.

## Where to Go from Here

This article demonstrated a method of modifying your Captivate content to simulate right-click events. The solution may not be perfect but it's a step in the right direction. I hope it will help take your Captivate projects further and allow you to develop even more effective soft- ware simulations.

This code was developed and tested exclusively with Microsoft Internet Explorer 6 on Windows XP; compatibility with other versions or browsers is highly unlikely—or at least unknown at this time. Creating a cross-browser alternative would be very useful to allow non-IE6 users to enjoy this right-click func- tionality.

If you have any suggestions on how to improve any aspect of this method, including cross-browser support, feel free to contact me at [steve@suncitydesign.com](mailto:steve@suncitydesign.com). I would be happy to discuss any further enhancements or problems you might have to make this method as use- ful as possible. ☺

*Steven Shongrunden has been developing leading-edge e-learning solutions for the past three years. He is the owner of Sun City Design, an Internet marketing and consulting firm in Canada. He manages all aspects of the company's web develop- ment and technical training solutions.*

**Incorrect**  
Right-click-box.swf is touching the start of the slide

**Incorrect**  
Right-click-box.swf is touching the end of the slide

**Correct**  
Right-click-box.swf is not touching either end of the slide

# 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](http://events.sys-con.com)**

Sponsored by

WebServices  
JOURNAL

XML  
JOURNAL

NET  
JOURNAL

eclipse  
developer's journal

WebSphere  
JOURNAL

Information  
STORAGE+SECURITY  
JOURNAL

wldj  
THE LEADING  
INTEGRATED  
WEB SERVICES  
JOURNAL

JDJ

Linux  
WORLD

MX  
developers journal

asp.net  
PRO

SD  
Times

CoDe

Software Test  
& Performance

\*Call for Papers email: [events@sys-con.com](mailto: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



# Integrating Remote Shared Objects with Flex and Flash Communication Server

*In this project, a remote shared object on a Flash Communication Server contains data that changes frequently*  
by michael labriola



Communication Server (FCS) together provide an amazing toolbox that will undoubtedly provide inspiration to thousands of developers and projects. Unfortunately, without concrete examples and guidelines for good practices on their integration, we spent many hours on tiny issues that, with additional information, could have been easily circumnavigated.

Those hours of trail-blazing are the inspiration for this article. This article tackles the basic integration of how to use a remote shared object on FCS with Flex. Our goal was to make this integration as simple and straightforward as using a web service from within a Flex MXML file.

Flex and FCS provide the foundation for Internet applications that deliver continually changing data to the desktop. This technology, called data push, allows a developer to update information on a user's screen without a browser refresh. In our company, we use this technology to build interactive auction sites that, within moments of a new bid, deliver updated prices on thousands of items to users around the world and update business systems that track critical orders, from estimate through final delivery. This coupling of technology can help you build any application where presentation, collaboration, and up-to-the-moment information is paramount.

Before continuing, I must emphasize that this is a basic implementation and a basic example. If you want to dig into more complex implementations, read, "Developing Rich Clients with Macromedia Flex" and "Programming Flash Communication Server." These

books have become invaluable to our organization, and I can't say enough positive things about both of them.

## The Project and Approach

In this project, a remote shared object on a Flash Communication Server contains data that changes frequently. It is imperative that the client immediately receive this data without any need to refresh.

In our approach, we connect to FCS server, attach to the remote shared object, and display the data in a Flex DataGrid control. Then, using FCS and Flex, we push changes to the client with Flex data-binding capabilities, so that the DataGrid displays updated data as changes occur.

## The Project Scope

You can accomplish 95% of this project solely through MXML constructs, without ever writing a line of ActionScript; however, there are a few issues to understand.

- Flex has components available for WebService and RemoteObject connections that you can easily access through MXML. Our team wanted to offer this same feature for FCS connections and gain some event listener and broadcasting capabilities.
- The SharedObject class is static, and therefore it is not used from within MXML in the same way as familiar Flex components. Our team thought that a wrapper for this class that interacted with FCS and provided the user with viable events would be a useful.

If the above points are a bit foreign right now, that is okay. They will become

clearer in the course of this and other articles; also, using the following components, you can still work with FCS.

Please take a moment and review FCSService and SharedRemote class documentation, available from the fcs\_flex\_sample.zip that you downloaded at the beginning of this article.

- FCSService documentation (fcsservice\_class\_readme.html)
- SharedRemote documentation (sharedremote\_class\_readme.html)
- Examples of using the two classes together (two\_classes\_example.html)
- FCSService.as
- SharedRemote.as

Now read on to assemble the application.

## The Layout

The following example creates a DataGrid control that displays in 90% of the available horizontal and vertical space.

You can create the following MXML file and name it flexFCS\_01.mxml, or you can open the solution file (flexFCS\_01.mxml) available in the ZIP file that you downloaded in the Requirements section of this article.

### Example 1: flexFCS\_01.mxml

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.
macromedia.com/2003/mxml"
                xmlns="*"
                creationComple
e="initializeApplication(event)">

    <mx:Script>
    <![CDATA[
```

```

    var myData:Array;
    function initializeAp-
plication( event )
    {

        //We are called by the
creation complete event of the appli-
cation tag
        //Instantiate a new Array
        myData = new Array();
    }
}]>
</mx:Script>
<mx:DataGrid id="display_grid"
dataProvider="{myData}" width="90%"
height="90%" />
<!--Created a DataGrid and bound
the dataProvider to myData-->
</mx:Application>

```

The example specifies the dataProvider property for the DataGrid control as a variable named myData. In the creationComplete event of the <mx:Application> tag, the example calls initializeApplication, which instantiates myData as an instance of Array.

When you run this example, you a DataGrid control and other information displays. Next, you will add the functionality.

## Flash Communication Server

Flex and FCS create a client/server environment. Using the MXML code in the flexFCS\_01.mxml file, Flex creates a client application that executes on a user's machine. FCS acts as the server in this relationship, providing data and updates to the client.

This article is about interfacing two components, not programming FCS. For more on FCS check out the books mentioned in the beginning of the article. For now, I provide a basic server-side script to demonstrate the interface.

Place file, main.asc (included with the ZIP file in the Requirements section) on FCS server in a directory called random in the \Flash Communication Server MX\applications\ directory.

This file sets up a simple shared object on FCS server. The object has 10 slots, named 0-9. Each of these slots will have a random number. This random number changes every two seconds and FCS pushes the results to the client.

## The Connection

At this point you have a simple client application, created in Flex, and a FCS application. Now you will make the client display the changing data from FCS. You accomplish this in two phases. First, you connect to FCS and the Remote Shared Object. Second, you receive information back from FCS and process it for display purposes.

First, add the following highlighted lines to Example 1 (flexFCS\_01.mxml) or open the solution file, flexFCS\_02.mxml, which is available from the fcs\_flex\_sample.zip that you downloaded at the beginning of this article.

### Example 2: flexFCS\_02.mxml (additions are highlighted)

```

<?xml version="1.0" encoding="utf-8"?>

<mx:Application xmlns:mx="http://www.
macromedia.com/2003/mxml"

    xmlns=""

    creationComplete="initializeApplicati
on( event )" >

<mx:Script>

<![CDATA[
    var myData:Array;
    function initializeApplication( event
)
    {
        //We are called by the creation com-
plete event of the application tag

        //Instantiate a new Array
        myData = new Array();

        //Connect to the flash communication
server.
        //we are using rtmp as our first
protocol attempt

        my_fcs.connect('rtmp://[your_server]/
random' );
    }
}]>
</mx:Script>

<mx:DataGrid id="display_grid"
dataProvider="{myData}" width="90%"

```

```
height="90%" />
```

```

<!--Created a DataGrid and bound the
dataProvider to myData-->
<!--Instantiate an FCSService compo-
nent-->

```

```

<FCSService id="my_fcs"
    closed=" alert( 'server closed con-
nection' )"

```

```

    rejected=" alert('Server rejected
connection' )"

```

```

    failed=" alert('Server connection
failed' )" />

```

```

<!--Instantiate an SharedRemote com-
ponent and bind it to the FCSService
tag above-->

```

```

<!--The name property of this tag
must match the argument in the
SharedObject.get command on the server
file main.asc-->

```

```

<SharedRemote id="my_remote1"

```

```

    name="RandomNumbers"

```

```

    service="{my_fcs}"

```

```

    failed="alert('Shared Object
Failed' )"

```

```

    status="alert('Status ' + event.info.
level + ' ' + event.info.code );"/>

```

```

</mx:Application>

```

In this version, I added three important blocks of code:

- An FCSService component
- A SharedRemote component
- A call to the FCSService object's connect method

First, examine the FCSService component. The FCSService has an id of my\_fcs, which you use to reference the instance going forward. You also defined event handlers for the closed, rejected, and failed events. If an end user encounters any type of problem, an alert box will display.

Second, the SharedRemote component has an id of my\_remote1. The service property is set to {my\_fcs}, the id of the FCSService component. This allows any FCSService component to have multiple shared remote objects defined. The SharedRemote component also defines event handlers for the failed and status



messages. If an end user encounters a problem with the shared object, an alert box displays on the screen.

Finally, the initializeApplication handler calls my\_fcs.connect('rtmp://yourServerName/random'). The name, random, is the name of the application created when you placed the main.asc file into the random folder on your FCS.

When you invoke the connect method, you attempt to make a connection initially through RTMP to FCS. In reality, you actually have several protocols/port combinations which will be tried before a connection occurs. While this concept is outside the scope of this article, please read "Tunneling Macromedia Flash Communications Through Firewalls and Proxy Servers" for more information.

If you run the application at this point, you will see a DataGrid, much like the previous example.

If an alert box appears, ensure that your Flash Communication Server is running and that you placed the main.asc file in the proper directory as explained in the "Flash Communication Server" section.

### The Good Stuff

After the last round of revisions, your application can receive updates from the server, but can't do anything with them yet. Now you'll add that functionality.

Add the highlighted code below to Example 2 (flexFCS\_02.mxml) or, open the solution file, flexFCS\_03.mxml, which contains all of the changes. The flexFCS\_03.mxml file is included in the fcs\_flex\_

sample.zip that you downloaded in the Requirements section of this article.

#### Example 3: flexFCS\_03.mxml (additions shown in bold):

```
<?xml version="1.0" encoding="utf-8"?>

<mx:Application xmlns:mx="http://www.
macromedia.com/2003/mxml"

    xmlns="*"
    creationComplete="initializeApplicati
on( event )">

    <mx:Script>

    <![CDATA[
        var myData:Array;
        function initializeApplication( event
        )
        {

            //We are called by the creation com-
            plete event of the application tag
            //Instantiate a new Array
            myData = new Array();
            //Connect to the flash communication
            server.
            //we are using rtmp as our first pro-
            tocol attempt
            my_fcs.connect('rtmp://[your_server]/
            random');
        }

        function syncData( event )
        {
            var currentIndex:Number;
            var currentNode:Object;
            for ( var i=0; i<event.actions.
            length; i++ )
```

```
{

    /* This is cheap and cannot be relied
    upon, however, for this example: our
    slot 'names' are all numeric and can
    be used as Indexes into the array.
    These names are set when we do the
    random_so.setProperty("0", ... ); in
    the main.asc file.The "0" is the slot
    name
    So, in this case the slots are named
    0,1,2,3 and so on. */
    currentNode = event.actions[i];
    /* Whenever we receive a synchroniza-
    tion event from FCS, it is provided
    as an array.Each element in the array
    has a code, which defines the type of
    event we are receiving.
    In this case, we are only looking
    at two of the possibilities, a slot
    changed or the whole array should be
    cleared */
    switch (currentNode.code)
    {
        case "change" :
            //This slot has changed and needs to
            be
            //updated

            if ( myData[ currentNode.name ] ==
            undefined )

            {
                //Even though FCS sees this data as
                //a change, this is the first time we
                have
                //data in the slot
                //So we add it to our data structure
                myData.addItemAt( currentNode.name,
```

label	value
Random Number 0	51
Random Number 1	10
Random Number 2	90
Random Number 3	35
Random Number 4	70
Random Number 5	33
Random Number 6	87
Random Number 7	4
Random Number 8	14
Random Number 9	15

# Welcome to the Future

**REGISTER NOW!**  
[www.iTVcon.com](http://www.iTVcon.com)

**CALL FOR PAPERS NOW OPEN!**

**LIVE SIMULCAST!**  
AROUND THE WORLD ON [SYS-CON.TV](http://SYS-CON.TV)

# of Video on the Web!

**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](mailto: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](http://iTVcon.com)  
[jeremy@sys-con.com](mailto: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



```

    { label:event.data[ currentNode.name
].label,

    value:event.data[ currentNode.name
].value } );
//Notice above that we are using the
{ } as a shortcut to create a new
object from the data we received from
FCS
//This is do to the databinding prop-
erties of flex.
//If we do not create a copy of the
object, flex will try to update FCS
server everytime we make a change to
the object
//This can be desireable, but, it is
not for this example

}
else
{
//We have seen data for this slot
before, so we update our data struc-
ture
myData.replaceItemAt( currentNode.
name,
{ label:event.data[ currentNode.name
].label,
value:event.data[ currentNode.name
].value } );
}
break;
case "clear" :
//FCS has instructed us to clear all
of our existing data

myData.removeAll();

break;
}
}]]>
</mx:Script>
<mx:DataGrid id="display_grid"
dataProvider="{myData}" width="90%"
height="90%"/>
<!--Created a DataGrid and bound the
dataProvider to myData-->
<!--Instantiate an FCSService compo-
nent-->

<FCSService id="my_fcs"
closed=" alert( 'server closed con-
nection' )"

rejected=" alert('Server rejected
connection')"

failed=" alert('Server connection
failed')" />
<!--Instantiate an SharedRemote com-
ponent and bind it to the FCSService
tag above-->
<!--The name property of this tag
must match the argument in the
SharedObject.get command on the server
file main.asc-->
<SharedRemote id="my_remote1"
name="RandomNumbers"
service="{my_fcs}"
failed="alert('Shared Object
Failed')"

status="alert('Status ' + event.info.
level + ' ' + event.info.code );"

```

```

sync="syncData( event )"/>
</mx:Application>

```

In this version, you added two important items. You specify an event handler for the sync event of the SharedRemote component and added the code for that handler.

First, let's talk about the syncData function. This is the function that performs all of the work in preparing the data for display. The event object passed to this function always contains two very important properties: actions and data.

The actions property is an array that will contain a variable number of elements based on the changes that occurred on the Flash Communication Server. FCS adds one object to this array for every action it implements on the client. Examples of actions include: clear all of the data, delete a single element, change the value of an existing element, and so forth.

When you look at the individual elements of the actions array, you will always find a single property named code, and for some codes an additional property called name. The data structure looks like the following:

```

event
  actions
    0 0 -
    code
    [name]
    0 1 -
    code
    [name]
    0 2 -
    code
    [name]

  data

```

At the start of the syncData function, the code loops through the actions array and examines each element. You perform a switch statement on the code property, which defines the type of action you can perform. For this example, you are only interested in two specific codes: change and clear.

A clear code simply means that all of the current data should be cleared. The

# “Flex and FCS provide the foundation for Internet applications that deliver continually changing data to the desktop”





# Visit the *New* **www.SYS-CON.com** Website Today!

The World's Leading *i*-Technology  
News and Information Source

# 24/7

## FREE NEWSLETTERS

Stay ahead of the *i*-Technology curve with E-mail updates on what's happening in your industry

## SYS-CON.TV

Watch video of breaking news, interviews with industry leaders, and how-to tutorials

## BLOG-N-PLAY!

Read web logs from the movers and shakers or create your own blog to be read by millions

## WEBCAST

Streaming video on today's *i*-Technology news, events, and webinars

## EDUCATION

The world's leading online *i*-Technology university

## RESEARCH

*i*-Technology data "and" analysis for business decision-makers

## MAGAZINES

View the current issue and past archives of your favorite *i*-Technology journal

## INTERNATIONAL SITES

Get all the news and information happening in other countries worldwide

## JUMP TO THE LEADING *i*-TECHNOLOGY WEBSITES:

*IT Solutions Guide*

*MX Developer's Journal*

*Information Storage+Security Journal*

*ColdFusion Developer's Journal*

*JDJ*

*XML Journal*

*Web Services Journal*

*Wireless Business & Technology*

*.NET Developer's Journal*

*Symbian Developer's Journal*

*LinuxWorld Magazine*

*WebSphere Journal*

*Linux Business News*

*WLDJ*

*Eclipse Developer's Journal*

*PowerBuilder Developer's Journal*



The World's Leading *i*-Technology Publisher



code executes a removeAll method on the data structure.

A change code indicates that a slot has changed on the server and provides the name of this slot through the name property. In this case, the slot names are 0-9.

The data array is the other extremely important part of this equation. The data is an associative array of the slots defined on FCS. So, in this example, data has slots named 0-9.

When you receive a change code, you also receive the name of the changed slot. This name allows you to look up the new slot information in the data array.

There are two more issues to understand, however. First, FCS informs the Flex client of changes to the data, however, it is up to the client program to analyze this data and determine whether it is new content to store or existing content to replace.

In this simplified example, you simply check to see if data of this slot name exists in the myData array. If it does, you replace that item with the new data. If it does not, you add it.

Second, and perhaps more complicated, is the interaction of the data from FCS with the Flex data binding mechanism.

### Data Structure Review

An object is a collection of properties and it is one of the basic Flex data types. However, objects behave differently than numbers or strings.

For example, examine the following piece of code.

```
myNum1 = 5;
myNum2 = myNum1;
myNum1 = 6;
```

At the end of this code, myNum1 contains the number 6, and myNum2 contains the number 5; however, this same example does not hold true for objects. If you assign one object to another, as in the following example:

```
myObj2 = new Object();
myObj2.test = 3;
myObj2.case = 4;
myObj1 = myObj2;
```

Flex doesn't actually make a copy of the object, but rather myObj1 contains

a reference to myObj2. So, there is one copy of all the data stored in the object and both names can be used to access this data. Therefore, myObj1.test and myObj2.test are really the same variable.

This can cause an issue when you receive a new object from FCS. For example, if you receive a change for Slot 1, the intuitive approach might be to assign the data coming from the FCS shared remote object directly to your myData array using an assignment (=) operator.

However, as noted above, if you simply set two objects equal to each other, Flex will be unable to make a copy, but merely provide a new reference or way for you to access the same data. This means that any time you attempt to make a change to an element in your myData array, in reality, you are attempting to change the shared remote object that FCS created.

The implication is that every user running your application will receive a change made by a single user. When used properly this can be beneficial, however, in many cases it can cause significant performance reduction and potentially prevent users with slower connections from accessing the application.

A quick way to resolve this issue is to create a brand new object every time a new piece of content arrives from Flash Communication Server and then use that newly created object in the myData array. Any changes made from within a single client's application will affect a user's

copy of the data but will not change values in the shared remote object used by everyone.

The final step required to display incoming data from FCS is to provide the name of the function syncData to the sync event of the component. Now, whenever data changes on the server, a sync event will be broadcast from the RemoteShared component and the syncData function will be called to process the received data.

### The Result

When you run this application, you have 10 rows in a DataGrid. Every two seconds, when the data changes on the Flash Communication Server, this data will be pushed to all connected clients, keeping them in sync.

While this is a rudimentary example, it shows how you can incorporate a Flash Communication Server remote shared object into your application in a way that is familiar to you, and keeps the metaphor established by the WebService tag. It demonstrates the very basic technique of dealing with returned data and illustrates the general concept of the server technology. ☁

*Michael Labriola is a founding partner and senior consultant at Digital Primates IT Consulting Group. Digital Primates analyzes client business processes and develops custom solutions that extend the latest technology.*

## Advertising Index

Advertiser	URL	Phone	Page
CFDynamics	www.cfdynamics.com	866-233-9626	20
Community MX	www.communitymx.com		27
EdgeWeb	http://edgewebhosting.com	866-334-3932	5
Flashforward	www.flashforwardconference.com		15
Hosting	www.hosting.com		31
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	35
Stream57	www.stream57.com	212-909-2550x1012	6
Vitalstream	www.vitalstream.com	800-254-7554	2
WebAppCabaret	www.webappcabaret.com/f/dj.jsp	866-256-7973	51

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.

Complex JAVA J2EE Hosting made easy.

**WebAppCabaret<sup>sm</sup>**

<http://www.webappcabaret.com/jdj.jsp>  
**1.866.256.7973**



## JAVA J2EE-Ready Managed Dedicated Hosting Plans:

### Xeon III

**SAMEDAY  
SETUP**

Dual 3.2 GHz Xeons  
4GB RAM  
Dual 73GB SCSI  
1U Server  
Firewall  
Linux  
Monitoring  
NGASI Manager

**\$399**  
monthly

### Pentium 4 I

**SAMEDAY  
SETUP**

2.4 GHz P4  
2GB RAM  
Dual 80GB ATA  
1U Server  
Firewall  
Linux  
Monitoring  
NGASI Manager

**\$199**  
monthly  
**2nd month  
FREE**

### 4Balance I

1 Database Server  
and 2 Application  
Servers connected  
to 1  
dedicated  
load  
balancing device.  
Dual Xeons.  
High-Availability.

**\$1724**  
monthly

**NEW!** Geronimo 1.0 Hosting . Virtual Private Servers(VPS) starting at \$69/month

At **WebAppCabaret** we specialize in **JAVA J2EE Hosting**, featuring **managed dedicated servers** preloaded with most open source JAVA technologies.

#### PRELOADED WITH:

JDK1.4 . JDK1.5 . Tomcat . Geronimo . JBoss . Struts . ANT . Spring . Hibernate  
Apache . MySQL . PostgreSQL . Portals . CRM . CMS . Blogs . Frameworks  
All easily manage via a web based control panel.

#### Details:

- All Servers installed with the latest Enterprise Linux
- Firewall Protection
- Up to 60 GB daily on site backup included at no extra charge per server.
- Database on site backup every 2 hours
- Daily off site database backup
- A spare server is always available in case one of the server goes down
- Intrusion detection.
- 24x7 Server and application monitoring with automatic self healing
- The Latest Bug fixes and Security updates.
- Tier 1 Data Center. 100% Network Uptime Guarantee
- Guaranteed Reliability backed by industry-leading Service Level Agreements

Log on now at <http://www.webappcabaret.com/jdj.jsp> or call today at **1.866.256.7973**

**WebAppCabaret<sup>sm</sup>**

## JAVA J2EE Hosting

Prices, plans, and terms subject to change without notice. Please log on to our website for the latest price and terms. Copyright © 1999-2006 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](http://www.macromedia.com/go/studio8_mxdj)

macromedia®  
**STUDIO 8**