FLASH SECRETS: REVEALED P. 28

4.3

MX developer's journal

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

volume 4 issue 3 2006 www.mxdj.com

Adobe Flex

Developing Flex RIAs with Cairngorm Microarchitecture

DREAMWEAVER

Configuring and Testing PHP Servers
for XSL Support

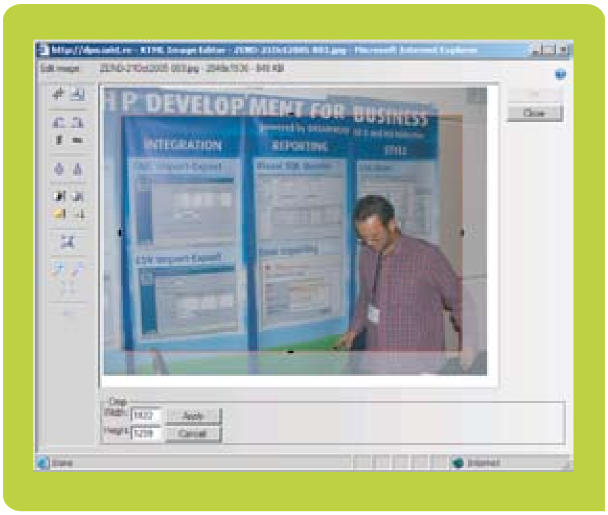
FLASH

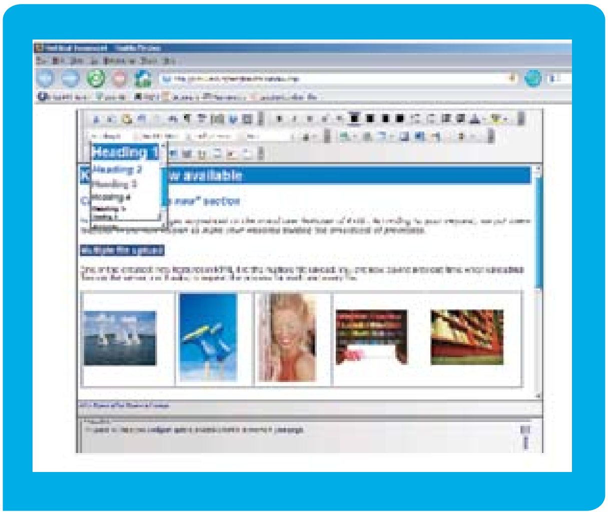
Best Practices for Encoding Video
with the VP6 Codec

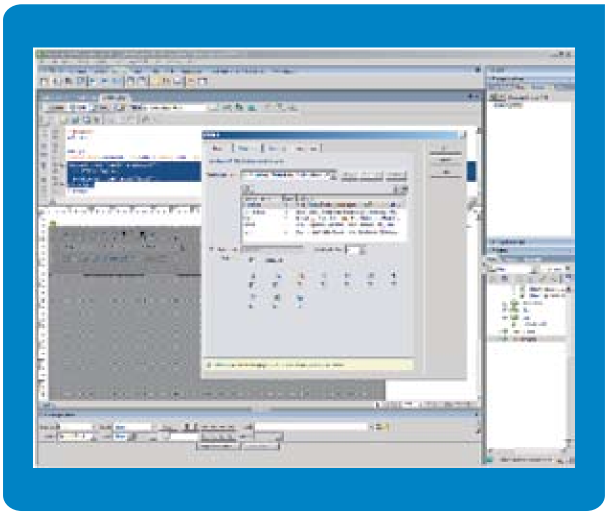
DIRECTOR

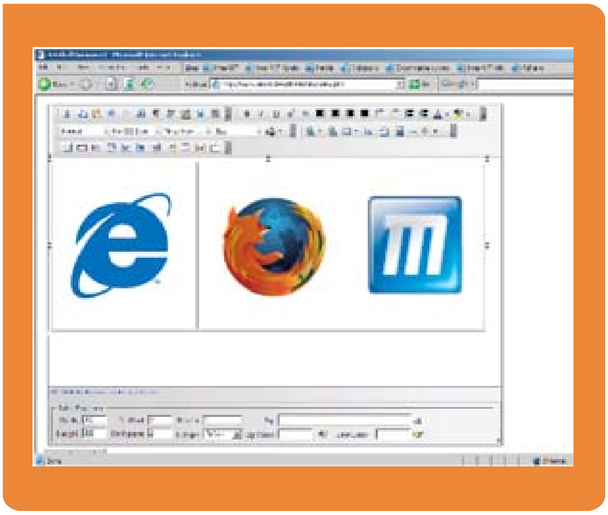
SCORM 1.3 SCORM SCO
Presentation Engine

Rev Up Your Flash Video









MX
developer's journal
march 2006

For answers call us at 1-866-EDGEWEB

3 3 4 3 9 3 2

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

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

By the Numbers:

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

EDGE
WEB HOSTING

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

http://edgewebhosting.net

CFDJ READERS' CHOICE AWARD
MXDJ READERS' CHOICE AWARD
2003 - 2006

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

Group Publisher Jeremy Geelan
Art Director Louis F. Cuffari

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

Technical Editors
Jesse Warden • Sarge Sargent

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

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

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

Worldwide Newsstand Distribution
Curtis Circulation Company, New Milford, NJ

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

Promotional Reprints
Megan Mussa
megan@sys-con.com

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.

Configuring and Testing PHP Servers for XSL Support

Transformations in a snap

marius zaharia & cristian ivascu

The major breakthrough in Macromedia Dreamweaver 8 is visual authoring with XML data. You can now perform both client-side and server-side XSL transformations in a snap. I covered XML syntax in one of my previous articles. I also presented the XSL syntax and covered the differences between a server-side and a client-side transformation in my article, XSL Overview. Finally, in the article, Consuming a Remote RSS Feed with Dreamweaver 8, I show you how to consume a remote feed in your site, using the XSL Transformation server behavior in Dreamweaver 8.

This article explains how to install and configure XML and XSL support for your web server, in order to be able to perform server-side XSL transformations. The article covers the configuration of PHP application servers—versions 4 and 5—for Windows, Macintosh, and Linux operating systems.

In a server-side XSL transformation, the server performs the actual work of producing HTML output using pre-installed libraries for XML and XSL support. In a client-side transformation, however, the client's browser does the work of fetching the XML and XSL files and producing the HTML output. The downside of this approach is that not all browsers have XML/XSL support, so some of your clients might not be able to see your pages. There are also other advantages to server-side XSL transformations:

- The file containing the XML data can reside on your own server or anywhere else on the web.
- Any browser can display the transformed data.

Quickly Testing Your PHP Server for Server-Side XSL Transformation Capability

Your PHP server may already be set up to do XSL transformations. You can quickly test it by following these steps:

- Download the sample files linked below.
- Unzip the ZIP file into your web root and upload the entire xslt_test folder to your web server running PHP.

Open a browser and browse to the test.php file on your server. If you receive an error, please continue reading this article. I will show you how to configure and test your PHP server for XSL transformations. If the test.php page returns a list of articles from the Macromedia Developer Center, I encourage you to read my article, Consuming a Remote RSS 1.0 Feed with Dreamweaver 8, in which I will show you how to build the test.php page.

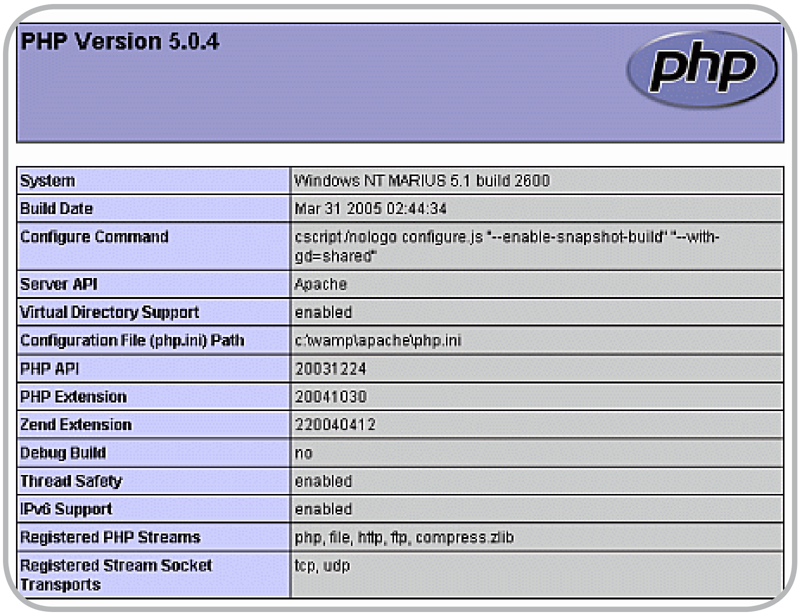
Determining Your PHP Server Version and Whether It Supports XSLT

How Do I Know Which PHP Version I Have?

To find out which libraries you need for performing server-side XSL transformations, you need to determine which PHP version is currently running on your web server. To do that, follow these steps:

- Open Dreamweaver.
- In a site defined with a testing server, create an empty PHP page (File > New).
- Switch to Code view.
- Inside the body tag, type the following line of code:
- <?php phpinfo() ?>
- In your browser, this function will display detailed information about the current PHP installation and settings, modules, libraries, and so on.
- Save your page to something like test.php and upload it to the testing server.
- Load the page in your browser, to see the results, using F12 (Windows) or Option + F12 (Macintosh).

This article originally appeared on www.macromedia.com/devnet. Reprinted with permission.



PHP Version 5.0.4

System	Windows NT MARIUS 5.1 build 2600
Build Date	Mar 31 2005 02:44:34
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--with-gd=shared"
Server API	Apache
Virtual Directory Support	enabled
Configuration File (php.ini) Path	c:\wamp\apache\php.ini
PHP API	20031224
PHP Extension	20041030
Zend Extension	220040412
Debug Build	no
Thread Safety	enabled
IPv6 Support	enabled
Registered PHP Streams	php, file, http, ftp, compress.zlib
Registered Stream Socket Transports	tcp, udp

Note: On some systems, the phpinfo() function is disabled for security reasons. In this case, you will receive a warning that this function doesn't exist or your browser will display a white page. You'll have to remove this function from the disable_functions list in the php.ini file or ask your system administrator to do it.

The PHP version is listed on top of the page. Depending on your current PHP version, you need the following XML/XSL libraries:

- For PHP 4.2 – Sablotron
- For PHP 4.3+ – DOM/XML
- For PHP 5 – DOM and XSL extensions

In the next section, you will learn how to check if your server already has these libraries installed and configured.

How Do I Know If I Have XML/XSL Support?

On the same page that you have created and loaded in the browser previously, test.php, look for these settings, depending on your PHP version:

- For PHP 5:
  DOM/XML
  libXML
  XSL
  libxslt
- For PHP 4.3+:
  DOM/XML
  DOM/XSLT
  libxslt
  libxml
  zlib (on Mac OS only)
- For PHP 4.2:
  Expat
  XSLT Support

If these settings are marked as enabled or active, then you have what you need to process XML documents on the server.

Note: For libxml, libxslt, and Expat, only the version number may be listed (instead of the enabled/disabled status); this indicates they are present on your server.

If any of these libraries are missing or are disabled on your server, then your server is not configured to enable XSL. If you usually dive head-first into your projects and don't like to spend time configuring your server, you might have already encountered this error:

MM_XSLTransform error.

The server could not perform the XSL transformation because an XSLT processor for PHP could not be found. Contact your server administrator and ask them to install an XSLT processor for PHP.

The error is displayed on the dynamic page where you have applied the XSL Transformation server behavior in Dreamweaver 8.

The error message is an indication that your server does not have the required XML/XSL libraries.

To learn how to apply the XSL Transformation server behavior, see the section, Applying a Server-Side XSL Transformation to the RSS Feed, in my previous article.

In general, PHP servers require two components to enable XSL:

- a library (that usually comes with your PHP installation for your operating system)
- a PHP extension using that library

These two components may vary, according to your current PHP version and may also have other prerequisite modules. Go to the next section to find how to install what you need.

Configuring PHP 4.2 Servers

If you're using PHP 4.2.x, I advise you to upgrade to the latest PHP version for the following reasons:

- XML and XSL libraries for PHP 4.2 are deprecated and never really reached a truly stable state.
- Support for XML and XSL has been rewritten from scratch in PHP 5 to offer greater speed, improved memory management, and compliance to W3C standards. You can read more about the XML improvements in PHP 5 in Christian Stoker's article.
- PHP 5 offers more XML support by default and is a lot easier to configure.

If, however, you are restricted to using PHP 4.2.x as your application server, you must have the following libraries for performing XSL server-side development:

- Expat
- iconv
- Sablotron

Expat is an open-source XML parser. It must be installed prior to Sablotron, as it is required by this library. To learn more about Expat, visit the SourceForge website. Sablotron is an open-source XSLT processor. You can find out more about Sablotron from the Ginger Alliance website. To install these libraries on your server, follow the instructions corresponding to your operating system. It is likely that Expat is already built-in into your PHP or Apache installation.

Configuring PHP 4.2 Servers on Windows

The following instructions assume you have already installed a web server with PHP language support on a computer running Windows. Before proceeding with the installation steps, check if the required libraries are not already installed on your server by loading the PHP test page in your browser.

- If Expat is already enabled, you can skip steps 1-3.
- If both Expat and Sablotron are installed, but XSL support is not enabled, you can skip steps 1-7.
- If both Expat and Sablotron are installed, and XSL support is enabled, you can skip this section altogether. Your server is already configured for server-side development using XML data.
- Download the latest version of Expat for Linux from the SourceForge.net website. The file you're looking for is named expat_win32bin.exe.
- Run the downloaded file to install Expat to a folder of your choice on the server.
- Locate the expat.dll file in the libs subdirectory of the Expat installation, and copy it to the system32 subdirectory of your Windows installation (usually, C:\Windows\system32\).
- Copy the iconv.dll file from the dlls folder of your PHP installation to the system32 subdirectory of your Windows installation. This is a character set conversion library required by Sablotron.
- Download Sablotron 0.90 from the Ginger Alliance website. Look for the Windows binary release.
- Extract the ZIP file contents to a folder of your choice on your computer. This creates a folder called Sablot-0.90.
- Locate the sablot.dll file in the bin subdirectory of your Sablotron installation and copy it to the system32 subdirectory of your Windows installation (usually located at C:\Windows\system32\).
- Locate the php.ini configuration file for your current PHP installation (usually located in C:\Windows\) and open it using any text editor (such as Notepad).

Note: You can find out where the PHP configuration file (php.ini) is located on your system by loading the PHP test page containing the server information in your browser. Look for the Configuration File Path setting:

- Search for the following line:
- extension_dir
- Make sure the value of this directive is set to the extensions directory of your PHP installation and that this line is uncommented. For instance, if your PHP installation is located in C:\PHP, the line should be as follows:
- extension_dir = C:\PHP\extensions
- Search for the following line in the php.ini file:
- extension=php_xslt.dll
- If this line is missing, locate the Windows Extensions section of the php.ini file and copy it there. If this line is commented, that is, it starts with the semicolon (;) character, uncomment it by deleting the semicolon to enable XSLT support.
- Save the php.ini file.
- Restart your web server.

To check if the configuration went smoothly, read the section, Testing PHP 4.2 Servers.

Configuring PHP 4.2 Servers on Linux

The following instructions assume you have already installed a web server with PHP language support on a computer running Linux. Before proceeding with the installation instructions, check if the required libraries are not already installed on your server by loading the PHP test page in your browser.

- If Expat is already enabled, you can skip steps 1-3.
- If both Expat and Sablotron are installed, but XSL support is not enabled, you can skip steps 1-6.
- If both Expat and Sablotron are installed, and XSL support is enabled, you can skip this section altogether. Your server is already configured for sever-side development using XML data.
- Download the latest version of Expat for Linux from the SourceForge.net website. The source file you're looking for is called expat-1.95.8.tar.gz.

Note: The version number may be different at the time you download the package. The version used in the following instructions was the latest version at the time of writing this article. Please make sure you use the version number you have in the next instructions.

SYS-CON MEDIA
President & CEO
Fuat Kircaali, 201 802-3001
fuat@sys-con.com
Group Publisher
Jeremy Geelan, 201 802-3040
jeremy@sys-con.com

ADVERTISING
Senior Vice President, Sales & Marketing
Carmen Gonzalez, 201 802-3021
carmen@sys-con.com
Vice President, Sales & Marketing
Miles Silverman , 201 802-3029
miles@sys-con.com
Advertising Sales Director
Robyn Forma, 201 802-3022
robyn@sys-con.com
Advertising Sales Manager
Megan Mussa, 201 802-3023
megan@sys-con.com
Associate Sales Manager
Kerry Mealia, 201 802-3026
kerry@sys-con.com

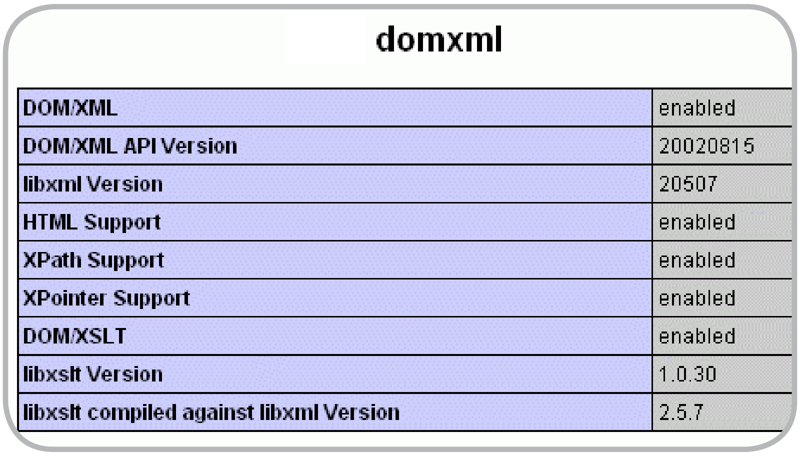
PRODUCTION
Lead Designer
Louis F. Cuffari, 201 802-3035
louis@sys-con.com
Art Director
Alex Botero, 201 802-3031
alex@sys-con.com
Associate Art Directors
Tami Beatty, 201 802-3038
tami@sys-con.com
Abraham Addo, 201 802-3037
abraham@sys-con.com
Assistant Art Directors
Andrea Boden, 201 802-3034
andrea@sys-con.com
Video Production
Ryan Palmieri, 201 802-3036
ryan@sys-con.com

SYS-CON.COM
Consultant, Information Systems
Robert Diamond, 201 802-3051
robert@sys-con.com
Web Designers
Stephen Kilmurray, 201 802-3053
stephen@sys-con.com
Wayne Uffleman, 201 802-3057
wayne@sys-con.com

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

CUSTOMER RELATIONS
Circulation Service Coordinators
Edna Earle Russell, 201 802-3081
edna@sys-con.com
JDJ Store Manager
Brundila Staropoli, 201 802-3000
bruni@sys-con.com

SYS-CON MEDIA



domxml	
DOM/XML	enabled
DOM/XML API Version	20020815
libxml Version	20507
HTML Support	enabled
XPath Support	enabled
XPointer Support	enabled
DOM/XSLT	enabled
libxslt Version	1.0.30
libxslt compiled against libxml Version	2.5.7

- Extract the archive contents to a directory of your choice using the following command:
- tar xzvf expat-1.95.8.tar.gz
- Unix creates a directory called expat-1.95.8, which contains the files for compiling and installing Expat.
- To compile and install Expat, you need root privileges (you may have to log on again using the root user).
- Go to the expat-1.95.8 directory and run the following commands at the command prompt, in the given order:
- ./configure –-prefix=/usr
- make
- make install
- Expat is now installed on your server.
- Download Sablotron 0.90 from the Ginger Alliance website. The source file you're looking for is called Sablot-0.90.tar.tar.

Note: The version number may be different at the time you download the package. The version used in the following instructions was the latest at the time of writing this article. Please make sure you use the version number you have in the next instructions.

- Extract the source archive as you did for Expat, using the following command:
- tar xzvf Sablot-0.90.tar.tar
- Linux creates a directory called Sablot-1.0.2, which contains the files for installing and compiling Sablotron.
- Go to the Sablot-0.90 directory and run the following commands at the commands prompt, in the given order (make sure you are logged in as root):
- ./configure –-prefix=/usr
- make
- make install
- Sablotron is now installed on your server.
- Recompile PHP with XSLT support. To do that, configure it with the following commands, in this sequence:
- ./configure \
- --prefix=/usr/local/php \
- --with-config-file-path=/usr/local/php \
- --with-apxs=/usr/sbin/apxs \
- --enable-xslt \
- --with-xslt-sablot=/usr

Note: The flags used above only enable Sablotron and XSLT support. Add any other flags that might be necessary for your own configuration, such as MySQL support, LDAP, FTP, and so on.

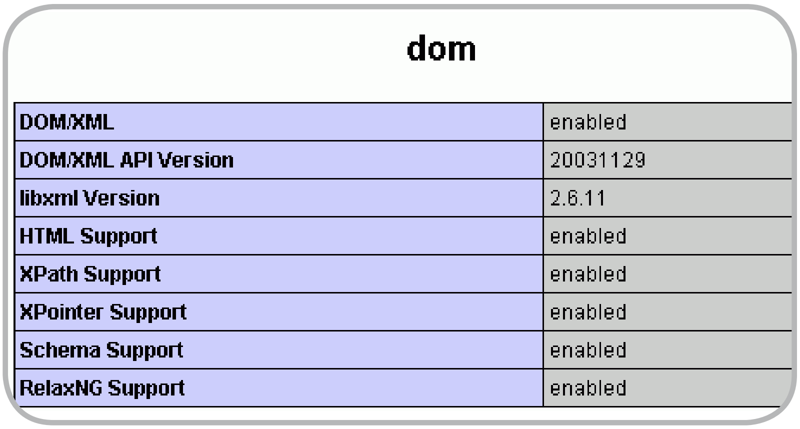
- Also, the path to your current PHP installation may be different than /usr/local/php, depending on your settings or the Linux/Unix version you are using.
- Compile PHP:
- make
- Reinstall PHP:
- make install
- Restart your web server.

To check if the configuration went smoothly, read the section, Testing PHP 4.2 Servers.

Configuring PHP 4.2 Servers on Mac OS

The following instructions assume you have already installed a web server with PHP language support on a computer running Mac OS. Before proceeding with the installation instructions, check if the required libraries are not already installed on your server by loading the PHP test page in your browser.

- If Expat is already enabled, you can skip steps 1-6.
- If both Expat and Sablotron are installed, but XSL support is not enabled, you can skip steps 1-13.
- If both Expat and Sablotron are installed, and XSL support is enabled, you can skip this section altogether. Your server is already configured for sever-side development using XML data.
- Download the latest version of the Expat source code package from the SourceForge.net website. The file you need is expat-1.95.8.tar.gz (the latest available version as of writing this article).
- Extract the library to a folder of your choice, using StuffIt Expander, or the tar command-line utility:
- tar xzvf expat-1.95.8.tar.gz
- Open a Terminal window and navigate to the created folder (usually Expat-1.95.8).
- Configure the Expat library for use with your system, by executing the following command:
- ./configure –with-prefix=/usr
- Compile the library:
- sudo make
- To install the library, you must run the following command:
- sudo make install
- The Expat library should be correctly installed.
- Download Sablotron 0.90 from the Ginger Alliance website and save the tar.gz file to your local drive.
- Extract the Sablotron library to a folder of choice, using StuffIt Expander, or the tar command-line utility:
- tar xzvf Sablot-0.90.tar.gz
- Open a Terminal window and navigate to the created folder (usually Sablot-0.90).
- Run the configuration scripts for your system, by executing the following command:
- ./configure –with-prefix=/usr
- Compile the library:
- sudo make
- Execute the following command to install the library:
- sudo make install
- Configure PHP for XSLT support. To do that, execute the following commands in this sequence:
- ./configure --enable-xslt --with-xslt-sablot --with apxs



dom

DOM/XML	enabled
DOM/XML API Version	20031129
libxml Version	2.6.11
HTML Support	enabled
XPath Support	enabled
XPointer Support	enabled
Schema Support	enabled
RelaxNG Support	enabled

Note: The flags used above only enable Sablotron and XSLT support. Add any other flags that might be necessary for your own configuration, such as MySQL support, LDAP, FTP, and so on.

- Recompile PHP:
- sudo make
- Install PHP (overwriting any other existing version):
- sudo make install
- Restart your web server.

To check if the configuration went smoothly, read the section, Testing PHP 4.2 Servers.

Testing PHP 4.2 Servers

The first proof of a successful installation is that your web server doesn't throw any errors after restarting. To check if everything went smoothly, load the PHP test page in your browser. You will see a section similar to the following image:

Check if these settings are present and enabled:

- Expat
- XSLT Support

For Expat and Sablotron, only the version number may be listed (instead of the enabled/disabled status); this indicates they are present on your server.

Congratulations! Your server is now prepared for server-side development using XML and XSL.

Configuring PHP 4.3+ Servers

Before you continue reading this section, you should first consider upgrading your server to PHP 5 for the reasons explained earlier.

If, however, you are restricted to using PHP 4.3.x as your application server or you don't plan for an upgrade, you need to have the following libraries for perform XSL server-side development:

- libxml (you need at least version 2.4.14)
- DOM XML
- libxslt (you need at least version 1.0.18)
- DOM XSLT
- zlib (on Mac OS only).

DOM XML and DOM XSLT are two PHP extensions used for processing XML files. To learn more about DOM XML and DOM XSLT, visit the PHP.net website. To run properly, these extensions require the libxml and libxslt libraries from the Gnome project. You can find out more about libxml on the xmlsoft.org website. To learn about libxslt, visit this web page.

Configuring PHP 4.3+ Servers on Windows

The following instructions assume you have already installed a web server with PHP language support on a computer running Windows. Before proceeding with the configuration steps, check if the required libraries are not already enabled on your server by loading the PHP test page in your browser.

As of PHP 4.3, the libxml and libxslt libraries are built-in in the PHP installation. The DOM XML and DOM XSLT support is also included by default, but may be disabled in your particular server configuration. To enable DOM XML and DOM XSLT on Windows operating system, follow these instructions:

- Check if the iconv.dll file is located in the system32 subdirectory of your Windows installation. If it's not, copy this file from the dlls folder of your PHP installation to the specified path. This is a character set conversion library required by DOM XML.
- Locate the php.ini configuration file for your current PHP installation (usually located in C:\Windows\) and open it using any text editor (such as Notepad).

Note: You can find out where the PHP configuration file (php.ini) is located on your system by loading the PHP test page containing the server information in your browser. Look for the Configuration File Path setting:

- Search for the following line:
- extension_dir
- Make sure the value of this directive is set to the extensions directory of your PHP installation and this line is uncommented. For instance, if your PHP installation is located in C:\PHP, the line should be:
- extension_dir = C:\PHP\extensions
- Search for the following line:
- ;extension=php_domxml.dll
- and uncomment it by removing the semicolon (;) character, to activate DOM XML.
- Save the php.ini file.
- Restart your web server.

To check if the configuration went smoothly, read Testing PHP 4.3 Servers.

Configuring PHP 4.3+ Servers on Linux

The following instructions assume you have already installed a web server with PHP language support on a computer running Linux. Before proceeding with the installation instructions, check if the required libraries are not already installed on your server by loading the PHP test page in your browser.

- If libxml is already enabled, you can skip steps 1-4.
- If libxsl is already enabled, you can skip steps 1-7.
- If all required libraries are enabled, you can skip this section altogether. Your server is already configured for server-side development using XML data.
- Download the latest version of libxml for Linux from the xmlsoft.org website. The source file you're looking for is called libxml2-2.6.20.tar.gz.

Note: The version number may be different at the time you download the package. The version used in the following instructions was the latest at the time of writing this article. Make sure you use the version number you have in the next instructions.

- Extract the archive contents to a directory of your choice using the following command:
- tar xzvf libxml2-2.6.20.tar.gz
- Linux creates a directory called libxml2-2.6.20, which contains the files for compiling and installing libxml.
- To compile and install libxml, you need root privileges (you may have to relogin using the root user).
- Go to the libxml2-2.6.20 directory and run the following commands at the command prompt, in the given order:
- ./configure -–prefix=/usr
- make
- make install
- Libxml is now installed on your server.
- Download the latest libxslt version from the xmlsoft.org website. The source file you're looking for is called libxslt-1.1.14.tar.gz.

Note: The version number may be different at the time you download the package. The version used in the following instructions was the latest at the time of writing this article. Make sure you use the version number you have in the next instructions.

- Extract the source archive as you did for libxml, using the following command:
- tar xfz libxslt-1.1.14.tar.gz
- Linux creates a directory called libxslt-1.1.14, which contains the files for installing and compiling libxslt.
- Go to the libxslt-1.1.14 directory and run the following commands at the commands prompt, in the given order (make sure you are logged in as root):
- ./configure –-prefix=/usr –with-libxml-prefix=/usr
- make
- make install
- Libxslt is now installed on your server.
- Configure PHP with DOM XML and DOM XSL support. To do that, execute the following commands, in this sequence:
- ./configure \
- --prefix=/usr/local/php \
- --with-config-file-path=/usr/local/php \
- --with-apxs=/usr/sbin/apxs \
- --with-xml=/usr \
- --with-libxml-dir=/usr \
- --with-dom=/usr \
- --with-dom-xslt=/usr

Note: The flags used above only enable DOM XML and DOM XSL support. Add any other flags that might be necessary for your own configuration, such as MySQL support, LDAP, FTP, and so one. Also, the path to your current PHP installation may be different than /usr/local/php, depending on your settings or the Linux/Unix version you are using.

- Compile PHP:
- make
- Install PHP:
- make install
- Restart your web server.

To check if the configuration went smoothly, read Testing PHP 4.3 Servers.

Configuring PHP 4.3+ Servers on Mac OS

The following instructions assume you have already installed a web server with PHP language support on a computer running Mac OS. Before proceeding with the installation instructions, check if the required libraries are not already installed on your server by loading the PHP test page in your browser. In addition to the libraries mentioned earlier, you may also need to install zlib prior to the libxml installation.

- If zlib is already enabled, you can skip steps 1-5.
- If libxml is already enabled, you can skip steps 1-11.
- If libxsl is already enabled, you can skip steps 1-17.
- If all required libraries are enabled, you can skip this section altogether. Your server is already configured for severside development using XML data.

Note: The required libraries are presented in the order they must be installed. Therefore, if you try to install libxml without having zlib, your browser will display an error message and you will have a corrupt installation.

- Download the latest source code package for the zlib library. The file you're looking for is zlib-1.1.4.tar.gz.
- Extract the zlib library to a folder of your choice, using StuffIt Expander, or the tar command line utility:
- tar xzvf zlib-1.1.4.tar.gz
- Open a Terminal window and navigate to the created folder.
- Compile the library, using the following command:
- sudo make
- Install the zlib library:
- sudo make install
- Download the latest source code version of the libxml2 library.
- Extract the library to a folder of your choice, using StuffIt Expander, or the tar command-line utility:
- tar xzvf libxml2-2.6.20.tar.gz
- Open a terminal window and navigate to the created folder.
- Configure the libxml library with the command:
- ./configure –with-prefix=/usr
- Compile the library for your system by running the command:
- sudo make
- Install the library:
- sudo make install
- Download the latest source code version of the libxslt library.
- Extract the library to a folder of choice, using StuffIt Expander, or the tar command line utility:
- tar xfz libxslt-1.1.14.tar.gz
- Open a Terminal window and navigate to the created folder.
- Configure the libxslt library, by issuing the following commands:
- ./configure --with-prefix=/usr/local \
- --with-libxml-prefix=/usrl/local /
- --with-libxml-include-prefix=/usr/local/include \
- --with-libxml-lib-prefix=/usr/local/lib
- Compile the library for your system, by running the following command:
- sudo make
- Install the library on your server (you need to know the root password):
- sudo make install

Note: Alternatively, you can use a bundled package of the libxml and libxslt libraries. Installation instructions are provided here.

- After installing all the required libraries, you need enable them in PHP, by running the following commands:
- ./configure –with-zlib-dir=/usr/local \
- --with-dom –with-dom-xslt=/usr/local

--with-apxs

- Note: The flags used above only enable XSL support. Add any other flags that might be necessary for your own configuration, such as MySQL support, LDAP, FTP, and so on.
- Recompile PHP:
- sudo make
- Install PHP (overwriting any other existing version):
- sudo make install
- Restart your web server.

To check if the configuration went smoothly, read the section Testing PHP 4.3 Servers.

Testing PHP 4.3+ Servers

The first proof of a successful installation is that your web server doesn't throw any errors after restarting. To check if everything went smoothly, load the PHP test page in your browser. You should see a section similar to the following image:

Check if these settings are present and enabled:

- DOM/XML.
- DOM/XSLT.
- libxslt.
- libxml.
- zlib (for Mac OS).

For libxml and libxslt, only the version number may be listed (instead of the enabled/disabled status); this indicates they are present on your server.

Congratulations! Your server is now prepared for server-side development using XML and XSL.

Configuring PHP 5 Servers

On PHP 5 application servers, XML and XSL support has been greatly improved:

- Server-side XSL transformations are a lot faster.
- XML functions are W3C compliant.
- XML functionality is usually built-in and enabled by default.

> "For libxml and libxslt, only the version number may be listed this indicates they are present on your server"

Here is what you need for server-side XSL development on PHP 5:

- DOM extension, which, in turn, requires libxml2.
- XSL extension, which, in turn, requires libxslt.

The lixml2 and libxslt libraries by the Gnome project are usually built-in for Windows, so you don't need to worry about them. You may need to compile these libraries into your PHP installation on Linux-based systems, although this is unlikely in most cases. Go to the section corresponding to your operating system for detailed configuration instructions.

Configuring PHP 5 Servers on Windows

The following instructions assume you have already installed a web server with PHP language support on a computer running Windows. Fortunately, there are a lot of free tools on the Internet that help you install and configure a web server automatically. My personal favorite is WAMP Server, which also sets up a MySQL server, phpMyAdmin and SQLiteManager, in addition to Apache and PHP. For detailed instructions on installing WAMP Server, visit this web page.

Before proceeding with the configuration instructions, check if the required libraries are not already enabled on your server by loading the PHP test page in your browser.

As of PHP 5.0, libxml and libxslt libraries, and DOM extension are built in the PHP installation. The XSLT extension is also included by default, but may be disabled in your particular server configuration. To enable the XSLT extension on Windows, follow these instructions:

- Locate the php.ini configuration file for your current PHP installation (usually in C:\Windows\) and open it using any text editor (such as Notepad).
- Note: You can find out where the PHP configuration file (php.ini) is located on your system by loading the PHP test page containing the server information in your browser. Look for the Configuration File Path setting:
- Search the following line:
- ;extension=php_xsl.dll
- and uncomment it by removing the ; character.
- Save the php.ini file.
- Restart your web server.

To check if the configuration went smoothly, read the section Testing PHP 5 Servers.

Configuring PHP 5 Servers on Linux

The following instructions assume you have already installed a web server with PHP language support on a computer running Linux. Before proceeding with the configuration instructions, check if the required libraries are not already enabled on your server by loading the PHP test page in your browser.

As of PHP 5.0, libxml and libxslt libraries, and DOM extension are built-in in the PHP installation. The XSLT extension is also included by default, but may be disabled in your particular server configuration. To enable the XSLT extension on Linux, follow these instructions:

- Configure PHP with XSL support. To do that, execute the following commands, in this sequence:
- ./configure \
- --prefix=/usr/local/php \
- --with-config-file-path=/usr/local/php \
- --with-apxs=/usr/sbin/apxs \
- --with-libxml-dir=/usr
- --with-xsl=/usr

Note: The flags used above only enable XSL support. Add any other flags that might be necessary for your own configuration, such as MySQL support, LDAP, FTP, and so on. Also, the path to your current PHP installation may be different than /usr/local/php, depending on your settings or the Linux/Unix version you are using.

- Compile PHP:
- make
- Install PHP:
- make install
- Restart your web server.

To check if the configuration went smoothly, read the section Testing PHP 5 Servers.

Configuring PHP 5 Servers on Mac OS X

The following instructions assume you have already installed a web server with PHP language support on a computer running MAC OS X. In the default configuration, the Apache web server is installed, but PHP support is disabled. To learn how to enable it, read the article by James Pelow.

Before proceeding with the configuration instructions, check if the required libraries are not already enabled on your server by loading the PHP test page in your browser.

You can also download a package that will install PHP 5.04 (as of writing this article) with default support for XML and XSLT from the entropy.ch website.

- If libxml2 is already enabled, you can skip steps 1-7.
- If both libxml2 and libxslt are already installed, but XSLT support is not enabled in your PHP installation, you can skip steps 1-12.
- Download the latest source code version of the libxml2 library from the Gnome Project website.
- Extract the library to a folder of choice, using either StuffIt Expander, or the tar command line utility:
- tar xzvf libxml2-2.6.20.tar.gz
- Open a Terminal window and navigate to the created folder.
- Configure the libxml2 library with the following command:
- ./configure –with-prefix=/usr/local
- Compile the library:
- make
- Install the libxml library on your server (you need to know the root password):
- sudo make install
- Download the latest source code version of the libxslt library from this website.
- Extract the library to a folder of your choice, using either StuffIt Expander, or the tar command line utility:
- tar xfz libxslt-1.1.14.tar.gz
- Open a Terminal window and navigate to the created folder.
- Configure the libxslt library, by issuing the following command:
- ./configure --with-prefix=/usr/local \
- --with-libxml-prefix=/usr/local /
- --with-libxml-include-prefix=/usr/local/include \
- --with-libxml-lib-prefix=/usr/local/lib
- Compile the libxslt library:
- make
- Install the libxslt library on the server (you will need the root password):
- sudo make install
- You need to recompile PHP to enable XML and XSL support. Configure PHP with at least the following flags:
- ./configure --with-xml –with-zlib –with-xslt

Note: The flags used above only enable XSL support. Add any other flags that might be necessary for your own configuration, such as MySQL support, ldap, ftp, and so forth.

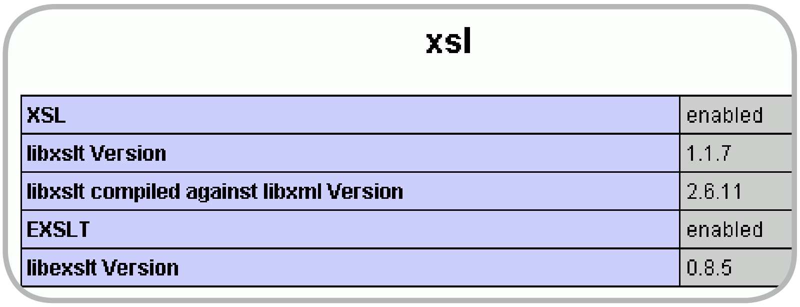
- Compile PHP:
- sudo make
- Install the new PHP version. It will overwrite any previous version.
- sudo make install
- Restart your web server.

To check if the configuration went smoothly, read the section Testing PHP 5 Servers.

Testing PHP 5 Servers

The first proof of a successful installation is that your web server doesn't throw any errors after restarting. To check if everything went smoothly, load the PHP test page in your browser. You should see a section similar to the following image, if the DOM extension was successfully enabled.

If XSL support was successfully enabled, too, you should also see a section similar to this one:



xsl

XSL	enabled
libxslt Version	1.1.7
libxslt compiled against libxml Version	2.6.11
EXSLT	enabled
libexslt Version	0.8.5

Check if these settings are present and enabled:

- DOM/XML.
- libXML.
- XSL.
- libxslt.

For libxml and libxslt, only the version number may be listed (instead of the enabled/disabled status) – this indicates they are present on your server.

Congratulations! Your server is now prepared for server-side development using XML and XSL

Where to Go from Here

In this article, I have explained how to prepare your PHP server for server-side XSL transformations. Now you have everything you need to start exploring the new XSL authoring features in Dreamweaver 8. A good point to start my next article, Consuming a Remote RSS Feed with Dreamweaver 8.

You can also have a look at these additional resources:

- The article Displaying XML Data walks you through using client-side XSL transformations to process XML documents.
- John Skidgel shows how a restaurant website can use XML and XSL to display its daily menu, in his article, Creating XSLT Fragments for Server-Side Transformations.
- Marc Liyanage's website offers a wealth of resources for MAC OS users.
- The XSL FAQ
- The Macromedia XML and XSL Developer Center Resource page

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.

Consuming a Remote RSS 1.0 Feed with Dreamweaver 8

XML and XSL-based web projects

by marius zaharia

Macromedia Dreamweaver 8 is a major breakthrough for XML development, just as Dreamweaver MX 2004 was for CSS. The development team behind this release decided to offer people the entry-point support they needed to tackle XML and XSL-based web projects. The approach Dreamweaver 8 takes to XML and XSL is no surprise:

- Visual XSL transformations using drag and drop from the Bindings panel.
- Easy-to-use objects accessible from the Insert bar.
- Code highlighting and code completion functions for the more experienced developers.

Jen Taylor presents a brief overview of the new XML/XSL features in Dreamweaver 8 in her article, "Dreamweaver 8 New Features and Benefits." In this article, I use the new XML features in Dreamweaver 8 in a real-life situation: consuming a remote RSS feed using server-side XSL transformations. It is a useful exercise if you plan to include news headlines, articles lists, blog posts, or other syndicated content in your personal website, in a portal, or company site.

Dreamweaver 8: Workspace Orientation

First, I'll take you on a quick tour of the new XML authoring features in Dreamweaver 8 to help you become familiar with the workspace.

Creating XML and XSL Documents

With Dreamweaver 8, you can create XML and XSL documents from scratch, just as you would create any other type of page. Open Dreamweaver 8 and select File > New. The New Document dialog box appears.

Notice that you can also create XSLT fragments that you can later use to process XML data in your dynamic pages. An XSLT fragment is a file that does not contain the <head> and <body> tags like a full XSLT page. It is a simple piece of code that is later inserted in a dynamic page.

If you already have a static site set up and want to go dynamic, you can easily convert HTML pages into XSL templates, from the File menu by selecting File > Convert > XSLT 1.0.

Visual Authoring

The best thing about Dreamweaver 8 is that you get a smooth landing if you plan to take up XML development. You can see a tree-like representation of the XML data source that you're using, right in the Bindings panel:

You can drag nodes from the tree and drop them onto your XSL stylesheet, as you will learn later in this article.

You can use any of the objects from the XSLT tab on the Insert bar to display repeating nodes, define simple or multiple conditions, and comment your code.

Towards the end of this tutorial, you'll also learn how to use the XSL Transformation server behavior (from the Application panel) in your dynamic pages to process and display XML data:

Coding Made Easy

If you'd like to tweak the code or try some XSL tricks of your own, you can do this painlessly using the code hints and code auto-completion features in Dreamweaver 8.

You'll discover and master all these features in no time, as you follow this tutorial.

Application Scenario

In this article, you will consume Macromedia's RSS feed (available here: weblogs.macromedia.com/dev_center/index.rdf) in a mock-up site that can be anything from a blog, to a company site, or a personal site. Macromedia makes its Developer Center resources available to the public as an RSS feed. It is a great way to display the latest articles from Macromedia right in your site and let other developers know about them.

In this tutorial, you will:

- Understand RSS feeds and syndication
- Connect to the Macromedia Developer Center's remote RSS feed
- Process the RSS feed with XSLT using the new features in Dreamweaver 8
- Apply a server-side XSL transformation to include the feed in your mock-up site, under the "Related Articles" section of the generic page. Dreamweaver 8 supports server-side XSL transformations for ColdFusion, ASP, ASP.NET, and PHP pages.

What Is Syndication?

Web syndication is a popular method of making content available to other websites simultaneously. Syndication is achieved using web feeds or channels, which are written in a variety of standards RSS 0.9, 1.0, 2.0, or Atom. Despite the differences between the various standards and specifications, it is important to keep in mind that all feeds are actually XML documents that contain list-oriented

This article originally appeared on www.macromedia.com/devnet. Reprinted with permission.

information. The site that makes its content available as a source of information is said to "publish a feed," while the sites that republish that content are said to "consume the feed."

Any information on your site that visitors might be interested in reusing is a good candidate for an RSS feed. This includes news headlines, article lists, press releases, job listings, bookmarks, conference events, playlists, or software releases.

The benefits of syndication are obvious:

- Imported content is updated automatically when the remote feed changes.
- Content is made available to more people, as sites republish the same feed over and over. For instance, Macromedia's Developer Center resources are made available by DreamweaverFAQ.com, dopostback.com, actionscript.it, and edcox.net.webwatch.
- Content is served in a format that matches your site's look and feel. Notice how differently the same information is displayed in the websites above.

What's in an RSS Feed?

RSS is just another flavor of XML. The term "RSS" is actually used to refer several different formats that are used in parallel, and therefore can have different meanings:

- ***RSS 0.9x:*** In these versions, it stands for Really Simple Syndication.
- ***RSS 1.0:*** Includes RDF, a W3C standard for meta data. In this version, RSS stands for RDF Site Summary. RSS 1.0 also incorporates the Dublin Core Module, a standardized set of meta data used to describe both the feed and the individual items.
- ***RSS 2.0:*** This is the latest version, which does not include RDF.

If you feel baffled by all the different versions and standards available for RSS, don't worry: they are similar enough that you can treat them as simply any other XML document.

Note: If you want to learn more about the differences between various RSS dialects, check out this article from the O'Reilly XML.com site.

An RSS feed is made up of a channel, which has a title, a link, and a description followed by a series of itemseach of which have a title, a link, a description and an optional author and publishing date. To see what I mean, download a copy of the Macromedia Developer Center RSS feed and open it in Dreamweaver 8.

Notice how Dreamweaver 8 uses code coloring to make the document easier to read:

- Tags and attributes are blue
- Attribute values are green
- Content is displayed in black

The document starts with an XML declaration followed by the RSS declaration, which includes references to the RDF namespace, the Dublin Core Module namespace, and the RSS namespace.

For a reminder of namespaces, revisit this section of my previous article, XSL Overview.

The channel node contains generic information about the RSS feed: title, link, and a short description:

```
<channel rdf:about="http://www.macro-
media.com/devnet/">
    <title>Macromedia Developer Center
RSS Feed</title>
    <link>http://www.macromedia.com/
devnet/</link>
    <description>Macromedia Developer
Center is your center for the tutori-
als, articles, and sample applications
you need to master Macromedia prod-
ucts. </description>
```

Also included in the channel node is a table of contents for the current feed, specified by the items element. This is actually an ordered list of links to the Developer Center articles. To indicate that the order is important, a sequence container is used: rdf:Seq. The items in the sequence are specified by rdf:li elements.

```
<items>
    <rdf:Seq>
      <rdf:li rdf:resource="http://
www.macromedia.com/devnet/mx/flashcom/
articles/comm_components.html" />
<rdf:li
rdf:resource="http://www.macromedia.
com/devnet/mx/flashcom/articles/flv_
bestpractices.html" />
<rdf:li
rdf:resource="http://www.macromedia.
com/devnet/mx/flashcom/articles/dyn_
buffering.html" />
<rdf:li
rdf:resource="http://www.macromedia.
com/devnet/mx/coldfusion/articles/adv_
flashforms.html" />
    </rdf:Seq>
  </items>
```

Notice that each item's rdf:resource URL must be the same as the associated item element's rdf:about URL, much like in a table of contents, where each chapter is located at a specified page number.

Note: Items appearing in the feed but not as members of the channel's table of contents are likely to be discarded by RDF parsers.

Finally, the actual Developer Center articles are listed in the feed, as item elements. An item is made up of a title, a link, a description, a subject (which indicates the topic covered by the article), an author, and a publishing date:

```
<item
rdf:about="http://www.macromedia.com/
devnet/mx/flashcom/articles/comm_com-
ponents.html">
<title>Building Communication
Components</title>
<link>http://www.macromedia.com/
devnet/mx/flashcom/articles/comm_com-
ponents.html
</link>
<description>This book excerpt from
O'Reilly describes the first major
step in building a full-fledgedappli-
cation.</description>
<dc:subject>Flash Communication
Server</dc:subject>
<dc:creator>Brian Lesser</dc:creator>
<dc:date>2005-07-25T17:31:46-08:00</
dc:date>
</item>
```

Now that you have learned how to read the code that makes up an RSS feed, you can start processing it to display the articles in your site.

Creating an XSLT Fragment

You'll start by creating an XSLT fragment that displays the article headings in the Related Articles area of the mock-up site:

- Download and unzip the tutorial sample files in your site root (the site must be running a ColdFusion, PHP, ASP or ASP.NET testing server). The sample files are linked at the beginning of this article. Check out How to define a site in Dreamweaver (TechNote 14028) for instructions on correctly defining your site.
- Open the homepage.html page in Dreamweaver 8.
- Select the list under the Related Articles heading, by putting your cursor in the that area of the page and clicking the corresponding <ul> tag form the tag selector:
- Press Delete to remove this section. You will later replace it with the Developer Center headlines.
- Select File > Save As and save the current page as homepage.cfm, homepage.php, homepage.asp, or homepage.aspx, depending on your server model.
- Select File > New and in the displayed dialog box, select Basic Page > XSLT (Fragment), and click Create. Dreamweaver prompts you to associate an XML file as the data source for the new XSLT fragment.
- Select "Attach a remote file on the Internet" and paste the following URL in the text box:
- http://weblogs.macromedia.com/dev_center/index.rdf

Note: Make sure you remove the http:// text before pasting the URL so those characters are not duplicated in the URL.

- Click OK to attach the XML source. Notice that a tree-like representation of the RSS feed is displayed in the Bindings tab of the Application panel (If you cannot see the Bindings tab, select Window > Bindings).

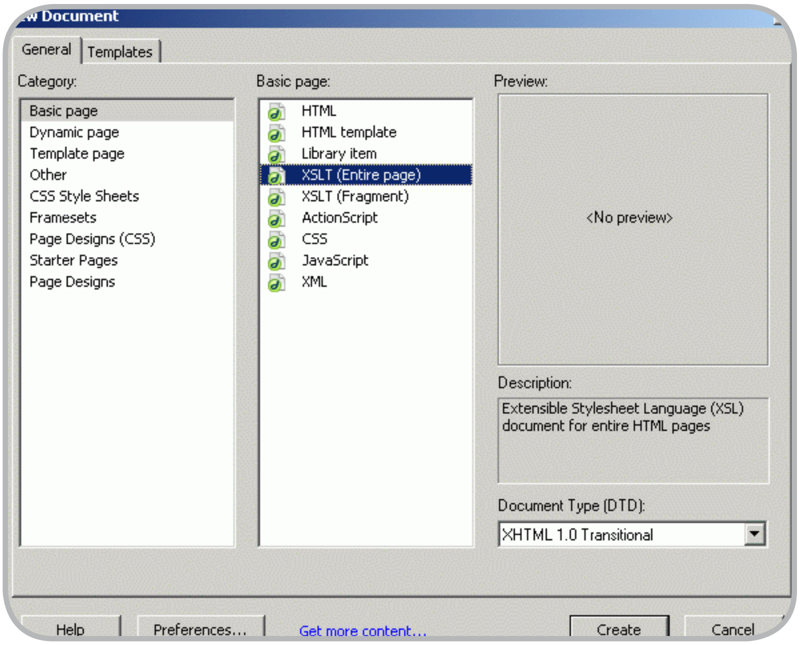
Note: You will need to be connected to the Internet for a remote RSS feed to populate the Bindings panel.

- Dreamweaver uses the following symbols to display XML trees:
  <> for an element that appears exactly once within its parent node
  <>+ for an element that appears one or more times within its parent node
  <>? for an optional XML element
  @ for an XML attribute
- Save the file as headlines.xsl.

The XSLT fragment that you will create next will display the article titles and their authors. The titles will be linked to the actual articles from the Developer Center.

You can achieve this by dragging and dropping from the Bindings panel:

- Drag the rss:title node (having the parent rss:item) from the XML tree and drop it onto the page. Dreamweaver inserts an XML data placeholder in your page, which is, in fact, the XPath expression corresponding to the node.



Description:
Extensible Stylesheet Language (XSL) document for entire HTML pages

Document Type (DTD): XHTML 1.0 Transitional

Note: You must use the rss:item node from the rss:item parent node and not from the rss:channel parent node.

- Place the mouse cursor after the XML placeholder and press Shift-Enter (Windows) or Shift-Return (Macintosh) to begin a new line immediately below it.
- Type by on the new line and press the spacebar.
- Drag the dc:creator node from the Bindings panel and drop it after "by."
- Preview the page in your browser using F12 (Windows) or Option + F12 (Macintosh). It is pulling the latest article added to the Developer Center RSS feed and the feed is updated regularly, so your browser will display a different article than the one shown here.

Notice that Dreamweaver automatically generates a temporary file for previewing the transformation results. However, this is neither a client-side, nor a server-side transformation. Dreamweaver internally does the transform using its built-in XSLT processor and generates a temporary file for preview purpose. Dreamweaver is performing the translation. You will apply the actual server-side transformation later on in this tutorial. Also notice that the CSS styles that are applied to the homepage are not available in the transformation. They will be after you include the XSLT fragment in the homepage. Finally, notice that only one article is displayed in the transformation result page. To display all the articles from the feed, you need to create a repeat region that loops through all the items in the feed and displays them.

Creating the Repeat Region

To display all the articles from the RSS feed, follow these steps:

- In Dreamweaver Design view, select the title and creator elements as well as the "by" text from the headlines. xsl document and right-click them (Control-click on Mac).
- From the displayed menu, select Paragraph Format > Paragraph. This will display each article as a new paragraph. Paragraph may already be selected in the contextual menu.
- Go to the XSLT tab from the Insert bar, and click the Repeat Region button.

Note: The default view for the Insert bar is menu view. If you are in menu view, select Show as Tabs from the list of Insert bar options. If you are in Tab view, right-click (Control-click on Mac) any of the tabs and select Show as Menu.

- In the XPath Expression Builder dialog box, select the repeating node that you want to loop through, which is rss:item. Be careful not to confuse this with the rss:items node.
- Click OK. Notice that Dreamweaver displays a gray border labeled xsl: for-each, around the selected text, to indicate a repeat region.
- Save your page and preview it in a browser. Again, since the Developer Center resource feed is updated regularly, the list of articles will be different.

Again, Dreamweaver is doing the XSL transformation and displaying the result in a temporary HTML file. In the next section, you'll learn how to link each article to its corresponding URL, provided via the RSS file.

Creating Links with XSL

To link the title of each article to the corresponding resource on the acromedia Developer Center, follow these steps:

- In the headlines.xsl page, select the XML data placeholder for the article title.
- In the Property inspector, click the folder icon next to the Link text box, to define a link on the selected text.
- Select the Data Sources option in the Select file name from section of the Select File dialog box.
- Each article title should point to the URL contained by its rss:link node.
- Select the rss:link node from the XML tree. Be sure to select the rss:link node under the rss:item parent node and not the rss:link node under the rss: channel parent node.
- Click OK. Dreamweaver generates the link.
- Save the page and preview it in the browser.

Click on the links to test it out. In the next section, you will apply the XSL Transformation server behavior to display the articles on the homepage.

Applying a Server-Side XSL Transformation to the RSS Feed

To list the Developer Center articles in the Related Articles section of your homepage, follow these steps:

- Open the dynamic homepage file (PHP, ASP, or ColdFusion page) you created earlier in the tutorial.
- Place the mouse cursor below the "Related Articles" heading.
- Go to the Application tab/menu on the Insert bar, and click the XSL Transformation button.

Note: You can access the same command from the Server Behaviors tab, as well.

Note: If you are working on a ColdFusion server, Dreamweaver might prompt you to specify a RDS login password. Type in the password or contact your network administrator, if you don't know it.

- In the XSL Transformation dialog box, click the Browse button next to the XSLT file text box. Then browse to and select the headlines.xsl file. Notice that Dreamweaver automatically detects the associated XML data source and fills in the XML URI text box.
- Click OK to apply the transformation. Dreamweaver displays the XSL fragment directly in your dynamic page. It also generates a folder called "includes" in your site root. This folder contains the library that performs the server-side transformation and must be uploaded on your testing tester. Notice that the CSS styles from the stylesheet attached to the home page are automatically applied to the imported XSL fragment.
- In the Files panel, find the includes folder Dreamweaver created and upload it to your server.
- Save the page and display it in the browser (Make sure you have uploaded the "includes" folder to your testing server).

What Is Really Happening

Dreamweaver generates a runtime library file that is used by the application server to process the XML data and output HTML content to the client's browser. The runtime library is the file in the includes folder. The XSLT fragment, the dynamic page containing it, and the runtime library from the "includes" folder must all be on the server for your page to display correctly. To review the server-side transformation workflow, visit this section of my previous article.

Notice that all the items from the RSS feed are displayed. If the feed has 10 or 20 articles, this may lead to a very long home page, which might interfere with your design requirements. In the next sections, you will learn how to limit the number of items that are displayed from an RSS feed using parameters and you will learn how to sort items.

Using Parameters in an XSL Transformation

The processed RSS feed currently displays several articles on the homepage. Suppose you only want to list the latest four articles from the Developer Center. To do this, you need a parameter that tells the XSL transformation to display only the first four items. Here are the steps to do that.

- Open the headlines.xsl page and click inside the XSL Repeat Region.
- Select the <xsl:for-each> tag from the tag selector.
- In the Property inspector, click the lightning bolt icon to define the condition that displays only the first four items:
- In the XPath Expression Builder dialog box, click Build Filter, then click the plus (+) button to define a new XPath condition. See the section in my article, XSL Overview, for a review of XPath.
- Edit the default condition in the Where field. Select that field and type position(). In the Operator menu, select <= (less or equal). Finally, in the Value field, type $ItemsPerPage.
- The position() function returns a number equal to the position of the current node. This value must be less or equal than the value specified by the ItemsPerPage parameter. You will define this parameter later, in your dynamic home page file.
- Click OK to apply the condition.
- Save the XSL document, and preview it in the browser using F12 (Windows) or Option + F12 (Macintosh). Your browser will throw an error similar to this one:
- XSLT Error: The variable 'ItemsPerPage'

is not defined.Source tree node: #document.

- This is because Dreamweaver performs the XSL Transformation internally, using nothing but the XSLT fragment you have created and the XML source file. The ItemsPerPage parameter that you have used to limit the number of displayed items is defined in xslt page, but its value is passed to the XSLT page by a dynamic page (homepage.php, homepage.asp, homepage.aspx, or homepage.cfm). To make the browser preview work and to get used to the code hints in Dreamweaver 8, type the following line of code in the headlines.xsl file right before the xsl:template tag:
- <xsl:param name="ItemsPerPage" select="4" />
- This code defines the value of the ItemsPerPage parameter and gives it a default value of 4.

Note: The default value of the ItemsPerPage parameter specified in the XSLT fragment will be ignored at run-time, when the server-side transformation is performed. The value specified in the XSL Transformation server behavior will be used instead.

You can now preview the XSLT fragment in your browser without errors.

- Switch to the homepage dynamic page.
- In the Server Behaviors tab from the Application panel, double-click the XSL Transformation server behavior to edit it.
- Dreamweaver opens the XSL Transformation dialog box. In the XSLT Parameters area, click the XSLT parameters plus (+) to define a new parameter.
- Select ItemsPerPage from the Name pop-up menu, and type 4 for its value.
- Click OK. If 4 appears with double quotes around it in the XSL Transformation dialog box, click the Edit button, remove the quotes, and click OK.
- Click OK again to dismiss the XSL Transformation dialog box. Save your page and preview it in your browser. Make sure you have uploaded the headlines.xsl, homepage file and includes folder to your testing server.

Only the first four articles display in the browser. You may need to refresh your browser if all of the articles are still showing.

But how do you know these are the latest articles from the Developer Center? Since it is not always safe to assume RSS feeds list items in chronological order, you will learn how to sort them by date.

Sorting Items

In my previous article, XSL Overview, I explained the syntax for sorting XML nodes. You will apply that knowledge to sort articles by date.

- With the headlines.xsl page open, switch to Code view.
- Locate the following line of code:
- <xsl:for-each select="rdf:RDF/rss:item[position() <= $ItemsPerPage]">
- You can use Edit > Find and Replace to look for it.
- Press Enter (Windows) or Return (Macintosh) after this line and start typing <xsl:s. Dreamweaver suggests hints to fill in your code.
- Select xsl:sort and press Enter (Windows) or Return (Macintosh). If you are unable to see the code hints menu, you can press Control + Spacebar to display it.
- As you type a space, Dreamweaver suggests the attributes of the <xsl:sort /> tag. Choose select and type dc:date as its value.
- Type another space and select order from the list of suggested attributes. Choose descending for its value using the Dreamweaver code hints again.
- Close the xsl:sort tag by typing />. The line should look like the following:
- <xsl:sort select="dc:date" order="descending" />

Dreamweaver can really make coding XSL documents child's play. If you're not sure which function or XSL construct you need, you can always check out O'Reilly's XSLT Reference on the Reference panel by selecting Window > Reference and selecting O'REILLY XSLT Reference from the Book pop-up menu.

You can even copy-paste code samples from the Reference panel directly in your code.

Save the file and put your files to the server and preview them in your browser. You will see that the list of articles is sorted in descending order.

Creating a Detailed Display of the RSS Feed

The links to the Developer Center articles are okay, but what if you need a detailed page that displays all articles, grouped by subject, listing their descriptions and their authors? In this section, you will exercise your newly-acquired XSLT skills to create a detailed view of the RSS feed and you will learn how to use conditional regions to display specific content.

Follow these steps to create the XSLT fragment that displays the detailed view of the articles:

- Select File > New to create a new XSLT fragment.
- At the Dreamweaver prompt, attach the remote RSS feed, as described previously.
- Save the page as all_articles.xsl.
- Drag the following nodes from the XML tree in the Bindings panel and drop them onto your page, each on a new line:

```
 rss:title (from the rss:channel parent node).
 rss:description (from the rss:channel parent node).
```

- Format the channel title as Heading 2.
- Suppose your site focuses on Flash technology. Therefore, you would like to display Flash-related articles first, and all the others after under a separate heading. Type the following text below the channel description, on two separate lines:
- Flash
- Other Technologies
- Format each of the two lines as Heading 3.
- On a new line below the Flash heading, drag and drop the article title node (rss:title) from the rss:item parent node.
- Link the article title to the corresponding resource, as described previously.
- On the same line, insert an em dash after the article title from the Text tab of the Insert bar.

Note: If the em dash button is not showing on the Insert bar, you may need to select it from the more characters menu at the right edge of the displayed characters.

- Type by after the em dash, and then drag and drop the dc:creator node after it.

- Place the mouse cursor at the end of the line and press Shift + Enter (Windows) or Shift + Return (Macintosh) to begin a new line immediately below it.
- On the new line, drag and drop the article description node (rss:description) from the rss:item parent node..

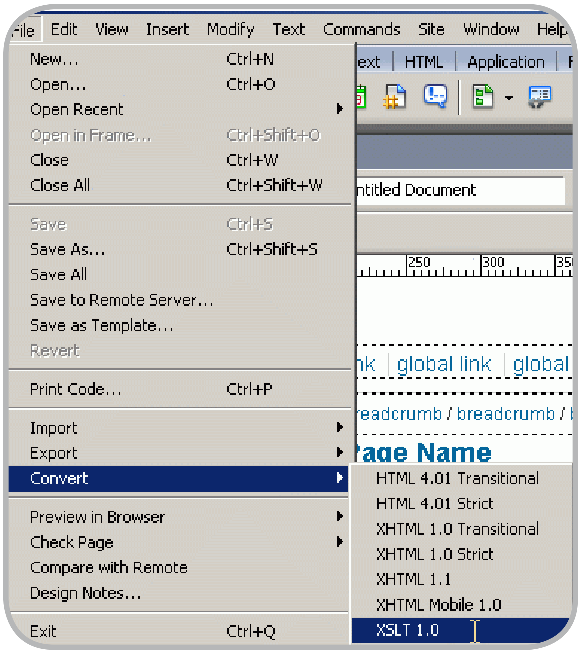
In the next section, you will learn how to use conditional regions to group articles by topic.

Using Conditional Regions

To display only Flash-related articles under the Flash heading, you need to define a conditional region. I discussed the XSL syntax for building conditional regions in my previous article, XSL Overview .

Now you will learn how to do this visually, using the visual XML authoring features in Dreamweaver 8:

- Select the paragraph containing the article details by clicking the corresponding <p> tag from the Tag Selector.
- Go to the XSLT tab from the Insert bar and click the Conditional Region button:
- In the Conditional Region dialog box, type the following line:
- contains(dc:subject,'Flash')
- This is an XPath expression that returns true when the subject node contains the word "Flash." When this condition is satisfied, the article is displayed.



- Click OK to dismiss the Conditional Region dialog box. Dreamweaver displays a gray border labeled xsl:if around the selected text.

To display all the articles about Flash, you need to loop through all the items from the RSS feed and evaluate the test condition each time using a repeat region. To do this, follow these steps:

- Select the <xsl:if> tag in the tag selector.
- Apply a repeat region, as I explained earlier in this article.
- Save your page and preview it in the browser. Yours will probably have more articles since there have been more Flash articles since the time of writing.

Before completing this XSLT fragment, you need to display the remaining articles under the "Other Technologies" heading:

- Select the <xsl:for-each> tag from the Tag Selector.
- Copy the selected content and paste it on a new line below the "Other Technologies" heading using the Copy and Paste commands from the Edit menu.
- The conditional region must display only articles that do not contain the word "Flash" in their subject node. To do this, you need to change the test condition. Select the <xsl:if> tag corresponding to the second conditional region on the tag selector.
- Dreamweaver displays the test condition for the selected region in the Property inspector.
- Change the condition to this one:
- not(contains(dc:subject,'Flash'))
- This is simply the negation of the initial XPath expression. It says: "if the subject of the article does not contain the word Flash, then display the article."
- Save the XSLT Fragment and upload it to your testing server.

To display the detailed feed in a dynamic page, you'll use the XSL Transformation server behavior again:

- Open the homepage.html page and save it as a dynamic page, corresponding to your server model ColdFusion, PHP, ASP, ASP.NET.
- Switch to Code View, select the code lines 19 to 28 and delete them, to remove the contents from the main area of the page.
- Go back to Design view, and place your mouse cursor inside the empty <div> tag that should display the content:
- Apply the XSL Transformation server behavior .
- Save your page and upload all files on your testing server. Preview the page in the browser and view the results:

Where to Go from Here

In this article, you have learned how to use Dreamweaver 8 to consume a remote RSS feed in your dynamic pages, and process it in various ways. With the visual XML authoring features, you can now easily display XML data, define repeat and conditional regions, and build complex XPath expressions. The ways you can use XML and XSL in your web applications are only limited by your imagination. The following articles can give you more clues:

- In the article, Displaying XML Data, Jon Varese walks you through using client-side XSL transformations to process XML documents.
- John Skidgel shows how a restaurant website can use XML and XSL to display its daily menu, in his article, Creating XSLT Fragments for Server-Side Transformations.
- Check out the Dreamweaver XSL FAQ.
- Visit the Dreamweaver Developer Center XML and XSL category page for more resources on learning XSL transformations.

You can learn more about the RSS tools out from this article on RSS validators, aggregators, and clients. If you need a tool for creating your own RSS feeds, check out MX RSS Reader-Writer, a Dreamweaver extension from InterAKT Online.

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

Best Practices for Encoding Video with the VP6 Codec

A simple yet powerful tool for Flash developers

by william caulfield

Flash Video is rapidly changing the landscape of video on the web. Developers will need a new set of skills and knowledge relating to video editing, encoding, and delivery. This article introduces the video encoding process using the Flash 8 Video Encoder, a simple yet powerful tool for Flash developers.

In this article I will explain some of the basics of video encoding and give a step-by-step tour of the new Flash 8 Video Encoder with the On2 VP6 codec. The On2 VP6 codec enables Flash Video encoding at lower bit rates and larger frame sizes. At the end of this article, you will find a wide range of recommended settings for VP6 encoding. These settings provide a reference point for you as you create your own custom projects.

A Short History of Video on the Web

In 1998 web video was all about Real Media, which provided the best codecs combined with the best server. Within a few years, Microsoft had moved to the forefront with Windows Media. Apple was also on the move with QuickTime. For many years, these three companies had cornered the web video market. All had their pluses and minuses depending on the type of projects and client requirements. What they all had in common were modern video and audio codecs and the ability to deliver video through progressive download or streaming technologies. What they lacked was an easy way to add rich media and interactivity to the audiovisual experience. Developers were learning to use HTML plus TIME for Windows Media, SMIL for Real, and all the various technologies for QuickTime. None of them provided a perfect solution for delivering web video.

This article originally appeared on www.macromedia.com/devnet. Reprinted with permission.

Experience with Flash Video

During the same period Flash became the web standard for motion graphics and interactivity on the web. A painless plug-in installation process made Flash Player practically ubiquitous on desktop and laptop computers. Developers used various approaches to integrate video with Flash. Many used Flash to make players for Windows Media, Real, and QuickTime. Others vectorized video to create an early version of Flash Video. Some began creating the video as vectorized Flash, later exporting the contents to video formats. The need for a fully integrated Flash Video codec soon became apparent.

The Spark codec brought integrated video to Flash. It marked a tremendous first step and was instantly accepted as a new alternative for web video. Our clients now use it to display television promos, music videos, and educational clips.

The New On2 VP6 Codec and How It Differs

Spark had its limitations at first. The limiting factor on quality in web video is the wide variance in download bandwidth among clients. The most important quality for a video codec intended for the web is encoding efficiency—the codec should be able to squeeze the most quality into the smallest possible bit rate. This is especially important when using streaming video. The new On2 VP6 codec brings Flash Video up to speed with the industry leaders in encoding efficiency so that Flash Video projects can keep the bit rate down and the quality up.

What You Should Know About Video

If you are a Flash developer but new to video, here are a few tips and pitfalls to avoid:

- ***Frame size:*** Standard-definition NTSC video—the format most commonly used in the United States—is usually digitized to 720 x 480 pixels. Ignore these dimensions. Before encoding the video, resize it to a 4 x 3 ratio—for example, 640 x 480, 320 x 240, and so on (see the next section). Or else crop and resize it to a 16 x 9 ratio—for example, 640 x 360, 320 x 180, and so on. In case you ever work with various wide-screen formats—such as 2.35:1—crop the letterboxing, assume a 640-pixel width, and get out the calculator.

  Although these ratios are standard, and should be used to avoid distorting the video, the size of the encoded video is not set in stone. The original web video sizes used heights and widths that were evenly divisible by 16. This was mandatory for many early codecs. Although this is not necessary for modern codecs, you should stick to even heights and widths.
- ***Interlacing:*** When video is intended for television, it is interlaced. Each frame is made up of two fields of interlaced lines. On a computer monitor this is seen as a comb effect because one field is 1/60 of a second ahead of the other. Interlaced video should be deinterlaced before encoding.

  If the original source was film,

the interlacing may occur in a three-progressive-frames, two-interlaced-frames pattern. This common method of telecining converts the original 24 frames per second (fps) film frames to 29.97 fps video frames. It is possible to use inverse telecine to return the video to its original 24 frames.

- ***Frame rate:*** Your video projects will usually be at 30 fps or will have been inverse-telecined to 24 fps. If you want to lower the frame rate, you should use equal divisions of the source frame rate. For 30 fps, use 15 fps, 10 fps, 7.5 fps, and so on. For 24 fps, use 12 fps, 8 fps, 6 fps, and so on.

Converting Frame Sizes

Frame size (mentioned in the previous section) brings up a number of confusing issues. Most confusing is the concept of square and non-square pixels.

Television presents video as horizontal lines. Computers present video as a grid of pixels. Both use a 4 x 3 ratio for standard video playback. The digitized version is referred to as a "square pixel" resolution.

Video is often digitized at what's called a D1 resolution, which means 720 x 480 for NTSC and 720 x 576 for PAL. Within the world of the 4 x 3 standard, these are referred to as "non-square pixel" resolutions—NTSC pixels are wider at a ratio of 1.1:1 or 11/10, while PAL pixels are slightly taller at a ratio of about 0.915:1 or 54/59.

So why convert your video to square pixels and take the 720 x 480 frame to 320 x 240 or other legacy format? On one hand, that's the way it's always been done. But there's also no reason to use the non-square pixel dimensions if the encoded file will be played on a computer screen.

Since the early days of video, D1 has been the "legal" resolution. However, these are not the resolutions at which the vast majority of people watch video. As I mentioned earlier, television presents video at a 4 x 3 ratio screen size. When computers came along, screen resolutions did the same. Early computer monitor resolutions were 480 x 360 and 640 x 480 pixels; my current favorite is 1280 x 960.

Practically speaking, you should use the D1 resolution for videotape and DVD; use a 4 x 3 ratio for computer playback. A standard conversion from D1 to 4 x 3 would begin as follows:

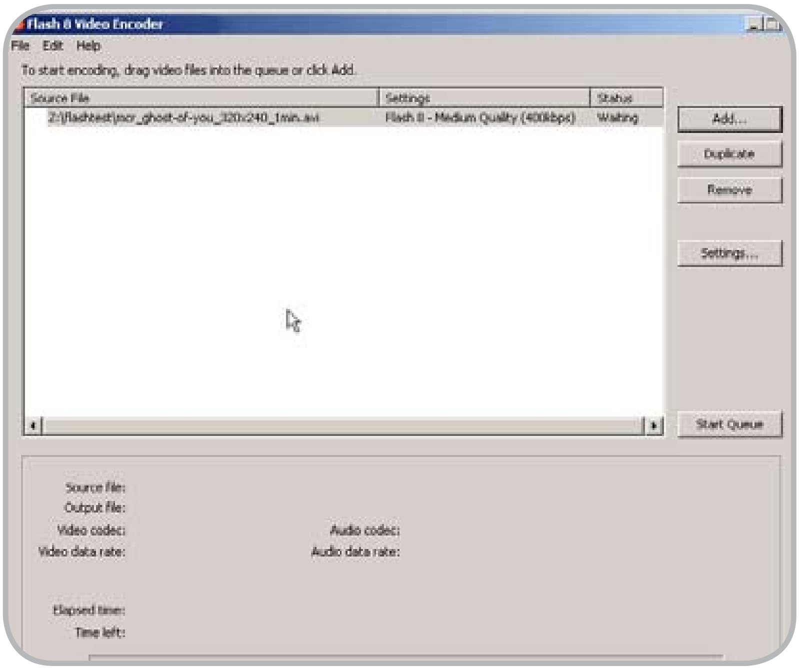
- Capture at 720 x 480 pixels.
- Deinterlace or inverse-telecine the video (see the previous section).
- Crop eight pixels from the left and right sides (so it's 704 x 480).
- Resize the video to 640 x 480 pixels.

From here you may crop out further edge noise, crop out letterboxing, do further resizing, and so on. Everything from Step 4 onward is based on a 4 x 3 ratio.

There are times when I want to capture and de-interlace on the fly. In that case, the process would be as follows:

- Capture at 360 x 240 pixels.
- Crop four pixels from the left and right sides (so it's 352 x 240).
- Resize to the video to 320 x 240 pixels.

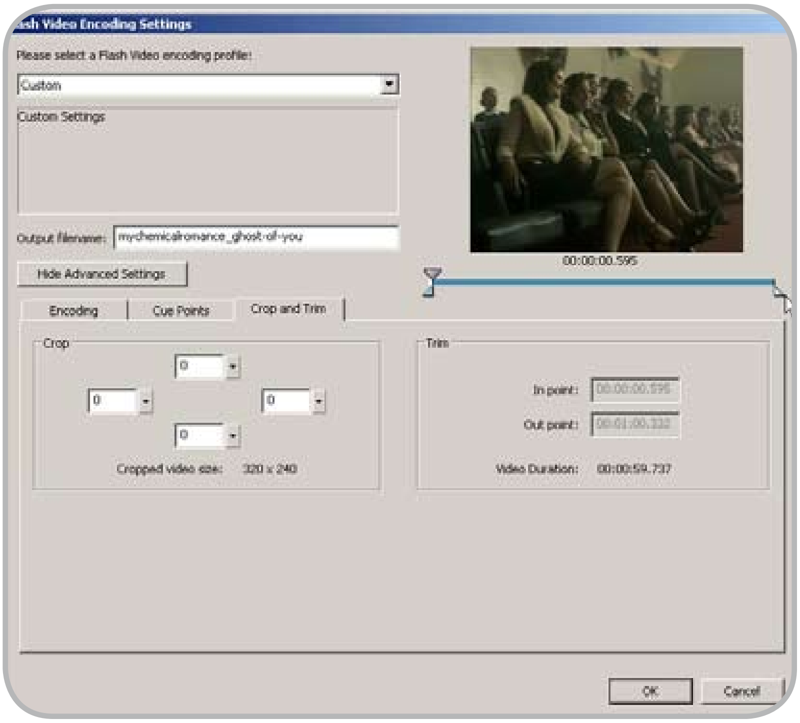
Once again, the target is a 4 x 3 resolution and the 360 x 240 resolution is only an intermediary step.



To start encoding, drag video files into the queue or click Add.

Source File	Settings	Status
Z:\flashtest\mcr_ghost-of-you_320x240_1min.avi	Flash 8 - Medium Quality (400kbps)	Waiting

Source file:
Output file:
Video codec:
Audio codec:
Video data rate:
Audio data rate:
Elapsed time:
Time left:



Please select a Flash Video encoding profile: Custom

Custom Settings

Output filename: mychemicalromance_ghost-of-you

00:00:00.595

Crop: 0 0 0 0

Cropped video size: 320 x 240

In point: 00:00:00.595

Out point: 00:01:00.332

Video Duration: 00:00:59.737

When you compare 720 x 480 and 640 x 480 video side by side on a computer screen, you'll see the most practical answer of all as to why you want to make the conversion: "That one looks stretched out!"

Working with the Flash 8 Video Encoder

The Flash 8 Video Encoder is installed in a separate directory as part of your Flash 8 Professional installation. This section walks you through some of its capabilities and features. After you know where and how to change the settings, check out the tables at the end. They give you some recommended settings for various types of projects.

When you launch the Flash 8 Video Encoder, you are presented with a simple interface. You can add content to the encoding queue by clicking the Add button or by dragging content to the queue window. The default setting for encoding is Medium Quality 400 Kbps. If your source is 320 x 240 pixels or less, you may be ready to click the Start Queue button now. These default settings will give you good-quality VP6 video at the same frame size and frame rate as the source.

If you need to change the settings or use more advanced settings, highlight one or more source files and click the Settings button.

Tip: If you want to encode the same source with different settings, highlight the source, click the Duplicate button, and then adjust the settings separately for each instance of the source.

Within this simple settings dialog box you can set your source to encode to any of the Flash 7 or Flash 8 presets. You can also rename the output file and adjust (or trim) the in/out points for the encoding process. If you need to adjust further, click the Show Advanced Settings button to reveal more encoding options.

The Crop and Trim tab gives you advanced options for cropping the frame as well as information on the trim settings.

Tip: To avoid any distortion when cropping video, crop according to the ratio of the source. For example, when cropping a 4 x 3 source (240 x 180, 320 x 240, and so on) crop three pixels from the height for every four pixels cropped from the width. For extremely accurate trim settings, click one of the trim tabs on the bottom of the scrubber and use the left and right keypad arrows to adjust the settings.

On the Encoding tab you can create a custom setting by making adjustments to the frame rate, keyframe interval, frame size, and the audio or video bit rate.

Tip: Never resize larger than the cropped frame size of the source.

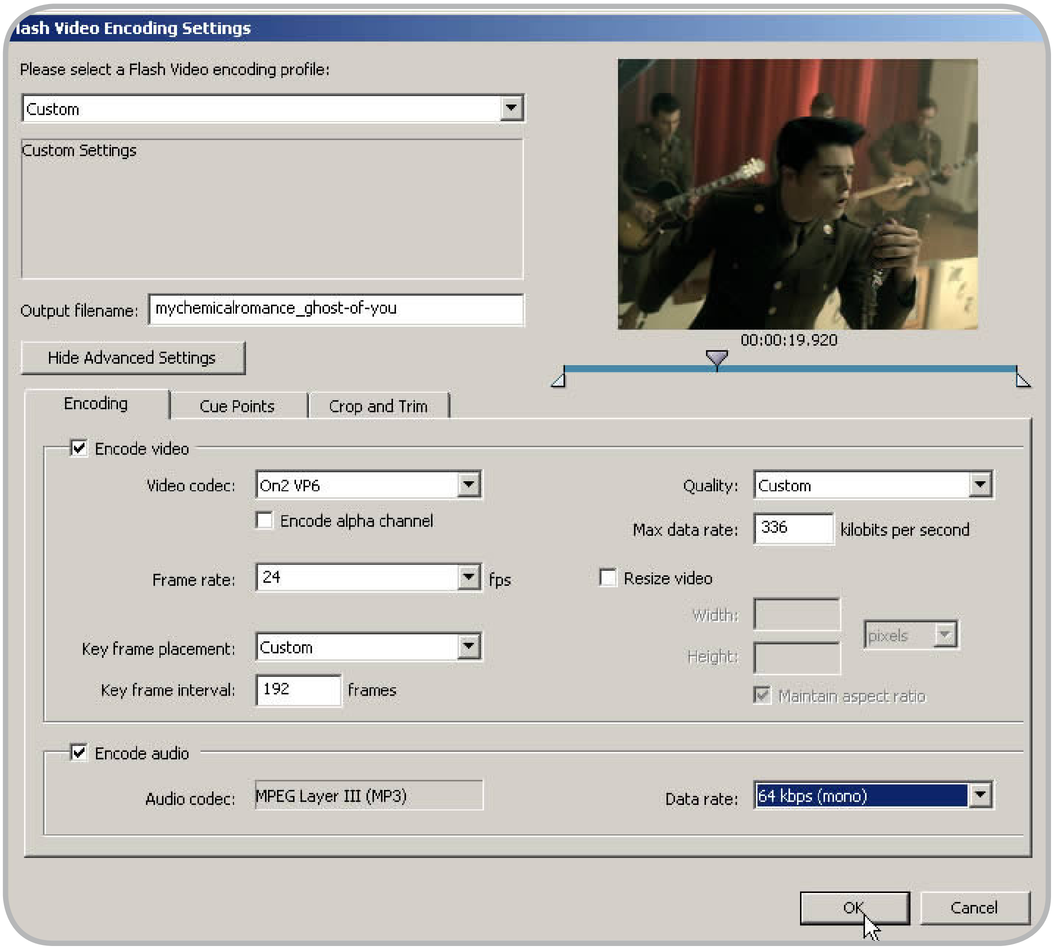
The most important variable is the amount of movement in the clip. Fast-moving music videos need a higher bit rate. Talking-head videos can be encoded using the lower bit rate settings. Adjust audio settings according to how important the audio is to your project. Music should be in stereo and go higher than 64 Kbps. Speech clips can be in mono and go as low as 16 Kbps.

Where to Go from Here

Video for the web is an exciting field, and there's much more to learn about it. You may be shooting your own video or capturing from various sources. You may need to learn more about the pre-processing methods that yield a perfect encode from a raw capture. You may be ready to build your first Flash Video player.

Many of these subjects are covered in the Flash Video Developer Center, and much more information will be coming there. Start with the Flash Video Learning Guide, which introduces Flash Video and provides you with tools to develop your skills.

William Caulfield is video encoding manager at Metro Encoding in Los Angeles. His experience in video compression goes back to 1998, a lifetime in web years. After spending three years as lead video encoder for Starmedia Networks, he started Metro Encoding in 2001 to encode video exclusively for the web. Metro Encoding currently encodes video for many of the major film studios and record labels in Los Angeles.


lash Video Encoding Settings

Please select a Flash Video encoding profile: Custom

Custom Settings

Output filename: mychemicalromance_ghost-of-you

☑ Encode video

Video codec: On2 VP6

☐ Encode alpha channel

Frame rate: 24 fps

Key frame placement: Custom

Key frame interval: 192 frames

Quality: Custom

Max data rate: 336 kilobits per second

☐ Resize video

Width:

Height:

pixels

☑ Maintain aspect ratio

☑ Encode audio

Audio codec: MPEG Layer III (MP3)

Data rate: 64 kbps (mono)

Designing and Animating Characters in Flash 8

Versatile workflow and production process

by chris georgenes

Many animators use Macromedia Flash for detailed vector-based animation. One of the strengths of Flash is its versatile workflow and production process. I have developed some custom techniques that can help you create and manage the various moving parts of a complex animated character. In this article I reveal some of my best-kept Flash secrets while taking a single character from pencil sketch to fully animated Flash cartoon.

Drawing in Flash

Macromedia Flash offers several drawing tools for creating cool characters and more. Each tool differs from the others and yields different stylistic results. Some tools create larger file sizes while other tools result in smaller files. As you become more familiar with these tools, you will learn to choose the tool that best suits the artistic style you want and any file size requirements you may have to abide by. It's simply a trade-off between file size and download times and playback performance. In this section, I'll take a look at each of the Flash drawing tools and discuss its pros and cons.

Brush Tool

The Brush tool is probably the most versatile of all the tools, especially when combined with a pressure-sensitive tablet.

Drawing with the Brush tool is the same as drawing with shapes. It's the tool that feels the most natural when used due to the support of pressure sensitivity and tilt features. Wacom makes a series of popular tablets that differ in size, include a cordless and battery-free pen called a stylus, and work great with Flash. Since a Wacom tablet can work in conjunction with your existing mouse, or replace your mouse completely, its use goes far beyond that of Flash alone. Many digital designers use a tablet with any number of graphics editors and animation programs, including Adobe Photoshop, Adobe Illustrator, and Macromedia FreeHand. But fear not: If you do not currently own a tablet, you can still use the Brush tool with one exception—both the tilt and pressure sensitivity features will not be available.

If you do have a pressure-sensitive tablet installed, selecting the Brush tool will provide you with two subselections that affect your line quality. At the very bottom of the toolbox you will notice the pressure sensitivity and tilt options. With one or both of these options selected, varied results occur when you use the Brush tool in Flash.

The result on the left is a shape—meaning, it has vector points on all sides. The more points an object has, the bigger the file size. I drew this particular shape using a pressure-sensitive tablet. Notice the tapered ends and inconsistent line weight that result with varied amounts of pressure applied to the stroke. I also drew the shape on the right with the Brush tool; in this case, however, I turned off the pressure sensitivity setting.

The Brush tool offers several subselections to help you draw certain effects:

- Paint Normal: Paints over lines and fills on the same layer
- Paint Fills: Paints inside a fill color and outside of a shape (on the Stage)
- Paint Behind: Paints behind existing fills and strokes
- Paint Selection: Enables you to paint inside a selected fill only
- Paint Inside: Keeps you from going outside the lines after you start painting inside a fill

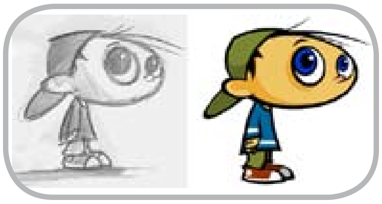
Object Drawing

Object Drawing is a new feature in Flash 8. It enables you to draw shapes as individual objects that remain independent of each other. Gone are the days when you had to draw each shape on a separate layer to avoid one shape "cutting" into another. With Flash 8 you can simply turn Object Drawing on or off as a subselection of any of the drawing tools (Brush, Pencil, Pen, Oval, and Rectangle) under Options. For those of you familiar with Macromedia FreeHand or Adobe Illustrator, this is a familiar and welcome addition to Flash.

Drawing a shape over an existing shape without Object Drawing merges the shapes together, causing one to cut into the other. This is called Merge Drawing mode. This mode can be useful for cutting into shapes to create new shapes.

When you draw a shape in Object Drawing mode, it automatically becomes an Object Drawing and cannot be merged with other shapes. You can overlay Object Drawings without fear of cutting into them. This is useful in situations where you may want to reposition objects or simply need to keep them independent.

I created my trademark boy character entirely with the mouse and the rectangle and oval tools. I never had to use my Wacom tablet in combination with the Brush tool. I could have easily chosen the Brush tool, but using the shape tools resulted in the ultra-clean line quality I was trying to achieve. Often times the Brush tool can create several unnecessary points that bloat file sizes and create headaches when trying to edit the shapes



This article originally appeared on www.macromedia.com/devnet. Reprinted with permission.

later on. For the boy graphic, I was looking for simple shapes and as small a file size as possible.

Pencil Tool

The Pencil tool is the perfect tool if you want a consistent line weight throughout your image and if you need to keep the file size as small as possible.

Lines drawn with the Pencil tool consist of fewer vector points. A perfectly straight line has only two points, one on either end. A curved line has three points or more depending on the number of curves.

The Pencil tool also offers several line quality modes:

- ***Straighten:*** This is a great option if you want to draw very straight lines by freehand. Even with a shaky hand, your lines will "snap" straight after you finish drawing your stroke.
- ***Smooth:*** If you use this option, your lines will automatically be smoothed while retaining the basic shape you intended to create.
- ***Ink:*** This option leaves your line quality unaffected. This is great for freehand or loose-style drawings.

Shape Tools

You can select the fill and stroke color through the Property inspector. Clicking either swatch opens the color panel and enables you to edit the stroke or color choice.

You can even specify no color for the stroke or fill by selecting the square with the red diagonal line through it.

Pen Tool

To draw precise paths as straight lines or smooth, flowing curves, you can use the Pen tool. Create straight or curved line segments and adjust the angle and length of straight segments and slope of curved segments.

With the Pen tool, you click to create points on straight line segments, and click and drag to create points on curved line segments. You adjust straight and curved line segments by adjusting points on the line. You can convert curves to straight lines and the reverse.
When creating curved lines you will notice that Flash creates tangent handles for each anchor point. The lines can be edited later with the subselection tool (White Arrow) by selecting an anchor and dragging the tangent handles. Moving the handles in relation to the anchor point reshapes the curve.

Onion Skin Tool

The Onion Skin tool is located at the bottom of the Timeline panel.

When you select the Onion Skin tool, the application adds Onion Skin markers to the playhead indicator of the Timeline. You can drag these markers to expand the number of frames before and after the current frame.

Flash provides two flavors of onion skinning:

- ***Normal onion skinning:*** The current frame is shown in full color while the previous and future frames are dimmed progressively. This gives an impression of a series of drawings created on semi-transparent paper and then stacked on top of each other.
- ***Onion skinning in outline mode:*** The current frame is shown in full color while previous and future frames are shown as outlines. This is easier on the eyes when working with several different keyframes before and after the current frame.

Working with Symbols

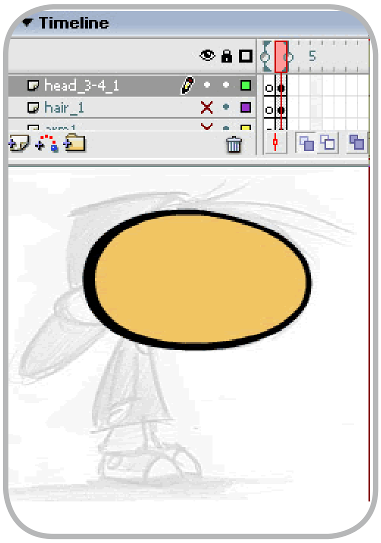
Symbols are the very essence of what makes Flash ... well, Flash. You can make anything you draw or import into a symbol. And in almost all cases you should. Here's why. When an object is converted into a symbol, it automatically becomes an item in the Flash document's library. Every Flash document has its own library from which you can drag a symbol to the Stage. When you do, the object on the Stage is now referred to as an instance. No matter how many instances of a symbol reside on the Stage, Flash only needs to load it once. This is how Flash delivers streaming animations while maintaining small file sizes. It's extremely efficient to reuse symbols as many times as possible. You can also apply effects to instances such as Scale, Tint, Alpha, and Brightness, and apply motion tweens in combination with one or more effects.

However, before I get too far ahead of myself, I'll introduce you to symbols and their behaviors.

Create an object—anything, a simple shape will do. Select it (Ctrl+A) and then convert it to a symbol by choosing Modify > Convert to Symbol or pressing the F8 key. This opens the Convert to Symbol dialog box.

In the Convert to Symbol dialog box you can type a name for your symbol, select one of three behaviors, and determine the registration point of your object. The following list explains what each behavior is and what it means:

- ***Movie clip:*** Movie clips are dynamic, which means they can be targeted with ActionScript, the Flash programming language. They can have any number of layers and frames, but their timelines are independent of all other timelines. Think of a solar system: Each planet is a movie clip, looping endlessly and independently around a sun, which is your main Timeline.
- ***Button:*** Buttons have four states: Over, Up, Down, and Hit. These are represented as keyframes in a button symbol. You can place graphics in any of these states and then apply ActionScript to the instance of a button to add interactivity to your Flash movie.
- ***Graphic:*** Graphic symbols are very similar to movie clips with the exception that they are not dynamic and cannot be targeted with ActionScript. However, you can place a graphic symbol inside a movie clip symbol. Graphic symbols can have any number of frames and layers. The most important feature is that they will always be in sync with the main Timeline and each other. This is very important when trying to create time-based animations.

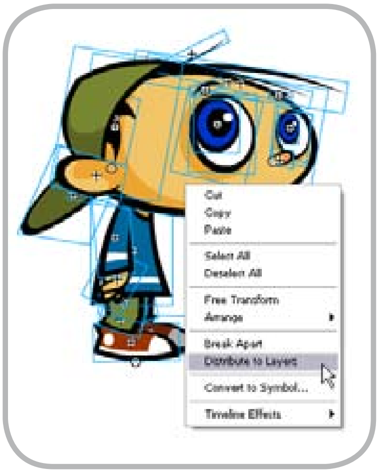


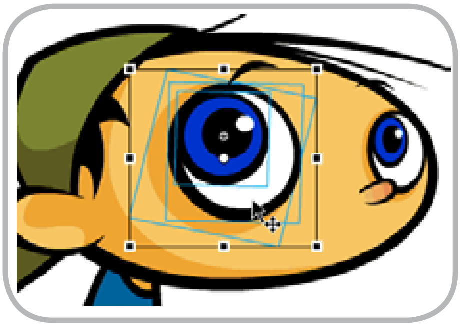
In this animation tutorial, I recommend using the Graphic behavior. This enables you to scrub the Timeline to see your animation play while inside the Flash authoring environment. Scrubbing refers to moving the playhead back and forth manually to play back the contents of the Timeline. Movie clip symbols do not play beyond Frame 1 of their contents unless you test your Flash movie (Ctrl+Enter) or export your movie as a SWF file.

Cleaning Up Your Sketches

Every good design usually starts with the time-tested pencil and paper. I've had some of my most creative moments during long and boring management seminars and sometimes a doodle on a cocktail napkin can be the inspiration for a great character. Fact is, you never know when or where you'll be inspired, so keep a pencil in your back pocket or behind your ear at all times.

You can sketch in Flash using any of the tools I already discussed. If you have drawings on paper you'd like to use as the basis for a Flash character, you'll need a scanner to scan and save them as graphics files. Most scanners come with software that will make this process easy. Some of the more widely used graphic formats that Flash can import are PNG, GIF, JPEG, TGA, and TIFF.





After you import your sketch into Flash, think about how to break it apart into individual pieces. This is the hard part. This is a very conceptual process that depends ultimately on your style of animation and the style of your character.

Form follows function, if I may borrow a phrase from Volkswagen, and this is crucial to your characters' design. You must visualize how you think your character might move, which ultimately dictates its overall design.

This character was made entirely with the Oval and Rectangle tools. Notice how I used the original sketch as a reference, making subtle changes and adjustments as I fine-tuned the image, tracing it with the Flash drawing tools.

After creating a blank keyframe (F7) next to your sketch, you can turn on the Onion Skin feature to see a ghost of your sketch. You can use any of the drawing tools to draw, using the onion-skinned sketch as your guide. I used the Oval tool to create the head of the boy character. When creating body parts and various accessories for your character, try to imagine and then draw the entire shape. You must visualize your character as if it is truly a 3D object existing in space. Even though this is a two-dimensional format, thinking like a 3D artist can help you visualize how certain parts will be attached to others.

When you finish drawing an object of your character, copy and paste it to a new layer. I like to select it and convert it to a symbol at this time. You can also name your layers.

Saving Time Placing Symbols on Layers

I always convert my objects to symbols and name them with a simple yet descriptive naming convention. For example, head1, eye1, mouth_wide, and so on. However, I do not create names for my layers at this time because there's a much easier and faster way. After all my symbols are created and named appropriately, I simply select them all (Ctrl+A) and copy (Ctrl+C) them. I now create a new layer and paste them in place (Ctrl+Shift+V). This places them all on one layer, yet still as separate symbols. Delete all your other layers so you end up with just the one layer containing all of your character symbols.

The next step is the coolest part. Select all again and then right-click over your character and select Distribute to Layers.

Voilà! Flash has not only placed each symbol on its own layer but has named each layer based on their symbol names! Nice, huh?

Saving Time Through Good Document Management

If you want to edit the name of a layer, double-click the layer name and type in a new one. Giving layers a descriptive name based on the kind of object it contains is good practice in terms of Timeline and file management. This is especially important when working with more than one artist—and even more so when working within team environments.

Since Flash MX (version 6), you've had the option to create layer folders. A layer folder is simply a new layer that acts like a folder to hold other layers. They can be expanded and collapsed. This is extremely useful when dealing with multiple layers for several characters. You can create a layer folder for each character and place ll layers inside these folders, giving you the option to collapse them all except for the character you are working on. This prevents endlessly scrolling through the Timeline and is a huge timesaver.

Another way to manage your Flash document is building an organized library. As I mentioned earlier, when you create or convert something to a symbol, it automatically becomes an object in the library of your movie. To open the library select Window > Library or press Ctrl+L. The library gives you a variety of information and options for each symbol contained in it. You can select each one by clicking a symbol name and view a thumbnail in the library preview window.

If the symbol contains an animation within it, you also see a Stop and Play button in the upper right corner of the preview window. These buttons enable you to preview the animation within this preview window.

The upper right corner of the library contains a pop-up menu with several options. You can create a new symbol, folder, font, or video. You can also rename a symbol, move symbols to folders, duplicate or delete a symbol, or edit and obtain the properties of a symbol.

Introducing "2.5D"

Just because tweening in Flash is easy does not mean you can always rely on it for creating convincing animations. Don't get me wrong: Tweening can be a great timesaver and can get the job done, as long as the job doesn't require anything more than simple objects moving around the Stage.

But what if you could push the tweening method to give more realism to your character? What if you could harness its simplicity and make it work in ways not too many other Flash users have thought of? What if you have learned everything there is to know about tweening, go back to the first 10% of that knowledge, and take a left turn? Where would that take you?

In this lesson, I'm going to reveal a truly killer Flash animation technique that will actually create what some consider a convincing 3D effect. The cool thing is you never leave the Flash environment; you remain in the 2D realm. You are now in a dimensional limbo. If it's still 2D but looks like 3D, then what exactly is it?

Welcome to the world of 2.5D animation.

Yup, 2.5D animation. I made that name up. I don't claim to have invented this technique but so far the name, two-and-a-half D, may be mine.

I have turned on the outlines feature so you can see how each part of the character's body has been separated. This is very important because motion tweens require only one symbol per layer during the tween. If you try to have two different symbols on the same layer within the same motion tween, the tween will break.

I have even taken the time to name each layer. Although this is not necessary, it does help to have layer names because, as your Timeline grows, you eventually start to scroll the Timeline more and more. It is also important when working in team environments where file sharing is common. You can never be too organized.

After your Timeline is set up, you can begin to apply motion tweens to your character. For this example, I animated only the symbols that make up the character's head, so I made sure all other layers were locked to avoid making changes to the wrong layer or layers.

Turning Heads–Part 1

First I created my keyframes where I wanted the head to start and stop turning. Click and drag vertically from the top to the bottom layer and convert to keyframes (F6). Do this twice, keeping about 15 to 20 frames in between keyframes. In my example, I made keyframes on frames 15 and 30.

Keeping the playhead on the frame at which I wanted the character's head turn to stop, I edited each of the symbols on the Stage using the Free Transform tool.

Follow my lead. I started with one of the eyes. You need to imagine the eyes wrapping around a sphere (the head). If this were truly a 3D object, one eye would get bigger as it comes closer towards you. It also wouldn't be as egg-shaped, so making it wider is also a good idea. Because my character's eye consists of three different symbols, I scaled them all at once by holding down the Shift key and selecting all three symbols. Then I used the Free Transform tool to scale them at the same time. Don't worry about being perfect at this point. You can always fine-tune your animation later. This is almost blind work, as you won't know exactly how it will look until you apply the motion tweens.

Let's move on to the nose. Because the character is turning towards us, everything closest to us should move from right to left. If it were truly a 3D object, the nose would be sticking straight out from the head. To make this a little more convincing, I turned it slightly downwards as I moved the nose from right to left. Don't forget to scale it up a bit as it is also coming closer towards the viewer.

Next I moved the other eye over while scaling it larger and making it less of an oval shape and more circular. I also moved over both eyebrows and rotated them slightly while scaling them up.

At this point I applied motion tweens to each of the layers where I made these edits. It's time to reveal what I have actually done and determine whether it needs further adjustments.

Don't worry, there's almost always a lot of tweaking to get this effect just right.

View the file mudbubble_boy.swf to see the final animated effect.

Not a bad start, is it? With a few simple edits with the Free Transform tool, you can start to see the effect come to life. It's not perfect but at this point it doesn't have to be. You can continue to tweak the amount of scaling, positioning, and skewing of your elements until you achieve the desired effect.

Turning Heads–Part 2

Next I address the remaining assets of the character's head to further emphasize his head turning.

The most important part of this illusion is convincing the viewer that this flat, 2D object has volume and mass. You also need to trick the user's eye into perceiving depth in the animation. You need to imagine this character as being built from a variety of 3D objects in space. If the character were truly turning his head in a 3D space, what is in the foreground would travel in the opposite direction from what is in the background. To help you imagine this, picture the earth spinning on its axis in space. The earth essentially spins counter-clockwise. This means the surface closest to us travels from right to left. The surface farthest from us travels from left to right. Let me apply this to my character's head.

Because the head symbol itself is simply a flat oval shape, all I really needed to do is skew it a little bit. I also chose to nudge it a little to the right. I did this because when someone turns to face you, you see more flesh to the right of the left eye. Also, because this head shape is farther back from the facial features, it should move slightly to the right a few pixels. Don't over-exaggerate some of these movements. This is an effect best executed with many subtle movements rather than a few major changes. It's the sum of its parts that make up the whole because several subtle animations complete the overall dramatic effect.

Next I rotated the brim towards the right side of the character's head. Remember, objects farther back move in the opposite direction to objects in the foreground.

I also used the arrow keys to nudge the brim over to the right so it appears to be behind the character's neck even more.

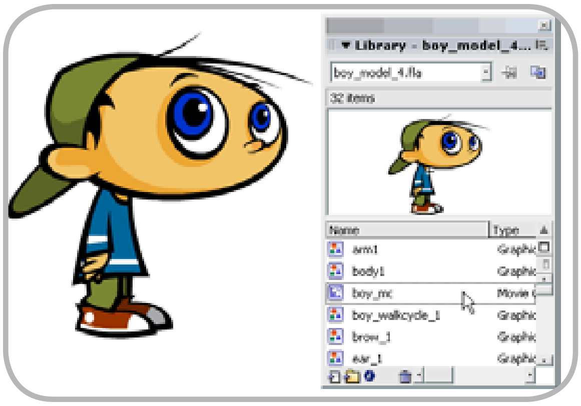
The same principle applies to his hat. I skewed the hat to make it thinner while moving it over to the right a few pixels.

The ear got skewed a bit as well as it moved upward and to the left.

Don't forget the hair. I designed the hair as three individual symbols so they could be edited independently. As the head turns towards the viewer, I skewed them down towards the character's eyes and made them shorter as if they were pointing towards the viewer a little more. This is cheating the perspective a bit and can be difficult to achieve. There is a limit to how far you can skew and scale a symbol. Remember, this effect is better when used sparingly and in a subtle way. You can only push the envelope so far. Less is more.

Next I added some motion tweens to the rest of the layers to which I made edits. If you followed my lead so far, play back the results.

Not too shabby, is it? You clearly can see his head turn as if it were a 3D object. Remember that most characters, if human, have two ears. To have the character's second ear coming around the right side of his head, create a new layer below all other layers. Copy the existing ear and paste it to this new layer in a blank keyframe midway between the motion tweens (around Frame 22).


▼ Library - boy_model_4...

boy_model_4.fla

32 items

Name Type

arm1 Graphic

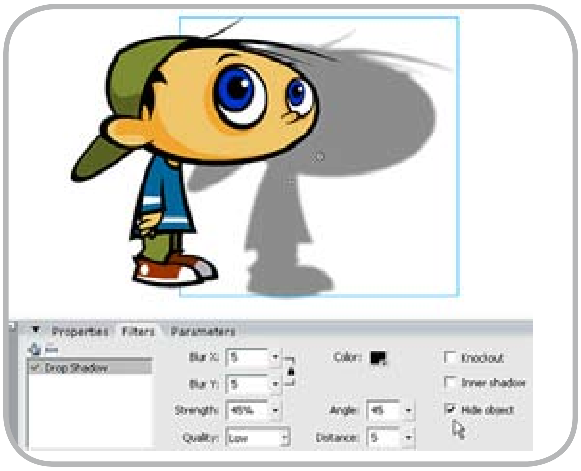
body1 Graphic

boy_mc Movie

boy_walkcycle_1 Graphic

brow_1 Graphic

ear_1 Graphic



☑ Drop Shadow

Blur X: 5

Blur Y: 5

Strength: 45%

Quality: Low

Color: ■

Angle: 45

Distance: 5

□ Knockout

□ Inner shadow

☑ Hide object

Then select Modify > Transform > Flip Horizontal to flip it.
Position the ear on the right side behind the head so it's sticking out about half way. This is about as much of the ear you need to see.

Add a second keyframe on Frame 30 (where all the other keyframes reside).

Go back to the ear's first keyframe and reposition the ear almost completely behind the head. I have converted some of the other layers to outlines so you can see the ear's new position.

Now just apply a motion tween. Play back your animation to see the ear reveal itself as the head "turns" towards you. This small detail adds an enormous amount of realism to the head turn as it helps create the illusion of the head being a rounded sphere (this requires Flash Player 8).

Feel free to scrub the Timeline and make adjustments to your symbols if something doesn't feel quite right. This depends on how much of a perfectionist you are as well as on the complexity of your character. The boy character in this example is fairly complex, given the amount of objects I used in his design.

Custom Ease In/Ease Out Panel (Flash Professional 8 Only)

I'd like to add one more finishing touch to the head turn. As you know, motion tweens can look pretty static due to their constant rate of speed. They can look pretty jerky in their initial state. Thanks to a new feature in Flash Professional 8, however, you can add some "easing in and easing out" to make the transition softer. The trick is how to ease in and out inside of a single tween.

The first step is to apply the motion tween. You can do this in one of two ways:

- Right-click the frames in between the two keyframes. From the pop-up menu, select Create Motion Tween.
- Select all layers with the selection tool by clicking and dragging, highlighting them all in black. From the Tween drop-down menu in the Property inspector, select Motion.

Notice the Edit button in the Property inspector.

Clicking this button opens the new Custom Ease In/Ease Out panel.

With the new Custom Ease In/Ease Out panel you have a lot more control over easing than you did with previous versions. The Custom Ease In/Ease Out panel displays a graph that represents the degree of motion over time. The horizontal axis represents the frames, while the vertical axis represents the percentage of change to the object.

The graph's curve dictates the rate of change of the object. When the curve is horizontal (no slope), the velocity is zero; when the curve is vertical, you don't have any easing or delay to the objects' movement.

If you select the Use One Setting for All Properties check box, the current curve is applied to all properties (Position, Rotation, Scale, Color, Filters). If you deselect this check box, you can apply a separate curve to each of the properties, as follows:

- ***Position:*** Specifies custom ease settings for the position of an animated object on the Stage.
- ***Rotation:*** Specifies custom ease settings for the rotation of an animated object. For example, you can fine-tune how quickly or slowly an animated character turns around to face the user on the Stage.
- ***Scale:*** Specifies custom ease settings for the scale of an animated object. For example, you can customize the scale of an object more easily so it appears to be moving away from the viewer, then coming closer, and then moving away again.
- ***Color:*** Specifies custom ease settings for color transitions applied to an animated object.
- ***Filters:*** Specifies custom ease settings for filters applied to an animated object. For example, you can control the ease setting of a drop shadow that simulates a change in the direction of a light source.
- ***Play and Stop buttons:*** Lets you preview an animation on the Stage using all the current velocity curves defined in the Custom Ease In/Ease Out dialog box.
- ***Reset button:*** Lets you reset the velocity curve to the default, linear state.

Click the diagonal line once to add a new control point. You can precisely control the motion of an object by dragging the positions of the control points.

Using frame indicators (represented by square handles), you click where you want an object to ease in or ease out (accelerate or decelerate). Clicking the handle of a control point (the square handles) selects that control point, and displays the tangent points (hollow circles) on either side of it. You can use the mouse or the keyboard's keys to drag the control or tangent points to new positions.

Tip: By default, the control points snap to a grid. You can temporarily turn this off by holding down the X key while dragging the control point.

Clicking on the curve away from any control points creates a new control point to the curve. Clicking anywhere away from the curve and control points deselects the control point that is currently selected.

To make an animation start gradually and end gradually. I highly recommend experimenting by creating and editing control points and tangents and playing back your animation using the Stop and Play buttons in the lower left corner of the panel. It won't be long before you have a good feel for the relationship between the curve and its effect on the animation.

Now play back your animation. Notice how the head-turn eases in and then out as it approaches its end. To animate the head turning back again to its original position, you only need to copy frames from Frame 1 and paste them after the tween you just created. Repeat the same procedure by adding a motion tween, and for easing in and out and you have a character that can turn its head back and forth in a cool way.

Applying Filters

Yes, the impossible is now possible in Flash. Flash 8 introduces graphic filters and blend modes, which the Flash community has accepted with open arms. You can now apply effects such as Drop Shadow, Blur, Glow, Bevel, Gradient Glow, Gradient Bevel, and Adjust Color to any movie clip instance. So have fun with your character by applying a drop shadow.

The first thing to do is place your character in a movie clip symbol. In the Timeline, click and drag to select all frames across all layers. (Hint: start on Frame 1 in the upper left corner and drag diagonally to the last frame on the bottom layer.) This highlights your frames and layers in black. Select Edit > Timeline > Copy Frames (or press Ctrl+At+C) to copy the entire Timeline to the Clipboard. (You could also right-click anywhere on your highlighted frames and select Copy Frames from the context menu.) Now hold that thought until the next step.

Open your Library (Ctrl+L) and from the upper right drop-down menu, select New Symbol > and then select Movie Clip behavior and click OK. You are now in Edit mode for this new symbol. Select the first frame of the Timeline and select to Edit > Timeline > Paste Frames (or press Ctrl+Alt+V) to paste your selection. (You could also right-click over Frame 1 and select Paste Frames from the context menu.)

Return to the main Stage and create a new layer. Locate the movie clip symbol you just created in the Library and drag it to the Stage on this new layer.

Delete all other layers that contain your original character. At this point your Timeline should have one layer and one movie clip symbol containing your character animation.

There are two kinds of shadows you can create. A standard drop shadow creates a shadow below the instance that is slightly offset from the original. This type of effect is often very flat and similar to a shadow cast against a wall or any other flat surface. It is not very convincing if you are trying to achieve a shadow cast on the ground. For a more realistic shadow effect, you need to add a few simple steps.

Duplicate the movie clip instance and place it behind the original instance. (You can keep the duplicate on the same layer and arrange it behind the original instance, or paste it to a new layer below the original.)

Select the duplicate instance and, in the Filters tab of the Property inspector, select Drop Shadow from the Plus (+) menu. Next, adjust the strength level to around 45% and select Hide Object. This hides the movie clip object but reveals the Drop Shadow Filter properties. You can make further adjustments by setting Quality to Low, Medium, or High, or modifying the amount of blur, color, angle, or distance of the blur effect from the original graphic.

To prevent this effect from looking too much like a drop shadow against a flat wall behind the original object, use the Free Transform tool to skew the filtered movie clip instance. This technique requires some trial and error before you can apply it successfully. Squash and skew the instance until it looks as if the shadow is cast on a floor. You may need to reposition the instance to make this look right.

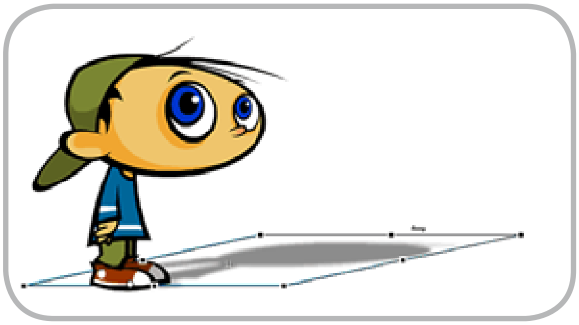
Test the movie and see the effect come to life.

When combining looping animations with several other looping animations, you can achieve some sophisticated visual results, (this requires Flash Player 8).

In this sample you can see how the drop shadow effect adds an exciting dimension to the animation. This animation is a series of movie clips containing looping animations of the boy and dog characters. In a future article I will explain how small animated loops can yield a lot of mileage to your movies.

I hope this article has helped you sharpen your Flash design and animation skills. There are several more animation techniques that you can achieve with Flash, such as full animation (often referred to as frame-by-frame), shape tweens, and image sequences from video or 3D animation programs. There are also a number of cool plug-ins available from other designers and animators that can help speed up your Flash workflow. Look for future articles that explore these techniques and tools.

Chris Georgenes is a full-time freelance artist, animator, and all-around designer for the web, CD-ROM, and television. His clients include Pileated Pictures, LucasArts, Universal Records, Plot Developers, and AOL, among others. He maintains www.mudbubble.com as his online portfolio and www.keyframer.com as his Flash tutorial website. Chris is also a member of Team Macromedia.



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
SEMINAR

DoubleTree Hotel — **April 24, 2006**
San Jose, CA

The Roosevelt Hotel — **June 5-6, 2006**
New York, NY

AjaxWorld Conference & Expo — Rich Internet Applications — **Oct 9-10, 2006**
San Francisco, CA

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...

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

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.

Jeremy Geelan

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

Register at...
www.AJAXSeminar.com

Before April 24, 2006 (San Jose, CA) (On-site Price: $1295) Call for group discounts	**$995**
Early Bird Price (New York, NY*) (Before March 31, 2006) Call for group discounts	**$1,295**
Discounted Price (New York, NY*) (After March 31, 2006) Call for group discounts	**$1,495**
On-Site Price (New York, NY*) (On June 5-6, 2006) Call for group discounts	**$1,695**

*Includes SOA Web Services Edge plus Enterprise Open Source Conference & Expo

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

SPONSORED BY Adobe telerik BACKBASE TIBCO Perficient Microsoft

AND XML ASYNCHRONOUS JAVASCRIPT AND XML ASYNCHRONOUS JAVASCRIPT AND XML ASYNCHRONOUS JAVASCRIPT AND XML ASYNCHRONOUS JAVASCRIPT AND XML ASYNCHRONOUS JAVASCRIPT AND XML ASYNCHRONOUS JAVASCRIPT AND XML ASYNCHRONOUS JA-

What you'll walk away with...

A New understading of AJAX and how to make it work for you, plus:

A) **Conference Proceedings Binder**
Full-conference proceedings in a self-contained commemorative binder.

B) ***Ajax in Action* Book**
A copy of the Best-Selling AJAX book by Dave Crane.

C) ***Real-World AJAX "Secrets of the Masters"* Book + DVD**
New AJAX book edited by Dion Hinchcliffe (Release Date: Summer 2006)

D) **Notebook**
Executive style keeps you looking sharp for note-taking wherever your business might take you!

E) **Computer Back-Pack**
A hip new way to conceal your laptop! The Urban Wonder Compu-Pack is an ingenious creation - a 600-denier polycanvas and nylon knapsack!

F) **T-Shirt**
100% preshrunk 6.1 oz. heavyweight cotton tagless shirt features shoulder-to-shoulder tape, and double-needle sleeves and bottom.

G) **Porcelain Mug**
Lightweight and durable, you can use it as a teacup, coffee mug or keep it as a pencil holder.

H) **Satin Silver Contemporary Pen**
Satin-silver plastic barrel with the look and feel of brass. Rubber comfort grip improves writing control and comfort.

I) **Real-World AJAX Seminar on DVD**
Watch complete seminar video and slide presentations (Release Date: Summer 2006)

Featuring... Real-World AJAX Rock Stars! FACULTY

Jesse James Garrett (San Jose April 24)
Father of "AJAX" Who Coined the Term in 2005

Jesse James Garrett is the Director of User Experience Strategy and a founding partner of Adaptive Path, the world's premier user experience consulting company. He is author of The Elements of User Experience (New Riders), and is recognized as a pioneer in the field of information architecture. Jesse's clients include AT&T, Intel, Crayola, Hewlett-Packard, Motorola, and National Public Radio. Since starting in the Internet industry in 1995, Jesse has had a hands-on role in almost every aspect of Web development, from interface design and programming to content development and high-level strategy. Today, information architects around the world depend on the tools and concepts he has developed, including the widely acclaimed "Elements of User Experience" model. He is co-founder of the Information Architecture Institute, the only professional organization dedicated to information architecture. He is also a frequent speaker and writer whose work has appeared in numerous publications, including New Architect, Digital Web, and Boxes and Arrows.

Adam Bosworth (San Jose April 24)
Vice President of Engineering, Google
One of the Fathers of XML & the Creator of MS Access

Adam Bosworth is Vice President of Engineering, Google. He joined Google in 2005 from BEA Systems, where he was Chief Architect & Senior VP of Advanced Development. Prior to joining BEA, Bosworth co-founded Crossgain, a software development firm acquired by BEA. Known as one of the pioneers of XML, he previously held various senior management positions at Microsoft, including General Manager of the WebData group, a team focused on defining and driving XML strategy. While at Microsoft he was also responsible for designing and delivering the Microsoft Access PC Database product and assembling and driving the team that developed the HTML engine of Internet Explorer 4.0.

Dion Hinchcliffe (San Jose April 24)
Cofounder & CTO, Sphere of Influence Inc.
Editor-in-Chief, Web 2.0 Journal

Dion Hinchcliffe, newly appointed editor-in-chief of SYS-CON's pioneering Web 2.0 Journal, is cofounder and chief technology officer for the enterprise architecture firm Sphere of Influence Inc., in McLean, Virginia. A veteran of software development, Dion works with leading-edge technologies to accelerate project schedules and raise the bar for software quality. He is highly experienced with enterprise technologies and he designs, consults, and writes prolifically. Dion actively consults with enterprise IT clients in the federal government and Fortune 1000. He is a frequent speaker on AJAX, Web 2.0 and SOA and is currently the top-read SYS-CON.com blogger.

Christophe Coenraets (San Jose April 24)
Senior Technical Evangelist, Adobe
AJAX/Flex Integration Guru

Christophe Coenraets currently works as a Senior Technical Evangelist at Adobe. Before joining Adobe, Christophe was an evangelist at Macromedia, focusing on Rich Internet Applications and Enterprise integration. Prior to Macromedia, Christophe was the head of Java and J2EE Technical Evangelism at Sybase, where he started working on Java Enterprise projects in 1996. Before joining Sybase in the US, Christophe held different positions at Powersoft in Belgium, including Principal Consultant for PowerBuilder, and Manager of the Professional Services organization. Before joining Powersoft, Christophe worked as a developer and architect on several retail and BPM projects. Christophe has been a regular speaker at conferences worldwide for the last 10 years.

Paul Rademacher (San Jose April 24)
Google, Creator of HousingMaps.com

Paul Rademacher is the creator of HousingMaps.com, which combined Craigslist and Google Maps for the first web mashup. Paul holds a Ph.D. in Computer Science from UNC-Chapel Hill, and worked as an R&D Engineer at Dreamworks Animation on such movies as Shrek 2 and Madagascar. Since creating HousingMaps, Paul is now at Google.

Jouk Pleiter (San Jose April 24)
Co-Founder & CEO of Backbase

Jouk Pleiter is the CEO of Backbase, a leader in the field of Rich Internet Applications and AJAX development software. Backbase's clients include ING, ABN AMRO, TNT, KPN, Comsys and Heineken. Backbase operates globally with offices in San Mateo (North America) and Amsterdam (Europe). Since 1995, Jouk has been an entrepreneur: he founded three successful software companies. Prior to Backbase, Jouk was part of the founding team at the web content management company Tridion, where he led the product management operations, and was driving the company's efforts to become a leader in the European WCM software market. Jouk previously was part of the founding team at the Interactive Agency Twinspark where he grew the company to a leading market position in Europe and was instrumental in the sale of Twinspark to Agency.com. He has an MBA from the University of Groningen.

Kevin Hakman (San Jose April 24)
Director of Product Marketing for TIBCO
General Interface TIBCO Software

Kevin Hakman is the director of product marketing for TIBCO General Interface, the award winning AJAX and Rich Internet Application framework and toolkit. Kevin Hakman pioneered AJAX in the enterprise co-founding General Interface in 2001. Since that time General Interface (aka 'GI') has been powering Web applications that look, feel and perform like desktop applications, but run in the browser at Fortune 500 and U.S. Government organizations. General Interface was also the first to use its own toolkit to provide full visual tooling for AJAX when it released it's 2.0 Version in 2003. TIBCO acquired General Interface in 2004 to extend its vision for service oriented applications to the end user. Kevin is a contributor to the SOA Web Services Journal and the AJAX Developer's Journal.

Shanku Niyogi (San Jose April 24)
Product Unit Manager of the UI Framework
and Services Team Microsoft Corporation

Shanku is Product Unit Manager of the UI Framework and Services (UiFX) team, which is responsible for delivering high-productivity UI framework technologies for the .NET platform, including ASP.NET, Atlas, Windows Forms, and frameworks for smart clients. Prior to his current role, Shanku was Group Program Manager of the Web Platform and Tools team on the Whidbey release of ASP.NET and Visual Web Developer. Shanku joined Microsoft in 1998 as a developer, having spent several years shipping products in the Windows ISV industry. Shanku holds a Bachelor of Mathematics degree in Computer Science from the University of Waterloo.

Coach Wei (New York June 5-6)
Chairman, Founder and CTO, Nexaweb
The Creator of First Commercial AJAX Applications

Coach Wei combines in-depth IT industry expertise with extensive education and research experience at MIT to drive technology innovation and business direction for Nexaweb. He founded Nexaweb in 2000 and served as CEO until summer 2003. Before founding Nexaweb, Coach architected and designed enterprise software for managing storage networks at EMC Corporation. As a graduate researcher at MIT, Coach developed software and hardware systems for non-destructive evaluation as well as signal/image processing algorithms. Coach was a finalist in the 1999 MIT $50K entrepreneurship competition and holds several U.S. patents. An accomplished writer and speaker, Coach has published numerous articles on topics including: AJAX, J2EE and .NET, RIA development, XML, signal/image processing, composite materials and ultrasonic imaging. He has spoken at top industry events, such as JavaOne and Web Services Edge. Coach holds an MS in information technology from MIT.

Ajit Jaokar (New York June 5-6)
CEO, futuretext
Author, "Mobile Web 2.0"

Ajit Jaokar, based in London (England), is the CEO of a publishing company, futuretext (www.futuretext.com). He is currently writing a book about Mobile Web 2.0 (Mobile Web 2.0: The Innovator's Guide to Developing and Marketing Next Generation Wireless / Mobile Applications). Ajit also chairs Oxford University's Next-Generation Mobile Applications Panel and, since January 2006, has been a member of the Web 2.0 Workgroup. In his "Real-World AJAX" conference session, Ajit will discuss the "AJAX Use in Mobile Applications" as part of the wider impact of Web 2.0, sometimes referred to as the "Global SOA."

Jonas Jakobi (New York June 5-6)
AJAX Evangelist and Co-Author, "Ajax and JSF: Friend or Foe?"
Jonas will autograph a copy of his book for all delegates!

Jonas Jacobi is a principal product manager and evangelist for Oracle's Java/J2EE tool offering, JDeveloper, and over the past three years has been responsible for JavaServer Faces, Oracle ADF Faces, and Oracle ADF Faces Rich Client development features within Oracle JDeveloper. Jonas has been in the software business for 15 years. Prior to joining Oracle, he worked at several software companies in Europe, covering many roles including support, consulting, development, and project team leadership. Jonas' new book "Ajax and JSF: Friend or Foe?" released by Apress on February 25, 2006.

John Fallows (New York June 5-6)
AJAX Evangelist and Co-Author, "Ajax and JSF: Friend or Foe?"
Jonas will autograph a copy of his book for all delegates!

John Fallows, former lead developer for Oracle ADF Faces Rich Client, has been working in distributed systems for over a decade. After five years spent focused on designing, developing the JavaServer Faces standard to provide AJAX functionality, playing a leading role in the Oracle ADF Faces team, he recently joined an AJAX start-up. Originally from Northern Ireland, John graduated from Cambridge University in the United Kingdom and has worked in the software industry for more than ten years. Prior to joining Oracle, he worked as a research scientist for British Telecommunications Plc.

Steve Benfield (New York June 5-6)
Well-known AJAX Evangelist and CTO of Agentis Software
Steve's first talk on "Aspect-Oriented Programming & AJAX"

Steve Benfield is CTO of Agentis Software and one of the pioneers of AJAX technology, a gifted writer and a technical visionary. A technology marketeer and strategist with 20 years of software entreprenuerism experience, a combination of qualities that made him the perfect choice of editor-in-chief for SYS-CON Media's inaugural publication 12 years ago. Steve's proven ability to determine marketing and technology strategies that align with market needs led to successful stints at SilverStream, where he started as technology evangelist and ended as CTO, and at ClearNova, an open source AJAX company, where he was CTO and AJAX evangelist.

Jeremy Geelan
Conference Chair
Group Publisher & Editorial Director, SYS-CON

Jeremy Geelan is group publisher and editorial director of SYS-CON Media, and is responsible for all print titles and online i-technology portals for the firm. He regularly hosts SYS-CON.TV, is executive producer of the "Power Panels with Jeremy Geelan" iTV series, and represents SYS-CON at conferences and trade shows, speaking to technology audiences both in North America and overseas. His i-Technology Blog is at jeremy.linuxworld.com and he is conference chair of the upcoming iTVcon - "Internet TV Conference & Expo 2006".

For more information...

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

SYS-CON EVENTS **For more great events visit www.EVENTS.SYS-CON.com**

NOTE: SPEAKER LINE-UP SUBJECT TO CHANGE WITHOUT NOTICE
VISIT WWW.AJAXSEMINAR.COM FOR THE MOST COMPLETE UP-TO-DATE INFORMATION

Director and SCORM 1.3 SCORM SCO Presentation Engine (S2PE)

Sharable Content Object Reference Model

by kraig mentor

The Sharable Content Object Reference Model (SCORM ®) Shareable Content Object (SCO) Presentation Engine (S2PE) is a SCORM content presentation application prototype, which allows content to be abstracted from the playback mechanism. The engine removes the burden of programming Learning Management System (LMS) communication from the content author by automatically handling all communication for them. Rather th an programming content, the author describes content through Extensible Markup Language (XML) files, which the engine interprets to render the content. The engine is robust in its support for graphics, text, video, audio, 3D, Shockwave, and Flash content. The engine also supports quizzes, interactivity, and synchronization of events.

Preface

After the September 11 attack against the United States, the U.S. Department of Defense initiated a pilot university program for first responders. The initial classroom course, created as a collaboration between the U.S. National Guard, Indiana University of Pennsylvania, and Concurrent Technologies Corporation, was a three week paperwork-intensive program covering topics related to chemical, biological, radiological, nuclear and high-yield explosive attacks. Post review of the pilot program indicated the need to bring the students least familiar with the topics to a higher understanding prior to entering the classroom for advanced work.

As part of the Department of Defense (DoD), the National Guard delivers learning material through a distributed LMS learning network based on the SCORM standard. These initial lessons potentially would cover thousands of pages of material. Traditional methods of SCORM content creation would require hand building each lesson with HTML, Director, Flash, and so forth. As a proof of concept, the SCORM SCO Presentation Engine (S2PE) was created using Macromedia Director from Adobe. The purpose of the concept was to demonstrate that individuals could create engaging, voluminous SCORM 1.3 content quickly with little to no programming skills.

Introduction

This article presumes the reader has a basic understanding of SCORM. For specific information about SCORM visit www.adlnet.org.

The introduction of SCORM as a delivery mechanism for learning content has created many new challenges for developers and instructional designers. Developing and implementing even straightforward content has become difficult for many individuals and groups who are otherwise competent at their craft. The ability to understand and merge the necessary multiple technologies is a requirement that not all are able to match. These factors slow the implementation and acceptance of SCORM outside of those groups who are mandated to use it.

There are limited existing applications that can be of assistance in addressing areas of authoring content for SCORM. In the commercial software arena, most companies have only stuck a toe in the SCORM waters to test it out. Presently there is no complete development tool for the creation of SCORM content. No product addresses LMS communication and content authoring.

To facilitate the growth of SCORM and the spread of the LMS to the point of being as common as the web browser, users need simplified way to create content. Thus, the S2PE was created, using Director, as a proof of concept to meet that need.

The S2PE engine allows content authors to describe SCO content through associated XML files. Interpretation of those files, content layout, and all LMS communication is automatically handled by the presentation engine. The content is described external to the application, creating a natural separation from the learner interface.

As the application stands alone, anyone with nominal XML skills can create SCORM content successfully. Instructional Designers can now implement their vision without relying on a programmer. Additional benefits are rapid creation times for content, simple content updates, and reduced downloads for the learner.

Presentation Engine

The Presentation Engine (PE) is lightweight at about 100 Kbytes. This includes varying levels of support for graphics, 3D, text, audio, video, Flash, and Shockwave. The additional custom JavaScript libraries, necessitated by SCORM, add an additional one time download of another 100 Kbytes.

The PE is divided into two parts. The Player Module (PM) is responsible for determining the order of play and intra-

This article originally appeared on www.macromedia.com/devnet. Reprinted with permission.

SCO branching. The Content Module (CM) is responsible for LMS communications and displaying content to the learner.

During initial loading of a SCO, the PM is loaded into an empty HTML shell page. This shell page contains a placeholder where the Shockwave file is eventually loaded. The scr tag for the HTML shell page is initially set to an empty string value (Listing 1).

```
<param name="src" value="">
<embed src="" ... >
```

When the shell page is loaded, the manifest item's parameter tag is used to point to the Shockwave PM dcr file for loading into the scr placeholder tag. In this example, the filename of the PM is player.dcr (Listing 2).

```
<item identifier="ITEM-9298b6c0-
a695-11d9-9" identifierref="intro"
parameters="?dcr=player.dcr&xml=:::
pl_intro.xml">
     ...
     ...
</item>
```

The learner initiates the above sequence of events by making a SCO selection from the LMS Table of Contents (TOC).

In much existing SCORM content, it is not unusual to find a unique HTML file for each page within a SCO. The approach presented here allows all screens for every SCO to share this single HTML page. This can eliminate hundreds or even thousands of HTML files from needing to be authored and downloaded for a SCO. The result is simplified development. Since the PM is a dedicated playback file common to all SCOs, it is reasonable to ask why its path would need to be specified for each manifest item's parameters. While it is true that the playback file remains the same for each SCO, there may in fact be a situation where a unique PM file will need to be written that has capabilities beyond or completely unrelated to the generic PM. Although including an identical path to the PM dcr may seem redundant, it leaves the door open to accommodate alternative scenarios should they arise.

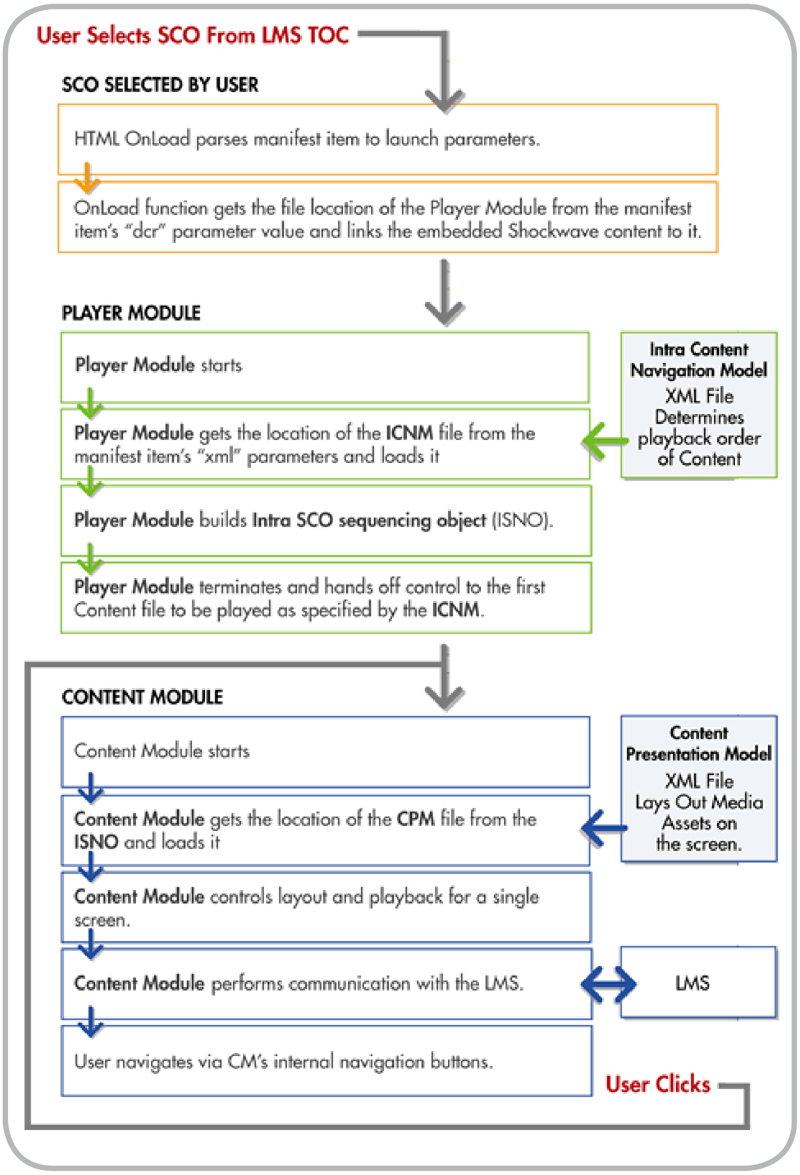
Once the PM is loaded into the shell it reads in the second manifest item's parameter, an XML file path. This XML file is referred to as the Intra Content Navigation Model (ICNM) (Listing 3).

```
<item identifier="ITEM-9298b6c0-
a695-11d9-9" identifierref="intro"
parameters="?dcr=player.dcr&xml=:::
pl_intro.xml">
     ...
     ...
</item>
```

This ICNM contains the sequence of screens that comprise the selected SCO. The sequence of screens to be delivered is called a playlist. For example, the simplest of all SCOs has but a single entry in its playlist. This equates to a SCO having only one screen (Listing 4). Once the PM has successfully loaded the playlist, an internal navigation structure is built and used for intra-SCO navigation. When that navigation structure is completed, the PM automatically loads the content of the first item in the playlist for presentation. For the moment, the important point is that for each item in the playlist, there is one corresponding screen presented to the user. I explore how that happens later.

```
<playlist type="linear">
    <item dcr=":::content.dcr"
xml=":::intro_01.xml"/>
</playlist>
```

A more complicated SCO might have several screens to present to the user (Listing 5). In this case, each time the learner selects the SCO's internal Next button, the next item in the list is loaded. The playlist would present the user with four successive screens for this SCO.


User Selects SCO From LMS TOC
SCO SELECTED BY USER
HTML OnLoad parses manifest item to launch parameters.
OnLoad function gets the file location of the Player Module from the manifest item's "dcr" parameter value and links the embedded Shockwave content to it.
PLAYER MODULE
Player Module starts
Player Module gets the location of the ICNM file from the manifest item's "xml" parameters and loads it
Player Module builds Intra SCO sequencing object (ISNO).
Player Module terminates and hands off control to the first Content file to be played as specified by the ICNM.
Intra Content Navigation Model XML File Determines playback order of Content
CONTENT MODULE
Content Module starts
Content Module gets the location of the CPM file from the ISNO and loads it
Content Module controls layout and playback for a single screen.
Content Module performs communication with the LMS.
User navigates via CM's internal navigation buttons.
Content Presentation Model XML File Lays Out Media Assets on the screen.
LMS
User Clicks

```
<playlist type=”linear”>
     <item dcr=”:::content.dcr”
xml=”:::intro_01.xml”/>
     <item dcr=”:::content.dcr”
xml=”:::intro_02.xml”/>
     <item dcr=”:::content.dcr”
xml=”:::intro_03.xml”/>
     <item dcr=”:::content.dcr”
xml=”:::intro_04.xml”/>
</playlist>
```

While this is a straightforward concept for linear presentations, a complete solution needs to include SCO screens, which contain buttons allowing branching within the SCO. In fact, the implementation of the playlist allows the embedding of playlists within playlists to n levels (Listing 6).

In the code sample below, the learner views the content of XML file 01.xml, then proceeds to 02.xml. At that screen there are two navigation buttons, each taking the user down a different path. See the button identifier in the the playlist tag as button=”A”, and so forth. When the button attribute is included in a playlist tag, the parent screen automatically renders the necessary navigation controls. You do this through the CM, which I discuss later.

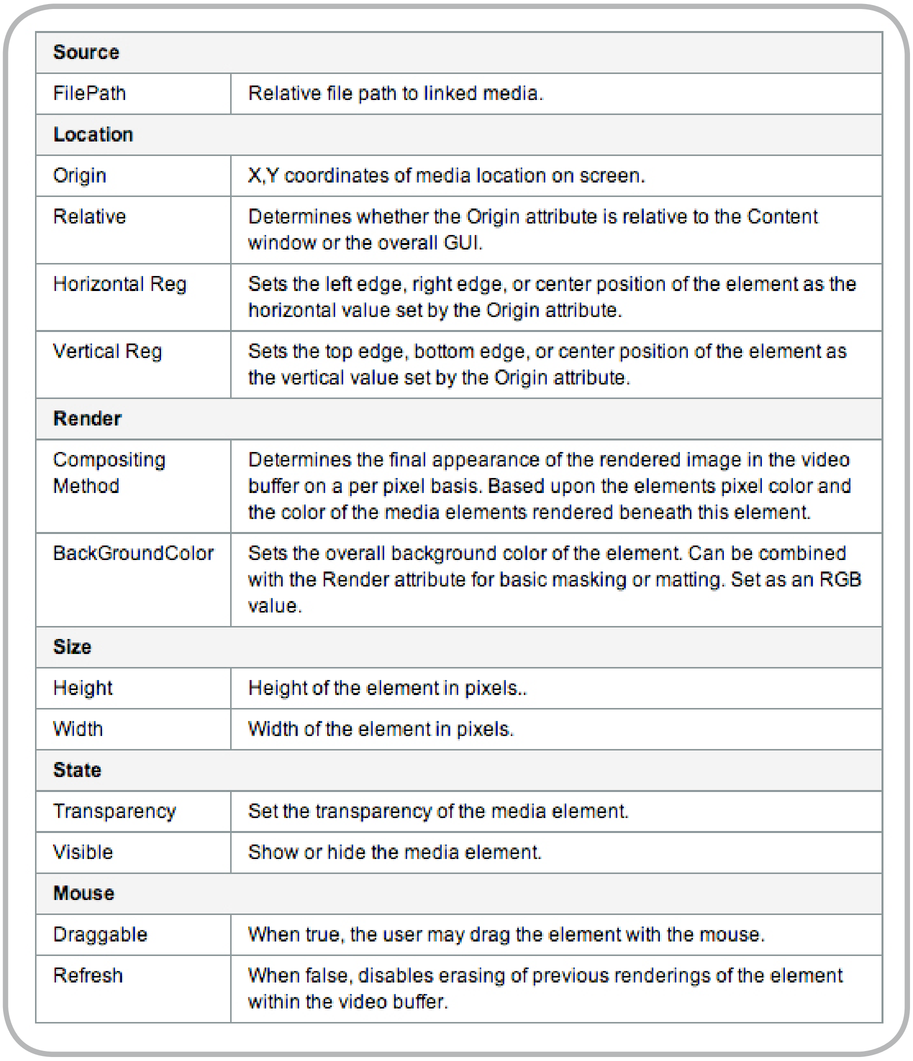
```
<playlist type=”linear”>
     <item dcr=”:::content.dcr”
xml=”:::01.xml”/>
     <item dcr=”:::content.dcr”
xml=”:::02.xml”>

          <playlist type=”linear”
button=”A”>
               <item dcr=”:::content.
dcr” xml=”:::03.xml”/>
               <item dcr=”:::content.
dcr” xml=”:::04.xml”/>
          </playlist>

          <playlist type=”linear”
button=”B”>
               <item dcr=”:::content.
dcr” xml=”:::09.xml”/>
               <item dcr=”:::content.
dcr” xml=”:::10.xml”/>
               <item dcr=”:::content.
dcr” xml=”:::11.xml”/>
          </playlist>

 </item>

 <item dcr=”:::content.dcr”
xml=”:::05.xml”/>
     <item dcr=”:::content.dcr”
xml=”:::06.xml”/>
</playlist>
```



Source	
FilePath	Relative file path to linked media.
Location	
Origin	X,Y coordinates of media location on screen.
Relative	Determines whether the Origin attribute is relative to the Content window or the overall GUI.
Horizontal Reg	Sets the left edge, right edge, or center position of the element as the horizontal value set by the Origin attribute.
Vertical Reg	Sets the top edge, bottom edge, or center position of the element as the vertical value set by the Origin attribute.
Render	
Compositing Method	Determines the final appearance of the rendered image in the video buffer on a per pixel basis. Based upon the elements pixel color and the color of the media elements rendered beneath this element.
BackGroundColor	Sets the overall background color of the element. Can be combined with the Render attribute for basic masking or matting. Set as an RGB value.
Size	
Height	Height of the element in pixels..
Width	Width of the element in pixels.
State	
Transparency	Set the transparency of the media element.
Visible	Show or hide the media element.
Mouse	
Draggable	When true, the user may drag the element with the mouse.
Refresh	When false, disables erasing of previous renderings of the element within the video buffer.

Regardless of the playlist structure, the PM automatically loads the first item in its playlist. It does so by passing control of playback to the CM. The PM is released and the CM is in control for the duration of the SCO’s life. It’s important to the content author that he or she can add content screens to a SCO without rewriting or recompiling any code. With just an XML editor or a simple text editor, all the desired changes can be made and instantly reviewed within the LMS environment.

The CM is responsible for presenting the content to the learner and handling any intra-SCO navigation. Each screen the user sees is a new instance of the CM. The CM renders everything that the learner sees on the screen. The CM loads its layout and content information from an external, XML-based Content Presentation Model (CPM) file. The CPM file specifies content layout information for a single screen. The CPM contains information about text, graphics, video, positioning, fonts, and so forth

An entry in the CPM file might specify an image to load and the location on the screen to render the image. Or, for text, a CPM entry could specify the text string, its font, leading, color, and so forth. During playback, the CM is responsible for communicating with the LMS. If a designer had placed an image at a horizontal location of 300 pixels but later decides it should be at 290 pixels, there is no need to open Director. Just edit the appropriate XML source file and the change is automatic.

Sharing the playback engine across all SCOs ensures repeatability and control. Updates to the PE are universal and immediately reflected in all SCOs upon

Enter the premier ColdFusion Conference

CFUNITED is the only conference of its kind that is run by developers, for developers. What this means is that speakers aren't dictated what they can and cannot talk about and all of the speakers are encouraged to base their sessions on real world experience. There are real users in the case studies so you can see what your colleagues at other organizations are doing. It also means that the marketing fluff stops at the door – so you can learn more from other developers!

www.cfunited.com

> *"I attended CFUNITED along with a number of my colleagues from the university. Some were experienced developers, some were novices, some were IT managers or server admins. The consistent thread of all our post-conference conversations regarding CFUNITED was the number of ideas we came away with. Whether it was a new solution to an old problem or just having the possibilities of ColdFusion opened up before us, we all felt it was well worth the trip. Of course our blood sugar levels after eating that CF birthday cake might have contributed some to that feeling as well :-)"*
> *- Bob Flynn, Indiana University MMUG*

Accessibility/Usability - Section 508, CSS and disabled access
Advanced CF - Advanced ColdFusion topics
Boot Camp - Basic ColdFusion and Flash Topics
Deployment/Platform - Integrating with SQL Server, Windows and .NET
Flex - Learn Flex 2 by experts at Universal Mind & Adobe
Manager/Emp Programming - Fusebox and Project management topics

8th Annual Event
Over 1,000 attendees

Pre-Conference Classes: June 26th & 27th
Main Event: June 28th - July 1st
Washington DC Area

Adobe
New Atlanta COMMUNICATIONS
TeraTech Programming
HostMySite.com
Microsoft

re-import into the LMS. The process removes typical SCORM authoring methods of unique code for every SCO and HTML page.

Playback Support

One goal of the S2PE is to leverage the content creator's vision. The support for a variety of media asset types includes graphics, 3D, text, video, audio, Flash and Shockwave. Their implementation by the author is not limited to merely placing elements on the screen. Support for event synchronization allows media elements to have their location, transparency, visibility, z-ordering, playback rates, video locations, and so forth set and tweened automatically at runtime. All necessary mathematical in-between calculations are performed by the PE. The author need only specify key event times.

For many designers, they're content appears on screen as delivered to the programmer. Yet, with the S2PE they are able to easily set many properties that they relied on the programmer for. For example, the "render" tag maps directly to Director's sprite "ink" property. "Blend", "background" color ,etc are all exposed to encourage creativity. For many designers this is something new to work with. In general, properties of sprites map to a specific tag within the CPM XML file.

Multi-state buttons, graphical and textual hotspots, pop-up images and pop-up text all offer additional capabilities in support of the author. All on screen items are referred to as media elements. Nearly all media elements share a set of common attributes.

Specific media elements have their own applicable attributes.

Graphic

Supported Formats (PNG, JPG, GIF, BMP, TIF, PSD) Note: PSD (Photoshop) files depend on the state of the PSD when it was last saved. Hidden layers are discarded and the visible layers are flattened to yield the final image.

Inherits from Media Element Properties

Any media element may have hotspots assigned to it. Hotspots are described as areas within a sprite area. This is similar to an HTML image map. For textual elements, it is done on a per word, or word range basis. Hotspots support unlimited pop-up graphics and text for each spot.

Though a basic overview, the previous information allows inspection of a simple CPM XML file (Listing 7).

```
<graphics>
    <graphic filename=”:::ashleym.png”
origin=”320,200”/>
</graphics>

<text>
    <head render=”bgtransparent”>
Adobe > Thank You Director</head>
    <prompt color=”255,255,255”
align=”center” render=”bgtransparent
”/>
</text>
```

All graphics are enclosed within the graphics tag. In this case there is only the single ashleym.png file. Also included with that graphic's information is its layout location.

This XML file does not indicate any text to be rendered on screen but does take advantage of the built-in support for header text and a user prompt. The next example includes audio, graphics, and a linked Director movie (LDM) containing an independent 3D world (Listing 8).

```
<audio>
    <synch filename=”:::shelbym.mp4”
start=”-1” stop=”-1”/>
</audio>

<graphics>
    <graphic filename=”:::type4.jpg”
origin=”center,center” blend=”60”/>
</graphics>

<ldm>
    <ldm filename=”:::butzi.
dcr” origin=”center,center”
visible=”true”/>
</ldm>

<text>
    <head>Compound/Element >
Compound</head>
    <prompt>Click Continue.</prompt>
</text>
```

Note that in the audio section there are start and stop times listed. The values of -1 indicate play from start to end. However, any value in 1/60 second increments may be used.

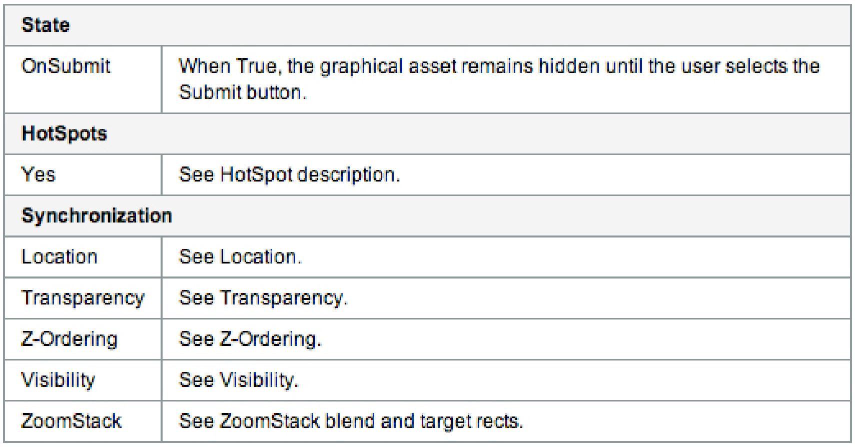
The graphic section places the location of the image at "center, center". The XML specification allows this value as shorthand. Besides specifying the term "center" or absolute pixel locations, you can also use fractions such as "1/4" to indicate one-fourth of the way in the specified direction.

Finally, Listing 9 is a more complex XML file. This is a partial list of a file used to run a 60-second sequence of animation, art, video, and music. The main feature to note is the synch attribute, which is used to interpolate media element states in reference to the movie time of the audio filename at the top of the file.

```
<audio>
    <synch filename=”:::shortPhili.
mp4” start=”-1” stop=”-1”/>
</audio>



<video>
```



State	
OnSubmit	When True, the graphical asset remains hidden until the user selects the Submit button.
HotSpots	
Yes	See HotSpot description.
Synchronization	
Location	See Location.
Transparency	See Transparency.
Z-Ordering	See Z-Ordering.
Visibility	See Visibility.
ZoomStack	See ZoomStack blend and target rects.

```
  <video filename=”:::drmiller.mp4”
origin=”center,center”  width=”640”
height=”420” visible=”0”>
    <synch>
        <playrate>
            <playrate event=”start”
playrate=”1.0”/>
            <playrate event=”100”
playrate=”6.0”/>
            <playrate event=”2800”
playrate=”0”/>
        </playrate>
        <visible>
            <visible event=”start”
visible=”1”/>
            <visible event=”100” vis-
ible=”0”/>
            <visible event=”110” vis-
ible=”1”/>
        </visible>
        <playtime>
            <playtime event=”start”
playtime=”1100”/>
            <playtime event=”1850”
playtime=”4600”/>
            <playtime event=”2450”
playtime=”11000”/>
        </playtime>
    </synch>
</video>

<graphics>
  <graphic filename=”:::collage_
01.jpg” origin=”center,center” vis-
ible=”0”>
    <synch>


        <zoomstack timebase=”audio”>
            <zoomitem file-
name=”:::collage_01.jpg” target-
rect=”85,110,90,115”  method=”rect”
direction=”in” prescale=”true”
steps=”30” blendinat=”-1” blend-
outat=”925”   startzoomat=”900”  stop-
zoomat=”951”/>
            <zoomitem filename=”:::
common:images:collage_02.jpg” target-
rect=”85,110,90,115”  method=”bestfit”
direction=”in” prescale=”true”
steps=”30” blendinat=”951”   blend-
outat=”1100”   startzoomat=”925” stop-
zoomat=”1150”/>
            <zoomitem filename=”:::
common:images:collage_03.jpg” target-
rect=”85,110,90,115” method=”bestfit”
direction=”out” prescale=”true”
steps=”30” blendinat=”1200” blend-
outat=”1400” startzoomat=”1150” stop-
zoomat=”1500”/>
        </zoomstack>
    </synch>


  </graphic>

  <graphic  filename=”:::collage_
10.jpg” origin=”200,-100” width=”100”
height=”75” blend=”100”>
    <synch>
        <zorder>
            <zorder event=”3130”
top=”1”/>
        </zorder>
    </synch>
    <hotspots click=”0”>
        <hotspot item=”left”
rect=”10,10,90,65”  >
            <popimage filename=”:::
sm_collage_10.jpg” origin=”350,130” />
            <popText origin=”220,40”
width=”275” regH=”left” regV=”top”
italic=”true” pt=”18” text=”end”/>
            <popText align=”center”
origin=”250,95” width=”250”
regH=”left” regV=”top”  pt=”26”
bold=”true”  text=”End”/>
        </hotspot>
    </hotspots>
  </graphic>
<graphics>
```

With regards to quizzes and interaction within the LMS, all scoring, determination of correctness, objective mapping, and LMS interaction is handled automatically. An initial library of approximately 150 JavaScript functions dedicated to LMS interaction supports all required messaging between the S2PE and the LMS. For clarity, the libraries are split into SCORM sections dealing with the API, Comments From Learner, Comments From LMS, Date and Time, Interactions, Objectives, SCO’s, and Sequencing and Navigation. Whether you implement SCORM content with Director or some other method, JavaScript is a requirement for communication with the LMS.

Future Vision

The current working version supports limited messaging between media elements. Increasing that support will allow media elements to interact with greater detail and robustness.

The inclusion of support for speech and accessibility standards will round out the needs of most content. Director offers the most flexible 508 authoring solutions available. It is a little known fact that the Windows version of Director supports the Microsoft SAPI XML TTS. As such, Director is a natural for notably exceptional speech within SCORM content.

Throughout it all, the vision remains clear, bringing LMS and SCORM into the public arena by facilitating the creation of robust content that can present one’s vision.

Additional Resources

- Advanced Distributed Learning The official site for all things SCORM. http://www.adlnet.org/
- Advanced Distributed Learning SCORM 2004 Plug-In Technologies Example. http://www.adlnet.org/downloads/204.cfm Includes both Director and Flash examples of LMS communication.

US Army Training Support Center SCORM v1.2 Business Rules and Best Practices http://www.atsc.army.mil/itsd/imi/bus_rules.asp Refers to SCORM 1.2 but relevant.

Kraig Mentor is a developer for Concurrent Technologies Corporation (CTC) of Johnstown, PA (www.ctc.com). CTC is an independent professional services organization that provides management and technology-based solutions to a wide variety of clients representing state and federal government as well as the private sector. Mr. Mentor recently presented the content of this article at the International Plugfest II in Taipei, Taiwan. He has spoken at other SCORM conferences on the topics of JavaScript LMS communication and has written articles for the Advanced Distributed Learning (ADL) Web site. His work on various projects has earned him eight Addy awards as well as an Excellence in Education Award from the National Hydropower Association. Prior to joining CTC, Mr. Mentor worked as a developer on the Director product line at Macromedia, San Francisco. Other past areas of work include automation, robotics and electronics. He may be contacted at kraig@ctc.com.

Developing Flex RIAs with Cairngorm Microarchitecture

Using the Cairngorm Store sample application

by steven webster

This series presents an open-source architectural framework to Flex developers called Cairngorm. In this series I explain the thought leadership behind Cairngorm, the design challenges that Adobe feels Cairngorm addresses best, and the projects for which Cairngorm is an appropriate skeleton for development.

Using the Cairngorm Store sample application, this series explains what Adobe Consulting thinks about scoping, estimating, and delivering a Rich Internet Application (RIA) when basing it on Cairngorm from the start. I also explain the various Cairngorm concepts and take a deep dive into the implementation of the Cairngorm Store.

Finally, I demonstrate some of the principal benefits of delivering an RIA based on this established microarchitecture by adding a new feature to the existing Cairngorm Store application from the point of view of a Cairngorm developer. By this stage in the series, you see the benefits for yourself.

Cairngorm isn't the only way to build a Rich Internet Application, of course. However, Adobe Consulting has helped numerous customers and partners succeed in delivering large-scale Flex RIAs by building upon their preexisting Flex application development knowledge using the information contained in this series.

This comprehensive introduction covers the full spectrum of Cairngorm, from understanding the motivation and concepts of Cairngorm to architecting your own applications upon this established and supported microarchitecture.

Instead of delving into code from the outset, the first part offers a context and background for understanding the Cairngorm architecture. I discuss frameworks and clarify the difference between an application framework and an architectural framework. I then explore design patterns and introduce the microarchitecture concept. Finally, I give a brief background on the emergence of Cairngorm: its history and where it is headed—its roadmap.

You will develop a retail commerce application using both Flex and Cairngorm on the client-side tier and a new or existing J2EE infrastructure on the server-side tier.

This article originally appeared on www.macromedia.com/devnet. Reprinted with permission.

Clarifying the Definition of Frameworks

In software development, the term framework is one of the most overloaded and overused terms within and between development teams. When developers write large pieces of code that they consider significant enough to consider leveraging on other projects, they tend to supplement the code with this term. Thus there are many types of frameworks: persistence frameworks, transaction frameworks, logging frameworks, aspect-oriented frameworks, animation frameworks, unit-testing frameworks, and the like.

Before I delve into the discussion of the Cairngorm framework, it's important that I explain an important distinction that the Adobe Consulting team shares with customers and partners about frameworks—specifically the distinction between application frameworks and architectural frameworks.

Application Frameworks

Flex is a tremendous example of an application framework. In fact, the forthcoming release of Flex 2.0 actually distinguishes the application framework piece—often called the "app model" within Adobe—in its architecture. The Flex Framework 2.0 provides a rich collection of class libraries that provide highly granular functionality that developers can use to create custom code. For instance, the Flex 2.0 Collections API provides application developers with all the base-level functionality needed to create managed data collections. Application developers then compose these collections together into higher-level objects that are relevant to their particular application. Furthermore, application frameworks such as Flex typically expose application-level services such as history management, layout management, cursor management, exception handling, internationalization, logging, and so forth.

When a framework provides highly granular class libraries that give a high degree of flexibility to application developers, or when a framework provides application-level services that are useful across multiple developer projects, I call it an "application framework."

Another excellent example of an application framework is the FAST framework that Adobe Consulting has used with customers and partners with great success. The FAST framework, as John Bennett explains in his article, Faster Development with the Flex Application Starter Toolkit (FAST), provides application services for logging and tracing, and value-add class libraries that extend the Flex Framework's own implementation of RPC data services in Flex 1.x.

Architectural Frameworks

Architectural frameworks are different beasts entirely. Typically the job of an architectural framework is not to provide any additional services to application developers except an infrastructure around which an application can hang—to provide the skeleton or internal structure of moving parts, around which the flesh and muscle particular to a business domain can be layered.

In other words, an architectural framework provides a generic starting point for the technical architecture of your application.

Applying Design Patterns

It is hard to talk about technical architecture without paying attention to an important movement in software engineering called design patterns.

Without going into detail about software design patterns, let me just say that the expression "there is nothing new under the sun" is never more true than in the discipline of software engineering. Developers often find themselves addressing engineering problems that appear with regular consistency in application development. Almost as consistent as their appearance is the repetition of solutions for these problems. Wherever such recurrences occur, you can identify the solution as a "pattern," indicated by facing the design challenge and finding the appropriate design solution.

The Lure of Design Patterns

Now a warning: When software engineers first encounter design patterns, the realization of a catalog of solutions to their engineering problems can be a powerful one. Often developers recognize a subset of problems they have encountered and then seek to understand other design patterns and where they might apply them. However, the old adage "when all you have is a hammer, everything looks like a nail" can apply here. You can often find "pattern overload" in an application, where developers abdicate responsibility for classes and collaborations and, instead, shoehorn everything into a Factory, Flyweight, Observer, or Decorator.

Used appropriately, however, design patterns can be a powerful tool in a developer's toolbox. Design patterns not only offer common solutions to problems, but the way developers apply them to an application indicates the intention of the implementation. For instance, whenever you see a Singleton in a code base, you understand that this is a class for which there should be only one instance. Likewise, whenever you come across a Factory, you recognize that there are a number of different objects that a manufacturing class can construct.

Rising through the technical architecture—from the ground-zero of detailed implementation towards the helicopter view of high-level system design—you can begin to appreciate the design in other, recurring ways. Like the Mandelbrot, you begin to recognize higher order structure as your lower level structure collaborates. Thus design patterns begin to coalesce in repeatable ways, offering higher level solutions to higher level design problems.

Microarchitecture as a Composition of Design Patterns

Although a design pattern might have offered a specific solution to a specific problem (in this case, using the Singleton pattern for the problem of ensuring that only one instance of a class ever exists), when a collection of patterns regularly collaborate with one other, their assembly is for the greater good of a greater aim: "How do I intercept user gestures and ensure that a worker class assumes responsibility?" or "How do I centralize the business logic that might be used in more than one context on the client, even though it is actually implemented as a collection of service calls to a number of different types of services on a collection of servers?"

When you start to assemble collections of design patterns into these higher order but nonetheless highly generic systems, you communicate these as "microarchitectures."

Let's take this high-level discussion of application frameworks, architectural frameworks, design patterns, and microarchitectures and put it into context. When Adobe Consulting speaks about the Cairngorm framework, we are speaking about an architectural framework—a starting point for technical architecture that is generic enough to apply in most cases to medium-to-complex Flex RIAs.

Furthermore, when we speak about the Cairngorm framework, we do not mean some monolithic architecture that limits your freedom as a developer to solve problems. Rather, when we advocate the Cairngorm framework, we mean the following:

- A lightweight framework that offers a little prescription for some of the challenges consistent with the Flex RIAs we have encountered
- Using a small number of relevant design patterns, where the moving whole is ever so slightly greater than the sum of its static parts
- A microarchitecture for RIA development—a starting point for your technical architecture that solves the problems as they have been solved successfully before

If we can see further, it is because we are standing on the shoulders of giants.

A Short History of Cairngorm

iteration::two, a software consultancy that Alistair McLeod and I cofounded, recognized that many of the design challenges faced and successfully solved in the world of J2EE application development were still relevant issues in the world of RIA development. We came to this realization back in the days of RIA development with Flash, Flash Remoting, and J2EE—a period of RIA history that we'll look back upon with the same fondness and nostalgia as a C++ programmer has for 6809 assembly language.

Borrowing a subset of a collection of design patterns advocated by Sun Microsystems as the Core J2EE Pattern Catalog, we first presented these patterns to the Flash community in our book, Reality J2EE: Architecting for Macromedia Flash MX (Pearson Education, 2003). To accompany the release of Flash MX 2004, we presented the patterns in a chapter called "ActionScript 2.0 Design Patterns for RIA Development" in Macromedia Flash MX 2004 ActionScript 2.0 Dictionary (Macromedia Press, 2003).

As the RIA technology platform matured from the design-centric approach of Flash towards the declarative programming model of Flex, most of the motives for applying these patterns remained. However, the Flex programming model offered us ever more elegant ways to implement these patterns. Furthermore, some patterns that we considered hugely valuable to Flash RIA developers (such as the ViewHelper pattern) became less relevant in the Flex RIA world, allowing us instead to advocate new Flex-specific patterns of our own, such as the ModelLocator pattern.

At MAX 2004 we announced our decision to release the Cairngorm framework as an open-source project for Flex; a decision that has led very much to its widespread adoption within the community.

What Cairngorm Teaches You

Cairngorm is a microarchitecture that addresses three key areas that we have always tried to provide as best-practice recommendations for our customers and partners:

- Handling user gestures on the client
- Encapsulating business logic and server interactions
- Managing state on the client and representing this state to the user interface

Cairngorm offers a microarchitecture (collection of design patterns) that aids us in consistently solving these recurring design challenges.

As you read this series of articles, you will learn the following:

- How the Front Controller and Command patterns implement a "Service to Worker" microarchitecture for listening and responding to user requests.
- How the Business Delegate and Service Locator patterns interact in such a way that you can reuse the business logic and encapsulate it so that it establishes a clear contract between client and server-side development teams, independent of server-side technical implementations such as web services, Enterprise JavaBeans, ColdFusion components, or even RESTful architectures using XML over HTTP.
- How the Value Object pattern from J2EE can collaborate with the Model Locator pattern first implemented in Cairngorm 0.99 to be an elegant strategy for maintaining a stateful client with a rich and cinematic user experience.

Current State of Cairngorm

Cairngorm is currently in version 0.99. It never made it to a 1.0 release because iteration::two became immersed in a trail of acquisitions, first by Macromedia and then by Adobe Systems. Furthermore, with Flex moving towards a 2.0 release of its own, the introduction of ActionScript 3.0, a number of improvements to the underlying Flex application framework, an exciting array of new services (including Flex Enterprise Services), and the evolution of the Cairngorm Committee—now comprised of a committed core of Adobe consultants and Adobe engineers, as well as numerous community members behind it—Adobe Consulting is focused on aligning a comprehensive Cairngorm 2.0 release that leverages the underlying Flex 2.0 framework.

The core Cairngorm concepts remain the same, however. Only the underlying implementation changes due to additional thought leadership and best practices on how to leverage Flex 2.0 features such as Flex Data Services and Flex RPC Services within a Cairngorm application. For those working with Flex 2.0 in Labs, we will also make available regular alpha releases of Cairngorm 2.0.

Note: In the final article of this series, I explore the criteria you can follow to decide when to use Cairngorm–and when not to use it. We do not advocate for a second that Cairngorm is the only way to build Rich Internet Applications. Nor do we even suggest that our best practices are the only best practices. In fact, others may promote best practices that seem to contradict what Cairngorm suggests.

All we ask is that you first try to understand the problems that Cairngorm tries to solve before preemptively solving them with Cairngorm. If you are learning Flex for the first time and consequentially building one of your first Rich Internet Applications, I strongly suggest that you become comfortable with the myriad of new tools and techniques that the technology and platform offer you before complicating your learning curve with Cairngorm.

This is not to say that Cairngorm is complex. On the contrary, it is important that you be confident and comfortable building simple RIAs that do not require the benefits of a technical architecture such as Cairngorm before taking advantage of the benefits that Cairngorm prescribes.

Future parts of this series will ensure that both seasoned and novice Flex developers understand the RIA challenges that created the problems for Cairngorm to solve. We want to explain as clearly as we can, and with detailed code-level examples, the elegant way in which Cairngorm can help you solve them.

Where to Go from Here

The aim of this article was to introduce you to the Cairngorm framework by clarifying the ambiguity around the definition and usage of frameworks, why you might use a framework, and the origins and the future of the Cairngorm framework.

Cairngorm is an architectural framework that provides a suggested implementation of a technical architecture upon which you can build your own, more complex, application-specific architectures.

By basing your application on the Cairngorm architecture, however, there are some generic and fundamental design challenges that you won't have to solve by yourself while leveraging the benefits inside Flex, such as how to handle user gestures in an elegant fashion, how to handle server interaction and business logic in an elegant and scalable fashion, and how best to manage state on a rich and immersive client application.

I gave a brief overview of the engineering discipline of software design patterns and how iteration::two borrowed those that were useful to us as J2EE developers, and which are still useful for our RIA development. Having an aerial view of the technical architecture usually helps developers identify recurring collaborations of design patterns. The concept that such collaborations, known as microarchitectures, exist is the foundation of the Cairngorm framework.

When consulting, I often say that the difference between theory and practice is that, in theory, there's no difference between theory and practice. I depart from the discussion of context and background of design patterns and microarchitecture, and instead present Cairngorm in terms of providing four key benefits:

- Maintaining state on the client
- Architecting the user experience
- Implementing feature-driven development
- Invoking server-side services

You'll get a hands-on feel for the ways we leverage the Cairngorm framework in application development, the problems that Cairngorm solves, and the ease with which you can consistently deliver highly scalable, maintainable, and robust Rich Internet Applications with Flex.

Introducing the Cairngorm Store

The Flex Store application showcases the features of the Flex application framework. It demonstrates how to use various layout containers, navigator containers, controls, effects, data binding, the Drag-and-Drop manager, form validation, and the History manager. Furthermore, Flex Store articulates the component-development model of Flex, showing how to create loosely coupled components that integrate through an event-driven architecture. The Flex Store application is an excellent way to become familiar with declarative layout using MXML and business logic development using ActionScript.

With the Cairngorm Store, I take this familiar application and completely rebuild it as an enterprise rich Internet application—an application that is rapid, scalable, and maintainable—through Cairngorm.

The Cairngorm Store borders on the complexity of a rich Internet application that would benefit from a Cairngorm architecture. I demonstrate this tipping point by showing how, within the rearchitected Cairngorm application, you can add new functionality easily, more predictably, and with significantly less development risk than if you were to attempt to add functionality to a non-Cairngorm code base.

The Cairngorm Store: Four Key Challenges

Instead of an academic discussion of design patterns we selected for Cairngorm, I will explain the common high-level challenges that often occur when developing rich Internet applications and how to achieve these solutions with a Cairngorm development process. By describing the development process, I also explain the various constituent design patterns used in the Cairngorm framework.

A good technical architect first sees application development as a solution to a business problem, then as a system that realizes this solution, then as a technical architecture for the custom software in the system, and finally as the detailed class-level implementation of that technical architecture.

The approach taken here in Cairngorm and Cairngorm Store gives you clarity from 35,000 feet before a deep dive and race along the terrain at the code level.

From a high level, there are four key challenges that the Adobe Consulting team faces, whether building mortgage calculators, retirement planners, online banks, single-screen checkouts, full e-commerce applications, or interactive maps. These challenges are as follows:

- Keeping state on the client
- Architecting the view
- Driving feature-driven development
- Invoking server-side services

The challenges and requirements for the Cairngorm Store are no different.

The client presents the products to sell to the customer, the shopping cart remembers what the user purchased, and the user must complete multipage application forms during the checkout process. Throughout this entire process, your application must maintain state on the client.

The single-screen user experience has a number of different on-screen components. It has both graphical and list-based views of the products for sale, detailed product information, a shopping cart that the user can add to or drag products to, and a series of forms that the user must fill in to fulfil his or her order. You must architect the view.

There are a number of features that users demand in the application. The users must be able to do the following:

- View all products and get detailed information for any selected product
- Add products to, or remove products from, their shopping cart
- Purchase products in their cart
- Proceed to and complete a checkout process.

These features drive your development.

Finally, the products that the application sells change as inventory changes. Your application must fetch product information from a server-side infrastructure capable of pulling products from an inventory database. Furthermore, when customers place orders, your application must persist the orders in enterprise systems by committing these orders to a database or a system such as SAP. Your application must integrate with and invoke server-side services.

As you can see, the Cairngorm Store—though unique in its business domain, product selection, and look and feel—is just like any other rich Internet application when studying it against the four key challenges.

In this article, I take a look at the first of these four key challenges: keeping state on the client.

Keeping State on the Client

One key shift in mindset when you move from the development of web applications to rich Internet applications is that the client is stateful.

Web application developers are all too familiar with the stateless request/response approach. Therefore, they have created various approaches for maintaining state on the client. Whether you continually stuff data into an HTTP request as the client and the server exchange pages, or you use URL encoding or cookies to keep some limited state information on the client, or you take advantage of abstractions of these features such as J2EE sessions, the typical application developer's is to maintain state on the server, and make it available to the client.

With RIAs however, web application developers can break free from the shackles of HTTP request and response. No longer do you have to concoct solutions for the stateless client because the client is indeed stateful. If you have been building desktop or rich-client applications with technologies such as Swing or AWT, for instance, you'll shrug your shoulders right now because all of this will sound all too familiar—it is.

Difference Between State and the Model in MVC

Many developers are familiar with the concept of MVC or Model/View/Control in application development, and wonder where state fits into this discussion. Quite simply, the state is the model. As obvious as this may sound, it presents some further common challenges to solve in the implementation of the model in an RIA.

First, resist the opportunity to scatter state all over your application as strings, numbers, Booleans, and all manner of other primitive objects. It is not my purpose here to advocate best practice object-oriented development. Indeed, that you accept the benefits and practice the principles of OO are almost prerequisites to understanding this article.

However, I strongly advocate that the client hold the data—as state, model, or whatever you want to call it, as objects that have semantic meaning. What is semantic meaning in this context? I mean that if you sell fly-fishing flies online, your object model should include "flies," "patterns," "hooks," and "hackles." If you sell mortgages, your object model should include "products," "rates," "repayments," and "terms." If you sell maps, your object model likely includes "routes," "waypoints," "points of interest," and "gas stations."

Synchronizing Client and Server State

One challenge is relatively unique to the stateful client of a rich Internet application: The state on the client usually reflects the state held on the other side of the wire, in the server middleware. Let's examine this design problem some more.

In server-side application development, one key challenge is that state often sits within two tiers of an application architecture: the business tier and the integration tier.

The integration tier usually contains databases, mainframes, message queues, ERP solutions, CRM systems, and other enterprise information

BONUS TRACK ADDED! (FREE $1,295 VALUE!)
"Real-World AJAX" 2-Day Seminar

SOA WebServices & Enterprise OpenSource Conferences

The #1 i-Technology Educational and

Register at... www.SOAEOSConference.sys-con.com

The **10th International SOA Web Services Edge Conference**, colocated with the First Annual **Enterprise Open Source Conference**, will take place on **June 5-6, 2006** at the historical **Roosevelt Hotel in New York City**. SOA Web Services Edge and Enterprise Open Source Conference together deliver the #1 i-technology educational and networking opportunity of the year.

This year, more than 100 distinguished conference faculty members will present **96 cutting-edge SOA, Web Services and Enterprise Open Source topics** in educational classes, panels, and keynotes in six simultaneous tracks for two information-packed days, plus a bonus track, a **"Real-World AJAX"** two-day seminar, a $1,295 value!

The conference program does not merely present a comprehensive view of all the development and management aspects of integrating a **SOA strategy** and an **Open Source philosophy** into your enterprise, its organizing principle is that delegates will go away from the intense two-day program replete with why-to and how-to knowledge delivered first-hand by industry experts.

According to analyst firm Gartner Group, **by 2008 more than 60 percent of enterprises will use SOA** as the guiding principle when creating mission-critical applications and processes. **"Businesses that ignore the potential of SOA will find themselves outpaced by rivals who improve their agility and transform themselves into new kinds of enterprises,"** says Gartner analyst Yafim Natis.

Meanwhile, the **Open Source revolution** has spread into traditional areas of enterprise IT, with all major vendors and a global group of **Open Source entrepreneurs** developing the technology and delivering the solutions that are changing the way the world does business and the way governments serve their citizens.

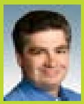
Receive FREE WebCast Archives of all Entire Conference!

The best news for this year's conference delegates is that your "Golden Pass" registration now gives you full access to all 96 conference sessions. We will mail you the complete content from all the conference sessions in seven convenient DVDs after the live event takes place.

► **This on-demand archives set is sold separately for $995**

Receive 8 Days Worth of Education F

FACULTY



Jeffrey Barr
Amazon

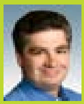
As Web Services Evangelist for Amazon.com, Jeff Barr focuses on creating developer awareness for the Amazon software platform. He has a longstanding interest in Web services and programmatic information interchange. Jeff has held development and management positions at KnowNow, eByz, Akopia, and Microsoft, and was a co-founder of Visix Software. Jeff's interests include collecting and organizing news feeds using his site, www.syndic8.com . He holds a Bachelor's Degree in Computer Science from the American University and has done graduate work in Computer Science at the George Washington University.

Israel Hilerio
Microsoft

Israel Hilerio is a program manager at Microsoft in the Windows Workflow Foundation team. He has 15+ years of development experience doing business applications and has a PhD in Computer Science.

Adam Kolawa
Parasoft

Adam Kolawa, Parasoft co-founder and CEO, is considered to be a visionary in his field. In 1983, he came to the United States from Poland to pursue his Ph.D. In 1987, he and a group of fellow graduate students founded Parasoft to create value-added products that could significantly improve the software development process. Kolawa's years of experience with various software development processes has resulted in his unique insight into the high-tech industry and the uncanny ability to successfully identify technology trends. As a result, he has orchestrated the development of numerous successful commercial software products to meet growing industry needs to improve software quality.



Jason Levitt
Yahoo!

Jason Levitt, Technical Evangelist on creating Flash-based Yahoo! Maps applications.

Duane Nickull
Adobe

As senior standards strategist for Adobe Systems, Duane Nickull is responsible for managing Adobe's participation in OASIS and UN/CEFACT, as well as ensuring that Adobe's enterprise solutions support emerging XML standards. Previously Mr. Nickull co-founded Yellow Dragon Software Corporation, a privately held developer of XML messaging and metadata management software, recently acquired by Adobe. Mr. Nickull currently serves as a vice chair of the United Nations Centre for Facilitation of Commerce and Trade (UN/CEFACT) where he oversees the United Nations Electronic Business strategy and architecture.

Bob Pasker
Azul

Bob Pasker is deputy CTO with Azul Systems. He has been designing and developing networking, communications, transaction processing, and database products for 25 years. As one of the founders of WebLogic, the first independent Java company (acquired by BEA Systems in 1998), he was the chief architect of the WebLogic Application Server. Bob has provided technical leadership and management for numerous award-winning technologies, including the TribeLink series of routers and remote access devices, and the TMX transaction processing system. Bob graduated magna cum laude and Phi Beta Kappa from San Francisco State University and holds a Masters degree from Brown University.

Brian Behlendorf
CollabNet

Brian Behlendorf founded CollabNet, with O'Reilly & Associates, in July 1999. The company provides tools and services based on open source methods. Before launching CollabNet, Behlendorf was co-founder and CTO of Organic Online, a Web design and engineering consultancy located in San Francisco. During his five years at Organic, Behlendorf helped create Internet strategies for dozens of Fortune 500 companies. During that time, he co-founded and contributed heavily to the Apache Web Server Project, co-founded and supported the VRML (Virtual Reality Modeling Language) effort, and assisted several IETF working groups, particularly the HTTP standardization effort.

Marc Fleury
JBoss

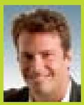
Born in Paris in 1968, Marc Fleury got his Ph.D in physics from the Ecole Polytechnique in Paris. He started in Sales at Sun Microsystems France and then moved to the US where he worked on early java enablement of SAP at SAPLabs. Marc started the JBoss project in 1999. An ex-Lieutenant in the paratroopers, Marc holds a degree in Mathematics from the Ecole Polytechnique, a master in Theoretical Physics from the Ecole Normale ULM and was a visiting scientist at MIT during his thesis. Marc's research interest focuses on aspect oriented middleware.

Andy Astor
EnterpriseDB

Andy is President and CEO EnterpriseDB, the world's leading enterprise-class, open source database company. Previously, Andy was vice president webMethods, leading the company's open source, standards, and Web services agendas. Andy was elected twice to the Board of Directors of the Web Services Interoperability Organization (WS-I), and led WS-I's marketing efforts. Prior to joining webMethods, Andy was vice president at D&B, where he led worldwide development of all on-line products. His work at D&B included the development and launch of one of the earliest commercial Web services.

Mike Milinkovich
Eclipse.org

Mike Milinkovich has held key management positions at Oracle, WebGain, The Object People, and Object Technology International Inc. (which subsequently became a wholly-owned subsidiary of IBM), assuming responsibility for development, product management, marketing, strategic planning, finance and business development. Mike earned his MS degree in information and systems sciences and a bachelor of commerce degree from Carleton University in Ottawa, Canada.



Peter Yared
ActiveGrid

Peter Yared is the founder and CEO of ActiveGrid. Most recently, he was CTO of Sun Microsystems's Liberty Network Identity initiative. Mr. Yared was also CTO of Sun Microsystems Application Server Division. Before its acquisition by Sun, Mr. Yared served as CTO of NetDynamics, which pioneered the then-leading J2EE application server. Earlier, Mr. Yared was founder and CEO of JRad Technologies, an enterprise Java company acquired by NetDynamics. Additionally, Mr. Yared was Chief Architect of client/server products at object-oriented tool maker Prograph International and the architect of several mission-critical systems deployed by U.S. government agencies and the GED Testing Service.

David Temkin
Laszlo

David Temkin is Chief Technology Officer of Laszlo Systems, Inc. In this role, he has positioned the company to become the next technology standard for rich Internet applications. Under his direction, Laszlo developed its patent-pending open-source product suite and extended operations to both coasts of the United States. Before founding Laszlo, Temkin was senior director of engineering at Excite@Home where he led a team of 55 engineers, designers and technical writers responsible for developing the company's consumer software. Prior to Excite@Home, Temkin was an engineering manager in the Newton division at Apple Computer and developed enterprise software at EDS.

Kevin Hakman
TIBCO

Kevin Hakman is Co-founder, TIBCO General Interface, TIBCO Software Inc. Prior to TIBCO General Interface, he was the co-founder of Versalent Inc. a leading provider of enterprise client technology. Prior to Versalent, he founded a series of successful emerging Internet technology and ecommerce ventures. He has also written for eBusiness Journal and HotWired.

Coach Wei
Nexaweb

Coach Wei currently serves as CTO for Nexaweb, which develops the leading XML-based rich client technology platform for building and deploying Enterprise Internet Applications. Previously, he played a key role at EMC Corporation in the development of a new generation of storage network management software. Coach is a graduate from MIT, holds several patents, and is an industry advocate for the proliferation of open standards.

Luis Derechin
JackBe

Luis Derechin is CEO and Co-Founder of JackBe. Mr. Derechin has over 12 years of entrepreneurial and management experience. He has been part of the founding team of successful startups, including a catalogue retail company that achieved $15M in sales.

Jouk Pleiter
Backbase

Jouk Pleiter is the CEO of Backbase, a leader in the field of Rich Internet Applications and AJAX development software. Backbase's clients include ING, ABN AMRO, TNT, KPN, Comsys and Heineken. Backbase operates globally with offices in San Mateo (North America) and Amsterdam (Europe). Since 1995, Jouk has been an entrepreneur: he founded three successful Software companies. Prior to Backbase, Jouk was part of the founding team at the web content management company Tridion, where he led the product management operations, and was driving the company's efforts to become a leader in the European WCM software market.

Networking Opportunity of the Year!

SOA WebServices Edge conference 10th International

2006 ENTERPRISE> OPENSOURCE CONFERENCE

The Roosevelt Hotel

JUNE 5-6, 2006

Roosevelt Hotel / New York, NY

For Sponsorship Information ▸ **Call 201 802-3022**

Early Bird Price (Before March 31, 2006)	
Incl. Golden Pass to Both Conferences*	$1,295
Discounted Price (After March 31, 2006)	
Incl. Golden Pass to Both Conferences*	$1,495
Conference Price (On June 5-6, 2006)	
Incl. Golden Pass to Both Conferences*	$1,695

*BONUS: "Real-World AJAX" Two-Day Seminar!

Attention Sponsors:

A Forum will be available to display leading Web services, OpenSource, and AJAX products, services, and solutions.

TOPICS INCLUDE...

SOA Web Services Edge:

> Transitioning Successfully to SOA
> ebXML
> Orchestration
> The Business Case for SOA
> Interop & Standards
> Web Services Management
> Messaging Buses and SOA
> SOBAs (Service-Oriented Business Apps)
> Delivering ROI with SOA
> Java & XML Web Services
> Security
> Systems Integration
> Sarbanes-Oxley
> Business Process Management

Enterprise Open Source:

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

AJAX User Experience:

> Rich Internet Applications
> Flash-AJAX Integration
> Developing Interfaces
> AJAX on Rails
> Best Practices
> Enterprise AJAX
> Technology Demos
> Lessons from the Frontline
> Bringing Desktop Apps to the Web
> AJAX Power Panel
> and more...

SYS-CON EVENTS www.EVENTS.sys-con.com

Packed into 2 Information-Filled Days!

systems that hold data on the objects and process particular to a business domain.

In enterprise application development, these systems are typically aggregated and presented to users through a middleware tier, also called the business tier. Many solutions address some of the challenges of keeping state consistent across the two tiers. First of all, the representation of data in these tiers may be dramatically different. For instance, the entities, relationships, views, and queries that exist in a relational database solution must be mapped from a relational model in a database to an object model in the middleware business tier. You can use technologies such as Enterprise JavaBeans, Hibernate (H8), and ADO.NET to map data across these architectural tiers.

Irrespective of the solution chosen, the application developer assumes a great deal of responsibility in ensuring the consistency of state across these tiers. As soon as you keep the same information in two places, there is a risk that changes to it in one place are not reflected in another place, or that changes in one place overwrite changes in another place. Suddenly, the developer must care about concurrency, locking, transactions, rollback, and other approaches for ensuring that the model is consistent in these different places in the architecture.

As RIA developers, you are abstracted from the complexities between the business and integration tier; your concerns focus more on how to ensure that the model in the business tier is consistent with the model in the presentation tier.

You can achieve this most easily by ensuring that you have a consistent object model on the client and server, and that you minimize the effort required to keep these object models synchronized.

Introducing the Value Object / Data Transfer Object Pattern

Whenever you want to represent the value of things on the client, you can do worse than to create classes or objects that represent the values of those things. For instance, in Cairngorm Store, one thing that you must represent is the products in our store. Each product has a number of attributes: name, description, price, an associated image for the detailed view, and an associated thumbnail image for the catalog view.

These value objects play a dual role in a rich Internet application, however. In addition to keeping the value of an item on the client, they are useful structures with which you can exchange or transfer data between different tiers in an application. In server-side development, consider a strategy in which you query a database using SQL, receive a result set, and then transform that result set into a collection of value objects representing the things you have pulled from the database.

By passing collections of these value objects between application tiers, you insulate each tier of the application from the underlying implementation of its neighboring tiers. The middleware need not care whether these value objects were created from a SQL query, stored procedure call, message queue, or web service call. By reusing these value objects, you decouple architecture by transferring data between tiers as objects with business meaning ("Products," "Fish," "Maps") rather than technical meaning ("RecordSets," "ResultSets," "Data Sets"). This has led to calling the Value Object pattern the Data Transfer Object pattern instead.

We conceived Cairngorm before the term data transfer object (DTO) became popular, so I call the business objects value objects or VOs.

For example, look at the VOs in the Cairngorm Store application, which reside in the org/nevis/cairngorm/samples/store/vo package:

```
class org.nevis.cairngorm.samples.
store.vo.ProductVO implements
ValueObject, Comparable
{
 public static var registered:Boolean
= Object.registerClass( “org.nevis.
cairngorm.samples.store.vo.ProductVO”,
ProductVO );

 public var id : Number;
 public var name : String;
 public var description : String;
 public var price : Number;
 public var image : String;
 public var thumbnail : String;
}
```

This is not rocket science. The ProductVO is nothing more than a collection of attributes that describe what makes a Product a product. Whenever you keep state on the client about Products, keep it as instances of the ProductVO. If the application must know what the "currently selected product" is, store that information on the client as a ProductVO instance. If the application needs to retain a list of the products in a customer's shopping cart, store that information on the client as a list of ProductVO instances.

Furthermore, if you have to transfer products from the server (and we do, but more on that later) or if you ever have to transfer products back to the server, transfer that data back and forth as value objects. Others might say that you're using data transfer objects, because you are. They're the same thing, remember?

Keeping a Consistent Object Model Between Flex and Java

When you develop middleware, a key artefact from the development process is typically a well-defined object model. In J2EE application development, there is often a package of Java value objects already on the server that correspond exactly to the value objects required on the client.

In this case, the Flex server does some pretty significant heavy lifting for the application. If you want to pass ActionScript value objects into Java, Flex can automatically create the equivalent Java objects on the server side for those objects that you placed on the client side. Similarly, as the Java middleware returns Java value objects to the client, Flex ensures that by the time the RIA uses these objects on the client side, Flex has already translated them into the equivalent ActionScript value objects.

Even more powerfully, if the objects are not just single value objects but complex graphs of value objects, Flex does all the work associated with traversing the graph of objects—not only translating the objects but maintaining the relationships between objects. Thus, when you have a ShoppingCartVO that contains a list of ProductVO objects, and each ProductVO contains a list of AddressVO objects (for delivery and invoice addresses) and an optional DiscountVO object, Flex ensures that the composition of objects stays consistent when passing it between the RIA client and middleware server.

To allow Flex to map between client-side and server-side value objects, you must provide some hints to the Flex server about the relationship between the classes. In Cairngorm, you achieve this by advocating that each and every instance of a value object class contains the following:

```
public static var registered:Boolean =
Object.registerClass( "org.nevis.cairngorm.
samples.store.vo.ProductVO", ProductVO );
```

This line ensures that a ProductVO class on the client maps to the Java class ProductVO.java that resides in the package org.nevis.cairngorm.samples.store.vo.ProductVO on the server.

Using XDoclet2 with ActionScript

XDoclet2 is a java-driven ActionScript code generator. Joe Berkowitz of Allurent created a labor-saving, open-source project for creating ActionScript value objects from their Java counterparts. You can find more information on the project, including a link to the source code on SourceForge, at www.allurent.com/joeb/xdoclet2.

To recap, value objects ensure that you do the following:

- Adhere to the best practice of representing client-side data with a meaningful object model
- Maintain a consistent object model on the client and server
- Leverage the capabilities of the Flex server to translate value objects between Java and ActionScript as they "cross the wire" between client and server

Getting data to the client that is consistent with the server is only the first part of the problem. What do you do with the client-side data and where you keep it to enable you to accomplish this task?

Binding the Model and View Together

One of the primary reasons for wanting data in the presentation tier of an n-tier application is to display that data. Furthermore, the R in RIA is about presenting data in a rich and immersive way.

One of the unsung features of the Flex application framework is the data binding capabilities of Flex. Data binding allows two objects to act as a source and a destination. Changes to the source object are reflected immediately in the destination object. When the destination object is a visual component and the source object is client-side data, you suddenly have a powerful means for ensuring that changes to the client-side state are rendered to the user in real time. The user's view is never stale; the model (the client-side state) is always notified by the view (the user interface) if you have used data binding to bind the model and view together.

At Adobe Consulting, we witnessed development teams convolute solutions for updating the user interface whenever the underlying model changed. Furthermore, these teams became tangled in a mess pretty quickly when displaying data in different parts of the view in different ways. Let's say you have a series of numbers that have been updated on a server. Depending on a user's location in the application, this may cause their table or graph display to update or update data in a dashboard summary about key assumptions for a business. By refactoring architecture towards keeping the data in a client-side model, and leveraging data binding between this model and the different view components, you eliminate much of the complexity around this kind of interactivity and responsiveness in an RIA.

Furthermore, we found that each developer kept the client-side state he needed for his parts of the application development (in his parts of the code base). Consequently, when another developer needed to use that data in another part of the application, she would have to create convoluted and complex solutions to ensure the data was visible from more than one place.

Often we found that developers referenced other data with long chains of component references, such as mx.core.Application.application.storeView.textualList.productGrid. Such strategies are incredibly brittle to change—adding or removing any components (storeView, textualList) would require that a developer update any references to productGrid.

Additional strategies include passing data up and down through MXML component instantiations. Consider the following graphical representation of MXML components, where data is in a leaf node of the tree and is required in another developer's leaf node. The only solution is to find a common branching point for both leaf nodes and then, from that branching point, pass the data down through the component chain with component instantiations, as follows:

```
<myView:MyComponent data="{this.data}" />
```

Once again, this is a strategy that is incredibly brittle to changes in the implementation of the view. It leads to regular errors because it incorrectly passes data down the chain somewhere. Debugging these errors is a laborious job of tracing the component paths to confirm that data has been correctly passed between components. It's a frustrating task.

Enter the Model Locator Pattern

Watching developers fall repeatedly into these traps, the Adobe Consulting team conceived the Model Locator pattern as a best practice for Flex developers to adopt. The Model Locator pattern is unique because it is not a pattern we borrowed from the Core J2EE Pattern catalog. Instead, we created this pattern particularly for Flex application development. Our motivation was to have a single place where the application state is held in a Flex application and where view components are able to "locate" the client-side model that they wish to render.

Our Model Locator pattern strategy encourages the use of data binding so that view components bind directly to the client-side state held in the single instance of the ModelLocator class. In this way, whenever the model is updated in ModelLocator, all view components binding to the model receive notifications (through the underlying data-binding mechanism) and update themselves to render the new model on the client.

In Cairngorm, we provide a marker interface for the Model Locator pattern, and in the Cairngorm Store we have a single class that implements this marker interface, org.nevis.cairngorm.samples.store.model.ModelLocator.

As you take this opportunity to review the source for this ModelLocator class, let me point out a few examples of its usage. In the Cairngorm Store, there is always a notion of the "currently selected product." This is the item that the user has selected. The application stores an instance of a ProductVO in the model locator, uses the selectedItem property in the application to render the ProductDetails panel, and uses the property to decide what to add to the shopping cart when the user presses the Add to Cart button.

See the ModelLocator source in the following entry:

```
public static var selectedItem : ProductVO;
```

You will learn how to use data binding to ensure that the currently selected item always appears in the product details view.

Additionally, through ModelLocator, you can find the list of products for sale in the Cairngorm Store from anywhere in the application by storing an ArrayList of these products on ModelLocator, as follows:

```
public static var products : Array; // con-
taining ProductVO objects
```

Also notice how the application stores constants in ModelLocator. In this application, these constants allow you to track the state of the application (there are three possible states) as follows:
public static var workflowState : Number;

```
//------------------------------------
------------------------

public static var VIEWING_PRODUCTS_IN_
THUMBNAILS : Number = 1;
public static var VIEWING_PRODUCTS_IN_
GRID : Number = 2;
public static var VIEWING_CHECKOUT :
Number = 3;
```

Finally, you have a complex ShoppingCart component that encapsulates all the functionality associated with adding and removing items to a list of items that the customer may want to purchase. You stored the shoppingCart instance on the ModelLocator and initialized it within the ModelLocator in the initialise() method.

Having all the attributes on the Model Locator pattern as static attributes ensures that the Model Locator pattern is a simple implementation of a singleton. You ensure, for instance, that one and only one instance of a ShoppingCart exists per user.

The single instance of ModelLocator is initialized immediately when the main application is initialized. In the Main.mxml file (the entry point for the Cairngorm Store) you have declared the following method as a handler for the creationComplete event on the main Application tag:

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

    import org.nevis.cairngorm.sam-
ples.store.model.ModelLocator;

    private function onCreationCom-
plete() : Void
    {
        ModelLocator.initialise();
    }

]]>
</mx:Script>
```

This code ensures that all the client-side state (such as our shopping cart) correctly initializes when the application starts.

Next, I'll take a look at data binding to the ModelLocator in greater detail. But to whet your appetite, let me say that when you create components that rely upon client-side data rather than adopt any of the convoluted strategies that result in a brittle implementation of the view, you can instead locate client-side state by binding to the singleton ModelLocator from any component in the application.

By way of example, we have a SideArea.mxml component in the Cairngorm Store that contains the ProductDetails and ShoppingCart components. ProductDetails requires the currently selected item, so that it can display the name, description, price, and image of the item the user has selected. Because the currently selected item is stored in ModelLocator as an instance Product VO called selectedItem, you can pass this item into our ProductDetails component as shown below:

```
<details:ProductDetails
        id="productDetailsComp"
        width="100%" height="325"
        currencyFormatter="{
ModelLocator.currencyFormatter }"
        selectedItem="{ ModelLocator.
selectedItem }"
        addProduct="addProductToShopp
ingCart( event )" />
```

The highlighted line of code shows that within the ProductDetails.mxml file, there is an attribute called selectedItem that is an instance of a Product VO. You have just used data binding (the curly brace notation) to ensure that anytime selectedItem on ModelLocator is updated, the ProductDetails view updates accordingly.

The continuation of this article can be found at www.macromedia.com/devnet.

Steven Webster is the practice director for Rich Internet Applications at Adobe. Previously, Steven was the technical director at iteration::two, a world-leading Rich Internet Application consultancy based in Edinburgh, Scotland. Steven is the author of Reality J2EE: Architecting for Flash MX and coauthored ActionScript 2.0 Design Patterns for Rich Internet Applications (ActionScript 2.0 Dictionary) and Developing Rich Clients with Macromedia Flex with Alistair McLeod. Steven speaks regularly at conferences and user group meetings on technical and business aspects of RIAs. Steven is the core contributor to the open-source Cairngorm project, a micro-architecture for RIAs based on J2EE patterns which was innovated by iteration:: two over a number of Flash and Flex RIA developments.

Advertising Index

Advertiser	URL	Phone	Page
CFDynamics	www.cfdynamics.com	866-233-9626	19
CFUNITED			39
Community MX	www.communitymx.com		23
EdgeWeb	http://edgewebhosting.net	1-866-334-3932	5
Hosting.com	www.hosting.com		27
InterAKT	www.interAKTonline.com/macromedia	4031 401.68.19	3
Intermedia	www.intermedia.net	1-888-379-7729	11
Macromedia Studio 8	www.macromedia.com/go/8_studio8	415-252-2000	52
Stream57	www.stream57.com	212-909-2550x1012	6
Vitalstream	www.vitalstream.com	800-254-7554	2
WebAppCabaret	http://webappcabaret.com/jdj.jsp	1-866-256-7973	51

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