SYS-CON MEDIA

MX TRACK: EDGE 2004 EAST P. 57

2.2

MX developer's journal

FOR MACROMEDIA MX DEVELOPERS & DESIGNERS

february 2004 www.mxdj.com

7 Habits of Highly Effective ColdFusion Developers

DREAMWEAVER
Take the Drag Out of Coding

FLASH
Web Typography

FIREWORKS
The Science of Color

FREEHAND
Library Power

DIRECTOR
A Little Something Xtra

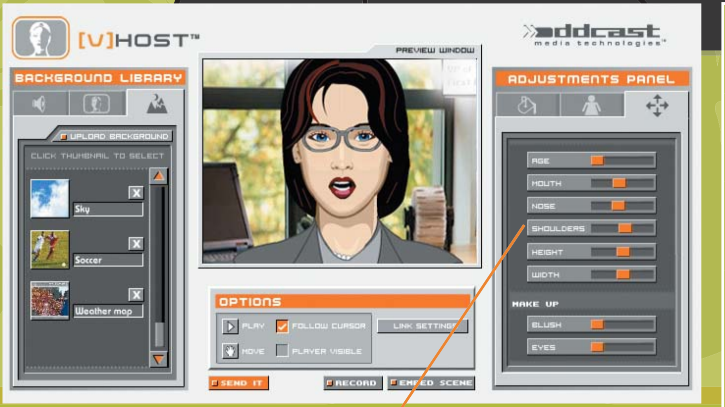
$5.99US $6.99CAN

02>

0 71486 02978 6

CREATE TALKING CHARACTERS IN A FLASH.

COLORING PANEL
CHOOSE THE FEATURE THAT YOU WOULD LIKE TO COLOR
EYES
HAIR
MAKE-UP
MOUTH
SKIN
Skin Color
CLICK THE BUTTONS BELOW TO ACCESSORIZE YOUR MODEL
TOPS
BEARDS
HAIR
HATS
SHADES
JEWELS
Clothing
Hair Styles



Main Features:

Export as swf
Lip Synchronization
Dynamic Text-To-Speech (with no Plug-in!)
Record by mic or telephone.
Advanced ActionScript and JavaScript APIs.
Advanced content management and workflow features
Usage and performance reports
Personification features:

- Set Age
- Set Color (Skin, Eyes, hair)
- Set Size (Shoulders, head, nose, mouth)
- Set Clothing
- Set Accessories
- Set Background
- Set Audio

oddcast
media technologies™

The [V]Host Studio is a tool that enables professionals to design and deploy interactive talking characters in minutes. Output as a .swf file, the characters can be embedded in any Flash movie. Record a voice or use our Text-To Speech features to read dynamic text, in 14 different languages, then watch the characters breath, talk and come to life. Non-programmers can learn to use the [V]HostT studio in minutes. Developers can use our ActionScript and JavaScript API's to make the character interact with users through wide range of actions and behaviors.

Winner of the following awards:
European Excellence in Digital Media Award 2003
Silver Medal, Spin Awards 2003
Bronze Medal, I.D. Magazine, Interactive Design 2003
Best Application award, FlashForward 2002

www.oddcast.com

Call (212) 375-6290

for developer's kit

Flexible E-Commerce that can adapt to ANY Product, ANY Customer, ANY ISP

Why Decide? *Open-source* PHP and ASP for the price other shopping cart vendors charge for one platform.

ComCity® Corporation has been building shopping carts and hosting E-commerce solutions since 1995. Our experience extends from small shops to multi-server, multi-geographic Enterprise E-commerce.

✱ ***ISP Friendly***™ script-based checkout system without complicated dlls, available for Windows (ASP) as well as UNIX (PHP) hosting providers – host with 99.9% of all ISP's without special server installation support.

✱ ***Dual product mode*** allows designers to create a cart quickly in *static* mode or use *dynamic* mode for script-based flexibility.

✱ ***Tight Dreamweaver MX integration*** allows developers to use Database Server Behaviors to modify Catalog design with ease and flexibility.

A Shopping Cart *Foundation* for Dreamweaver E-commerce Developers

- Intuit® Quickbooks Integration
- Affiliate Control Panel
- Customer Control Panel
- E-mail Marketing Tools
- Sales Summary Reports
- Market/Auction based pricing
- Online self-paced training

Buy SalesCart for Windows and LINUX at the same time and receive 25% off your combined base software order.

See http://www.salescart.com/syscon for details. Expires 3/1/04

SalesCart™

The first **Design-Tool E-commerce software** solution in the world™

For more information, visit www.salescart.com or call 1-888-826-6248 (Toll Free)

MX developer's journal

february 2004

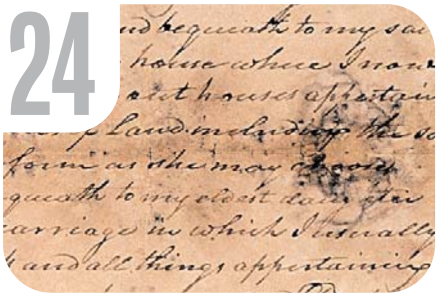
The Tag That Takes the Drag Out of Coding
Getting the most from Macromedia's DataSet tag
by heidi bautista

Web Typography
An oxymoron you can work with
by tom green

The Science of Color
Layers, coupled with blend modes and opacity, can add stunning effects to our graphics
by charles e. brown

10

24



34

20 JavaScript and Flash Panels
Increasing productivity and empowering innovation
by jesse r. warden

7 Getting the Most from the MX Community
There are a lot of resources out there
by robert diamond

Industrial-Strength multimedia
...without learning a new language."

Director now supports JavaScript syntax.

Thomas Higgins
Senior Engineer, Director MX 2004 product team

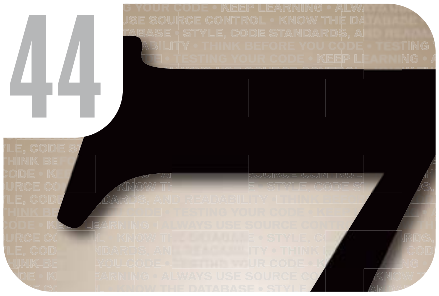
macromedia
Your Account | International
Home | Products | MX 2004 | Support | Developers | Solutions | Downloads | Company | Store
Search
Products
Studio MX 2004 with Flash Professional
Streamline web development with an integrated tool set
Buy | Try
Shortcuts
Download Flash Player
Download Shockwave Player
Free Trials
Free Online Product Seminars and Demos
Tutorials and Samples
Update available
Flash MX 2004 Updater Available
Update to Flash 7.0.1

Library Power
Symbology for beginners – the use of symbols and libraries can make your workflow much smoother
by ron rockwell

38

7 Habits of Highly Effective ColdFusion Developers
Good programmers will turn these winning CF strategies into habits
by robi sen

44

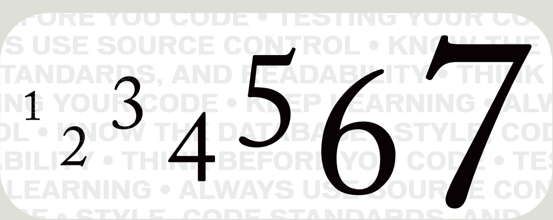


A Little Something Xtra
Cross-platform RAD that stretches to meet your needs
by tab julius

50

on the cover

A helpful quide to practical ideas and habits will allow you to develop and create better ColdFusion applications regardless of which CF version you use, hardware you run on, or methodology you employ.



32 xile
Cartoon
by louis f. cuffari

74 vanguard
Ascension
by robert immings

American Media

http://www.americanmediatraining.com

800 278 7876

Training

Authorized training partner since 1995

Experienced instruction from beginnner to expert level

Clients include movie studios, government agencies, fortune 500, military

The first step to certification

Always competitive pricing

Free technical support

Custom courses - tailored to suit

On-Site
Los Angeles
Denver
Las Vegas
Worldwide

Flash
ColdFusion
Dreamweaver
Fireworks
Contribute
Authorware
Director
Coursebuilder

Group Publisher Jeremy Geelan
Art Director Louis F. Cuffari

EDITORIAL BOARD
Dreamweaver Editor
Dave McFarland
Flash Editor
Jesse Warden
Fireworks Editor
Kleanthis Economou
FreeHand Editor
Louis F. Cuffari
ColdFusion Editor
Robert Diamond
Director Editor
Niqui Merret

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

EDITORIAL
Executive Editors
Gail Schultz, 201 802-3043
gail@sys-con.com
Jamie Matusow, 201 802-3042
jamie@sys-con.com

Editors
Nancy Valentine, 201 802-3044
nancy@sys-con.com
Jean Cassidy, 201 802-3041
jean@sys-con.com
Jennifer Van Winckel, 201 802-3052
jennifer@sys-con.com

Technical Editors
James Newton • 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
Carrie Gebert, 201 802-3026
carrieg@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.

MX
developer's journal

Getting the Most from the MX Community

There are a lot of resources out there

by robert diamond

When Macromedia first announced last year that it would be joining the old UCon and DevCon together, combining the best of the two events into "Macromedia MAX 2003" – subsequently held in Salt Lake City in November – there were the inevitable rumblings in the user/developer community.

By now, however, everyone has seen that not only did joining those two events make good business sense, it also made sense in the context of the enormous impetus that it gave to the MX product suite.

Last month's announcement of the "MX 2004" product line, with its new versions of Dreamweaver, Flash, and Fireworks, as well as Macromedia Flash MX Professional 2004, ought to have silenced anyone who still had any lingering doubts. Whether looking at new building blocks like the MX Elements for Flash and the MX Elements for HTML, or at the new look-and-feel for Internet applications, called Halo, MX developers – yes, let's use that term – are increasingly sitting in the catbird seat.

Flash Player 7 was also announced last month, with performance optimizations and support for new technologies. And let's not forget "Macromedia MAX 2004 Japan." Both developments confirm just how all-encompassing the commitment is to taking rich media apps to the max.

What does all this mean in real terms for you as a developer? It means, above all, that Macromedia is building out its offerings. But it is also building out the community of developers using them.

Let's look at just one of the many technologies within the MX family, ColdFusion, as it's the one I know best. Every month when writing the editorial for ***ColdFusion Developer's Journal***, one of ***MXDJ***'s sister publications in the SYS-CON Media family, I try to encourage our readers to get involved in the developer community. I get technical questions e-mailed to me all the time, with folks looking for advice on where to go to solve specific problems that they're having, or to get in touch with other members of the community to help them learn and grow. I often assume that everyone knows the major resources that are there to help, but it's good every now and then to have a reminder.

In addition to the magazine you're holding, and *ColdFusion Developer's Journal*, there's a wealth of community resources for both ColdFusion and MX developers at no, or little, cost that are well worth taking advantage of. Having covered many developer markets over the past several years, I can unequivocally state that we've got one of the finest groups of folks, more willing to share with and help each other than any other.

Here are the "Big Four":

- ***Macromedia's Forums:*** www.macromedia.com/support/forums/

Every self-respecting developer using one of Macromedia's products should visit their very extensive support forums on a daily basis. Each gets hundreds of posts a day on a variety of topics, and they're monitored by Macromedia as well. It's a great way to see who else is having the problems you are, as well as learn the techniques that others in similar situations are using.

There's no company out there (that I've found at least) that's more responsive to its core customer base than Macromedia, and we've seen several of the MX 2004 products (most notably Flash) receive updates over the past couple of months in response to customer problems that have now brought the MX 2004 line into many more production environments. You can get the latest information on these on the forums as well.

- ***User Groups:*** www.macromedia.com

Robert Diamond is editor-in-chief of ColdFusion Developer's Journal *and ColdFusion editor of* MX Developer's Journal. *Named one of the "Top thirty magazine industry executives under the age of 30" in Folio magazine's November 2000 issue, Robert graduated from the School of Information Studies at Syracuse University with a B.S. in information management and technology.*
robert@sys-con.com

/cfusion/usergroups/index.cfm

You can find user groups specializing in ColdFusion, or Macromedia products as a whole, in many local areas throughout the country and the world. These generally meet once a month and range in size from a handful to a hundred attendees, with periodic visits from Macromedia guests. These are great for networking, as well as technical tips and shared learning.

- ***Blogs:*** www.markme.com/mxna/index.cfm

Several Macromedia employees – as well as many independent developers and consultants – have gotten into this latest craze in Web publishing, posting about their daily trials, tribulations, and triumphs using the various MX products. Macromedia's XML news aggregator makes things even easier because you can view the latest posts in these by technology, or overall as they aggregate (as of this writing) 208 different blog feeds.

- ***Events:*** www.macromedia.com/macromedia/events**;** www.sys-con.com/edge2004/

Macromedia MAX is a don't-miss event for developers using any products in the Macromedia MX family. It's an invaluable resource for support, great technical sessions, demos of Macromedia products, third-party solutions, and lots more. Watch for news on next year's U.S. event.

In addition to MAX, there are dozens of national and local events that you can participate in. This month, for example, you can find many of the key players in developing MX technologies at Edge 2004 (East) in Boston, where there's an entire MX track reflecting the momentum that Macromedia is helping to build around its product line. The conference is the eighth successive "Edge" event produced by SYS-CON Events, a sister company to the publishers of *MXDJ*.

You'll be able to learn from senior evangelist Christophe Coenraets how to use Macromedia's "Flex" technology initiative to create rich Internet apps – using components, layouts, and managers – to build user interfaces, as well as using Flex's XML-based language, MXML, to create and manipulate client-side data models. Senior sales engineer Kevin Hoyt will teach you how the Flex initiative fits into today's new service-oriented architectures, with a look at design patterns for rich clients, accessing Web services, and securing your Flex application.

Simon Horwith will help you see how, compared to other application server platforms, ColdFusion MX makes creating and consuming Web services easy. True to the MX 2004 theme of interoperability, Simon's session will look at how Dreamweaver MX 2004 allows you to create a ColdFusion Component (CFC) in Dreamweaver, then expose that CFC as a Web service by just toggling one attribute of the component. His session will also explore built-in support for browsing and consuming Web services in Dreamweaver MX 2004, as well as the advantages of using Macromedia products to design and develop Web services–enabled applications from start to finish.

And for the ColdFusion faithful, there's a session on the basic principles of tuning software for performance, concentrating on tuning Web applications written in ColdFusion. Jim Schley talks from experience as the lessons he's learned and will discuss have come from working on the latest version of ColdFusion, CFMX 6.1 (Red Sky).

For those in search of a 35,000-foot overview, one of the Edge 2004 (East) keynoters is Macromedia's own David Mendels, who, since joining Macromedia in 1992, has played a major role in the development of product and business strategies. In his current role, David supervises the gamut of Macromedia's designer and developer products, including Flex, Flash, Dreamweaver, and ColdFusion – so his keynote presentation, intriguingly titled "SOA + RIA = ROI" is a definite must-see.

You can visit the Web sites above for full details on this and other conferences that Macromedia is participating in, either as a sponsor or a presenter. Additionally, Macromedia runs local training seminars and product showcases as new products are released. Generally run in a hotel or local training center in your area, they give you the opportunity to go and see the products firsthand, and to learn from the experts.

Until next time, happy developing – and don't forget to comment your code!

SYS-CON MEDIA
President & CEO
Fuat Kircaali, 201 802-3001
fuat@sys-con.com
Vice President, Business Development
Grisha Davida, 201 802-3004
grisha@sys-con.com
Group Publisher
Jeremy Geelan, 201 802-3040
jeremy@sys-con.com
Technical Director
Alan Williamson, 201 802-3040
alan@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
Director, Sales & Marketing
Megan Mussa, 201 802-3023
megan@sys-con.com
Advertising Sales Managers
Alisa Catalano, 201 802-3024
alisa@sys-con.com
Carrie Gebert, 201 802-3026
carrieg@sys-con.com
Associate Sales Managers
Kristin Kuhnle, 201 802-3025
kristin@sys-con.com

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

SYS-CON.COM
Vice President, Information Systems
Robert Diamond, 201 802-3051
robert@sys-con.com
Web Designers
Stephen Kilmurray, 201 802-3053
stephen@sys-con.com
Christopher Croce, 201 802-3054
chris@sys-con.com
Online Editor
Lin Goetz, 201 802-3045
lin@sys-con.com

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

EVENTS
President, SYS-CON Events
Grisha Davida, 201 802-3004
grisha@sys-con.com
Conference Manager
Lin Goetz, 201 802-3045
lin@sys-con.com
National Sales Manager
Sean Raman 201-802-3069
raman@sys-con.com

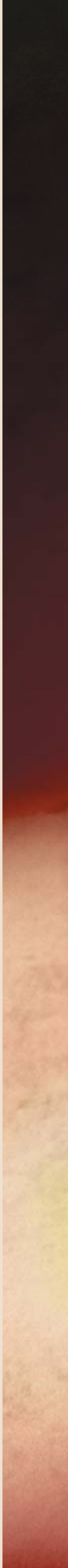
CUSTOMER RELATIONS
Circulation Service Coordinators
Shelia Dickerson, 201 802-3082
shelia@sys-con.com
Edna Earle Russell, 201 802-3081
edna@sys-con.com
Linda Lipton, 201 802-3012
linda@sys-con.com
JDJ Store Manager
Brundila Staropoli, 201 802-3000
bruni@sys-con.com

SYS-CON MEDIA

the tag that takes the drag out of coding

If you're using Dreamweaver to build ASP.NET-driven Web sites, you've undoubtedly encountered DataSets. The DataSet tag, one of Macromedia's custom tags, interacts with a database via SQL statements like SELECT, INSERT, UPDATE, DELETE. You can also use it to execute stored procedures. The DataSet tag is to ASP.NET what the recordset is to ASP. Except that it's better, of course.

by heidi bautista



The most obvious application of the DataSet tag is to select data from a table in your database. The resulting dataset can be displayed in a variety of ways: in a DataGrid, DropdownList, or pretty much wherever you want in your page.

Another handy, though less obvious, function of the DataSet tag is to change data in a table (e.g., insert, update, or delete). In this case, the DataSet tag doesn't need to create an actual dataset, just perform some function.

In this article you'll learn about the attributes and methods available with the DataSet tag. I assume you've already set up an ASP.NET site within Dreamweaver – using either the Visual Basic or C# ASP.NET server model. (If you're not sure how this is done, select "Getting Started and Tutorials" from the Help menu.)

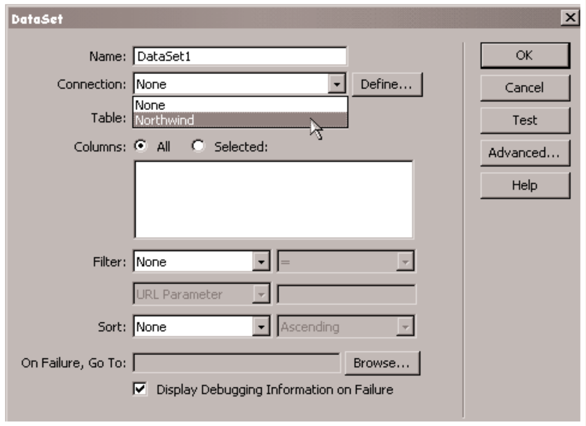
What's in a Name?

The DataSet tag written by Macromedia shouldn't be confused with the DataSet class included in the .NET Framework. Sure, they share the same name, but the two are quite different. Macromedia's DataSet is a custom tag. The Microsoft DataSet is a class in the .NET Framework. Macromedia's DataSet tag actually uses the .NET DataSet class within its implementation. In this article, the term DataSet can refer either to the DataSet tag or to the set of data created by the DataSet tag (I'll try to make it crystal clear which one I mean). But DataSet will *not* refer to the .NET DataSet class unless it is explicitly qualified by being preceded with ".NET."

DataSet Attributes

The DataSet tag has several attributes that return commonly needed information like the number of records, the current page, etc. Even though the DataSet has a slew of attributes, only three are required: CommandText, ConnectionString, and runat. (Table 1 lists all of the DataSet attributes.)

image I



DataSet

Name: DataSet1

Connection: None

None
Northwind

Table:

Columns: ● All ○ Selected:

Filter: None =

URL Parameter

Sort: None Ascending

On Failure, Go To:

☑ Display Debugging Information on Failure

Required attributes

CommandText

The CommandText attribute can be either a SQL statement or the name of a stored procedure. If you want to use a stored procedure, you also have to use the IsStoredProcedure attribute. Set the Boolean IsStoredProcedure attribute to true to indicate that the string in the CommandText attribute is the name of a stored procedure. It turns out that the default value of IsStoredProcedure is false, so if you just leave it off the CommandText is automatically interpreted as a SQL statement.

Here's an example of using a SQL statement for the CommandText:

```
CommandText=
   "SELECT LastName, FirstName FROM
    dbo.Employees ORDER BY LastName"
```

Tip: *The CommandText can also contain a binding expression. For example:*

```
CommandText='
<%# (bUseNickname) ?
"SELECT RealName, Nickname
FROM dbo.Employees ORDER BY Nickname"
:
"SELECT RealName, Nickname
FROM dbo.Employees ORDER BY RealName"
%>'
```

This ternary expression stipulates that:

- *If bUseNickname is true, the SELECT statement will be ordered according to the Nickname field.*
- *If bUseNickname is false, the SELECT statement will be ordered according to the RealName field.*

FYI: *You must put quotes around the attribute's value. That is, around the entire binding expression.*

You cannot break up the binding expression over multiple lines the way it's shown here. The entire binding expression has to be on one line. (You'll get a nasty runtime error otherwise.)

Parameterized SQL Statements

Certain portions of the SQL statement used in the CommandText can be parameterized. Parameterization can simplify the SQL statement, making it easier to read.

Only the value in a name/value pair within the SQL statement is legal for parameterization. Table II lists some of the name/value pairs that are candidates for parameterization.

The WHERE clause is a prime example of where a name/value pair is used. Keep in mind that the name and value need not be adjacent to each other (see the INSERT example). The ORDER BY clause cannot be parameterized since it doesn't follow the name/value paradigm.

The syntax for the parameter depends on which type of connection you are using: OLE DB or SQL Server.

Consider the following example of a SELECT statement:

```
SELECT *
FROM myTable
WHERE col1="one" AND col2="two"
ORDER BY col2
```

If you replace the values in the name/value pairs in the WHERE clause with parameters, the statement becomes:

SQL Server connection

```
SELECT *
FROM myTable
WHERE col1=@param1 AND col2=@param2
ORDER BY col2
```

- Parameters must be uniquely named and start with the @ symbol.
- SQL Server connection parameters can be in any order.

OLE DB connection

```
SELECT *
FROM myTable
WHERE col1=? and col2=?
ORDER BY col2
```

- All parameters use the ? symbol.
- With OLE DB connections, the order of the parameters is important. The placeholders used in the name/value pairs must correspond exactly to the order of the <Parameter> tags used to define the placeholders.

The parameters are defined using the <Parameter> tag. Each DataSet can have multiple <Parameter> tags defined and all must be enclosed within the <Parameters> tag.

Continuing with the example given above, here are the required <Parameter> tags to define the parameters (depending on the connection type used).

SQL Server connection

```
<Parameters>
 <Parameter Name="@param1" Value='one'
  Type="String" />
 <Parameter Name="@param2" Value='two'
  Type="String" />
</Parameters>
```

OLE DB connection

```
<Parameters>
 <Parameter Value='one' Type="String"
  />
 <Parameter Value='two' Type="String"
  />
</Parameters>
```

- Notice that the <Parameter> tags for the OLE DB connection appear in the same order as the parameters are used in the SELECT statement. This is very important since all the parameters share the same name (the question mark, ?).
- The data type stipulated for the parameters must match the data type of the parameter. For example, in the <Parameter> tags above, the type is "String" but if the Value of @param1 was the integer 1, then the data type would be Integer (OLE DB) or Int (SQL Server).

Table III lists the legal data types for OLE DB and SQL Server connections.

ConnectionString

The required ConnectionString parameter contains instructions for connecting to the database. You could hand-code this information, but the easier way to handle it is to create a connection. (Unless otherwise specified, the examples in this article use a SQL Server connection.)

In the DataSet dialog all you have to do is select one of the predefined connections from the Connection dropdown menu and voilá, the ConnectionString attribute is written for you (see Image I).

Here's an example of a ConnectionString:

```
ConnectionString=
  '<%#
   System.Configuration.
   ConfigurationSettings.
   AppSettings["MM_CONNECTION_STRING_
   Northwind"] %>'
```

(Keep in mind that you can't put a carriage return in the middle of a binding expression. Refer to the tip on page 12 for an explanation.)

runat="server"

The DataSet tag is a server-side control; it is processed on the server before being sent to the client. The default value for the runat attribute is "client" so you must explicitly set it to "server." Dreamweaver does this for you when it creates a DataSet tag.

Optional Attributes

The DataSet's optional attributes are grouped according to their function:

- Attributes that further refine the CommandText attribute
- Post-processing attributes
- Paging attributes
- Culling attributes
- Record-counting attributes
- Computed or created attributes

Attributes That Further Refine the CommandText Attribute

CreateDataSet

This optional Boolean attribute determines whether or not a dataset is created. That might seem odd since you'd think that the DataSet tag's whole purpose is to create sets of data. But remember that you can also use the DataSet tag to insert, update, delete, etc. Turns out that nothing bad will happen when you leave the CreateDataSet attribute at its default value of true when no records are returned. But it's good practice to set it to false in these cases.

The DataSet dialog does not have a checkbox for you to specify this value. If you want to set it to false, you have to do it directly in code view.

```
CreateDataSet="false"
```

Tip: *Notice that the value for this Boolean attribute is in double quotes. In order to be XHTML compliant, the values in all name/value pairs must be quoted (even if the value is a Boolean).*

Expression

The Expression attribute is a particularly powerful one. It's also a Boolean whose default value is true. The value of the Expression attribute determines if the CommandText is executed.

If a DataSet tag is sitting on your page and its Expression attribute is either absent or set to true, it will execute when the page loads. But what if the purpose of the DataSet tag is to update the database based on values the user is supposed to enter on the page? Instead of running each time the page loads, what you really want is for the DataSet to execute after the page has been submitted (e.g., posted back). You can accomplish that goal with the following:

```
Expression='<%# IsPostBack %>'
```

(Of course, you'll probably want to do some error checking to make sure all the required information is available first.)

The DataSet dialog does not have a checkbox for you to specify the value of the Expression attribute. You'll have to type it directly into the code if you want its value to be anything other than the default value of true.

IsStoredProcedure

And yet another Boolean attribute. The IsStoredProcedure attribute (default is false) is used in conjunction with the CommandText attribute. If your CommandText is the name of a stored procedure, then this attribute must be true.

```
IsStoredProcedure="true"
```

I bet you can't find a place on the DataSet dialog to set the IsStoredProcedure value. That's because Dreamweaver has an entirely separate Server Behavior for DataSets that execute stored procedures (see Image II).

It's still a DataSet, but Dreamweaver has simplified matters by providing a specific Server Behavior just for DataSets that execute stored procedures (see Image III).

Notice that it looks very similar to the DataSet dialog shown in Image I, but it prompts for the name of the stored procedure and its parameters rather than asking you to select a table and columns. In addition, the Stored Procedure dialog has a checkbox labeled Returns DataSet. The default is unchecked, which means table I

attribute name	data type	required?
Required attributes		
ConnectionString	string	yes
CommandText	string	yes
runat	string	yes
Refining the CommandText attribute		
CreateDataSet	Boolean	no
Expression	Boolean	no
IsStoredProcedure	Boolean	no
TableName	string	no
Post-processing		
Debug	Boolean	no
FailureURL	string	no
SuccessURL	string	no
Paging		
CurrentPage	int	no
PageSize	int	no
Culling		
StartRecord	int	no
MaxRecords	int	no
Record Counting		
GetRecordCount	Boolean	no
RecordCountCommandText	string	no
Calculated (read-only) or Created		
RecordCount	int	calculated
EndRecord	int	calculated
LastPage	int	calculated
DefaultView	DataView	calculated
theDS	DataSet	created

table II

SQL Statement	Name/Value Pairs Are Used In
SELECT, DELETE, etc.	WHERE clauses
INSERT	field name and its value
UPDATE	field name and its value

default value

""
""
"client"

true
true
false
"theTable"

false
""
""

0
0

0
0

true
""

0
0
0
---
new DataSet()

Example

```
SELECT * FROM myTable WHERE col1="one"
INSERT INTO myTable (col3, col4) VALUES (30, 40)
UPDATE myTable SET col5="five"
```

SQL Server connections

Type attribute values

Int
SmallInt
TinyInt
BigInt
Decimal
Float
Char
NChar
VarChar
NVarChar
Text
NText
Real
Money
SmallMoney
TimeStamp
DateTime
SmallDateTime
Bit

OLE DB connections

Type attribute values

Boolean
Decimal
Double
Numeric
Single
Integer
UnsignedInt
SmallInt
UnsignedSmallInt
TinyInt
UnsignedTinyInt
BigInt
UnsignedBigInt
Char
WChar
VarChar
LongVarChar
VarWChar
LongVarWChar
VarNumeric
Currency
Date
DBDate
DBTime
DBTimeStamp
Filetime
BSTR

that a DataSet is not created (in other words, the CreateDataSet attribute is false).

Dreamweaver's user interface separates the DataSet and Stored Procedure server behaviors into two separate commands for convenience and clarity in the dialogs. But keep in mind that they share the underlying code that supports the DataSet tag. Read through the DreamweaverCtrls.cs file to convince yourself.

If you installed Dreamweaver MX 2004 in the default location, you will find the Dreamweaver Ctrl.cs file in C:\Program\Files\Macromdia\Dreamweaver MX 2004\Configuration\Server Behaviors\Shared\ASP.net\Sripts\Source.

TableName

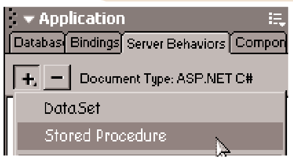
The TableName attribute is the name of the table that data is drawn from (default value is theTable). When you want information about the dataset most often you'll refer to the dataset's name rather than the table name. But the attribute is here for you in case you want to get into the guts of the custom DataSet tag. (Not surprisingly, the DataSet dialog doesn't have a spot on it for you to indicate the TableName. If you want to stipulate the name of the table you'll have to make this change directly in the code.)

Post-Processing Attributes

Debug

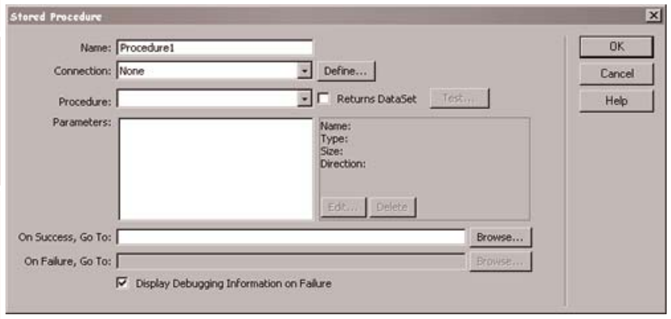
The Debug attribute should be used during development only. The default value is false. When you set Debug to true and an exception is thrown you'll get some fairly intelligible comments from Dreamweaver to assist you in debugging the problem. You'll want to set this attribute to false (or just delete it altogether) before deploying the final version of the page so your site visitors don't get what looks like gobbledygook to them. The DataSet dialog has a checkbox for this attribute.

image II



Document Type: ASP.NET C#

image III



Name: Procedure1
Connection: None
Procedure:
□ Returns DataSet
Parameters:
Name:
Type:
Size:
Direction:
On Success, Go To:
On Failure, Go To:
☑ Display Debugging Information on Failure

Also keep in mind that if you set Debug to true, the FailureURL attribute is ignored. When an exception is thrown, you'll get the debugging information instead of being redirected to the page specified by FailureURL.

Code 1 is an example of a DataSet tag with a problem with the ConnectionString attribute (it doesn't exist). Image V shows the debugging information you'll see when you run the page.

Obviously Dreamweaver hasn't pinpointed the exact problem, but it does give you a pretty broad hint that you should look at the ConnectionString attribute.

FailureURL

The FailureURL attribute specifies the page that will display when an exception is thrown from the DataSet tag. If you have Debug="true", the FailureURL attribute is ignored. Once you have the bugs worked out, set Debug to false and create a user-friendly FailureURL page.

SuccessURL

This attribute indicates the page to display when the DataSet's CommandText has executed successfully. For example, suppose that the purpose of your DataSet tag is to update data in the database. Once it has done that successfully the SuccessURL page will appear saying that the update was successful. If you don't set this attribute, the page simply redisplays again.

Paging Attributes

CurrentPage and PageSize

When you have CreateDataSet="true" the DataSet tag returns all the records corresponding to the SQL statement specified in the CommandText attribute. If you're working with large amounts of data the size of the resulting dataset can easily bog down the performance of your page. Since you're probably not going to display a zillion rows on the page anyway, why get all the records?

The CurrentPage and PageSize attributes are meant to be used together. They tell the DataSet tag how many rows to get (the PageSize) and from where in the table to get them (CurrentPage). Note that the PageSize is zero-based so the first page corresponds to CurrentPage="0".

Consider the following example: let's say that your CommandText's SQL statement returns 63 rows. If you specify a PageSize of 10, that means you will have seven pages. The first six pages will have 10 rows each and the seventh page will have the remaining 3 rows. If you want to see the page with the 22nd row, then the CurrentPage would be 2.

***Tip:** If you're creating a dataset to be used as the data source for a DataGrid, be sure to use Dreamweaver's user interface to create the DataGrid. They've built in the correspondence between the DataSet's PageSize attribute and the DataGrid's PageSize attribute.*

Culling Attributes

StartRecord and MaxRecords

Another way to indicate that you want just a subset of records rather than all the records corresponding to the SQL statement specified in the CommandText attribute is to use the StartRecord and MaxRecords attributes.

StartRecord is zero-based, so if you want the 10th through 25th record, then StartRecord="9" and MaxRecords="16". That is, StartRecord indicates the first record to get and MaxRecords indicates the total number of records to get.

***Tip:** Regarding the usage of the PageSize/CurrentPage and StartRecord/MaxRecords attributes:*

- *Use either PageSize and CurrentPage or StartRecord and MaxRecords, not both pairs of attributes. If both sets of attributes are specified, PageSize and CurrentPage are used and StartRecord and MaxRecords are ignored.*
- *If PageSize is greater than zero StartRecord is calculated automatically as (PageSize * CurrentPage).*

Record-Counting Attributes

GetRecordCount

This Boolean attribute defaults to true, meaning that the number of records in the DataSet will be calculated. If you don't need this value for your application, set the GetRecordCount to false. In fact, under certain circumstances you can

improve performance with GetRecordCount=false.

RecordCountCommandText

The RecordCountCommandText attribute allows you to specify an explicit SQL statement to compute the record count. Typically, the statement is something like this:

```
SELECT COUNT(*)
FROM myTable
WHERE ...
```

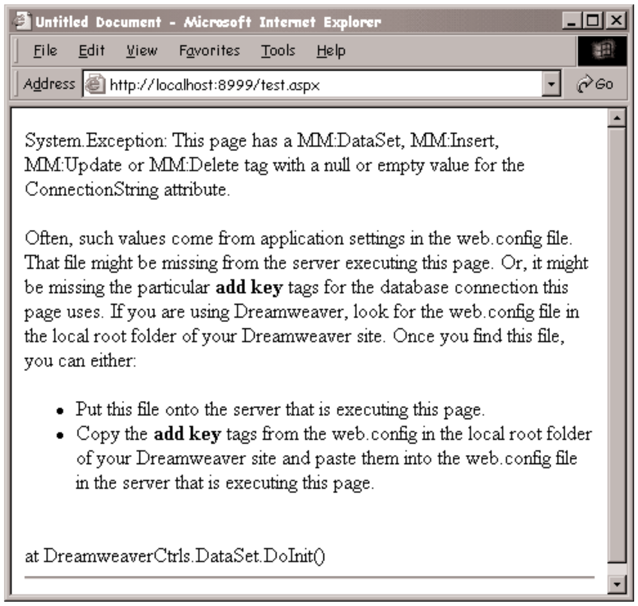
Using the RecordCountCommand Text attribute is particularly important (performance-wise) when you've set PageSize to be greater than zero.

It turns out that under the hood Dreamweaver uses the DbDataAdapter.Fill method to fill the .NET DataSet object with rows from the data source. The Fill method returns the number of rows added to the .NET DataSet (e.g., the record count). If PageSize is greater than zero, the value returned from DbDataAdapter.Fill is exactly equal to the PageSize. When you're looking for the record count, that's not the number you want. That means that the MM:DataSet code has to do some extra work to compute the actual record count.

This is how Dreamweaver handles record counting when PageSize is greater than zero:

1. If defined, the SQL statement in the RecordCountCommandText attribute is used to compute the record count.
2. Lacking the RecordCountCommandText attribute, Dreamweaver next attempts to parse the string in the CommandText attribute of the DataSet tag. This method works only if the CommandText uses a SELECT statement and doesn't contain a UNION operator.
3. Last and slowest, Dreamweaver will actually loop over the records in the .NET DataSet and count them.

image IV


Untitled Document - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
Address
http://localhost:8999/test.aspx
Go

System.Exception: This page has a MM:DataSet, MM:Insert, MM:Update or MM:Delete tag with a null or empty value for the ConnectionString attribute.

Often, such values come from application settings in the web.config file. That file might be missing from the server executing this page. Or, it might be missing the particular **add key** tags for the database connection this page uses. If you are using Dreamweaver, look for the web.config file in the local root folder of your Dreamweaver site. Once you find this file, you can either:

- Put this file onto the server that is executing this page.
- Copy the **add key** tags from the web.config in the local root folder of your Dreamweaver site and paste them into the web.config file in the server that is executing this page.

at DreamweaverCtrls.DataSet.DoInit()

So you can see why specifying your own RecordCountCommandText would be useful.

One last note on the RecordCountCommandText: you can't parameterize it.

Computed or Created Attributes

RecordCount

In order to get the record count, both the CreateDataSet (otherwise you don't have a dataset to count!) and GetRecordCount attributes must be true.

The simplest calculation for the record count occurs when both PageSize and MaxRecords are zero. The RecordCount corresponds to the number of records retrieved by the CommandText.

If MaxRecords is greater than zero, the value returned in the RecordCount attribute takes the MaxRecords and StartRecord values into account. RecordCount is equal to MaxRecords so long as MaxRecords is not greater than the number of records you actually have.

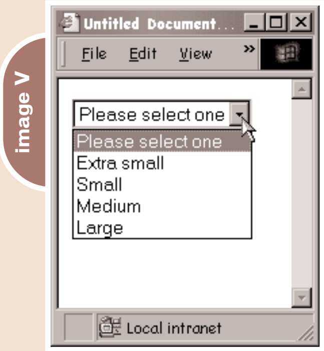
If PageSize is greater than zero, the RecordCount calculation totally ignores it. See the RecordCountCommandText section above, which describes how the record count is computed when the PageSize attribute is greater than zero.

***Tip:** For more information on the record count (including performance aspects), search the Dreamweaver-Ctrls.cs file for the CalculateRecordCount method and the GetRecordCount attribute.*

EndRecord

This calculated value is either [(CurrentPage + 1) * PageSize] or the RecordCount (whichever is smaller). You could easily compute this value yourself when you need it, but Macromedia decided to supply it for you.

image V



LastPage

The LastPage attribute is calculated only when the PageSize is greater than zero and GetRecordCount is true. It's zero-based and computed from the RecordCount and PageSize.

DefaultView

The data stored in the DataSet is just that: data. In order to use the data as a data source for another object (i.e., an asp:DataGrid or asp:DropdownList), you need to specify the DataSet's view. For example, to use an MM:DataSet in a DataGrid, the DataSource attribute would be:

```
<asp:datagrid
   ...
   DataSource="<%#
myDataSet.DefaultView %>
   ...
/>
```

The DefaultView represents all of the data retrieved by the DataSet tag and in the same order that it was retrieved. If you wanted a subset of the data and/or the data ordered differently, you would need to create a new view to associate with the DataSet or change the CommandText of the DataSet so that it gets the subset you want and/or orders it the way you want it ordered.

***Tip:** The MM:DataSet tag retrieves records from a single table in the database; but that's a limitation of the MM:DataSet tag, not of the .NET DataSet object. If you look at the documentation for the .NET DataSet object, you'll see that it can contain data from one or more tables. The theDS attribute of the MM:DataSet tag is equal to the DefaultView of the table 0 in the .NET dataset.*

```
myDataSet.DefaultView <==>
theDS.Tables[0].DefaultView
```

where:

- myDataSet is the ID of your MM:DataSet tag
- myDataSet.DefaultView is the default view of the data retrieved by the MM:DataSet tag
- theDS is the .NET dataset object returned by the MM:DatatSet tag
- theDS.Tables[0] indicates the first (and only) table within the .NET dataset object
- theDS.Tables[0].DefaultView is the default view of the first table of the .NET dataset object

Before you can define a new view, you need access to the .NET dataset. Fortunately for us, we have it with the theDS attribute.

theDS

Near the beginning of this article, I briefly mentioned that the Macromedia DataSet tag uses the .NET DataSet class. The theDS attribute *is* that .NET DataSet object. It holds the records obtained when the MM:DataSet CreateDataSet attribute is true. What's it good for, you ask? Well, truth is you probably won't need it on a regular basis, but theDS attribute does provide a powerful "hook" into the underlying .NET DataSet object.

Consider the following example: you have a dropdown list populated with the contents of a DataSet, but you want the first item (e.g., option tag) of the dropdown list to be "Please select one" and the option's value to be 0 (see Image V).

```
<select name="sizeDropdown"
id="sizeDropdown">
 <option value="0">Please select
 one</option>
 <option value="1">Extra
small</option>
 <option value="2">Small</option>
 <option value="3">Medium</option>
 <option value="4">Large</option>
</select>
```

You can programmatically add that first option tag by utilizing the theDS attribute to add a new row to the .NET dataset.

DataSet Methods

The DataSet tag has two methods to help you access particular pieces of data within a DataSet. The methods are FieldValue and FieldValueAtIndex (see Code II).

Both of the DataSet methods return a string corresponding to the value of a named field (FieldName) within some row in the DataSet.

The FieldValue method returns the value of the named field from the first row (row 0). FieldValueAtIndex takes an additional parameter that indicates the specific row number. Note that rows are zero-indexed.

Since FieldValue acts on row 0, the following two function calls are equivalent:

```
DS_uidAndPwd.FieldValue("emailaddr",
null)
DS_uidAndPwd.FieldValueAtIndex(0,
"emailaddr", null)
```

To Use a Container or Not

The last argument for both DataSet methods is the name of the container. You use either the word Container or null. If the call to FieldValue or FieldValueAtIndex is within a DataGrid, DataList, or Repeater tag, then use Container. Otherwise, use null. For example, if you're calling FieldValue from the Page_Load function, you don't have a container so you should use null as the last argument.

code I

```
<MM:DataSet
 id="DataSet1"
 runat="Server"
 IsStoredProcedure="false"
 ConnectionString=
 '<%# System.Configuration.
 ConfigurationSettings.
 AppSettings
 ["Fake_ConnectionString"] %>'
 DatabaseType=
 '<%# System.Configuration.
 ConfigurationSettings.
 AppSettings
 ["MM_CONNECTION_
 DATABASETYPE_Northwind"] %>'
 CommandText='<%# "SELECT *
 FROM dbo.Categories" %>'
 Debug="true"
>
</MM:DataSet>
```

code II

```
public string FieldValue(
  string FieldName,
  System.Web.UI.Control Container
)

public string FieldValueAtIndex(
  int Index,
  string FieldName,
  System.Web.UI.Control Container
)
```

Here's a quick example: you have a page with a DataGrid. The data source for the DataGrid is a DataSet that contains a list of e-mail addresses. Each row of DataGrid corresponds to a row in the DataSet so you have one e-mail address on each row.

The first thing you need to do in Dreamweaver MX is change the column data type from Simple Data Field to Free Form. This allows you to use an ItemTemplate. This is where you use the FieldValue("FieldName", Container) form of the method:

```
<asp:TemplateColumn
HeaderText="emailaddr" Visible="True">
  <ItemTemplate>
    <%# DataSet1.FieldValue("emailad
    dr", Container) %>
  </ItemTemplate>
</asp:TemplateColumn>
```

Using Container in the call to FieldValue ensures that the appropriate DataSet row is used.

Tip: *Curious about the meaning of Container? Check out the .NET documentation for the Eval method of the DataBinder class. The Container used in FieldValue and FieldValueAtIndex is identical to the Container used when you call DataBinder.Eval.*

On the other hand, you should use null when you want to use a value from the dataset somewhere outside of a DataGrid, DataList, or Repeater. The following example prints out a single e-mail address from the 5th row of the dataset.

```
<%# DataSet1.FieldValueAtIndex(4,
"emailaddr", null) %>
```

Conclusion

Well, there you have it. All the background information you need to start using the MM:DataSet tag. In its simplest form, the DataSet tag is quite easy to use. In its more complicated forms, the DataSet tag is very flexible. Everything that the DataSet tag does could be done without using Macromedia's custom tag but why bother when they've already done the legwork for you?

Heidi Bautista has contributed to a wide variety of Web sites as an independent consultant, building the dynamic chunks with ASP.NET typically in collaboration with HTML/CSS designers. You can find her articles at Macromedia's DevNet and on Community MX (www.communitymx.com) , where she provides in-the-trenches perspectives on topics like forms-based authentication and using Macromedia's custom tags. Heidi's professional nom de plume is Groovy bits (www.groovybits.com).

bautista@groovybits.com

JavaScript and Flash Panels

Increasing productivity and empowering innovation

by jesse r. warden

Macromedia products are ever more extensible in the 2004 releases. They make it easier for developers to create custom tools and extension scripts and then package those scripts and interfaces to distribute to their fellow designers and developers. Distribution comes in a few ways; in this article I discuss the use of JavaScript and Flash panels for use in Fireworks and Flash, and briefly cover the use of MXPs, the file format used in the Macromedia Extension Manager tool to allow developers to install extensions.

Although Dreamweaver is arguably more powerful in making use of such panels for coding purposes, Fireworks and Flash are a good pair to compare because one is currently a developer-oriented tool while the other one is designer focused…depending on who you ask, obviously. After reading this article, you'll have two distinct paths through which you can utilize the information.

The reasoning behind creating such tools deserves investigation. First, there are a lot of things developers and designers do that are unique to each company, each process or project, and almost always each developer. Realizing this, Macromedia has opened up the functionality of their products, allowing them to be scripted to be used in different ways, to be extended with custom menus and/or functionality blended with current functionality, and to automate repetitive tasks. Real-world examples include adding a custom tool to be used in the tools palette, adding an auto-color panel that automatically gives you a color scheme with swatches based on your color choice, and batch processing of images with custom settings. These not only allow you to get your job done faster, but also empower you to take matters into your own hands when there is certain functionality you need, or just plain want, that the application doesn't currently provide.

Now that you have your reasons, what tools are available? Most of Macromedia's products support extension through scripting in some way. Fireworks and Flash have a very similar JavaScript API that they expose to allow you to call application features through code. The code is a form of JavaScript and is procedural in nature. You can write functions and variables like you would in ActionScript, but there is really no point to OOP in terms of writing scripts. Because Fireworks has had its scripting engine longer, it comes with more example scripts upon first installation. There are more on Macromedia's site and on the Net; they're called commands and are saved as text files. For Fireworks the file format is .jsf, and for Flash the file format is .jsfl. You can use any text editor for either, but Flash has two advantages: Flash will provide code hints for the Flash JSAPI, and if you install the FWCommand-Components, an MXP found in the Extending Fireworks folder on the Studio MX CD, it'll install the code hints for Fireworks as well. These scripts can be run in a few ways. In Fireworks, you go to the Commands menu and select Run Command. This will open a dialogue and allow you to choose the JSF file to run. In Flash, you select Commands and choose Run Command, and through the same process pick your JSFL file.

If you have commands that you use often you can save them in the Commands folder, and when you reboot your programs they will be available as menu choices in your Commands menu. The Command folder for each respective program is where your Macromedia products are installed. In Fireworks on Windows XP, it's:

```
C:\Program Files\Macromedia\Fireworks
MX 2004\Configuration\Commands
```

In Flash:

```
C:\Program Files\Macromedia\Flash MX
2004\en\First Run\Commands
```

If scripts call each other via the runScript command and you don't provide a file path, as long as they are both in the Commands folder you're good to go. Additionally, placing scripts in here in Fireworks allows you to use them in batch commands (File>Batch), which you can then save as a script. You can see

> "there are a lot of things developers and designers do that are unique to each company, each process or project, and almost always each developer"

how this gets powerful quickly. Say you're exporting your image with very specific format, color, and matte color information, and you decide to place in a script because you're doing this often for your project. You can then make a batch command in Fireworks once you place said script in the Commands folder, and you can utilize the command you just made to batch a bunch of images using the same command, all in one fell swoop.

Now, that's uber-powerful…but uber-procedural, too. Flash panels provide a GUI to that functionality. By making a Flash movie with some new code, you can have that Flash movie control the IDE as well as call other commands and/or scripts. They appear in a normal Fireworks or Flash window and are accessible in Fireworks via Window> [Name of panel], and in Flash by Windows>Other Panels>[Name of panel]. A great example of this in Fireworks is the Align panel, which comes preinstalled. It is a Flash panel, written in Flash, that utilizes the Fireworks JSAPI to align whatever element on your document you've selected. For those coders who are intimate with auto-shapes and some of the bitmap effects you can do in Fireworks, creating

Advertising Index

Advertiser	URL	Phone	Page
American Media Training	www.americanmediatraining.com	800-278-7876	6
CFDynamics	www.cfdynamics.com	866-233-962-6427	49
ColdFusion Developer's Journal	www.sys-con.com	201-802-3022	56
ComCity	www.salescart.com	888-826-6248	3
Edge 2004 East	www.sys-con.com/edge	201-802-3069	57-73
Future Media Concepts	www.fmctraining.com	877-362-8724	22
HostMySite.com	www.hostmysite.com/mxdj	877-248-4678	9
Interakt	http://ktml.interaktonline.com		31
Macromedia	www.macromedia.com/go/2004		33
Macromedia	www.macromedia.com/into		76
NetQuest	www.nqcontent.com		75
Oddcast Media Technologies	www.oddcast.com	212-375-6290	2
PaperThin	www.paperthin.com	800-940-3087	55
Seapine Software	www.seapine.com	888-683-6456	17
TechSmith	www.techsmith.com/mxdev		28

an effects window is another way a Flash panel will allow you to deploy such an extension. Flash examples include the NetConnection Debugger, used for debugging Flash Remoting and Flash Communication Server, and a panel I created, the Key Tool, which shows you the keycode and ascii of the last key clicked. In Fireworks, they can also be used as modal dialogues.

Flash panels are made available by the Window SWF folder. On Windows XP, in Fireworks, it is:

```
C:\Program Files\Macromedia\Fireworks
MX\Configuration\Command Panels
```

In Flash it is:

```
C:\Program Files\Macromedia\Flash MX
2004\en\First Run\WindowSWF
```

Communicating from a Flash SWF to Fireworks or Flash is done via the MMExecute command. This command takes a JSAPI function as a string and runs it:

```
MMExecute("fl.runScript(\"my
command.jsfl\");");
```

Same for Fireworks:

```
MMExecute("fw.runScript(\"my
command.jsfl\");");
```

Notice two things. The Flash one utilizes "fl" in front of the command. This is how you write the JSAPI for Flash, while you use "fw" in Fireworks. You can also use "flash" for Flash, but who wants to type all that? The second thing to notice is the escape characters used to put quotes (") in the parameter of the JSAPI function. This can get tedious, but I'll show you how to get around it (no, not single quotes, that doesn't work).

For multiline statements that you deem unnecessary for an external script, you can do:

```
var str = "";
str += "var doc =
fl.getDocumentDOM();";
str += "for(var p in doc){";
str += "  fl.trace(p + \": \" +
doc[p]);";
str += "}";
MMExecute(str);
```

However, there is a much better way to write the code. You can do one of two things. When writing Fireworks Flash panels, you can install the FWCommandComponents, which have wrapper methods for just about every Fireworks function, so instead of writing this:

```
var path =
MMExecute("fw.appPatternsDir;");
```

you can do this:

```
var path =fwapi.getAppPatternsDir();
```

I've also written my own wrapper that does the same thing, but has two benefits:

1. It uses one function to deal with the function calls, therefore reducing overhead with all of those predefined functions.
2. You can register for events because Fireworks, not Flash (currently), has the added benefit of generating events if you have a function registered for it in your SWF. The events are detailed in the

Extending Fireworks PDF. This helps your Window SWF, in Fireworks anyway, to react to changes in the IDE. For example:

```
var b = {};
b.owner = this;
b.onFwStopMovie = function(){
          var so =
SharedObject.getLocal("settings");
          so.data.favoriteColor =
this.owner.selectedColor;
          so.flush();
};
fwapi.addListener(b);
```

Additionally, you can use the same API for Flash. So instead of:

```
MMExecute("fl.runScript(\"my
command.jsfl\");");
```

you can do:

```
flapi.runScript("my command.jsfl");
```

While developing a Window SWF, it is easiest if you set your Publishing Settings to have the SWF publish straight to the directory from which Fireworks and Flash actually pull the SWF. On Windows XP, in Fireworks, this is:

```
C:\Documents and
Settings\[user]\Application
Data\Macromedia\Fireworks MX
2004\Command Panels
```

In Flash:

```
C:\Documents and Settings\[user]\Local
Settings\Application
Data\Macromedia\Flash MX
2004\en\Configuration\WindowSWF
```

That way, you publish once, reboot Fireworks and/or Flash, and from then on can just close the Flash panel, publish, and then re-open it to test your updates.

Finally, to publish your work for ease of installation, updates, and uninstalling, using an MXP file is the best way to distribute it. MXPs are installer programs that run in Macromedia's Extension Manager. This program should already be installed if you're running any of the MX or MX 2004 products, Studio, or some of the version 4 products, although it never hurts to update your copy. This program allows installation and management of extensions to Macromedia products. Most Macromedia users have no idea they even have the program, but double-clicking the file will open the product if they do have it installed. A user can then uninstall your extension, toggle it on or off, and gleen information about how to use it. The files you want to install as well as where are defined in an MXI file; this file is an XML document that determines that information, and you can learn how to use the tags either by reading the MXI specifications PDF or looking at your currently installed extensions. For example, in Fireworks on Windows XP:

```
C:\Program Files\Macromedia\Fireworks
MX\Configuration\Extensions
```

The MXI Specifications PDF is available at http://download.macromedia.com/pub/exchange/mxi_file_format.pdf.

You use the Macromedia Extension Manager to package your MXP file, and then you can just distribute your MXP as your installation file.

Conclusion

As you can see, extending Macromedia products with your own functionality and GUI is made easy by the exposed JSAPI of both Fireworks and Flash, as well as the ability to create a GUI utilizing a SWF generated by Flash and have it utilize the JSAPI.

References

To get some ideas, read some of the methods in the Extending Fireworks PDF or the Flash JSAPI LiveDocs.

- www.macromedia.com/support/documentation/en/fireworks/
- http://livedocs.macromedia.com/flash/mx2004/jsapi/fm1.htm

Jesse R. Warden, MXDJ's *Flash editor, is a senior Flash developer at Surgical Information Systems, an operating room software company, where he currently uses Flash MX, Flash Remoting, .NET, and Oracle to create next-generation rich Internet applications for the OR. He contributed four chapters to the* Flash Communication Server MX Bible *and has written articles for various publications, including one for Macromedia for a DRK. jessewarden@sys-con.com*

MXDJ Section Editors

Dreamweaver
Dave McFarland
Author of Dreamweaver MX: The Missing Manual, *Dave can be relied upon to bring Dreamweaver MX to life for* MXDJ *readers with clarity, authority, and good humor.*

Flash
Jesse Warden
A multimedia engineer and Flash developer, Jesse maintains a Flash blog at www.jessewarden.com and says, referring to the MX product range, that "Things are changing, opportunity is on the frontier, a paradigm shift is occurring for Web design, Web applications, et al."

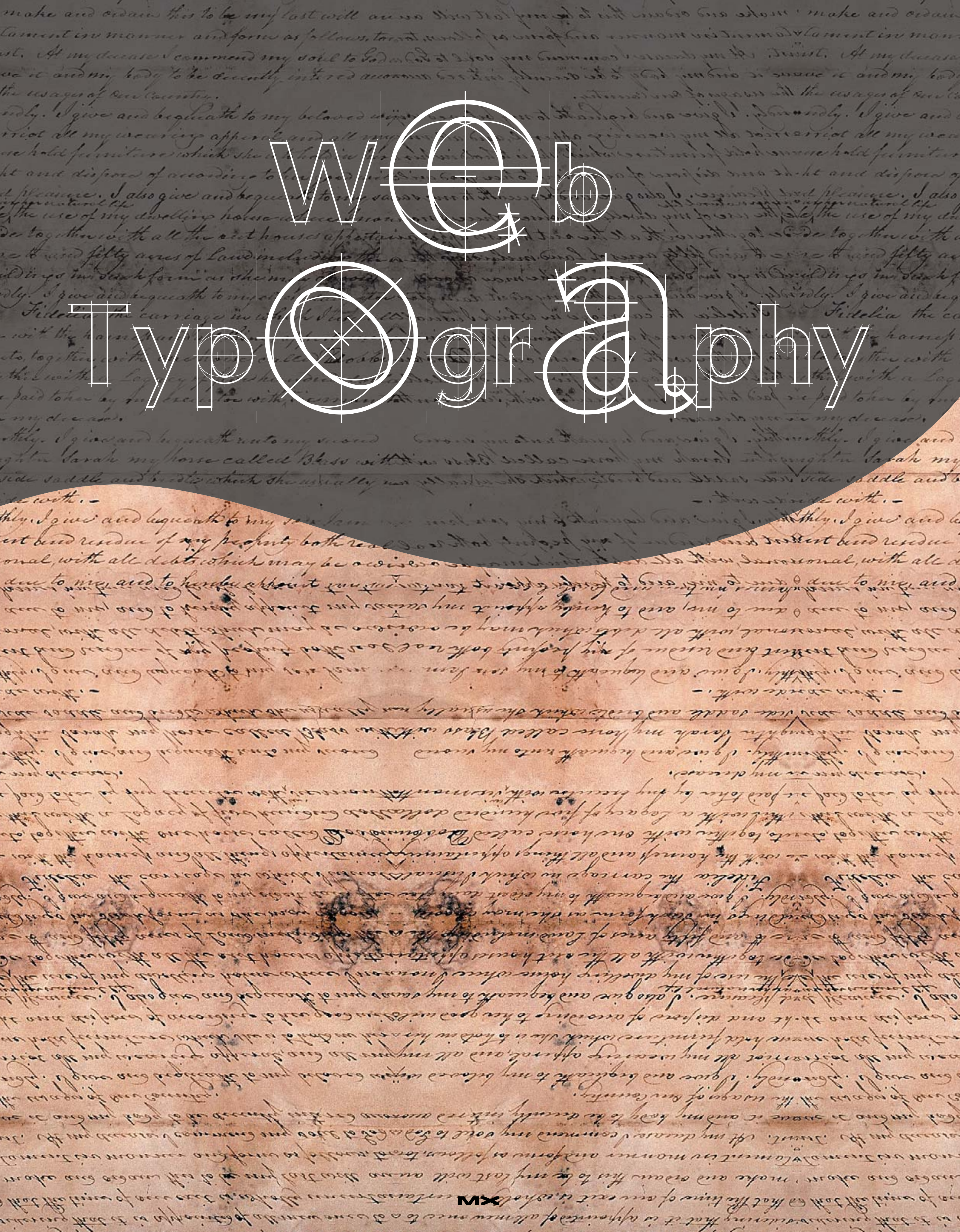
Fireworks
Kleanthis Economou
A Web developer/software engineer since 1995, now specializing in .NET Framework solutions, Kleanthis is a contributing author of various Fireworks publications and is the technical editor of the Fireworks MX Bible. *As an extension developer, he contributed two extensions to the latest release of Fireworks.*

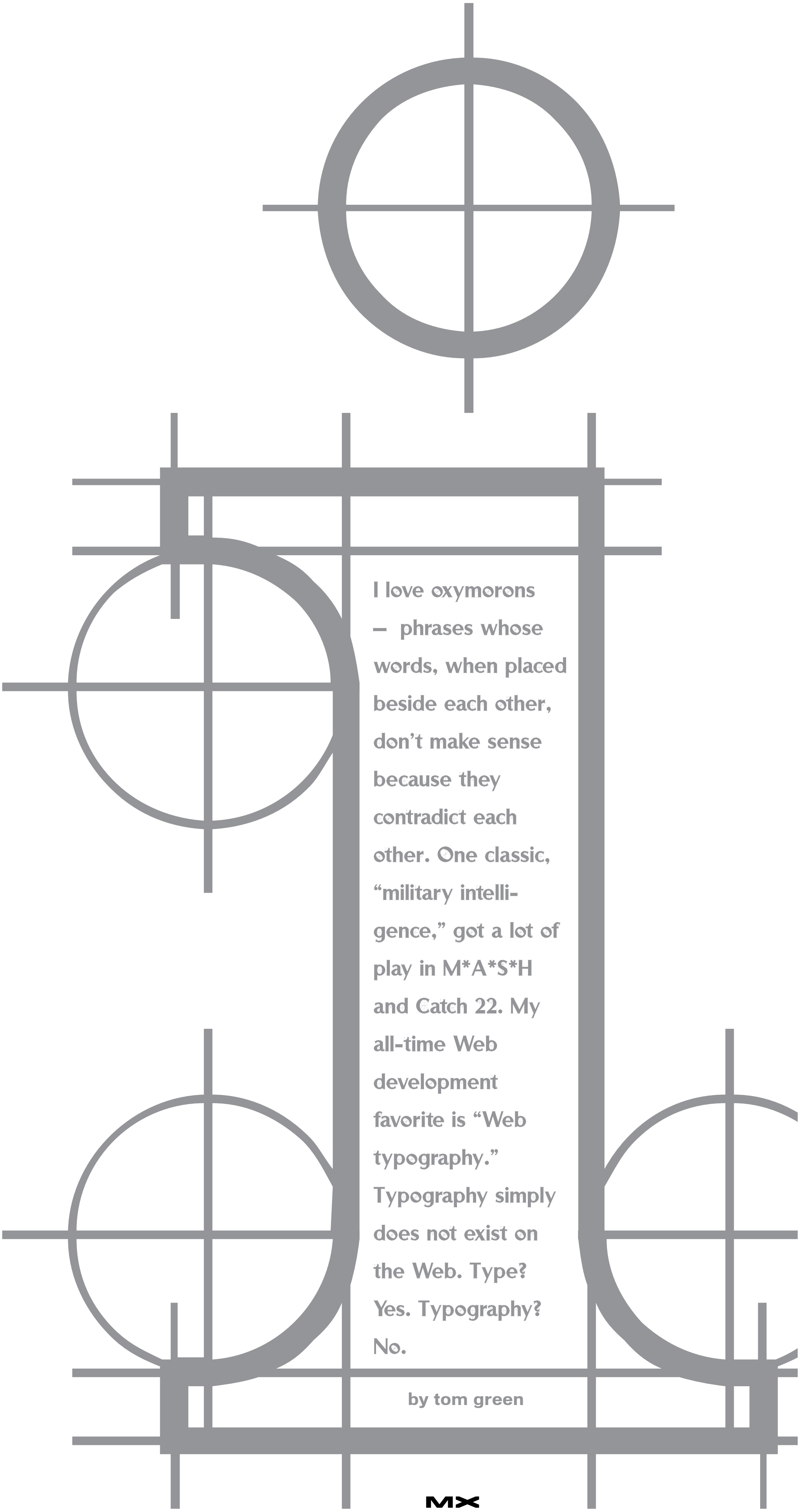
FreeHand
Louis F. Cuffari
Cofounder and art director of Insomnia Creations (www.insomniacreations.com), Louis has spent most of his life as a studio artist, including mediums from charcoal portraits to oil/acrylic on canvas. In addition to studio art, he has been involved in several motion picture projects in the facility of directing, screenwriting, and art direction. Louis's creative works expand extensively into graphic design, and he has expertise in both Web and print media. He is deputy art director for SYS-CON Media and the designer of MX Developer's Journal.

ColdFusion
Robert Diamond
Vice president of information systems for SYS-CON Media and editor-in-chief of ColdFusion Developer's Journal, *Robert was named one of the "Top thirty magazine industry executives under the age of 30" in* Folio *magazine's November 2000 issue. He holds a BS degree in information management and technology from the School of Information Studies at Syracuse University. www.robertdiamond.com*

Director
Niqui Merret
After many years as an instructor and developer in South Africa, Niqui now works at London-based rich Internet solutions firm, Prismix. She feels that she is a Director addict in a Flash-mad world.

web Typography





I love oxymorons – phrases whose words, when placed beside each other, don't make sense because they contradict each other. One classic, "military intelligence," got a lot of play in M*A*S*H and Catch 22. My all-time Web development favorite is "Web typography." Typography simply does not exist on the Web. Type? Yes. Typography? No.

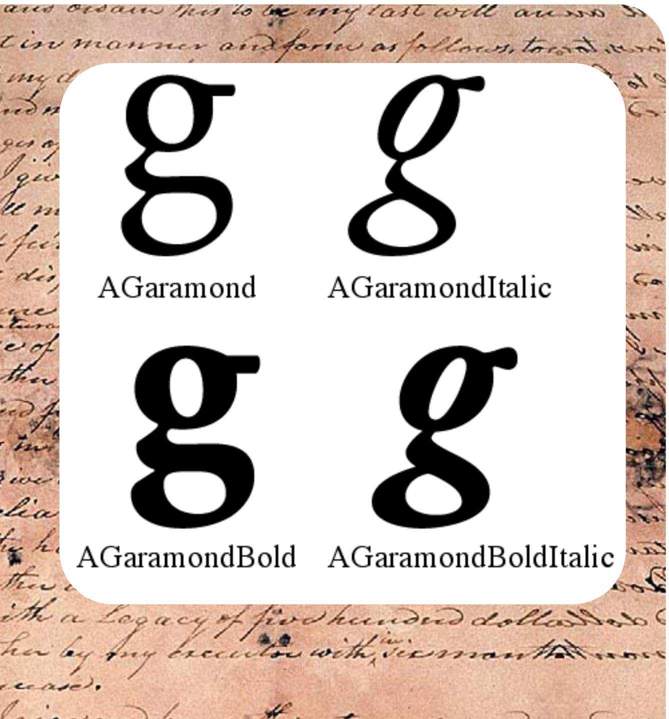
by tom green

Flash developers, for example, seem to have embraced 6-point for everything. What you see, in this instance, are Web sites that look more like the fine print at the bottom of used-car financing ads than digital media communications vehicles. Many who develop in the HTML universe still haven't figured out that line length increases with a corresponding increase in point size. Both groups have developed an unnatural fixation on using light grey type on a dark grey background. To use a Web dev buzz phrase: "What's with that?"

The answer lies in the fact that typography, in the print universe, is fundamentally different from typography in the Web universe. Typography is the official name for the design and use of text on a printed or digital page. The problem is just that: the digital page.

As Dave Berlow of the Font Bureau says in the article "Type On The Web," found in a book named *Websights: The future of business and design*, "You have control of print and no control of the Web. In fact, there is no typography on the Web at all. There is only type." This is an important distinction. Print is a "control" medium where all aspects of the printed word, from font choice to line length, are controlled by the designer in order to make the words on the page legible and understandable. In the Web paradigm, this simply does not exist. HTML is a markup language, and CSS is more a presentation tool than a design tool. Typographic nuances don't exist on the Web. There is only type. Don't believe me? Design an entire Dreamweaver MX 2004 page using only Optima or Garamond and see what happens in the browser of a viewer who doesn't own those fonts. Does that occur in the magazine you're holding in your hands?

image I



Thus you have to change your thinking and look at the text on the Web page as being either words or art. Once you make that distinction a whole world of design possibilities opens. Those possibilities will become real when you discover the serious type tools Macromedia Studio MX 2004 has put in your hands.

The tools are FreeHand MX, Fireworks MX 2004, Dreamweaver MX 2004, and Flash MX 2004. Your tool choice will depend upon what you need to do. If the text is words, Dreamweaver and Flash will fit the bill. If the text is art, FreeHand MX, Flash, and Fireworks will satisfy that need.

In this article I demonstrate three examples that allow you to take many of the typographic techniques used by our print brethren and apply them to the Web.

Typographic Logo

The first project is the creation of a typographic logo using two intertwined letters. This logo, based on an idea from Dmitry Kirsanov, will sit over a gradient background, and the direction of the gradient will follow the slope of the italic letters. The plan is to use the "t" in my first name and the "g" in my last name.

To accomplish this, my tool choices are essentially made for me: FreeHand MX or Fireworks MX 2004. Fonts use vectors, and both applications permit me to convert the lettering used to vectors. This allows me to subsequently move the design into a Web page without concerning myself with whether the font chosen is resident on the user's machine.

In this example I use FreeHand MX to create the artwork. Though the art can just as easily be created in Fireworks MX 2004, I prefer to do much of my type work in FreeHand MX. I find that the typographic tools included with the application permit me a finer degree of control over such things as kerning, tracking, and leading than the sliders used in Fireworks MX 2004. I'll also convert the letters to their postscript outlines, and I prefer to use a vector tool for that task.

My font choice is a modern classic from the Old Style school – Adobe Garamond. Old Style lettering, especially the italic, tends to have a calligraphic look, and Adobe Garamond, drawn by Roger Slimbach, is a faithful reproduction of the original drawn by Claude Garamond in the 1500s.

In this project I use AGaramond Italic. This is important because fonts are traditionally drawn as a collection known as a family. Each member of the family is designed to stand on its own or work together with the other members of the family. For instance the italic I am using is not simply a slanted version of the Roman, and the bold is not just a thick version of the Roman. As you can see in Image I, the various fonts in the AGaramond family are quite distinct from each other. For example, in addition to looking completely different from each other, the major differences between the Roman and italic versions of the font are in the shape of the ear, the decorative handle at the top of each letter shown. In the Roman version the ear is rectangular; in the italic it has been rounded and lengthened. The other major difference is the loop at the bottom of the letter, which wraps around the hole, or counter, in the letter. Note how the loop and the counter change shape in each version of the letter shown. These nuances of type, which have been around for more than 500 years, are usually overlooked by Web designers and have contributed to the state of the typographic art on the Web today.

(Though the Web is littered with sites promoting fonts, discussing typography, and even teaching it, one of the best is http://counterspace.motivo.com/. This Flash-based site gives a great overview of the nuances and history of typography.)

I start by opening FreeHand and adding three layers to the document, named "Gradient," "T," and "G." I then lock the T and the G layers.

I also want to add the colors I use in my site to the FreeHand MX color palette. To do this I select Window>Swatches to open the Swatches panel. One of the great secrets of FreeHand MX is that it contains a palette of Web-safe colors. To open it, click the Swatch tab, click on the Panel Options menu in the Assets panel, and select Web Safe Color Library at the bottom of the list. This library is a bit different from what you may be used to. Instead of the traditional cubes and swatches used by the Macromedia MX products, this library treats the Web colors like a swatch book. What I particularly like about it is that the colors chosen are shown as Hexadecimal and RGB color. I entered the hex values for my colors – #333300, #666633, and #999966 – and they were added to the Swatch list for the document (see Image II).

Having prepared the work space, I get right to work. I select the T layer and lock the Gradient and G layers. Locking layers is a good habit to develop in FreeHand; if you select an object on the FreeHand page and select a layer, the selected object automatically moves to the selected layer. Not a good thing.

I then select the type tool, click once on the page, and enter a lowercase "t". The new FreeHand MX Object panel immediately changes to the Text properties. I select AGaramondItalic from the font list, set the point size to 200 points, and set the fill color of the letter to #333300 by clicking the Fill icon in the Object palette and selecting the color from the list. Another neat technique is to select the text object, drag the swatch from the Swatches panel, and drop it onto the selected object. I then lock the T layer, unlock the G layer, and add a 200-point AGaramondItalic lowercase "g" filled with #999966. I lock this layer.

To create the gradient in the background, I unlock the Gradient layer, select the Rectangle tool, and draw a rectangle behind the letters. This results in a solid-color rectangle filled with whatever background color is shown at the bottom of the toolbar. If the rectangle has a stroke, remove it by selecting the Stroke Color chip in the toolbar and clicking the "No Stroke" icon.

To apply the gradient, I select the rectangle and click once on the Fill icon in the Object panel. When the Object properties open, I click the Fill strip in Properties to open the Fill properties panel. I then click on and hold the Basic Fill button to open a drop-down menu of available fill options. Select Gradient to open the Gradient Fill panel – the rectangle fills with a black-to-white gradient. This is easily fixed. I select the #999966 color chip on my Swatches panel and drag and drop it onto the black chip in the Gradient Fill panel. I then replace the white chip with #333300 and my gradient is complete. The final step is to get the gradient to follow the angle of the italic text.

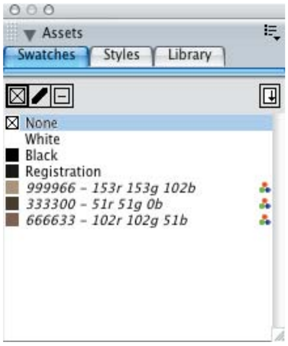
This step is where, as I tell my students, we separate the pros from the rest of the dreck out there. The gradient angle will approximate that of the stem in the letter "t." With a gradient as fine as the one chosen, I use the dark end of the gradient to set the gradient angle.

If the rectangle is selected there will be two handles joined by a dotted line running across the rectangle. The circular handle establishes the start of the gradient and the square one establishes the end and the angle of the gradient. I drag the round handle upward to a point just below the upper-left corner of the rectangle. I then drag the square handle to about the midpoint of the "g." I can now distinguish the vertical divide between the light and dark areas of the gradient. From there it's a simple matter of moving the square handle up and down to have the dark vertical match the angle of the stem. Once that's accomplished I move the square handle in and out to see how the lettering contrasts with the gradient. Once finished, I click on the page to deselect the rectangle and then lock the Gradient layer (see Image III).

As it currently stands the image is somewhat complete. What I don't have is the "t" and the "g" intertwined. The "g" is simply sitting in front of the "t". The plan is to have the "g" cover the cross stroke of the "t" and to have the serif of the "t" cross through the bowl of the "g".

First I unlock the T layer, select it, and select Duplicate from the Layer drop-down menu. This places a copy of the lowercase "t" in front of the "g". I then lock the T layer and select the "t" on the page. Next I select Text>Convert To Paths (Command-Shift-P if you use a Mac or Control-Shift-P if you use a PC). This converts the letter to art by using the postscript outline for the letter in the AgaramondItalic font.

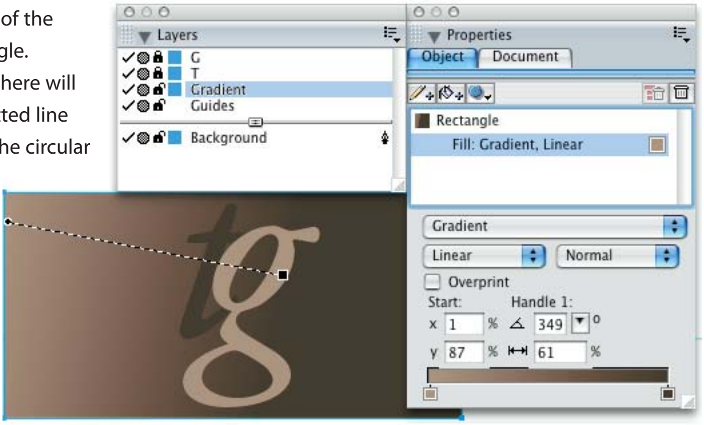
image II


Assets

Swatches | Styles | Library

☒ None
White
■ Black
■ Registration
999966 – 153r 153g 102b
333300 – 51r 51g 0b
666633 – 102r 102g 51b

image III


Layers

✓ G
✓ T
✓ Gradient
✓ Guides
✓ Background

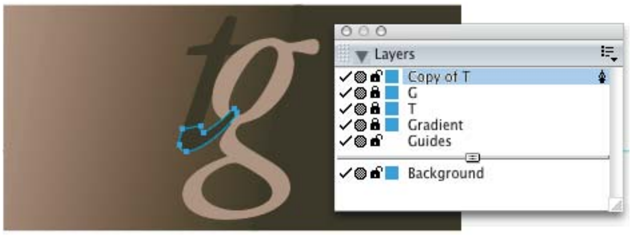
Properties

Object | Document

Rectangle
Fill: Gradient, Linear

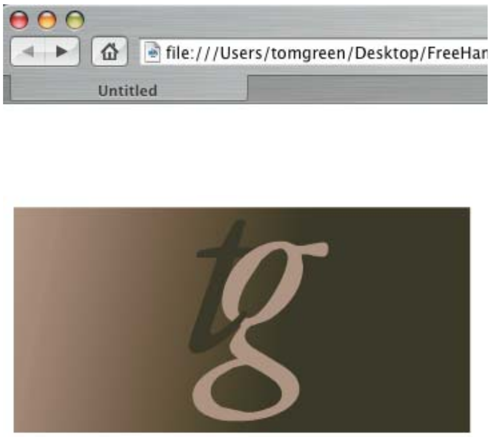
Gradient
Linear | Normal
□ Overprint
Start: Handle 1:
x 1 % 349 °
y 87 % 61 %

image IV


Layers

✓ Copy of T
✓ G
✓ T
✓ Gradient
✓ Guides
✓ Background

image V


file:///Users/tomgreen/Desktop/FreeHan
Untitled

I then select the Subselect tool in the FreeHand toolbox. I click on each point above the serif and press the Delete key (see Image IV). When I finish I have exactly the effect I'm looking for (see Image V).

Having completed the design, getting it onto the Web is not terribly difficult. There are a number of options available:

1. Save the image as a .jpg image for use in Fireworks MX 2004 or Dreamweaver MX 2004.
2. Save the objects as Symbols in FreeHand and export those Symbols into Flash MX 2004 or Fireworks MX 2004 for subsequent manipulation or animation.
3. Animate the lettering in FreeHand MX (use Release to Layers) and output a .swf file for use in Flash MX or Dreamweaver MX. All this does is to convert the page to a Flash .swf file with an accompanying HTML document.
4. Build a page around the image and select File>Publish as HTML.

Typographic Treatment of a Page

The next example, based on one from my book *Building Web Sites with Macromedia Studio MX*, addresses a question I have whenever I hit a page that has a Shockwave or Flash detector on the page. The text is usually set in whatever Web fonts happen to be hanging around the browser. The question is, why? If you have the space, why not use it for typographic treatment of the page?

The page uses two fonts: Helvetica Inserat for the branding and Albertus MT for the body copy. I have always liked Helvetica Inserat because it is a powerful font that should be used sparingly. In fact, "Inserat" is German for advertising; when released by the Stempel Foundry in 1966 its primary use was for ads, billboards, and newspaper headlines. Albertus is another display type that should be used sparingly on a page. This font was released by the Monotype Foundry in the 1930s and its first iteration was for use as a titling font. In 1940 a lowercase alphabet was added. I like Albertus because it isn't overpowered by Helvetica Inserat and for its sharp serifs. I find it to be elegant, aesthetically pleasing, and a good companion font to the Inserat.

In creating the "sniffer" page I open Fireworks MX 2004, set the background color to the mid-green I use on my site – #663333 – and set the page dimensions to 600 pixels wide by 400 pixels deep. Page size is immaterial at this stage of the process. The plan is to use the new Fit Page button on the Property Inspector to trim the canvas when I am finished.

The first step is to select the Branding text and set it in 60-point Helvetica Inserat, center align the text, and fill it with the red I use in my site's logo: #660000. The result, shown in Image VI, is okay but not what I want. I want the two lines closer to each other, and the words on the second line need to be spaced out a bit more.

The words are placed, formatted, and colored in Fireworks MX 2004 but still need some work.

I select both lines and reduce the leading between them in the Fireworks MX 2004 Property Inspector to 85%. This brings the second line up closer to the first one and avoids having the ascenders in the "t" and the "h" touching the letters in the line above (called a crash). The next step is to select the bottom line and to set the range kerning slider to a value of 7. This increases the spacing between the letters of "ontheweb". The line above it, being in uppercase letters, looks a bit more spaced out, which makes the line under it appear to have lettering that is squeezed together. The "th" and "we" letter pairs are still too close to each other, so I click once between the letters and add 3% kerning. The before and after results are shown in Image VII.

To complete the design, I use a variety of techniques to draw the user's eye to the important parts. For example, I use negative leading values to pull lines together and positive values to push them apart. I make the point size of the important lines larger than that of less important ones, and use the space bar to move them around on the page to avoid the straight lines you see when text is set as Flush Left. I also set the site's name in Albertus MT IT, track the letters out a bit more than the rest of the text, and give it the same color as the head to establish the relationship between the two elements.

With the element completed I can use the Fireworks MX 2004 slicing tool to slice it up for placement in Dreamweaver MX 2004.

Image VIII shows a Fireworks-produced typographic treatment of a "Flash Sniffer" page.

New to Fireworks MX 2004 is a beefed-up anti-aliasing feature. Macromedia kept its existing four options (None, Crisp, Strong, and Smooth) but added "System" (Quartz if you are on a Mac) and "Custom" to its anti-aliasing lineup.

Here's a brief description of what each anti-aliasing option, available through the Property Inspector, does:

- ***No Anti-Alias:*** Disables text smoothing
- ***Crisp Anti-Alias:*** Creates a sharp transition between the edges of the text and the background
- ***Strong Anti-Alias:*** Creates a very abrupt transition between the edges of the text and the background, preserving the shapes of the text characters and enhancing detailed areas of the characters
- ***Smooth Anti-Alias:*** Creates a soft transition between the edges of the text and the background
- ***System Anti-Alias:*** Uses the text smoothing method provided by Windows XP or Mac OS X

image VI

image VII

image VIII

- ***Custom Anti-Alias:*** Opens a dialog box with the following input areas:
  - *Oversampling* determines the amount of detail used for creating the transition between the text edges and the background.
  - *Sharpness* determines the smoothness of the transition between the text edges and the background.
  - *Strength* determines how much the text edges blend into the background.

Image IX shows the various anti-aliasing options applied to a 200-point Albertus MT "a" on a Macintosh.

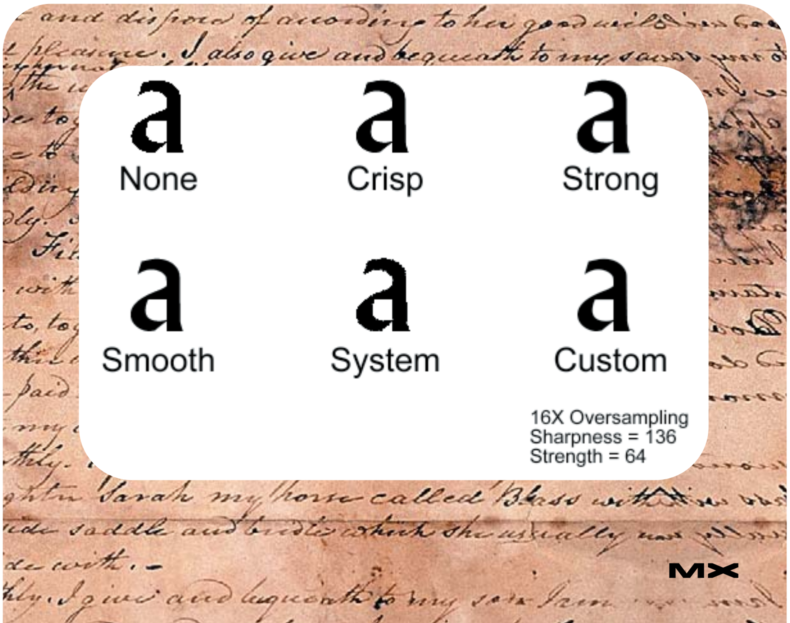
Alternative States

The final example shows how the type tools in FreeHand MX can be used to create a rather interesting effect in Flash MX 2004. This example is inspired by the movie *Altered States*. The title of the movie uses the font Avant Garde Demi, letters that are tightly kerned, and a rather interesting use of the uppercase A. The plan is to create the lettering for the word "ALTERNATIVES" in FreeHand MX and then use the word, as it was used in the movie's title, as the mask for an animation.

Image X shows the final Flash movie using the title created in FreeHand MX.

1. Open FreeHand MX, select the text tool, click once on the page, and enter the word "ALTERNATIVES".
2. When the Object Properties open (Window>Object), set the font to Avant Garde Demi and the point size to 90 points. If you don't have the font being used, feel free to use a Bold Helvetica, Arial, or Avant Garde (see Image XI).
3. The next step is to create the stylized "A". Select the first "A" in the word and cut it. Lock the layer the word is in and select the background layer.
4. Paste the "A" on the clipboard in the background layer. Add a new layer, select the magnifying glass, and zoom in on the letter in the background layer. The original title created stylized "A"s by removing the right stem of the "A" and placing the letter form up against the stem of the next letter. The

image IX



That "Em" Thing

When my students and I get talking about CSS the question that is inevitably raised is: How big is an em? The answer, on the surface, may appear to be a bit facetious because the real answer is, "As big as you make it."

Type is traditionally measured in two directions. If the measurement is vertical, the industry uses points and picas. Horizontal measurements are used for spacing between letters. That horizontal spacing is measured in ems and the em value is a moving target. One em equals the type size. In the exercise in this article I use 60-point type. Thus, one em equals 60 points. If I were to change it to 48-point type, the em would also change to 48 points. Thus an em is proportionately the same in any point size regardless of the font chosen.

By adding 7% range kerning between the letters, I was, in very simplistic terms, adding a further 4.2 points of space to the sides of the letters selected.

It gets a bit weirder when, in Fireworks MX 2004, you select a few words in a sentence with the intention of adding a bit of space between the letters and the words. This is called "range tracking" in Fireworks MX 2004.

Traditional typesetting divides a line of text into units – usually a percentage of the em value. Fireworks MX 2004 comes at this from a different angle. The unit used is a pixel. Thus a 72-point letter will be divided into 72 units. Applying a range kerning value of 10% would, again in very simplistic terms, spread out the letters and the spaces between the words by 7.2 pixels. This is a double-edged sword. It's great because the value is consistent. It's bad because an inch on my computer monitor may be different from an inch on yours.

In CSS ems are also used and, in many respects, are the only constant type value available to you. In Dreamweaver MX 2004 ems are used for:

- Font size
- Line height
- Horizontal position
- Vertical position
- All Box values except for Float and Clear
- All Positioning values
- Border width
- Word spacing
- Letter spacing
- Text indent

All are determined by the point size used for the default font set by the browser. If it is 12-point Times, then an em is 12 points. The problem is points. They aren't used on the Web. Pixels are used instead. If you do assign a value in CSS use percentages. This removes a potential IE bug and allows the browser to scale the font size in Internet Explorer (PC). Just remember that the em value set will ripple through the entire design and could have a profound effect on page layout.

For example, if your reference size for text is 12 points and line height is set to 2 ems, that is the same as setting the leading to 24 points. In essence you have added double spacing between your lines of text.

interesting aspect of this design is that the letterform remains intact because the human eye will still see the "A".

5. Select the pen tool and trace the letter without the right stem. When finished, the shape should resemble the one shown in Image XII.

   The parts of the letter needed for the title are drawn over a copy of the letter in the background layer.

6. Unlock the letters layer. Select the text tool and delete each "A" in the word. Click once in the location of the second "A" and press the space bar. This will leave a space for the object. Drag the A-piece to the start of the word. Line up the top of the shape with the top of the "L". Use your arrow keys to move the shape closer or further away from the "L".

7. Select the "LTERN TIVES" letters and set the tracking value to about –3%. This will reduce the spacing between the letters.

8. Select the A piece, press the Option/Alt key, and drag a copy of the object into the space between the "N" and the "T". The object should overlap the "T". For now just ensure that the tops of the "T" and the object are aligned.

9. Click once between the "N" and the object. Add a value of about 13% to the kerning area of the Object Inspector. If the space between the object and the "T" is too large, click between the object and the "T" and enter a small, negative kerning value such as –3. Finally, click on the text block and select Text> Convert to Paths.

10. Select everything, group it, and convert it to symbol in the library. (If you have a Brushes folder in the library, feel free to delete it. The folder isn't needed in Flash.) Save the image and quit FreeHand MX. The title as shown in Image XIII is complete and ready for placement into Flash MX 2004.

Before I complete the piece in Flash, I

image X

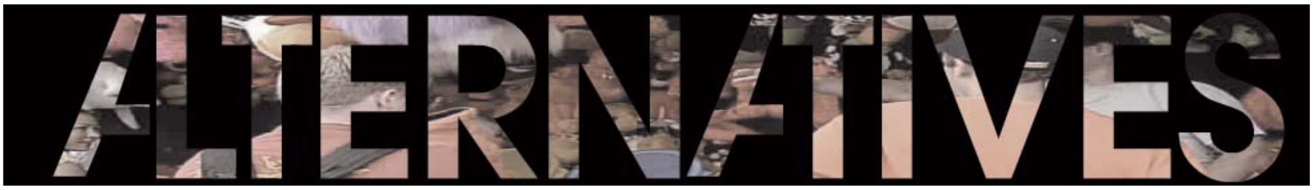
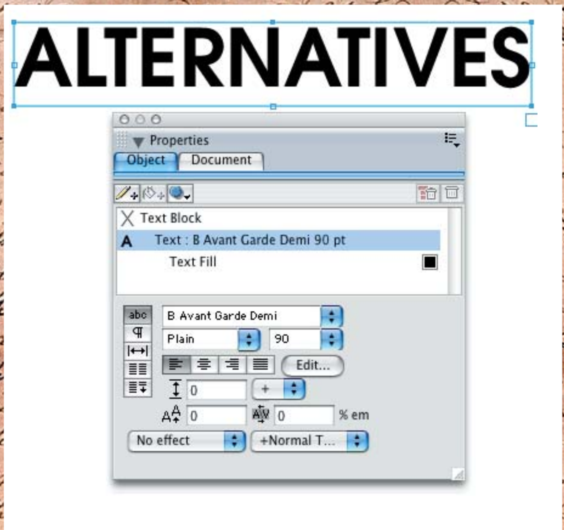
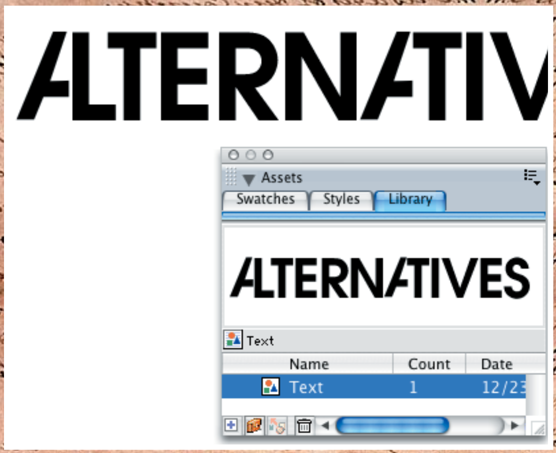


image XI


ALTERNATIVES

Properties
Object | Document
Text Block
A Text : B Avant Garde Demi 90 pt
Text Fill
B Avant Garde Demi
Plain | 90
Edit...
0 | +
0 | 0 % em
No effect | +Normal T...

image XIII


ALTERNATIV

Assets
Swatches | Styles | Library
ALTERNATIVES
Text

Name	Count	Date
Text	1	12/2

image XII

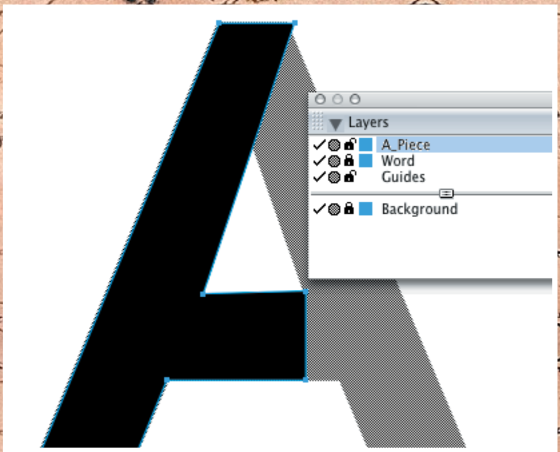
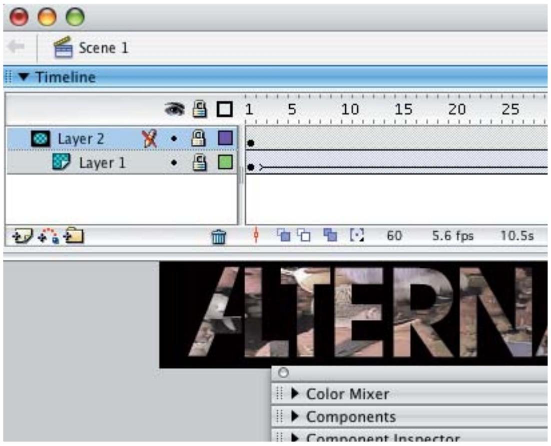

Layers
✓ A_Piece
✓ Word
✓ Guides
✓ Background

image XIV


Scene 1
Timeline
1 5 10 15 20 25
Layer 2
Layer 1
60 5.6 fps 10.5s
▸ Color Mixer
▸ Components
▸ Component Inspector

What's the easiest way to update a web site?

Through your browser

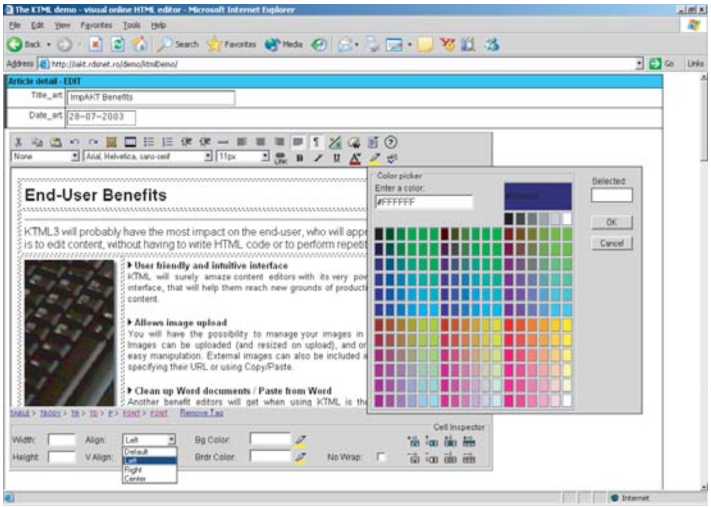
Introducing:

- Property inspector
- Spell-checker
- Image management
- Dreamweaver MX support
- ColdFusion, ASP, PHP

KTML WYSIWYG editor 3

Experience:

- "MS Word" style text editing
- Text formatting using CSS
- Intuitive table alignment


The KTML demo - visual online HTML editor - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://lakt.rdsnet.ro/demo/ktmlDemo/
Go Links
None
Arial, Helvetica, sans-serif
11px
Color picker
Selected:
OK
Cancel
TABLE > TBODY > TR > TD > P > FONT > FONT Remove Tag
Cell Inspector
Default
Left
Right
Center
Internet

Article detail - EDIT

Title_art: ImpAKT Benefits

Date_art: 28-07-2003

Enter a color: #FFFFFF

End-User Benefits

KTML3 will probably have the most impact on the end-user, who will appr
is to edit content, without having to write HTML code or to perform repetit

▸ User friendly and intuitive interface
KTML will surely amaze content editors with its very pow
interface, that will help them reach new grounds of producti
content.

▸ Allows image upload
You will have the possibility to manage your images in
Images can be uploaded (and resized on upload), and or
easy manipulation. External images can also be included i
specifying their URL or using Copy/Paste.

▸ Clean up Word documents / Paste from Word
Another benefit editors will get when using KTML is the

Width: Align: Left Bg Color:

Height: V Align: Brdr Color: No Wrap: □

Compatibile with:

- MSIE/Netscape
- Windows/MAC/Linux

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

http://ktml.interaktonline.com/

KTML3 - Coming Soon to a Browser Near You

Interakt
macromedia ALLIANCE PARTNER

macromedia COLDFUSION MX

ASP Active Server Pages

php

should explain why I use FreeHand MX and not Flash MX 2004 for the typesetting. The decision essentially comes down to one of control. The kerning feature of FreeHand MX has its roots in print and is, therefore, a bit more precise than its counterpart on the Flash MX 2004 Property Inspector. I also find tracing to be easier to accomplish in FreeHand than in Flash.

By converting the letters to paths, I avoid a potential font substitution issue if the font is not available to whoever does the Flash work. I also plan to use the words as a mask in Flash. This would require me to vectorize the text in Flash, meaning the text would eventually be converted to paths. Whether it is done in Flash or FreeHand MX is immaterial.

One final point before I move on. The choice of font really is a non-issue because Flash embeds fonts into the .swf when the movie is compiled. This essentially frees you from "Font Substitution Hell" if you use a font the user doesn't have on his or her machine. Still, treat this one judiciously as well. An embedded font adds weight to the file and all embedded text is anti-aliased, especially at the larger point sizes. In the case of our title, I use only 10 characters from the font. It seems silly to embed the font into a .swf if I'm using only a small percentage of the character set. Thus, the other reason for converting the text to paths.

1. Open a new Flash MX 2004 movie. Set the stage to 85 high by 550 wide. When the stage appears in the screen select File>Import> Import to Library.
2. When the import dialog box opens, navigate to the FreeHand MX file just created , select it, and click the "Import to Library" button. You will now see a FreeHand Import dialog box. Click OK. If you open the Library Panel you will see that the symbol from FreeHand MX is in the library.

   One of the quirks of Flash MX is using text or other artwork as a mask. Unlike Fireworks MX 2004 and other vector apps, you can't simply select an object and denote it as being a mask. It must be broken apart in Flash MX 2004.
3. Double-click the symbol in the Flash MX 2004 library. Select the object and keep pressing Control-B (PC) or Command-B (Mac) until the object looks pixelated.
4. Return to the main timeline and import an image into the library or create some sort of wild, color-filled graphic symbol.
5. Add a new layer to the timeline. Name the bottom layer "Image" and the top layer "Alternatives".
6. Drag the image symbol into the "Image" layer. Drag the FreeHand symbol into the "Alternatives" layer. Add a key frame at Frames 30 and 60 of the "Image" layer.
7. Select frame 60 of the "Alternatives" layer, right click the mouse (Control-Click …Mac) and select Add frame from the Context drop-down menu.

9. Place the mouse over the name of the "Alternatives" layer and right click. Select "mask" from the Context menu.

   The image in the "image" layer will appear as the fill for the words created in FreeHand MX. Image XIV shows the final Flash movie including timeline, masks, and library items.

Conclusion

In this article I have just scratched the surface of the possibilities available to you. As I said in my New Riders book, *Building Web Sites with Macromedia Studio MX*: "typographic art is a powerful communications vehicle. The choice of font, layout, and effect can instantly communicate a mood or message that is far more powerful than a series of photos or images."

Text is simply more than the grey stuff that goes around the pictures and animations on a Web page, and Macromedia Studio MX offers you a full set of tools that range from the rudimentary (Dreamweaver MX 2004) to sophisticated (FreeHand MX).

For example, if your reference size for text is 12 points and line height is set to 2 ems, that is the same as setting the leading to 24 points. In essence you have added double spacing between your lines of text.

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

xile
written & illustrated by louis f. cuffari
3
Check it out. I replaced my computer keyboard with my electric piano...
...so I can do computer work and play my music at the same time.
Great idea, but what key do you use for programming?
C sharp.

NORM MEYROWITZ
PRESIDENT OF PRODUCTS

WEB DEVELOPERS ARE USING OUR MX PRODUCTS IN WAYS WE NEVER DREAMED. JUST IMAGINE WHAT THEY'LL DO WITH THE NEW MX 2004.

I've been endlessly amazed by the things our customers have done using our MX generation of products. And with all the new features in Studio MX 2004, it's going to be even easier and faster for them to realize their visions.

Dreamweaver MX 2004 helps you get that picture in your head turned into a web site faster than ever. That should be welcome news to the millions of web professionals who use Dreamweaver to create sites and applications. We've added things like CSS support, target browser check and improved code hinting to help you get through projects in far less time. And with Fireworks MX 2004 you can optimize web graphics up to 85% faster.

The new Flash MX Professional 2004 takes our industry standard tool for building rich content and applications to a whole new level. It really helps stretch your abilities–no matter where you fall on the designer-developer continuum. For development, we've added things like data-aware components and an extensibility layer so you can add your own features. For design, we've added high-quality, long-format video that's really impressive.

These are just a few of the new features. The products are available individually, but they really work well together in Studio MX 2004. Don't take my word for it. Download a free trial and read more at our web site.

Let me know what you dream up. norm_01@macromedia.com

Introducing
Macromedia MX 2004.

www.macromedia.com/go/2004

the science of color

Ok **Fireworks** class, let's see a show of hands: How many people understand how to really use the color tools in Fireworks **MX**? About **80%** of you! Now, how many people understand the Hexadecimal code behind colors? Hmmm….about **50%**! Now, how many understand the mathematics behind Blending Modes? Wow…only **10%** of you?

by charles e. brown

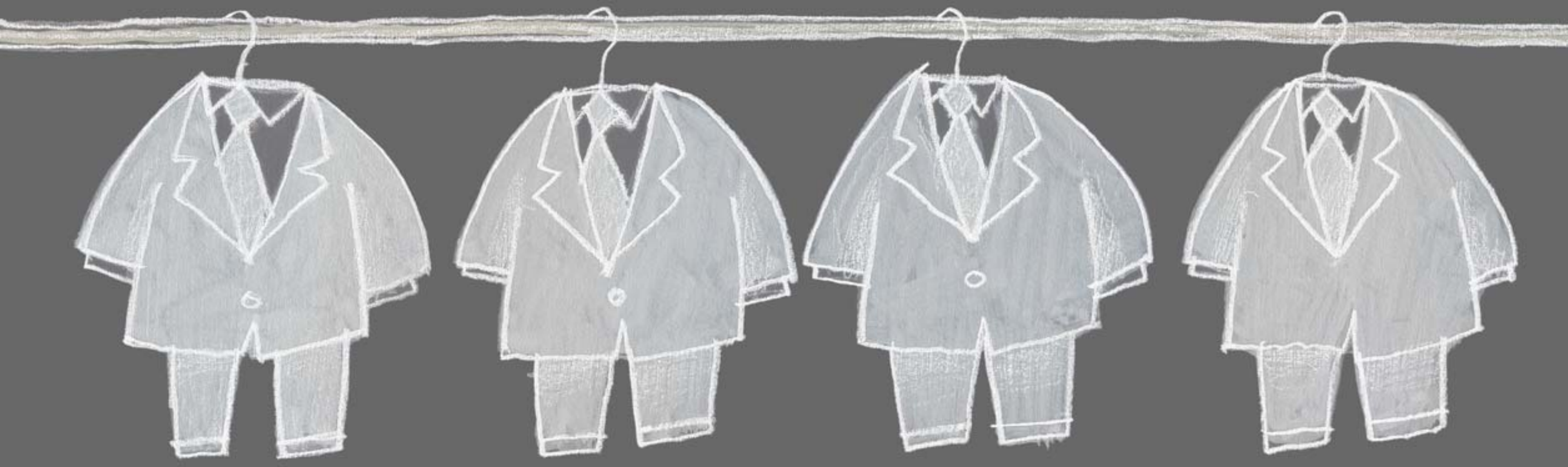


image I

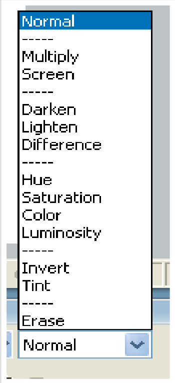


image II

While nearly all of us use Fireworks to create great looking graphics for our Web sites, a great many do not understand the science of color. I say science because there is a definite mathematics behind it. While I am not advocating doing complex math formulas every time we create a graphic, you will see that an understanding of the mathematical concepts will help us use the many color tools, available in Fireworks MX, better.

This month, we are going to take a slight departure from my usual hands-on approach to delve into the realm of the theoretical. However, you are certainly encouraged to experiment on your own and, hopefully, you will be able to do some great things with colors.

Color

Not that long ago, as Web graphic designers, we only had 216 colors to choose from. These were referred to as "Web-safe" colors that all browsers treated the same way. Any other color was, at best, a gamble as to what the browser would show.

That is no longer the case; today we can use millions of colors to tell our story, with a reasonable expectation of browser accuracy.

Despite the millions of colors, all Web colors are based on only three colors: red, green, and blue. We sometimes call these the Trichromatic colors. As we will soon see, what we call color is actually the term Hue. By changing the mix of those three hues, we can create brand new hues.

We use the hexadecimal system, or hex, for the amount of red, green, and blue in each hue. Let's take a look at how that works.

A hex represents a number between 1 and 15. The first 10 numbers are represented by 0–9. The next six numbers are represented by the letters A–F. Thus, A = 10, B = 11, C= 12, all the way to F = 15.

The amount of each of the three basic hues is measured on a scale from 0 to 255. But, if that is the case, how does a number between 1 and 15 add up to 255?.

Each color is represented by a two-digit hexadecimal representation. For instance: #AF, or #32, or #A3. We always begin with the number on the right; and that is always multiplied by 1. The number, or letter, on the left is always multiplied by 16. The results are then added together. Thus, #A3 would result in the following calculation:

```
(3 * 1) + (10 * 16) = 163
```

Remember, A = 10.

#FF would be:

```
(15 * 1) + (15 * 16) = 255
```

That's how we get our hue to 255.

But remember, we are working with three colors here: red, green, and blue. If we were to go into Fireworks and select the hue of Yellow, we might see a hex code of #FFFF00.

The first two hex digits are the amount of red, the next two the amount of green, and the last two the amount of blue. Thus, in this case yellow would have a red hue of 255, a green hue of 255, and no blue at all.

Just for a reference:

```
#000000 = Black
#333333 = Dark Gray
#CCCCCC = Light Gray
#FFFFFF = White
```

Try experimenting with different colors and work out the amount of RGB for each.

Blend Modes

When we work with a program like Fireworks MX, colors are not completely isolated from each other. We work with layers where objects sit on top of one another. We can control how the colors of these objects interact with each other by using the Blend Mode.

As a little experiment, try drawing three overlapping circles of different colors (see Image I).

We can then open the blending mode in the Property inspector (see Image II).

You can use the three circles, from Image I, to try out the blend modes above.

When we talk about Blending Modes, you need to know three terms:

- The *Base Color* can be thought of as

the background color. If you work with layers, think of it as a color in a lower layer.

- The *Blending Color* is the foreground color. Think of this as a color in a higher layer.
- The *Result Color* is the new color created from the blend.

Let's look at the mathematical relationships now.

Multiply takes the base color and multiplies it by the blend color so that:

```
Result = (Blend * Base)
```

Invert is a bit more involved. You have to multiply, then invert the result by 255. The resulting formula is:

```
Result =  1/255 (Blend*Base)
```

You will recall, from the hex discussion, that the smaller the number the darker the color.

Darken compares the Base and the Blend colors and uses whichever has the smallest value. This translates to the formula:

```
Result = min{Blend, Base}
```

Needless to say, *Lighten* does the opposite by selecting the maximum of the two colors.

```
Result = max{Blend, Base}
```

Difference subtracts the darker color (lower value) from the lighter color (higher value).

The other terms – *Hue, Color, Saturation,* and *Luminosity* – also have mathematical relationships. However, because of the complexity let's content ourselves with a description.

As we stated earlier, *hue* is the actual shade of the color. This blending mode takes the blend color and blends it with the luminosity and saturation of the base color.

Saturation measures the pureness of the color with respect to white. For instance, if you blend white with blue, blue's saturation decreases and moves toward pink. Without white, the saturation is 100 percent. This blend mode takes the saturation of the blend color and blends it with the luminosity and hue of the base color.

> "despite the millions of colors, all web colors are based on only **three colors: red, green, and blue**"

What we usually call *Color* is actually the result of blending the hue and saturation of the blend color and the luminosity of the base.

The *Luminosity* is actually the brightness. This is calculated by blending the luminosity of the blend color with the hue and saturation in the base.

The *Erase* feature erases the background color.

Finally, *Tint* is used to add gray to the background color.

Color Modes

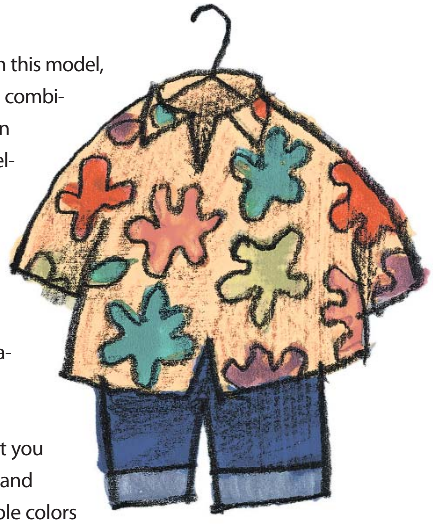
Now that we have an understanding of hexadecimal code, and blending modes, we can take a look at the other color modes that Fireworks MX has to work with.

If you open the Color Mixer and click on the Options Menu for the panel, you will see the alternative modes.

Like hexadecimal values, RGB defines colors as a ratio of red, green, and blue. The combination of the levels of the red, green, and blue is specified as a value between 0 (no color) and 255 (pure color). So, for example, pure red has a value of 255, 0, 0; pure blue is 0, 0, 255; pure black is 0, 0, 0; and pure white is 255, 255, 255.

CMY (Cyan Magenta Yellow) is used mostly in full-color printing, As opposed to the RGB model, CMY defines individual colors as a combination of cyan, magenta, and yellow. Again, values are between 0 (no color) and 255 (pure color). For example, pure magenta has a value of 0, 255, 0. In the reverse of the RGB model, however, pure black is 255, 255, 255, and pure white is 0, 0, 0.

The HSB (Hue Saturation Brightness) model defines colors in the same way that humans perceive color. In this model, each color is defined as a combination of the pure color in question (the hue), the relative brightness of that color from black (no brightness) to white (full brightness), and the saturation of the hue from gray tone (no saturation) to pure, vivid color (high saturation).



Grayscale is just what you think: you work in black and white, defining all possible colors as a percentage of black, where pure black is 100% and pure white 0%.

Opacity

We have one other tool available to us that we often don't associate with color: opacity. *Opacity* is actually a fancy word for transparency. An opacity of 100% is a solid color, and an opacity of 0% means completely transparent.

If you combine opacity with the Blend Modes, you can get some very fine gradations of blend. I strongly recommend that you use the three circle objects shown in Image I and experiment with blend modes *and* opacity.

Summary

This was a quick tour of the science behind colors. Now it's up to you to develop the art. While we saw the mathematical relationships of these various color tools, the mathematics now opens up endless possibilities.

Layers, coupled with Blending Modes and opacity, can add stunning effects to our graphics.

Charles E. Brown is the author of Fireworks MX From Zero to Hero *and* Beginning Dreamweaver MX. *He also contributed to* The Macromedia Studio MX Bible.
charles@charlesbrown.net

library power

by ron rockwell

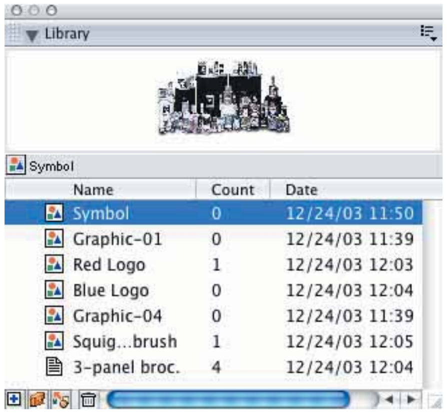
If your work involves company logos, a photo, or other graphic element that you use often, symbols can save you a lot of time. If you're in the beginning stages of creating a Web site or multipage document with recurring objects, a library of symbols can make layout changes quick and easy. Learning how to work with the library can mean hours of saved time.

What's a Library?

A library is a storage space within a FreeHand document that allows you to place various objects that you want to use more than once. Any vector graphic or bitmap object, block of text, or group of objects can be converted to a symbol that is stored in the library. Within the library you can group symbols in folders for easier viewing and access. Master pages are also stored in the library. A library from one document can be exported and then imported into another document on your computer or shared with a group of artists working on similar projects.

When a symbol is placed onto the page the placed object becomes an instance (child) of the symbol (parent). A symbol can be edited, and the changes will be reflected in every instance of that symbol within the document. This type of global editability is priceless when changes are required to many common objects in a multipage document. By releasing the child from the parent, individual instances of symbols can also be modified. Symbols can also be swapped with other symbols.

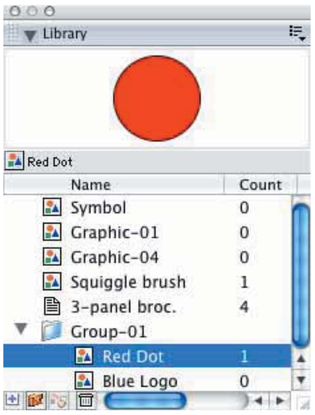
image I



Symbol

Name	Count	Date
Symbol	0	12/24/03 11:50
Graphic-01	0	12/24/03 11:39
Red Logo	1	12/24/03 12:03
Blue Logo	0	12/24/03 12:04
Graphic-04	0	12/24/03 11:39
Squig...brush	1	12/24/03 12:05
3-panel broc.	4	12/24/03 12:04

image II



Red Dot

Name	Count
Symbol	0
Graphic-01	0
Graphic-04	0
Squiggle brush	1
3-panel broc.	4
Group-01	
Red Dot	1
Blue Logo	0

What's in the Library?

You can place and find symbols and master pages in the library. If the library is not visible on your screen, choose Window>Library from the main menu bar, or click F11 for a keyboard shortcut. When you first open FreeHand MX, Library, Swatches, and Styles are docked in the Assets panel. To open or close a panel, just click the triangle to the left of the panel's name. In the default setup, FreeHand has a preview, and both graphics and master pages visible, as shown in Image I.

The preview window takes up the top of the Library panel. Its size doesn't change, but object previews expand to fit the window. A title bar beneath the preview window tells you which library object is selected. This may seem redundant, but if you have a long list, the object you've selected may not be visible in the list. The list contains the name of the object, the count – or number of times the object is used in the document, and the date the symbol was created. To save desktop space, you can grab the bottom-right corner of the panel and resize it, leaving only the name and count visible. The bottom border of the panel contains four buttons. Click the plus sign (New Symbol button) to copy a selected object in the document to the library as a new symbol. The original object is not an instance of the symbol and can be treated as any other FreeHand object. The next button is the New Group button. When you click it a folder is created (see Image II). Select any objects in the list and drag them into the folder. The folder can be opened or closed by clicking the flippy triangle. If you want to copy an object into a group, hold down the Option/Alt key as you select and drag. That allows you to have the symbol in two or more places at once. The third button is the Swap Symbol button. Clicking this button changes any selected objects in the document to the symbol you have selected in the list. Be careful with this button (see Changing Symbols). The last button is the Trash. Select an object in the library and click the Trash button to delete the object from the library. You will be given the opportunity to release all instances of the symbol by converting them to ordinary FreeHand graphics, or delete all instances altogether.

Symbols show up automatically in the list view area of the Library panel with generic names such as Graphic-01, Graphic-02, and so on. Master pages also appear, and if you create a custom brush tip it will be listed as a graphic. To give anything in the library a custom name, open the library options drop-down menu and choose Rename. Any object in the list can be moved up or down in the stacking order to make life more convenient for you.

What Is a Symbol?

Any vector object, bitmap object, or text in FreeHand can be made into a symbol. The benefit of using a symbol is that the library contains the information about the graphic object. When an instance of the symbol is placed on a page, FreeHand relays the file information from the library. Each time an additional instance is placed, FreeHand uses the same information, resulting in a smaller file size when the document is complete. You will be able to think of many ways in which you can use symbols in your workflow, but logos of different color combinations, type slugs, clip art, and animation elements for FreeHand animation can all be used effectively as symbols. For instance, your company may have a logo that is used on dark backgrounds and a different logo for light backgrounds, and yet another for black-and-white use or for use in a Flash animation. With each of those elements in a library – preferably in a group – you can grasp them instantly without searching your hard drive for a particular logo version. You can even import Flash movie and make symbols of them! It's great for testing the effectiveness or layout of a Web site. The Flash movie symbol can be placed anywhere in your design and resized. Then you can view the movie at the new size to see the impact you've made.

Master pages were covered in the last issue of *MXDJ* ("The Art of Pages," Vol. 2, issue 1) but in short, you can use master pages to create brochures, flyers, newsletters, animations, packaging,

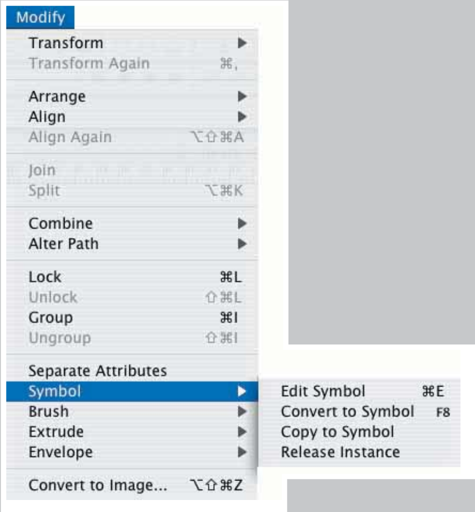


image III

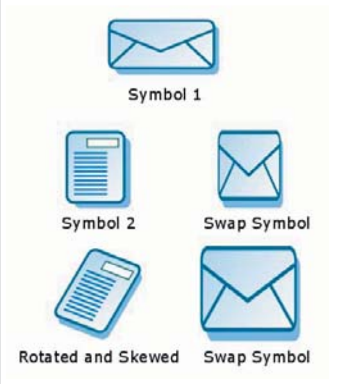


image IV

and Web sites. One slight note of caution however: if you place an action on an object in the creation of a master page, the action will not work unless you release the child page. At that point all your navigation will be functional – but you'll lose the ability to make global changes via modifying the master page.

When you go through the steps to create a custom brush tip, the tip will be added to the library and given a generic name. Drag an instance of the brush tip onto the pasteboard and release it from its parent. Then modify it and save it as a new brush tip. To me, it's quicker than going through the brush edit menu process. Graphic hose elements are not added to the library, but it's not a bad idea to add them manually due to the ease in which they can be modified and added to the Graphic Hose tool later. To add a symbol to the Graphic Hose, place the instance on the pasteboard and copy or cut it to the Clipboard. Then use the Paste Into button in the Graphic Hose dialog box. For instance, you could make symbols of different colored flowers and place them in the Graphic Hose and spray them throughout the document. Later you could decide to make all the blue flowers a dark violet and create the change by editing the symbol. The possibilities are almost endless.

How Are Elements Placed in the Library?

There are several ways to add a symbol to your library. The easiest is to select an object and press the F8 key, or click and drag the object from the pasteboard onto the library list. The object receives a generic name, and a thumbnail view of the new symbol appears in the library preview window. The original object becomes an instance of the symbol – the original object remains in the document, but it's not editable unless you release it from the parent. To have a choice in the matter, select the object and choose Modify>Symbol and make a further choice there as shown in Image III. Choose Convert to Symbol to change the object into an instance and create a symbol in the library – the same result as clicking and dragging as just described. If you choose Copy to Symbol, however, a symbol is created but the original object remains a normal FreeHand object and is editable without taking further steps. The last method of creating a symbol is to select an object and click the New Symbol button at the bottom-left corner of the Library panel. Doing so utilizes the Copy to Symbol process.

How Are Elements Retrieved from the Library?

To use a symbol, select it in the list view and drag its name, icon, or preview onto the pasteboard. That's all there is to it. The instance will be an exact replica of the original object. You can transform the instance as to scale, rotation, skew, and position, but there are a few things to keep in mind. First, if you change the scale nonuniformly, strokes change nonuniformly as well. For example, if you have an instance that is a square, and you stretch the width, top and bottom strokes will remain the same thickness, but vertical strokes will become wider.

Transforming instances will also impact any edits you make later to the symbol, or if you decide to swap symbols. A symbol has its own *aspect ratio* (relationship between height and width). If you modify the aspect ratio of an instance and then edit the original symbol or swap symbols, the replacement will take on the aspect ratio of the instance you modified. Doing so can create some pretty ugly surprises, so plan ahead. You can see an example of those surprises in Image IV. Symbol 1 is wide; symbol 2 is tall. Swapping symbol 1 for symbol 2 retains symbol 2's aspect ratio and becomes more or less square. When symbol 2 has been rotated and skewed as shown in the bottom of the image, and then swapped for symbol 1, the aspect ratio of symbol 2's bounding box is wider than before, and symbol 1 expands and shrinks to fit that shape. Notice, too, the width of lines and how they have been modified.

How Do You Edit a Symbol?

To edit a symbol, you can press Command+E (Control+E on Windows), or choose Edit Symbol from the Modify>Symbol menu. The process can also be started by double-clicking on the symbol's name in the library list. This brings up the Edit Symbol window, which looks just like any other FreeHand document, except there's no page outline, only the symbol itself. In the top left of the window there is an option to Auto-Update. Checking this option (it's on by default) will update the changed symbol in the document as you make the changes. If you uncheck Auto-Update, the changes happen to all the instances as soon as you close the Edit window. So, make any changes and modifications you wish, then click the Edit window's close button. Every instance of the symbol will reflect your modifications.

Say you want to change all instances of a logo to a new, improved logo.

Instead of editing the symbol, you can create a symbol of the new logo. Then select all the instances in the document (or plain FreeHand objects, it doesn't matter); choose the new symbol in the library list and click the Swap Symbols button in the bottom left of the Library panel. Everything you've selected will be an instance of the new symbol. Pretty slick.

When you edit a symbol, the changes are immediate in the current document, but no changes will occur to any other documents that contain instances of the symbol. In order to update those instances, you must export the edited symbol and import it into the older documents. You can export an entire library with the edited symbol, or if it moves you, just export the edited symbol with a new name and use Swap Symbol in the older documents after you've imported the new symbol.

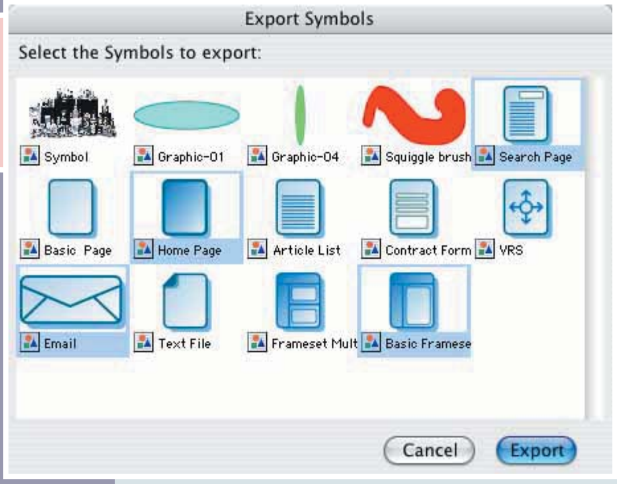


image V

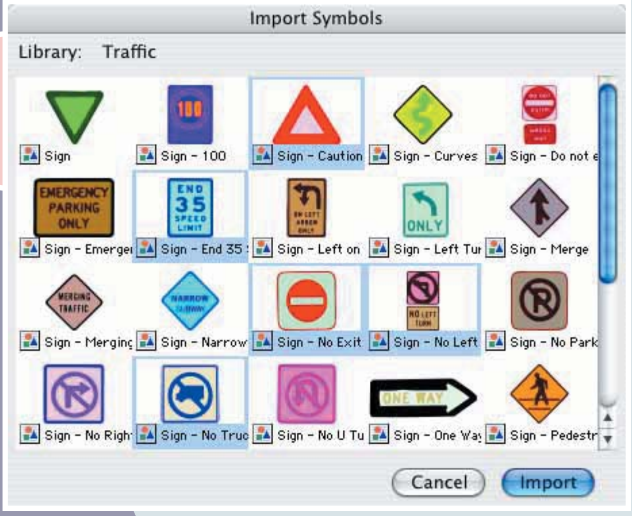


image VI

How Do You Release an Instance, and Why?

There are times when you need to release an instance. For example, you cannot use the 3D Rotation tool or the Fisheye Lens tool on a symbol. Then there's the chance that you'd like to change a color in one instance without modifying the symbol itself and changing all the other instances. When that happens, just select the instance and choose Modify>Symbol>Release Instance. This turns the instance into a simple grouped FreeHand object. Ungroup the object and modify as much as you like. There's no keyboard shortcut for Release Instance or Copy to Symbol, but you can create your own from Edit>Keyboard Shortcuts>Modify>Symbol. Doing so can save you a lot of time.

"The use of symbols and libraries can make your workflow much smoother"

How Do You Save a Library for Future Use?

To be able to use a library in other documents, you must first export the library. There are more dialog boxes than you would expect, but it's not too confusing. First, open the options menu in the Library panel and select Export. This will open the Export Symbols dialog box, which contains all the symbols in your library. You can select one or many of the symbols for export. Using the Shift key, you will select contiguous symbols, but with the Command/Control key you can select individual symbols from the window, as shown in Image V.

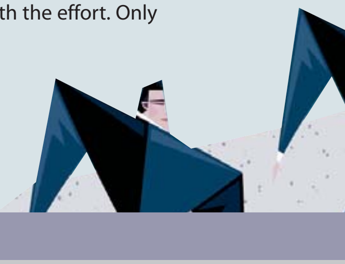
When you are finished with your selection, click the Export button. A standard Save window opens to the Symbols folder for FreeHand. You are prompted to name your library, and then you click on Save. At this point, you can import the library into any FreeHand document, make it part of your default page, and share it with co-workers. All they have to do is place the file in their Symbols folder. The actual folder is layered pretty deep on your Mac hard drive: HardDrive> Users>yourname>Library>Application Support>Macromedia>FreeHand MX>11>English>Settings>Symbols. The location is similar on a Windows machine.

Accessing a Library

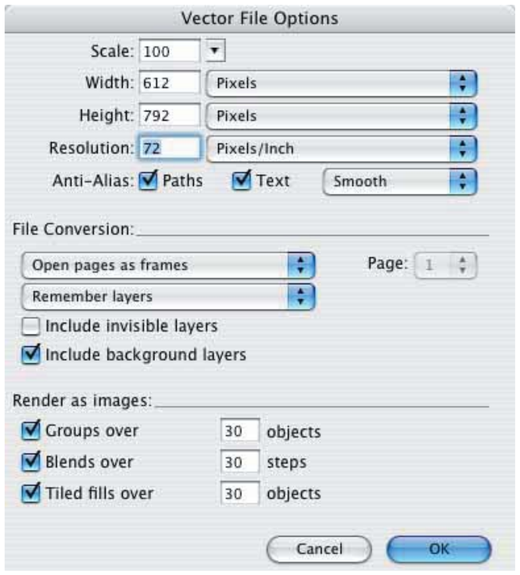
Importing a library is as simple as exporting one. Choose Import from the Library options drop-down menu. This opens the Symbols folder, and you choose the library you want. That opens the Import Symbols dialog box (see Image VI), which looks identical to the Export Symbols dialog box. Choose the symbols you wish to import, and click the Import button. All the symbols will appear in the library.

Libraries in Other Macromedia Programs

Although Flash and Fireworks have libraries, they work slightly differently than FreeHand's library. In Flash you can import a FreeHand library, but it's hardly worth the effort. Only



one element in the library will be imported to the timeline or library. Instead, import FreeHand objects as you would normally in your workflow. Fireworks allows you to import an entire library (or parts of one), though. The process is just a little different from importing a library into FreeHand. Open the Assets panel in Fireworks and click the Library tab. Then open the Assets options drop-down menu and scroll to Import Symbols. The hard part is navigating to FreeHand's Symbol folder (the path that was described earlier). Then just choose the symbol file you want to import and click on the Import button. The Vector File Options window opens. You can choose the size and resolution of the imports as shown in Image VII. Remember layers is not operable, as all objects within a single symbol are flattened automatically, and anything you may have had on a background layer in FreeHand has been moved up to the same layer as other objects. That said, when you click the OK button, the Import Symbols window opens with a list of all the symbols in the library you've chosen. Select a single file, or use Shift or Command/Control to select multiple files. Click the Import button and all the selected files will be in your Fireworks library.



Vector File Options

Scale: 100

Width: 612 Pixels

Height: 792 Pixels

Resolution: 72 Pixels/Inch

Anti-Alias: ☑ Paths ☑ Text Smooth

File Conversion:

Open pages as frames Page: 1

Remember layers

□ Include invisible layers

☑ Include background layers

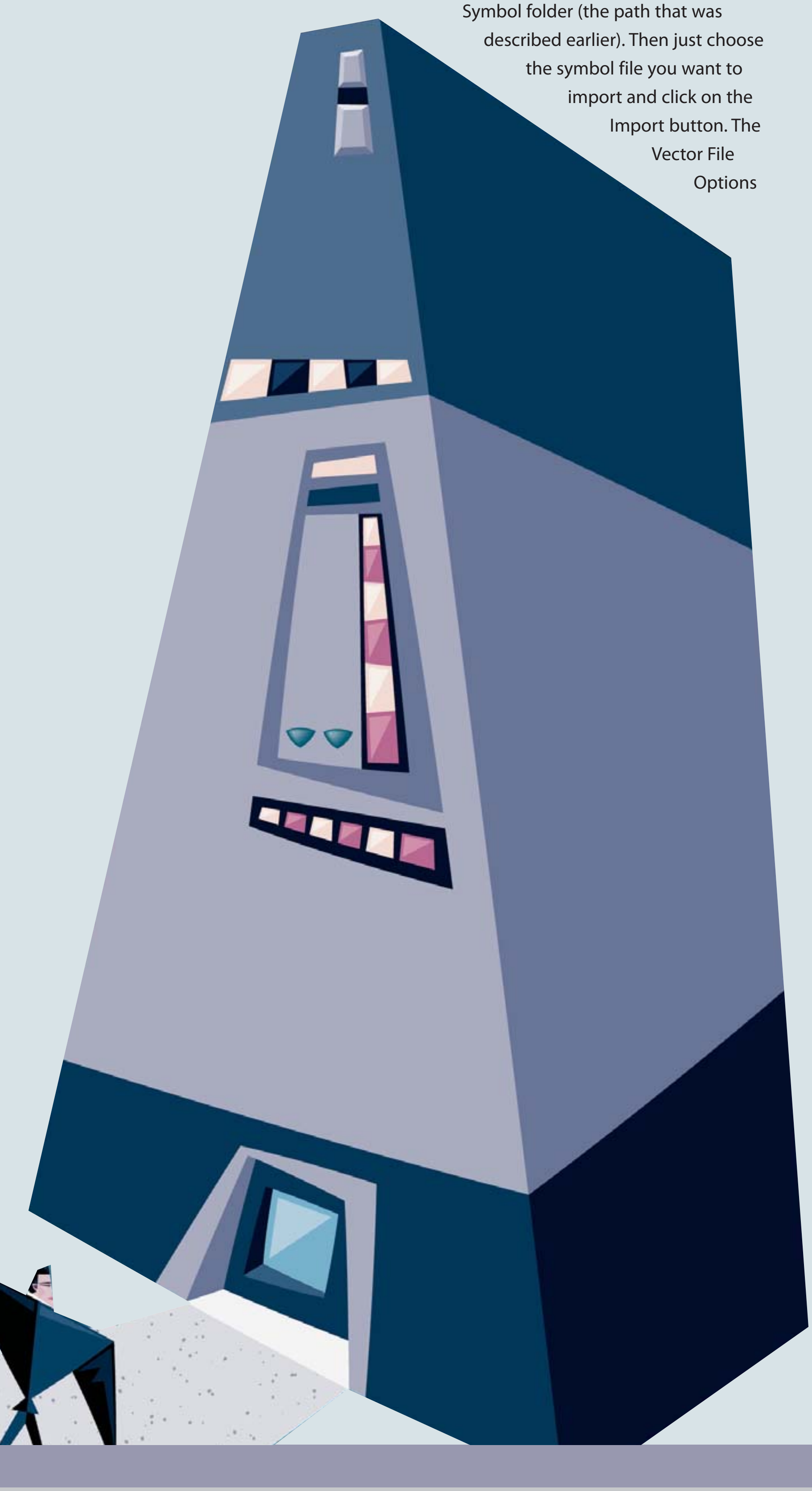
Render as images:

☑ Groups over 30 objects

☑ Blends over 30 steps

☑ Tiled fills over 30 objects

Image VII



The Science of Symbology

The use of symbols and libraries can make your workflow much smoother. When you become accustomed to their use, you will save time and money, and open new avenues of creativity. The first time you change 8 or 10 logos in a 32-page catalog with a single mouse click, you'll be a convert to creating and using symbols and libraries all the time.

Illustrator, designer, author, and Team Macromedia member Ron Rockwell lives and works with his wife, Yvonne, in the Pocono Mountains of Pennsylvania. He is the author of FreeHand 10 f/x & Design, *and coauthor of* Studio MX Bible *and* Digital Photography Bible. *He has Web sites at www.niduscorp.com and www.brainstormer.org. guru@brainstormer.org*

THINK BEFORE YOU CODE
TESTING YOUR CODE
KEEP LEARNING
ALWAYS USE SOURCE CONTROL
KNOW THE DATABASE
STYLE, CODE STANDARDS, AND READABILITY
7

Habits of Highly Effective ColdFusion Developers

by robi sen

This article will focus on practical ideas and habits that, when consistently applied, will allow you to develop and create better ColdFusion applications – regardless of which CF version you use, the hardware you run on, or the methodology you employ – as well as make your development easier.

1 Know the Database

Many ColdFusion developers do not take advantage of the power and features of their database, often trying to solve problems in ColdFusion that are better solved through the use of SQL and or specific features in their database.

The main reason for this is that many CF developers feel the need for more than an understanding of SQL syntax and basic relational database theory. As a CF developer you need to understand that each RDBMS has specific strengths and weaknesses, and powerful tools and features that can make your life easier and your application more scalable and robust. The database may also have features that, if not understood, can make development more difficult.

Make sure you have read up on your particular DB and are at least aware of what it can do for your application. For example, many developers try to port code developed for MS SQL to Oracle, and find their join statements throw exceptions, never realizing that while their SQL is correct in the generic sense, Oracle does not support the Join statement and requires you to use the symbolic representation for a join rather than the literal statement.

When building applications, try to build your data model and data access routines independently of your application code. In Web application projects, it can be impractical to totally abstract the data layer from your code, but even in smaller projects you can write many of your queries independently of your ColdFusion code and use literals in the place of variables. In larger applications, and depending on your architecture, creating CFCs, CF custom tags, or templates that you later include into other code that contains your data access and manipulation code, makes a lot of sense. It allows you to test your SQL, reducing the amount of redundant code written, and increasing code reuse. Finally, it also allows you to use database tools to help analyze your SQL to increase performance and find bottlenecks.

2 Style, Code Standards, and Readability

Developers tend to think about how to solve a particular problem in the most efficient or interesting way. They tend to forget that the most important thing in building an application that will be used and supported is maintainability.

The first thing you should always do when building any app is select or create style and coding guidelines. If you do not have one, Macromedia's internal Web development group has published their ColdFusion Coding Guidelines at http://livedocs.macromedia.com/wtg/public/coding_standards/contents.html. Regardless of what you choose, you need to make sure everyone in your development team adheres to it. Also remember that nothing ever stays in stasis and at times it makes sense to change your guidelines. When you do, be extra rigorous to make sure that people switch to the new guidelines.

Creating clean and understandable code is an art in itself and in most Web applications is far more important than creating the most efficient code. The reason for this is that with almost all Web applications, performance problems can be dealt with at a hardware and networking level, as long as your code and database are reasonably well designed (also, in my experience most performance issues reside at the database level). It is almost always easier to solve performance issues with more hardware than it is to try to rewrite your code to be more efficient. Also, hardware is a fixed capital cost, but labor isn't. In almost every project, labor associated with development is the single largest expense.

Furthermore, as applications grow and change over time, the underlying code often needs to be changed. Having code that does not conform to a documented standard makes it harder to collaborate and support any system regardless if it's ColdFusion or COBOL. While rigorous documentation can add to your development time, sometimes it actually speeds up development dramatically in that you actually know what you need to do when you sit down in front of your computer. Regardless, it's more than worth the effort to document your system to make it easier to support in the future.

> “What differentiates good prog
> average ones, is that good prog
> all of these strategies habits”

3

Think Before You Code

Before you ever write a line of code you should have a clear understanding of what you're about to do. There are many approaches to creating requirements, modeling systems, developing methodologies, etc. However, CF allows for such ease of development that many people start coding without clearly thinking about what the application needs to do. What's worse they tend to come back at a later date and slap on more code to create new features, without planning or thinking through what effects those changes will have on the system as a whole.

Before you start to work on your application you should create wireframe diagrams, logical flowcharts, and plainly written descriptions of what the application needs to do. You also need to document what your user is expected or required to do at each point in the processes that require user interaction. There is a large variety of methodologies for describing software systems, and some are very intense and demanding, such as the Universal Modeling Language and Unified Process. Others are much more flexible and easy to use, such as those described as Extreme Programming. Regardless of what you use, you should adopt some method of designing and describing your application and follow it all the time.

Most important, don't reinvent the wheel. If you're working on a problem and think it's something that people must have run into before, you're probably right. Make sure to use the tag libraries, your local user group e-mail list, and resources like CFCzone.org, CFlib.org, CFComet.com, Macromedia Dev Center, etc. Good programmers will always use or reuse good code when possible so they can focus on developing the application as a whole.

rammers from
grammers make

4

Exception Handling

People have different definitions of what exception handling is, and often make it synonymous with error trapping. In this article I will differentiate between simple trapping errors with CFTRY/CATCH blocks versus actually dealing with unexpected or unforeseen events, and creating a process to deal with those events. You can break down exception handling into several distinct categories: trapping, notification, and logging.

Trapping

This is what most people confuse with exception handling, which is the catching or trapping of errors, usually through a CFTRY/CFCATCH block and/or a CFTHROW command. You should use CFTRY/CFCATCH blocks in your code where an error or exception may occur, such as a CFQUERY, in that numerous unexpected things can happen in a query from a query failure to the database being down. Once you have trapped your error, you need to do something with it. The first thing you need to do is logically decide if the error needs to be dealt with in some way such as attempting to retry the failing process or notifying a system administrator.

Notification

Notifying a user of an error or a problem is often a good idea, but sometimes it can be a bad one. Many developers use e-mail to notify administrators of every exception that happens in an application – a classic mistake of beginning developers. In your planning stage you should think through what exceptions you need to notify a user about, or if you need to notify a user based on a threshold.

For example, you may have implemented a site-wide error-trapping system that e-mails a user whenever there is an exception. One day some new code is integrated that causes an exception every time a user comes to the site. Imagine if you have 70,000 users in an hour. You can easily see how this might bring your mail server to its knees, cause huge performance slowdowns in your server, and generally annoy everyone who is on the receiving end of the notification e-mail.

This is why you need to think through your notification and exception-handling strategy, and in this particular case, you may want to implement a threshold strategy where notification is sent based on severity and frequency. In addition, many errors should first be handled by the system and, if possible, the user should never know a problem has occurred. At the very minimum you can define site-wide error handling in the ColdFusion administrator and refer the user back to the home page. Users should never see a "raw" error message not only because it makes them uncomfortable, but also because it can expose information about the system that then creates a security risk. Many "crackers" purposely try to create exceptions in Web applications so that they can try to gather information that they can exploit to get access to the system.

Logging

Many CF developers confuse logging with notification and assume that since CF logs errors, and they have been notified by an error, that is all they need to do. In general, you want to think about what kind of exceptions the system might generate and what you are concerned about, and then log them when they are created. This can help you go back and reduce bottlenecks in your application and find hard-to-reproduce bugs, as well as look for hack attempts at your system. CFMX now makes use of CFLOG, which allows you to easily log your own information about any event you like and is a great way to log additional information about an exception when you trap it.

5 Testing Your Code

Most ColdFusion developers never really test their code, but would rather build their applications and work on a trial-and-error basis until the users are happy. This often leads to never-ending development cycles as well as bugs that appear after the project has been completed. Testing is a lot more than just making sure that the code runs, or changing features until the client is happy. Good testing can actually allow you to increase the speed with which you develop applications, reduce your frustration level, and increase the long-term maintainability of an application.

Testing forces you to think about the application and what you want your code to do beforehand, by forcing you to write your tests before or in parallel to your application code. If you cannot create a test for your code it often means you don't have enough information about what you need to develop, and should revisit your design and documentation.

Recently, the concept of unit testing and "test to code" practices have become more popular with ColdFusion, and now there are several useful tools. One of these is CFUnit, which is based on JUnit. It's available on the Macromedia DevNet Resource Kit (DRK) Volume 3, www.macromedia.com/devnet/mx/coldfusion/articles/cfunit.html. Also, Steve Nelson of Fusebox fame has developed a testing harness for Fusebox developers at www.secretagents.com/index.cfm?&fuseaction=tools.listtools. Finally, there's an open source ColdFusion testing initiative called ColdUnit, which shows great promise: http://sourceforge.net/projects/coldunit/.

6 Keep Learning

No matter what your level, you need to keep learning and pushing the envelope of your knowledge. Paradigms and practices in software development are always changing just like in any other discipline, and it's up to you to keep pace. Good programmers are always monitoring best practices, discussing problems with their peers, and learning new programming languages and technologies. One of the great things about ColdFusion is how easy it is to integrate Java, .NET, and XML. Other technologies and many problems that are extremely difficult to deal with using ColdFusion only are trivial when you use another language, but you won't know that if all you do is focus on ColdFusion.

Another great way to learn is to contribute to online discussions, and better yet, to contribute to your local ColdFusion or Macromedia user groups. If you want to learn a topic or really see if you know it, try teaching it to others.

7 Always Use Source Control

Most ColdFusion developers never use source control – even on large projects! While most ColdFusion developers may not think they need to use source control because their project is small, using source control can save you development time, allow you to track your work efforts, provide valuable backups, and allow you to easily manage many similar projects. This also applies to projects that you're doing yourself or at home. You'll save yourself a huge amount of frustration knowing that you can always revert to an earlier version of your code; also, the best time to build these habits is when you're working on your own. There's no good reason not to use source control and once you get into the habit, you'll find it takes no more time to use it than to not use it.

There are many excellent source control products out there, like Borland's StarTeam product, which is also a great tool for collaborative project development. Make sure to look also at Concurrent Version Control, which is the standard for concurrent version control and is available as a free open source tool from www.cvshome.org/. Windows versions are also available. If you're not using source control or version control, start now!

Conclusion

Probably the most important point of this whole article, and what differentiates good programmers from average ones, is that good programmers make all the above strategies habits! Many of us apply some or all of the techniques described above in a haphazard manner, and use them when it's convenient or when there is no pressure. Really good programmers do them as a matter of habit no matter what the pressures or time constraints are. The best way to improve as a developer is to always apply the same standards in development no matter the size of the project, even if you're just coding for fun.

Robi Sen is the vice president of services for Department 13 LLC, a technology consultancy, where he helps manage their large and government clients' projects. He has published a variety of articles on topics from .NET to managing outsourcing projects, and is a frequent speaker at ColdFusion User Groups across the United States. rsen@department13.com

A Little Something Xtra

Macromedia Director has proven itself in myriad applications as the development platform of choice. Although Director lets nonprogrammers create multimedia applications, there are a surprising number of developers with traditional software engineering backgrounds who use Director.

by tab julius

Why would software engineers, who are perfectly capable of writing in "professional" languages like C/C++, choose to develop applications in Lingo and Director instead of a "real" programming environment? Because Director, despite the occasional quirks, offers a significant advantage over more traditional languages.

I know this is why I originally chose Director as a development platform. After years of developing multimedia in C for DOS and Windows, I needed to find a way to create a product on the Macintosh. I was not in a position at that time to go up the Mac learning curve, so I started casting about for a cross-platform multimedia engine. And what I found was Macromedia Director v4.0, then new for Windows.

What intrigued me was not so much that Director could be used to develop cross-platfrom products – there were other programs that could claim as much – but that it had a powerful programming language, Lingo, *and* that it was extensible beyond that. Yes, extensible – if there were some feature I needed that Director didn't have, I could write that portion myself in C/C++, but I didn't necessarily have to write the whole application in C/C++. That was tremendously attractive.

Back then, extensions were done with XObjects, the precursor to Xtras. XObjects were much more limited than the current Xtras – you could write Lingo to call an XObject that could talk to the operating system and control attached devices or whatever you needed, and it could return information back to Director, but XObjects couldn't control things within Director itself (for instance, sprites, casts, etc.). It was a one-way street.

To make matters worse, the Windows and Macintosh SDKs for XObjects bore little relation to one another; many things were said about the Macintosh side in particular, few of them charitable.

All this changed with the introduction of Xtras with Director 5. Xtras in Director 5 was part of an effort to have all (or as many as possible) Macromedia products implement a common API called Macromedia Open Architecture, more simply known as MOA. One of the thoughts was that if you learned MOA, you could write plug-ins for multiple products. This would make things simpler for third parties, and presumably for Macromedia itself. One plug-in, if it didn't use product-specific functions, could work equally well in Authorware as Director. Another objective was to migrate certain key functions out of the runtime engine and into Xtras, so if an update was needed, just one Xtra needed to be updated and tested, but not the whole product.

And, in fact, this has all come true, with the possible exception of all products having been migrated over to MOA. Director supports MOA, as does Authorware, and certain of the server products.

What has also evolved over the years has been MOA itself. Originally an awkward collection of macros and involved COM-like C statements, it has slimmed down, is C++ friendly, and is much easier to start working with than it was originally. It has matured a lot, and nowadays it's comparatively easy to get up and running with the XDK (Xtra Development Kit).

In this article we'll look at just what is available via MOA and the XDK, and how you can put it to use. We won't actually try to write an Xtra, but I'll point you in the right direction in case you want to learn more, or take a crack at it yourself.

Types of Xtras

Macromedia Director recognizes four common types of Xtras. Scripting (or Lingo) Xtras let you develop your own Lingo commands, or make calls into the Xtra from Lingo. Sprite Xtras let you create a custom sprite type (for instance, a video sprite). Transition Xtras let you create your own custom transitions. Finally, Tool Xtras let you create tools that are available at authoring time (but not at runtime).

In truth there are actually five component types. The Sprite Xtra that I mentioned is really composed of two components – an asset component (this is the part that lives in the cast), and a Sprite Actor component (this is the part that exists on stage). Likewise, the Transition Xtra is really composed of an asset part and a transition actor part. You can, of course, use the actor and asset parts separately – someone might want to make a custom asset component to go with their scripting Xtra, say, but this is not common, although it is certainly allowable. It is far more common to find asset and actor components used in tandem to create Sprite Xtras or Transition Xtras.

Regrettably, the Director UI is misleading, as there is no one place to easily see what third-party Xtras are installed. They're scattered around the interface.

The understandably natural place to look first would be under the Xtras menu on the toolbar. This menu, in fact, lists only Tool Xtras, and Tool Xtras aren't the most common type of Xtra by a long shot. If you look there to see if your Xtra is installed, you'll probably be disappointed.

Where to look, then? Well, as mentioned, they're scattered about. You won't find transition Xtras until you go to insert a transition. Double-click on the transition frame and in the selection box you will find all the available transitions, including your third-party custom ones.

For Sprite Xtras, you have to go to the Insert menu, and there at the bottom you will find various types of Sprite Xtras that you can insert into the cast (technically you're creating an asset component when you do this). Once it's dragged onto the stage, that's the actor component. As mentioned, it's possible to create an asset without an actors should someone find some application where that would make sense.

Finally, Scripting, or Lingo, Xtras in Director MX you can now be seen them by clicking on the fifth Lingo button on the message window (or in the scripting window); the tooltip for this one says "Scripting Xtras" and if you click on it you will see a list of all available scripting Xtras, and their published commands. At the end of their command list is an entry that says "put interface". If you choose this, the command interface for the Xtra will be dumped to the message window.

Alternatively, from the message window, you can give the command "showxlib", which will list loaded Scripting Xtras. This is the old-fashioned (pre-MX) way. You can also show the interface for a scripting Xtra with the put interface command like so:

```
put interface(xtra "fileio")
```

Versions of Director prior to MX did not have the Scripting Xtras Lingo button, so users of Director 8.5 and earlier have to use this method to see scripting Xtras and their interfaces.

Writing Your Own Xtras

We don't have the space in this article for a how-to on writing your own Xtra, but I'll try to take some of the mystery out of it if you want to get started. We'll primarily look at Scripting and Sprite Xtras, with a quick glance at Tool and Transition Xtras.

First, you should at least know (or be willing to learn) C/C++ because Xtras are all done in C/C++. A few brave souls have tried to write Xtras in Delphi and Visual Basic, but it's not trivial to adapt the XDK for environments outside of the standard (the standard, by the way, is Microsoft Visual Studio on Windows, and CodeWarrior on Macintosh).

Second, you will need to have one of the aforementioned development environments, and the XDK from Macromedia. The XDK is free. To find it, go to the Macromedia site and type XDK in the search box. One of the top entries will be for the XDK for Director. The XDK typically lags behind versions of Director. As of this writing, Director MX is the most current release of Macromedia Director, but the most current XDK is for Macromedia Director 8.5.

> **"if there was some feature I needed that Director didn't have,** I could write that portion myself"

The XDK comes with an Examples folder that has examples of the various types of Xtras, but most important also has folders in each named "skeleton", which is really a template project that you can use for your own Xtras.

Xtras use GUIDs to identify themselves, and often each class (the registration class, the implemented class, etc.) will have their own GUID, so you will need to use the GUIDGEN.EXE tool that comes with Visual Studio to make these unique numbers. The original skeleton files had at least two GUIDs and multiple files (one for registration, one for the main class, etc.). The XDK for Director 8.5 has a "skeleton2" folder that is a new slimmed-down version of the skeleton, reworked to condense the Xtra down to just one .CPP/.H file pair, and just one GUID (for scripting). If you are a new Xtras developer, you might want to try the newer Skeleton2 template – it might be easier.

Xtras are really .DLLs on Windows, and Code Fragments on the Macintosh. They have a registration class, which Director will call upon starting up, and the registration class tells Director what the Xtra is capable of and what services it needs and offers. Beyond that, you can stuff whatever sort of class you want into an Xtra. Each type has its own specific calls that it should expect to receive.

Scripting Xtras

Scripting Xtras, originally called Lingo Xtras, are invoked only via Lingo commands. You can use them to make global, parent, or child functions available in Lingo.

A Global Lingo command (not to be confused with global Lingo variables) is one that is available to all users that have the Xtra installed, and is issued like other Lingo commands. If you made a global Lingo command called FixAllBugs, then you could issue it from the message window or in a handler just by giving the command FixAllBugs(). Each scripting Xtra provides Director with a list of commands it supports when it registers with Director. The command table (called the message table) has a special character in front of each command to indicate whether it is global, parent, or child. An asterisk in front indicates the command is global. Use global commands sparingly – there is a limited namespace – only one Xtra can have any given name of a command. Global commands are often preferred if the target user is generally not familiar with programming or instantiation of objects. In such cases, giving the command a unique name, possibly preceded by the initials of your name, your company name, or your Xtra. An example might be cfxLoadImage, as opposed to a more generic LoadImage name.

Lest you think it might be a good thing to lock up the market on generic command names, I would caution you that internal Lingo names take precedence. Were you to name your command the more generic "loadImage", and a later release of Director Macromedia implemented a new command called "loadImage", then yours would no longer work, and all programs written with your Xtra using that function would cease to work. I ran into this in the CastEffects Xtra we used to sell (it's now off the market). That Xtra had functions called GetPixel and SetPixel. The introduction of Imaging Lingo in Director 7 also saw the introduction of two Lingo commands, GetPixel and SetPixel. Suddenly the CastEffects commands were unreachable, and users saw errors on their calls. The commands were reimplemented with prefixes, as in cfxGetPixel and cfxSetPixel, which makes it much less likely that there will be a name conflict in the future.

Child functions have to be instantiated, but they allow the Xtra to keep local data per instance around. An example of instantiating an Xtra is:

```
global infileObj
infileObj =new(xtra "fileio")
```

This makes an instance of this Xtra, kept in the global variable "infileObj". Internally it has its own set of class variables. You could, from the same Xtra, make another instance, thus keeping two files open at the same time. Internally, the FileIO Xtra (which comes with Director) keeps track of each file separately via independent class variables.

Scripting Xtras have a simple interface, ::Call. When a user makes a function call that matches the message table your Xtra provided at registration, Director will call the Xtra and pass in a call pointer to transfer some parameters. One parameter, MethodSelector, is an index to the table so you know what command was requested (developers use an intermediate enum table to make this easier, as shown in the template and examples). Other parameters are an argument count and the arguments to the call themselves. Finally, there is a result value that is how you pass back whatever value you want to return to Lingo.

Sprite Xtras

Sprite Xtras are, as mentioned, really composed of two components – the asset component and the actor component. The asset component is that part that exists when placed in a castmember. At registration, the asset component tells Director what capabilities it has. Examples of capabilities it might declare are that it can animate the image (thumb) that you see when it's in the cast window, that it has a custom properties editor and/or media editor, that it provides its own about box, etc.

Unlike the simple scripting Xtra, the asset component of a Sprite Xtra has many entry points where MOA can call it. These include GetProp and SetProp so you can get (or set) a property on the castmember, for instance, perhaps the width of your asset. Other entry points include opportunities to stream media out or in, such as during save movie time, so your asset can save out information and load it back in the next time. This is really a two-part process, starting with Director asking how much information, if any, you need to save out, and then a second call asking you to actually write out the information.

There are calls to image or animate your thumbnail, to invoke your property editor or media editor, and so forth. By responding to these, and other, calls, you are creating the ties that integrate your Asset Xtra into Director. Finally, there are calls so you know when the Xtra has been dragged to the stage. It is through these that you would normally want to create a new actor instance, thus creating a sprite from your asset Xtra.

The SpriteActor class also has GetProp and SetProp (so you can, from Lingo, get a custom property from the sprite) but also functions specific to its life as a sprite, such as imaging and event handling.

There are really only two imaging callbacks. One is ::CollectChanges(), which is called to see if you have anything different to draw than what you've drawn already. A sprite with a static image might report no changes. A sprite with a dynamic image, such as video, might always report something changed. What you are reporting back is what rectangle in your draw area has changed, if any, and needs to be redrawn.

The other imaging callback is ::Image(), which is where you do your actual drawing. Director calls this when it needs your Xtra to render its image, which will be done to Director's internal compositing buffer, or Direct to screen, depending on how you declared you were going to handle your rendering. You get a pointer to the existing offscreen buffer and you then have an opportunity to modify your portion of it, thus drawing the image of your sprite onto the buffer.

The event handling callbacks include a very important one, ::GetCaps(). It is in the GetCaps callback that your sprite actor Xtra declares what it is capable of and/or what it wants. For instance, it can indicate that it wants to receive mouse events. Or if it has an unchanging, static image all the time. Or whether it prefers to draw directly to screen or offscreen.

Other callbacks are ::Event() which would receive any events it asked to be notified of (like keyboard or mouse events), and ::ContainsPoint(), which is where it can indicate if it contains the specified point.

This last one may seem a little unusual – you might think that Director would know if a point fell within a sprite's rectangle – but in reality that's true only for rectangular sprites. If your sprite imaged itself as anything other than a rectangle, then only your sprite knows whether or not a point would fall within the "live" portion of the sprite. The point test is for various things, including clicks and collision testing.

Transition Xtras

Transition Xtras are similar in structure to Sprite Xtras. They also have an asset component, and they have a transition actor component. We covered the asset component in the Sprite Xtras discussion; as far as actors go, the transition actor component is similar.

Transition actors are much simpler, though, because they're not interactive. A transition is a custom way to visually transition from one screen image to another. This is not an instant change, but usually some special effect. A simple example would be a crossfade. An involved example might be a burning page effect. Regardless, they all involve a starting point (the original image), an ending point (the destination image), and a series of steps to get from one to the other over a specific period of time.

As such, the transition actor has three main callbacks, ::Cue, ::Continue, and ::Finish.

Cue is used to tell the actor to get ready for the transition. You will get pointers to the starting and ending images, a rectangle indicating the portion to be transitioned, and a small info table providing information such as how long the transition is to take, in what size chunks, and so forth.

Continue is called repeatedly during the transition period, with each call affording you the opportunity to render the next step in your transition.

Finish is called when Director wants you to draw the final image, whatever it might be, and allow the Xtra to clean up after itself. Finish might be called even if you haven't done all the steps you intended – perhaps the time period expired or for some other reason the transition period is over. At any rate, the transition is over and that's that. Finish up.

Tool Xtras

A Tool Xtra is intended for use only at authoring time, not runtime. What your tool does is up to you – perhaps it organizes the score for you, or analyzes cast members, or converts them, or exports them to another format. A Tool Xtra has a simple interface, consisting of the calls ::GetEnabled() and ::Invoke().

GetEnabled is called when the user clicks on the Xtras menu on the toolbar. Director has to determine whether or not the menu item for your Tool Xtra is enabled or not. You might want some tools to always be enabled. You might enable others only if there is a selection made in the cast window, for instance. What you return here determines whether or not your tool will appear grayed out or active.

Invoke, as the name implies, means that the user has selected the Xtra from the Xtras menu, thus invoking it. Now your tool can go to work.

What Xtras Can Access

Because Xtras are typically written in C/C++, they can access all functions in the operating system. They can create threads, write files, trap messages, and so forth. But they also have access to various internal Macromedia functions. This is MOA, the public internals of Director.

Most of MOA is given in a COM-like format, with a long list of classes. We can't get into them in depth here, but as a quick overview:

- ***IMoaMmUtils2:*** Provides helpful utilities, mainly the ability to print messages in the message window.
- ***IMoaMmValue:*** Provides value conversions. Since Lingo is a loosely typed language – a variable can hold an integer one moment, a string the next – all data is passed around in MoaMmValue format, which is really a two-part structure – one part indicates what the value is (e.g., a string), and the other part is a data pointer to the information itself. IMoaMmValue provides the conversion functions to get information in and out of MoaMmValue format.
- ***IMoaMmList:*** Provides functionality to create and access Lingo lists (property lists and linear lists).
- ***IMoaDrPlayer:*** Provides access to the current player (the runtime engine). From here you can get a pointer to the current active movie (IMoaDrMovie), and then to the score (IMoaDrScore Access), the cast, and so on, or you can use it to call a handler in the current movie, allowing you to call Lingo from within the Xtra.

There are many, many more interfaces and classes. There are some for managing memory, for managing files, for streaming, for creating image objects, for traversing score frames, and so on. You can't get at absolutely everything in Director, but you can get at a surprising amount.

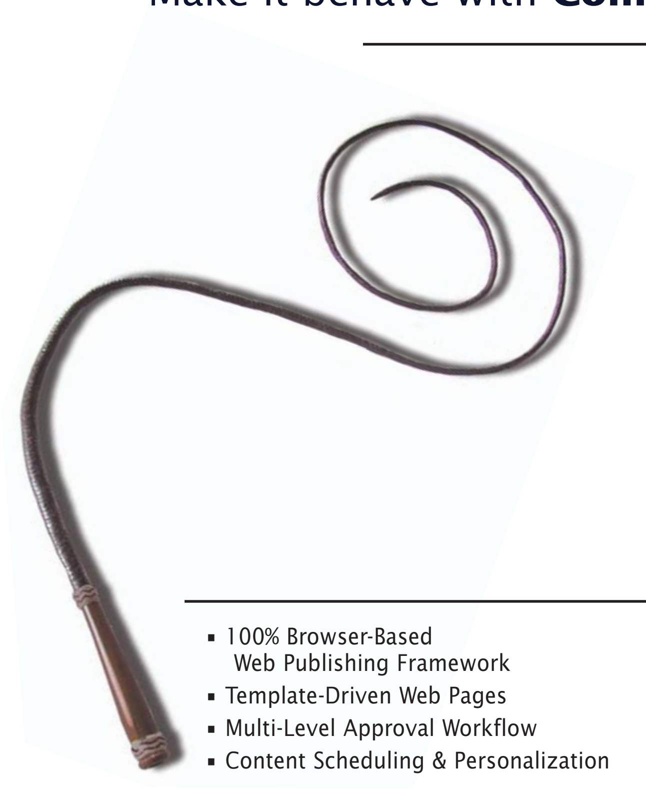
Where to Go From Here

If you are seriously interested in writing Xtras, you'll want to join the e-mail listserv mmxdk-l hosted by Macromedia. At the time of this writing they had just switched list-servers, and there was no public sign-up form for the new server yet. I would suggest just going to their site and searching on mmxdk; if that doesn't help, e-mail me at tab@penworks.com and I will see what I can find out.

Certainly you should get the XDK and read through the docs. Try compiling sample Xtras. You can also buy many off-the-shelf Xtras – check out Macromedia's Director Exchange, and you can have custom Xtras made as well if you're not in a position, time-wise or otherwise, to write your own.

The important thing to take away from all this is that Director is not a self-contained, static entity. You can extend it far and wide. For many years we've written custom Xtras for all sorts of applications – everything from controlling ZModem file transfers to controlling DV camcorders over FireWire, from exporting animations to proprietary formats, to music, to interfacing with Lotus Notes. There are few limitations, and since Shockwave runs in browsers, you effectively give yourself access to all this functionality in a browser environment as well as a desktop app. I can think of no other tool on the market that gives cross-platform rapid application development, in both browser and desktop, and still lets me extend the product as necessary to meet my needs. That's why I continue to use Macromedia Director to develop multimedia software products.

Tab Julius has been writing software since the mid-'70s, and now works for a software firm developing medical imaging applications, although he still does limited consulting on the side. tab@penworks.com



Now More Than Ever Before...

JAVA DEVELOPER'S JOURNAL

JDJ

Means **Business**

EXCLUSIVE: WHERE IS TECH GOING IN 2004?

JAVA DEVELOPER'S JOURNAL

JDJ

JANUARY 2004 VOLUME:9 ISSUE:1

SPECIAL REPORT: SCOTT McNEALY 2004 PREDICTIONS

PLUS...

Viewpoint: Deutsch's Fallacies, 10 Years After

Feature: Multiplatform Frames

Data Integration: Which EII Solution Is Right for You?

EDGE 2004 EAST International Conference & Expo BOSTON, MA FEBRUARY 24-26, 2004 See pages 47-48 for details

TAKE A RIDE WITH THE CEO OF BEA

RETAILERS PLEASE DISPLAY UNTIL MARCH 31, 2004 $5.99US $6.99CAN

THE WORLD'S LEADING i-TECHNOLOGY MAGAZINE WWW.JAVADEVELOPERSJOURNAL.COM

Subscribe Now!

$69.99 1 YEAR/ 12 ISSUES

www.SYS-CON.com/JDJ

or call

1-888-303-5282

The World's Leading *i*-Technology Magazine Just Got Better!

Now with expanded business coverage, *JDJ* is the world's premier independent, vendor-neutral print resource for the ever-expanding international community of IT professionals.

The Most Popular *i*-Technology Magazine in the World !

SYS-CON MEDIA

The World's Leading i-Technology Publisher

A LIMITED TIME SAVINGS OFFER FROM SYS-CON MEDIA

SUBSCRIBE TODAY TO MULTIPLE MAGAZINES

AND SAVE UP TO $400 AND RECEIVE UP TO 3 FREE CDs!*

RECEIVE YOUR DIGITAL EDITION ACCESS CODE INSTANTLY WITH YOUR PAID SUBSCRIPTIONS

CALL TODAY! 888-303-5282

Pick a 3-Pack, a 6-Pack or a 9-Pack

□ 3-Pack	□ 1YR	□ 2YR	□ U.S.	□ Can/Mex	□ Intl.
□ 6-Pack	□ 1YR	□ 2YR	□ U.S.	□ Can/Mex	□ Intl.
□ 9-Pack	□ 1YR	□ 2YR	□ U.S.	□ Can/Mex	□ Intl.

TO ORDER •Choose the Multi-Pack you want to order by checking next to it below. •Check the number of years you want to order. •Indicate your location by checking either U.S., Canada/Mexico or International. •Then choose which magazines you want to include with your Multi-Pack order.

□ MX Developer's Journal

U.S. - Two Years (24) Cover: $143	You Pay: $49.99 /	Save: $167 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $29.99 /	Save: $60
Can/Mex - Two Years (24) $168	You Pay: $79.99 /	Save: $137 + FREE $198 CD
Can/Mex - One Year (12) $84	You Pay: $49.99 /	Save: $40
Int'l - Two Years (24) $216	You Pay: $89.99 /	Save: $127 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $59.99 /	Save: $30
Digital Edition - One Year (12)	You Pay: $19.99	

□ Linux World Magazine

U.S. - Two Years (24) Cover: $143	You Pay: $79.99 /	Save: $63 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $39.99 /	Save: $32
Can/Mex - Two Years (24) $168	You Pay: $119.99 /	Save: $48 + FREE $198 CD
Can/Mex - One Year (12) $84	You Pay: $79.99 /	Save: $4
Int'l - Two Years (24) $216	You Pay: $176 /	Save: $40 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

□ Java Developer's Journal

U.S. - Two Years (24) Cover: $144	You Pay: $89 /	Save: $55 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $49.99 /	Save: $22
Can/Mex - Two Years (24) $168	You Pay: $119.99 /	Save: $48 + FREE $198 CD
Can/Mex - One Year (12) $84	You Pay: $79.99 /	Save: $4
Int'l - Two Years (24) $216	You Pay: $176 /	Save: $40 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

□ Web Services Journal

U.S.- Two Years (24) Cover: $168	You Pay: $99.99 /	Save: $68 + FREE $198 CD
U.S. - One Year (12) Cover: $84	You Pay: $69.99 /	Save: $14
Can/Mex - Two Years (24) $192	You Pay: $129 /	Save: $63 + FREE $198 CD
Can/Mex - One Year (12) $96	You Pay: $89.99 /	Save: $6
Int'l - Two Years (24) $216	You Pay: $170 /	Save: $46 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

□ .NET Developer's Journal

U.S. - Two Years (24) Cover: $168	You Pay: $99.99 /	Save: $68 + FREE $198 CD
U.S. - One Year (12) Cover: $84	You Pay: $69.99 /	Save: $14
Can/Mex - Two Years (24) $192	You Pay: $129 /	Save: $63 + FREE $198 CD
Can/Mex - One Year (12) $96	You Pay: $89.99 /	Save: $6
Int'l - Two Years (24) $216	You Pay: $170 /	Save: $46 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

□ XML-Journal

U.S. - Two Years (24) Cover: $168	You Pay: $99.99 /	Save: $68 + FREE $198 CD
U.S. - One Year (12) Cover: $84	You Pay: $69.99 /	Save: $14
Can/Mex - Two Years (24) $192	You Pay: $129 /	Save: $63 + FREE $198 CD
Can/Mex - One Year (12) $96	You Pay: $89.99 /	Save: $6
Int'l - Two Years (24) $216	You Pay: $170 /	Save: $46 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

□ WebLogic Developer's Journal

U.S. - Two Years (24) Cover: $360	You Pay: $169.99 /	Save: $190 + FREE $198 CD
U.S. - One Year (12) Cover: $180	You Pay: $149 /	Save: $31
Can/Mex - Two Years (24) $360	You Pay: $179.99 /	Save: $180 + FREE $198 CD
Can/Mex - One Year (12) $180	You Pay: $169 /	Save: $11
Int'l - Two Years (24) $360	You Pay: $189.99 /	Save: $170 + FREE $198 CD
Int'l - One Year (12) $180	You Pay: $179 /	Save: $1

□ ColdFusion Developer's Journal

U.S. - Two Years (24) Cover: $216	You Pay: $129 /	Save: $87 + FREE $198 CD
U.S. - One Year (12) Cover: $108	You Pay: $89.99 /	Save: $18
Can/Mex - Two Years (24) $240	You Pay: $159.99 /	Save: $80 + FREE $198 CD
Can/Mex - One Year (12) $120	You Pay: $99.99 /	Save: $20
Int'l - Two Years (24) $264	You Pay: $189 /	Save: $75 + FREE $198 CD
Int'l - One Year (12) $132	You Pay: $129.99 /	Save: $2

□ Wireless Business & Technology

U.S. - Two Years (24) Cover: $144	You Pay: $89 /	Save: $55 + FREE $198 CD
U.S. - One Year (12) Cover: $72	You Pay: $49.99 /	Save: $22
Can/Mex - Two Years (24) $192	You Pay: $139 /	Save: $53 + FREE $198 CD
Can/Mex - One Year (12) $96	You Pay: $79.99 /	Save: $16
Int'l - Two Years (24) $216	You Pay: $170 /	Save: $46 + FREE $198 CD
Int'l - One Year (12) $108	You Pay: $99.99 /	Save: $8

□ WebSphere Developer's Journal

U.S. - Two Years (24) Cover: $360	You Pay: $169.99 /	Save: $190 + FREE $198 CD
U.S. - One Year (12) Cover: $180	You Pay: $149 /	Save: $31
Can/Mex - Two Years (24) $360	You Pay: $179.99 /	Save: $180 + FREE $198 CD
Can/Mex - One Year (12) $180	You Pay: $169 /	Save: $11
Int'l - Two Years (24) $360	You Pay: $189.99 /	Save: $170 + FREE $198 CD
Int'l - One Year (12) $180	You Pay: $179 /	Save: $1

□ PowerBuilder Developer's Journal

U.S. - Two Years (24) Cover: $360	You Pay: $169.99 /	Save: $190 + FREE $198 CD
U.S. - One Year (12) Cover: $180	You Pay: $149 /	Save: $31
Can/Mex - Two Years (24) $360	You Pay: $179.99 /	Save: $180 + FREE $198 CD
Can/Mex - One Year (12) $180	You Pay: $169 /	Save: $11
Int'l - Two Years (24) $360	You Pay: $189.99 /	Save: $170 + FREE $198 CD
Int'l - One Year (12) $180	You Pay: $179 /	Save: $1

*WHILE SUPPILES LAST. OFFER SUBJECT TO CHANGE WITHOUT NOTICE

3-Pack

Pick any 3 of our magazines and save up to $275^00

Pay only $175 for a 1 year subscription plus a FREE CD

- 2 Year – $299.00
- Canada/Mexico – $245.00
- International – $315.00

6-Pack

Pick any 6 of our magazines and save up to $350^00

Pay only $395 for a 1 year subscription plus 2 FREE CDs

- 2 Year – $669.00
- Canada/Mexico – $555.00
- International – $710.00

9-Pack

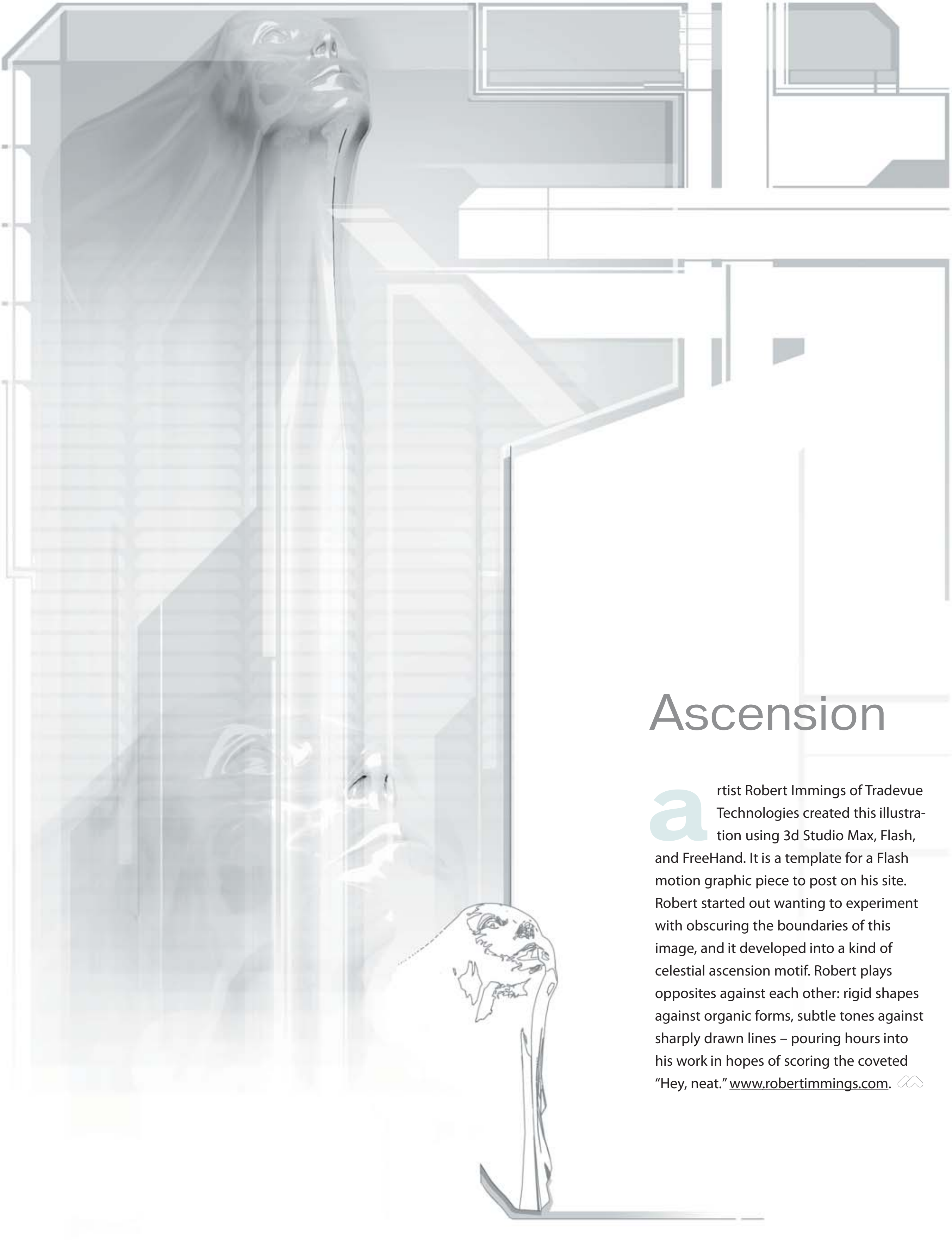
Pick 9 of our magazines and save up to $400^00

Pay only $495 for a 1 year subscription plus 3 FREE CDs

- 2 Year – $839.00
- Canada/Mexico – $695.00
- International – $890.00

Subscribe Online Today www.sys-con.com/2001/sub.cfm

SYS-CON MEDIA



Ascension

Artist Robert Immings of Tradevue Technologies created this illustration using 3d Studio Max, Flash, and FreeHand. It is a template for a Flash motion graphic piece to post on his site. Robert started out wanting to experiment with obscuring the boundaries of this image, and it developed into a kind of celestial ascension motif. Robert plays opposites against each other: rigid shapes against organic forms, subtle tones against sharply drawn lines – pouring hours into his work in hopes of scoring the coveted "Hey, neat." www.robertimmings.com.

Liberate your ColdFusion Talents

NQcontent v2

WEB CONTENT MANAGEMENT

The only CMS with the developer in mind

NQcontent is the ColdFusion based Content Management platform that eliminates repetitive development work and extends your talents to give you room to deliver robust, dynamic, websites that work all the time!

COLDFUSION Developer's Journal CFDJ Readers' Choice AWARD

- Best Content Management Tool 2002 **WINNER**
- Most Innovative Application 2002 **WINNER**
- Best e-Business Software 2002 **WINNER**
- Best Web Application 2002 **WINNER**

Visit **www.nqcontent.com** for more information

Consumer Products
Music
Fighting AIDS
Water