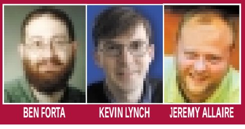
ANNOUNCING COLDFUSION EDGE 2001 P.8



U.S. $8.99 (Canada $9.99)

ColdFusion Developer's Journal

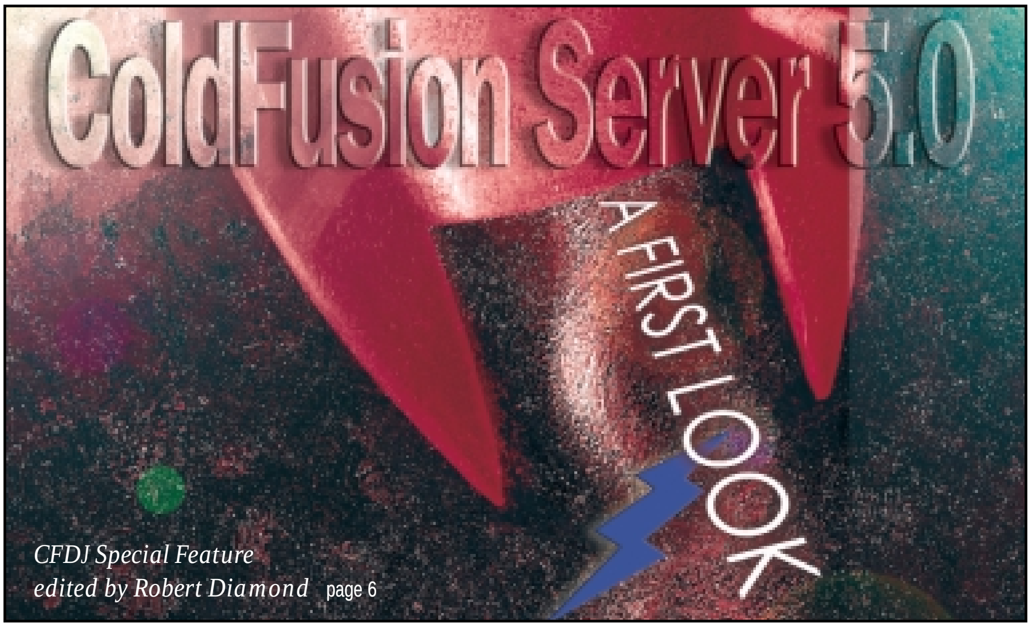
ColdFusionJournal.com

May 2001 Volume: 3 Issue: 5

Sept. 24–26, 2001 New York, NY
COLDFUSION EDGE fasttrack 8

Oct. 22–25, 2001 Santa Clara, CA
XML EDGE conference&expo 2001 14

Sept. 24–25, 2001 New York, NY
web services EDGE conference&expo
Oct. 24–25, 2001 Santa Clara, CA 53



CFDJ Special Feature
edited by Robert Diamond page 6

SYS-CON MEDIA

ColdFusion Developer's Journal

INTERNATIONAL ADVISORY BOARD
JEREMY ALLAIRE, CHARLES AREHART, MICHAEL DINOWITZ, STEVE DRUCKER, BEN FORTA, HAL HELMS, KEVIN LYNCH, KARL MOSS, AJIT SAGAR, MICHAEL SMITH, BRUCE VAN HORN

EDITOR-IN-CHIEF ROBERT DIAMOND
ART DIRECTOR JIM MORGAN
EXECUTIVE EDITOR M'LOU PINKHAM
EDITOR NANCY VALENTINE
MANAGING EDITOR CHERYL VAN SISE
ASSOCIATE EDITOR JAMIE MATUSOW
ASSISTANT EDITOR GREGORY LUDWIG
PRODUCT REVIEW EDITOR TOM TAULLI
TIPS & TECHNIQUES EDITOR MATT NEWBERRY

WRITERS IN THIS ISSUE
ANGELO ALVAREZ, DAVID ANDERSON, CHARLES AREHART, RAYMOND CAMDEN, ERON COHEN, ROBERT DIAMOND, BEN FORTA, CHRISTOPHER GRAVES, HAL HELMS, MICHAEL SMITH, MATT TATAM

SUBSCRIPTIONS
FOR SUBSCRIPTION REQUESTS PLEASE CALL
1 800 513 7111
OR GO TO: **WWW.SYS-CON.COM**

DOMESTIC $89.99/YR. (12 ISSUES)
CANADA/MEXICO $99/YR
OVERSEAS $129/YR
BACK ISSUES $10 U.S. $15 ALL OTHER

PRESIDENT AND CEO FUAT A. KIRCAALI
VP, PRODUCTION & DESIGN JIM MORGAN
SENIOR VP, SALES & MARKETING CARMEN GONZALEZ
VP, SALES & MARKETING MILES SILVERMAN
ADVERTISING ACCOUNT EXECUTIVE RONALD J. PERRETTI
ASSISTANT CONTROLLER JUDITH CALNAN
CREDIT & COLLECTIONS CYNTHIA OBIDZINSKI
ACCOUNTS PAYABLE JOAN LAROSE
ADVERTISING ACCOUNT MANAGER ROBYN FORMA
ADVERTISING ACCOUNT MANAGER MEGAN RING
ASSOCIATE SALES MANAGER CARRIE GEBERT
ASSOCIATE SALES MANAGER CHRISTINE RUSSELL
SALES ASSISTANT ALISA CATALANO
CIRCULATION MANAGER CHERIE JOHNSON
VP, EVENTS CATHY WALTERS
TRADE SHOW MANAGER DONA VELTHAUS
SALES EXECUTIVE, EXHIBITS MICHAEL PESNICK
SALES EXECUTIVE, EXHIBITS RICHARD ANDERSON
ART DIRECTOR ALEX BOTERO
ASSISTANT ART DIRECTOR CATHRYN BURAK
ASSISTANT ART DIRECTOR LOUIS F. CUFFARI
ASSISTANT ART DIRECTOR RICHARD SILVERBERG
GRAPHIC DESIGNER ABRAHAM ADDO
GRAPHIC DESIGNER AARATHI VENKATARAMAN
WEB DESIGNER STEPHEN KILMURRAY
WEB DESIGN INTERN PURVA DAVE
SYS-CON EVENTS MANAGER ANTHONY D. SPITZER

EDITORIAL OFFICES
SYS-CON MEDIA, INC. 135 CHESTNUT RIDGE RD., MONTVALE, NJ 07645
TELEPHONE: 201 802-3000 **FAX:** 201 782-9600

COLDFUSION DEVELOPER'S JOURNAL (ISSN #1523-9101) IS PUBLISHED MONTHLY (12 TIMES A YEAR) FOR $89.99 BY SYS-CON PUBLICATIONS, INC., 135 CHESTNUT RIDGE RD.,MONTVALE, NJ 07645

POSTMASTER
SEND ADDRESS CHANGES TO:
COLDFUSION DEVELOPER'S JOURNAL
SYS-CON MEDIA
135 CHESTNUT RIDGE RD.,MONTVALE, NJ 07645

WORLDWIDE DISTRIBUTION
BY **CURTIS CIRCULATION COMPANY** 730 RIVER ROAD, NEW MILFORD, NJ 07646-3048 PHONE: 201 634-7400

DISTRIBUTED IN USA
BY **INTERNATIONAL PERIODICAL DISTRIBUTORS**
674 VIA DE LA VALLE, SUITE 204, SOLANA BEACH, CA 92075
619 481-5928

Spectra…The Latest Trend

BY **ROBERT DIAMOND**

When Spectra was first released, ***CFDJ*** devoted a lot of editorial space to it. To say the least, we didn't receive the most positive feedback to articles on software that ran on top of ColdFusion, required an additional expenditure, and at first didn't fit into many people's development plans. Due to this response, we scaled back our coverage. As time went on, however, the trend started to turn, and we began receiving letters saying the opposite – asking us for more information on Spectra. In response to that is this month's focus issue.

Content management is something that everyone developing a large site has to consider at one point or another, and Spectra's content management features are certainly some of its best and most used highlights. For all those using them, as with any similar tools, one of the largest challenges is optimizing it and keeping it running smoothly.

Read on for David Anderson's fantastic piece on the subject, which details how to use Squid, a third-party solution to cache pages in RAM, greatly lightening the load on your Web server. If you run your own Web server, as we do here at ***CFDJ***, I certainly don't need to get into the advantages.

With the merger complete between Macromedia and Allaire, the next several months will bring increased integration between the two companies' formerly separate product lines. In this issue we bring you the first in a series of articles that will keep you up to date on the changes. Matt Tatam has written a fantastic overview on how to use Flash's abilities to dynamically communicate with server-side applications to set up graphical interaction with remote, data-based information. The article details how to effectively implement the "Model View Control Pattern," integrating it with Spectra's Site Layout Model, which is stored as an XML object. This feature really does cover it all.

Ray Camden weighs in with a comprehensive update on what's new with Spectra 1.5. The .5 is quite misleading in that it's a major update with several new features for productivity, caching, content versioning, authoring, and more. Also, when you get to the end of the article, you'll get a tiny peek at the next version of Spectra, coming soon to a server near you.

ColdFusion 5.0

And speaking of new software – this month we take our first official feature look at the newest version of the ColdFusion Application Server. It's got loads of enhancements that we've been hearing about for months – things like new tags, the ability to query a query, and a superb new scripting ability. That's just the beginning of it, though, with built-in graphing support and more new features than you can shake a stick at. The first of many articles on CF 5.0, it's a good overview of what Macromedia has done with the product since 4.5. I don't think you'll be disappointed….Read the article as you're on your way to buy it.

CFDJ: The Complete Works

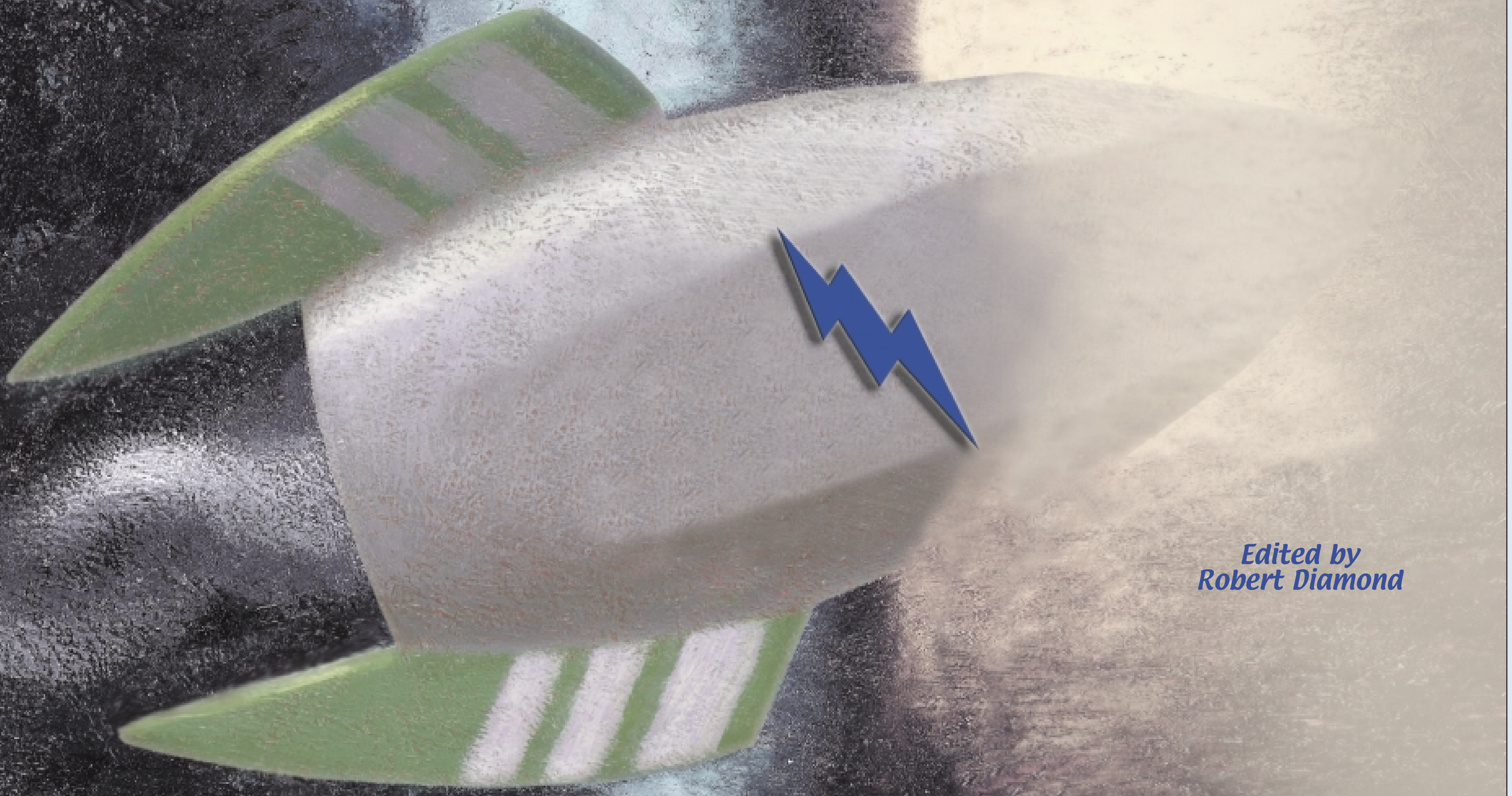
I want to close this month's editorial with a quick, shameless, commercial plug. Now available for purchase at JDJStore.com is ***CFDJ*: The Complete Works**, a CD collection of every article ever published in the magazine. Indexed by yours truly, the CD contains over 250 articles organized by category in an easily printable format. Also included is the source code for every article and PDFs of each issue. Best of all, it's completely searchable. Check it out, it's worth a look.

Robert Diamond

ROBERT@SYS-CON.COM

ABOUT THE AUTHOR
Robert Diamond is editor-in-chief of **ColdFusion Developer's Journal** *as well as* **SYS-CON**'s *newest magazine,* **Wireless Business & Technology**. *Named one of the "Top thirty magazine industry executives under the age of 30" in Folio magazine's November 2000 issue, Robert is currently a senior in the School of Information Studies at Syracuse University.*

ColdFusion Server 5.0 A FIRST LOOK



macromedia COLDFUSION 5

Edited by Robert Diamond

The biggest new release of ColdFusion in recent memory is now upon us. We've been hearing about it for months, saw it previewed at the Allaire Developer's Conference last year, and have been reading about it ever since – ColdFusion Server 5.0.

It will change forever the way you develop in CF

ColdFusion Server 5.0 A FIRST LOOK

macromedia COLDFUSION 5

Now it's finally ready – it will improve, enrich, and in some ways even change forever the way you develop in ColdFusion.

ColdFusion Server 5.0 delivers a wealth of new features and capabilities that make it easy to build powerful Web solutions. It includes a number of enhancements designed to increase productivity, improve performance, and simplify the management of ColdFusion applications.

The following provides a high-level overview of the most significant features and explains their benefits. Watch for expanded ongoing coverage, tips and tricks, and more in ***CFDJ*** throughout the rest of the year.

Here's the summary, straight from the horse's mouth.

User-Defined Functions

Developers can now use CFScript to define custom functions that are instantly available in any ColdFusion application. By creating UDFs for frequently used algorithms or procedures, developers can increase productivity and maintainability. UDFs offer the same or better performance than equivalent custom tags and enable the use of return values.

Query of Queries

One of the most clamored-for features is finally here – query of queries. Cross data source joins, cross data source unions, in-memory data denormalizations, and improvements in large-scale database queries are just some of the capabilities made available with the new query of queries feature. Using a subset of the Structured Query Language (SQL), ColdFusion applications can now query existing result sets in memory. Data in different RDBMSs can be combined and treated in the application as though they were from a single source. Accessing data directly in the CF process can also eliminate frequent connections to and from a database. In other words, cursorlike functionality can be achieved without maintaining a connection to the back end. The new CFSQL tag enables developers to efficiently integrate data from heterogeneous data sources and treat it as if it were from a single source.

Charting and Graphing

In one of the first visible results of the recent merger between Allaire Corp. and Macromedia, charting and graphing have been added. Thanks, Macromedia! Powered by Macromedia's Generator, the ColdFusion 5.0 charting engine can dynamically generate up to 13 different charts and graphs in either Flash, GIF, or JPEG format. Data can be drawn from nearly any source, including SQL queries, hard-coded data, and application variables.

New Verity VDK 2.6.1 and Verity K2 Server Integration

The full-text search capability provided by ColdFusion for documents and data sources now uses the Verity K2 Server. Indices can also be larger (up to 125,000 documents with ColdFusion Professional, 250,000 with ColdFusion Enterprise) and can now index the latest file formats, including Microsoft Office 2000. Multiple language capabilities are also now a standard part of the product.

Crystal Reports 8.0 Integration

Users can now create dynamic reports with a simple tag interface that passes data to the latest version of Crystal Reports. Used in conjunction with the ColdFusion charting engine, this feature provides a rich business reporting capability.

Incremental Page Delivery

The new CFFLUSH tag offers developers control over how a page is delivered to the browser. Thus, complex pages that take longer to build can be delivered in increments, giving users a more immediate response to their request. For example, a long report page could immediately send its header and a message stating, "Please wait while the report is built," until the full page becomes available.

Database Connectivity

ColdFusion 5.0 includes new Merant ODBC 3.7 Drivers on all platforms, including new wire protocol drivers that increase performance by interfacing directly with the native database protocols.

Application Deployment Services

Also new are site archiving features that enable easy backup and restoration of site configuration files and/or applications. This simplifies management by enabling the easy movement of applications between servers; for example, from testing to production or from one production server to another.

Enhanced Application Monitoring

ColdFusion 5.0 builds on the server's existing management capabilities by adding the ability to create an unlimited number of customizable alerts. These can be used in turn to send notifications via e-mail or pager, or to trigger automatic recovery procedures.

Server Log Analysis

New to CF 5.0, the Log File Analyzer provides a browser-based tool for filtering, searching, and querying log files in order to pinpoint application problems or locate performance bottlenecks.

SNMP Support

ColdFusion now supports integration with enterprise management systems such as IBM Tivoli, CA Unicenter, or BMC Patrol.

CFLOG Tag

This new tag has been added to the CFML language and allows developers to write a message to a custom user log or to a ColdFusion log at any point during the execution of a page. This can be of great value when debugging an application in development and for understanding patterns of usage in production applications.

Linux Improvements

ColdFusion applications can now be deployed on additional Linux distributions, including SuSE and Cobalt Qube, RaQ, and XTR servers. The server also shows significant performance improvements on the Linux platform.

Hardware Load Balancing Integration

The ColdFusion Server provides agent-based integration with hardware load–balancing solutions, such as Cisco's CSS 11000. By sending performance data to the load balancer, ColdFusion enables the switch to better distribute load among servers.

Enhanced COM Support

A number of improvements have been made with regard to implementing COM/COM+ in ColdFusion apps.

Enhanced CORBA Support

ColdFusion 5.0 changed the architecture involved with calling CORBA objects to dynamically load libraries using a connector. This new approach is more generic and doesn't tie the ColdFusion user to a specific Orb vendor. Macromedia provides implementations of the connectors for some of the popular Orbs and, for those that are not supported, Macromedia will make the source available under NDA to a select group of third-party candidates or Orb vendors.

A FIRST LOOK

ColdFusion Server 5.0

New Tags

CFGRAPH

Use the new tags, cfgraph and cfgraphdata, to generate dynamic charts and graphs in either Flash, GIF, or JPG format from various data sources, including database query results, hard-coded data, variables, and other data.

```
<CFQUERY NAME="AllMonths"
DATASOURCE="cffivedb">
SELECT Month, Amount, MonthNumber
FROM   GraphTest1
ORDER BY  MonthNumber
</CFQUERY>

<cfgraph type="bar"
  query="AllMonths"
  valueColumn="Amount"
  itemColumn="Month"
  title="Monthly Totals">
</cfgraph>
```

These attributes provide the basic information about the graph such as its type and the data it displays (see Table 1).

Query of Queries

Sample of cross data source joins:

```
<cfquery name="MailingList1"
datasource="SQLList" dbtype="OLEDB">
SELECT Name, Email
FROM MailingList
</cfquery>

<cfquery name="MailingList2"
datasource="OracleList" dbtype="Oracle80">
SELECT Name, Email
FROM MailingList
</cfquery>
```

> “The majority of all the new features and enhancements in CF 5 are user-requested, and that's an important thing to note when evaluating it. Everything in it has been asked for by the community, and Macromedia has certainly delivered.
>
> —Robert Diamond

```
<cfquery name="QofQUnion" dbtype="Query">
SELECT Name, Email
FROM MailingList1
UNION
SELECT Name, Email
FROM MailingList2
</cfquery>
```

User Defined Functions

```
<CFSCRIPT>
function myname(first, last)
{
VAR middle;
middle = "Homer";
return first & middle & last;
}
</CFSCRIPT>

<!--- "myname" can now be used like any
CFML Built in Function --->
<CFSET xxx = myname("John", "Doe")>
```

CFDUMP

Spectra's CFA_DUMP becomes part of the core CFML language. Use this tag to display the value of any variable in a formatted table.

```
<cfdump var="variable">
```

Attribute	Description
Type	Required. Type of chart to display. Must be one of the following: • Bar • HorizontalBar • Line • Pie
Query	Name of the query containing the data to graph. Required if you don't use cfgraphdata tags in the cfgraph tag body to specify the data values.
valueColumn	Query column containing the data values. Required if you don't use cfgraphdata tags in the cfgraph tag body to specify the data values.
itemColumn	Optional. Query column containing the item label for the corresponding data point. Item labels appear on the horizontal axis of Bar charts and the vertical axis of the HorizontalBar charts.

TABLE 1

CFFLUSH

The cfflush tag flushes the current ColdFusion output buffer to the Web server, which then sends it back to the client. The first instance of the cfflush tag on a page sends back all the HTTP headers as well. Subsequent flush tags send only the output buffered since the previous flush.

```
<cfflush interval = "bytes">
```

CFLOG

In ColdFusion 5.0, you can now generate application-specific, scheduled task, and custom logs. You can target files for storing logged information, and you can limit logs by size or entries by age. The ColdFusion Administrator now also supports filtering for log views. The ability to direct logging output on an application-specific basis should be especially valuable for ISPs who need to track application information on a customer-by-customer basis and provide log information to customers while protecting sensitive customer information. The user-defined log and user-defined message features allow you to integrate application specific management requirements into your ColdFusion Applications. For example, you can use the cflog tag to write a message to a custom user log or to a ColdFusion log. The user message repository allows you to standardize logging messages.

The log management supports limiting log entries by date or absolute size. When you specify file storage for logs, you can specify how many generations of log files are maintained.

cflog

```
<cflog text = "text"
     log = "log type"
     file = "filename"
     type = "message type"
     thread = "thread ID yes or no"
     date = "date yes or no"
     time = "time yes or no"
     application = "application name yes
or no">
```

text

Required. The text of the message to be displayed. Enter any text.

log

If you omit, the file attribute specifies the standard log file in which to write the message. Ignored if you specify a file attribute. Valid values are:

- Application – Default. Writes to the Application.log file, normally used for application-specific messages.
- Scheduler – Writes to the Scheduler.log file, normally used to log the execution of scheduled tasks.

file

The name of the file in which to log the message. All log files must have the suffix .log. You must specify only the main part of the file name, without the .log suffix. For example, to log to the Testing.log file, specify "Testing": The file must be located in the default log directory. You cannot specify a directory path. The file is created automatically if it does not exist. Default is Application.

type

The type (severity) of the message. Valid entries:

- Information – Default. All informational messages.
- Warning
- Error
- Fatal Information

thread

Specifies whether to log the thread ID. The thread ID identifies which internal service thread logged a particular message. Since a service thread normally services a CFML page request to completion, then moves on to the next queued request, the thread ID serves as a rough indication of which request logged a message. Leaving thread IDs turned on can help diagnose patterns of server activity. Valid values are Yes and No. Default is Yes.

date

Specifies whether to log the system date. Enter Yes or No. Default is Yes.

time

Specifies whether to log the system time. Enter Yes or No. Default is Yes.

application

Specifies whether to log the application name if one has been specified in a cfapplication tag. Enter Yes or No. Default is Yes.

Performance

Performance testing and final code improvements are not yet complete. Preliminary testing, however, demonstrates considerable gains across all platforms. On the Tack2Plus application, an e-commerce application that was also the basis for the ColdFusion 4.5.1 Performance Briefs and Whitepaper, we observed the following improvements over version 4.5.1 SP2:

- Up to 3.5X improvement on NT/Win2K (+226% with the untuned I/O-bound Tack2Plus app)
- Up to 4.5X improvement on Linux (+350% with the untuned I/O-bound Tack2Plus app)
- Up to 33% improvement on Solaris and HP-UX with the untuned I/O-bound Tack2Plus app

ColdFusion 5.0 Servers
ColdFusion Server Professional Edition – The full-featured server for departmental applications or moderate-size Web sites
ColdFusion Server Enterprise Edition – The enterprise-class server for maximum application performance, reliability, management, and security
ColdFusion Server Hosting Edition – The ColdFusion server for those providing commercial hosting services

TABLE 2

Server Memory Management Enhancements

ColdFusion 5.0 for Windows and Linux now features automatic footprint reduction to minimize the amount of memory used by the ColdFusion 5.0 Server on these platforms. ColdFusion 5.0 dynamically releases unused memory back to the operating system to minimize its process footprint. Customers with Web applications that occasionally use very large blocks of memory will see this memory returned to the operating system when the operation is complete.

Basic Security Enhancement

ColdFusion now allows you to disable certain attributes and tags. You can disable the following tags and attributes:

- cfcontent
- cfdirectory
- cffile
- cfobject
- cfregistry
- cfadminsecurity
- cfexecute
- cfftp
- cflog
- cfmail
- dbtype=dynamic attribute
- connectstring attribute

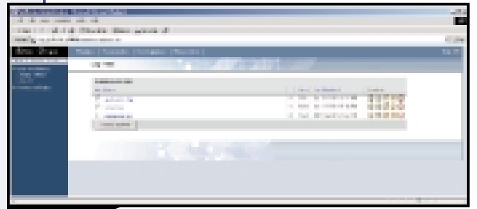


FIGURE 1: The updated Log Manager provides many new options not seen in 4.5

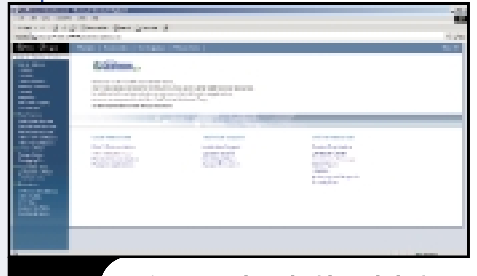


FIGURE 2: New features aplenty in this early look at the new administrator

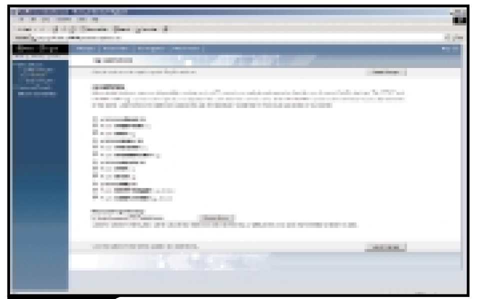


FIGURE 3: Added security options to protect your site from pesky hazards and hackers

FIGURE 4: Take off with ColdFusion 5.0!

Conclusion

All in all it seems like a great new upgrade, and here at CFDJ we're certainly looking forward to it! The majority of all the new features and enhancements in CF 5 are user-requested, and that's an important thing to note when evaluating it. Everything in it has been asked for by the community, and Macromedia has certainly delivered.

robert@sys-con.com

XMLEDGE2001 INTERNATIONAL CONFERENCE & EXPO

MEETING BUSINESS CHALLENGES WITH XML

XMLEDGE conference&expo 2001

The ONLY XML Event
Backed by the Power of
SYS-CON MEDIA and XML JOURNAL

CONFERENCE
October 22–25, 2001

EXPO
October 24–25, 2001

SANTA CLARA CONVENTION CENTER
Santa Clara, CA

Plan to Attend...This 4-Day Cutting-Edge Conference

Save the Dates!

XML...Dramatically Changing the Substance of Corporate Systems

- XML will fundamentally improve the speed, cost and flexibility of business applications.
- IT Managers expect XML Technology to enable performance of new operations and shorten application development time by 30% to 50%
- E-commerce transactions using XML are predicted to rise from .5% in 2000 to more than 40% by the end of 2003.
- XML products now under development will hit the market in the second half of 2001. *See them first at XMLEdge!*

www.XMLEdge2001.com

Conference Tracks

Track 1: PRESENTING XML

The latest tips, tricks and strategies for presenting XML in various front-end presentation environments

XSL / XSLT / CSS / XLINKS / XPOINTERS
XHTML
BROWSER SUPPORT FOR XML APPLICATIONS
WML AND WAP
VOICEXML
SCRIPTING AND XML (PERL / PHP / PYTHON / JAVASCRIPT)
WEB SERVICES
WIRELESS XML: WAP / WML, XML AND MOBILE COMMUNICATIONS

Track 2: REAL-WORLD XML

Actual Case Studies using XML and related technologies in various industry sectors

XML TOOLS / UTILITIES APPLIED TO ACTUAL APPLICATIONS
DOCUMENT AND CONTENT MANAGEMENT (EDITORS / PARSERS / SERVERS / TRANSLATORS)
REAL-WORLD APPLICATIONS
XML ISSUES IN THE REAL WORLD (PERFORMANCE / SECURITY / COMPRESSION / SERIALIZATION)
USE OF XML IN INDUSTRY SECTORS (MANUFACTURING / HIGH TECH, ETC.)

Track 3: XML & JAVA

The latest developments in XML and Java environments as well as insights into developing solutions using XML, Java and related technologies

DOM/SAX / JDOM
JAVA AND XML PROGRAMMING (JSP, JMS, ETC.)
PARSING XML IN JAVA
XML AND DATABASE PROGRAMMING (XQL)

Track 4: EAI with XML

Comprehensive coverage on developing Enterprise Application Integration solutions with XML

XML AND APPLICATION SERVERS
B2B SERVERS
PROCESS AND WORKFLOW INTEGRATION
XML-BASED INTEGRATION USING MESSAGING SYSTEMS
ERP INTEGRATION
DATABASE INTEGRATION

Track 5: XML STANDARDS

The latest news and updates on emerging XML standards from the industry gurus

W3C STANDARDS
XML DESIGN PATTERNS
DOM/SAX / JDOM
ROSSETTANET
SOAP / BIZTALK
DTDS AND XML SCHEMAS
XML REPOSITORIES (BIZTALK / OASIS)

SYS-CON MEDIA · JAVA DEVELOPER'S JOURNAL · XML JOURNAL · wireless · LINUX BUSINESS WEEK · COLDFUSION Developer's Journal · WebSphere DEVELOPER'S JOURNAL

The Largest International XML Conference of the Year! Reserve Your Exhibit Space Now!

Get the Competitive Edge...

- Over 100 exhibitors will have access to the most qualified audience of potential buyers and specifiers.
- They will demonstrate, meet the movers and shakers, solve their challenges and sell!

Plan to Sponsor – Many Sponsorships Are Exclusive!

- Demonstrate your commitment and capabilities to the largest XML audience to assemble on the West Coast in 2001!
- Rise above the noise. Deliver your message with the marketing support of SYS-CON Events and SYS-CON Media.

Who Should Attend XMLEdge2001

i-Technology Professionals	Technical Architects	Analysts, Consultants
Developers, Programmers, Engineers	Senior IT / IS Management	Product & Program Managers

Call 201 802-3069 to reserve your space today!

The only XML event backed by the full marketing power of **SYS-CON Media** and ***XML-Journal*** is your best opportunity to reach the largest, most qualified audience of software developers and *i*-technology professionals in 2001.

Owned and Produced by
SYS-CON EVENTS
135 Chestnut Ridge Road, Montvale, NJ 07645
201 802-3069 • Fax: 201 782-9601
Visit
www.XMLEDGE2001.com
for more information

Last Year's Exhibitors Include...

Baltimore Technologies	LearningPatterns.com	Search XML Resources.com
DNU Magazine	Mailers Software	Sequoia Software Corporation
EarthWeb	Merant	Skill Server
Eliad Technologies, Inc.	Microsoft	SoftQuad Software, LTD.
E-Numerate Solutions, Inc.	Mindgap Systems Ltd.	Sun Microsystems
E-Security Journal	*MSDN Magazine*	Talva Corporation
Fawcette Technical Publications	Nanobiz	The Gilbane Report
GeekCruises.com	Neocore, LLC	Tibco Software, Inc.
Hit Software, Inc.	Netfish Technologies, Inc.	Whitehill Technologies, Inc.
IAM Consulting	Oasis	X-Aware
IBM	Object Management Group	XML Global Technologies, Inc.
Infoshark, Inc.	O'Reilly Network	*XML-Journal*
Infoteria Corporation	Planet 7 Technologies Corp.	*XML Magazine*
Ingeniux Corporation	Progress Software Corporation	*XML Times*
Insight Software, Inc.	Pybiz	XML.com
Interknack Corporation	Radiant Logic, Inc.	XML.org
IXIASOFT, Inc.	Rogue Wave Software, Inc.	XML/EDI Group
JavaPro	*SD Times*	XMLResources.com
	Seagull	XYZ Find

XML-JOURNAL READERS' CHOICE AWARD

XML–JOURNAL
READERS' CHOICE AWARDS – THE "OSCARS OF THE SOFTWARE INDUSTRY" – RECOGNIZE 13 WINNERS AND 39 FINALISTS

AWARDS WILL BE PRESENTED AT XMLEDGE, THE LARGEST INTERNATIONAL GATHERING OF XML PROFESSIONALS ON THE WEST COAST OF THE U.S. IN 2001

CFDJ FEATURE

BY MATT TATAM

Spectra's XML Flashes

Using Flash to enhance the presentation of Spectra's services

"Know where you are heading and prepare yourself thoroughly and accurately by refining what is available."
—Bruce Lee, Philosopher

Macromedia and Allaire have merged. This is an ideal marriage. Allaire's server-side technology and Macromedia's client-side strengths are complementary, and the synergies between these two organizations will produce a strong alliance. In this scenario the whole is greater than its sum. I've familiarized myself with both technologies until the learning curve has started to slow. If you're a Spectra developer, this article may enhance your knowledge of the Model View Control Pattern and the new way Spectra stores the Site Layout Model (SLM).

If you're a Flash developer this may be a chance to be exposed to the eXtensible Markup Language (XML) object and how some of its methods can be used to deserialize the XML data into Flash's generic object.

Spectra was released in late 1999. Since then it has been recognized as a leader in commerce, content, and customer management (personalization). The other features of Spectra, which have not been widely recognized, but are of equal or greater importance, are:

- Workflow and process automation
- Business intelligence
- Role-based security
- Syndication
- XML data storage

Spectra stores data in XML format and manages this data in an object (Type) oriented system. One of these types (objects) is the Page object. This page object is associated with Spectra's SLM and allows, among other things:

- Content management
- Roles-based security
- Reporting
- Syndication
- Personalization
- Caching
- Business rules

In this article I'll explain how to use Spectra to retrieve the SLM from the Content Object Database (CODB), and extract the relevant properties of the page and section objects to deliver XML to a Flash interface. The article is written from the perspective of a developer rather than a creative designer.

Flash was first introduced as a multimedia component on the Internet in 1995. It allows a client-side dynamic multimedia medium that can be used for almost anything. Flash is a multimedia source that's encoded into HTML (similar to an audio or video file). It's a rich vector-based media program that makes animations and layered content that's ideal for Graphical User Interfaces (GUI). As stated on Macromedia's Web site (www.macromedia.com), "Macromedia Flash is the key to designing and delivering low-bandwidth animations, presentations, and Web sites."

Flash 5 allows for an effective client-side interface. This interface has both visual and functional strengths that have, until recently, been generally limited to operating system dependence. Successful GUIs have started to appear using Flash 5 due to its new capabilities (drag/drop, Actionscript, etc.). Because it can dynamically communicate with server-side applications, Flash enables an effective interaction between the users and remote information.

This article also aims to explore Flash's XML object by providing an example of one of its many uses in communicating to a server-side application.

Model View Control (MVC)

MVC is a programming pattern in which the programming processes are separated into three distinct parts:

1. ***The model:*** Maintains the data that describes the state of the program
2. ***The control:*** Provides the conduit by which data enters and is interpreted and directed by the program
3. ***The view:*** Provides one or more views of the data

When we choose to locate a book on a particular subject in a library, the easiest and most direct way is to ask a librarian. The librarian recognizes the request because it's in a language he or she understands. In order to realize the request the librarian needs to determine the location of a book, and consult the catalog system. The librarian would then request the information in a form which he or she understands, and upon receiving it, would write down the reference number on a piece of paper (in the requesting format and language), and pass it back to the person requesting the subject location.

This scenario has a number of processes:

1. The request for the information.
2. Recognizing where and how to locate the information.
3. Consulting, retrieving, and returning the data. Finally, displaying the information in a format consistent with the request. Using the library analogy we can generally associate this scenario with the MVC.

   The librarian is the controller, recognizing the request and knowing where, how, and in what format, to direct it. The controller interprets the request, then decides which action to take.

   The catalog the librarian consults is the data model, as it stores the data (it could be a microfiche, database, or catalog cards). The model consists of the data-specific logic that enables the modification, processing, forwarding, and/or retrieving of the required data.

   Writing down the information on a piece of paper allows the request to be viewed. It's the interface that allows the requester to view the information requested.

Site Layout Model Example

A MVC system can be effectively implemented using SLM and Flash. The SLM consists of a hierarchy of site elements. The top-level element is the site. A Spectra site element consists of pages, sections, and container elements. Pages and sections can exist within other sections. In Spectra 1 and 1.01 the SLM was stored in the sitecomposition table in the CODB. In the 1.5 version the storage is now within the objects table. However, in both versions a page was, and is, both an object in the CODB as well as a physical file (with a CFA_PAGE tag that identifies it). Both are linked through a property in the page's object instance (target). The object dump shown in Table 1 consists of all the properties of the Spectra site element page that we will access.

The data dump in Table 1 is an array element retrieved from the CODB using the <CFA_SITEMODELTRAVERSE> Spectra tag. This tag uses a ROOTELEMENTID parameter (to obtain the site descendants) that we'll provide from the Flash client. The array that's returned will be used to create XML data that Flash will load via its client-side XML object. Figure 1 is a MVC diagram showing the objective of this article.

In order to achieve the goal of this article, Flash will be used to request the SLM. The controller will then interpret the request and call the model. The model will then retrieve the data, modify it, and return it to the controller. The controller in turn passes the formatted data to the view interface. The view interface displays the XML for the Flash interface to retrieve. Flash loads the data into its own generic object, which can then be used to visually display the SLM or manage/edit the SLM.

COMPOSITIONID	E7068067-D78A-4F01-8335AAA5C57BB035
COMPOSITIONLABEL	Bones SkeletalSystem Body flashsiteObject
COMPOSITIONTYPE	PAGE
DEPTH	3
DESCRIPTION	
ELEMENTPOSITIONINDEX	1
PARENTCOMPOSITIONID	C01EBAF6-70DD-4732-9E5DEE9911F01053
STARTURL	/allaire/spectra/webtop/sitedesign/sitelayoutmanager/sitetree/visualizesite.cfm?thisNav=Site%20Design
TARGET	/SkeletalSystem/Bones.cfm

TABLE 1 A CFA_Dump of one array element returned by the CFA_SITEMODELTRAVERSE tag

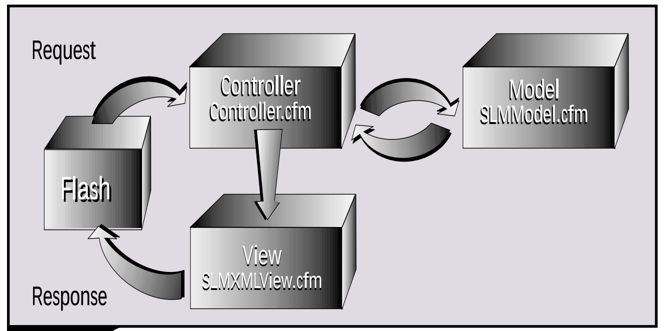


FIGURE 1: Component interaction in the MVC pattern

The Client Request

Flash 5 supports HTTP-based XML data transfer as well as XML-socket real-time continuous connections. Only a small number of methods from the XML object will be discussed in this article. The XML object gives Flash the ability to load, parse, send, build, and manipulate XML document trees.

To begin we must create an instance of an XML object that will send data to the controller page. This XML object may have security information that could be used by the server-side application to determine what information, if any, is returned:

```
siteModelXML = new XML();
```

Another XML object needs to be instantiated to receive the data when it's returned.

```
SiteElementXMLReply = new XML();
SiteElementXMLReply.onLoad =
onSiteElementXMLReply;
```

Once these two XML objects have been created we'll use the sendAndLoad method on the calling XML object. ***Note:*** the sendAndLoad method will immediately return, and processing will continue even if the server has not yet responded

For simplicity the URL will have the instructions for the controller page. The current site variable is a Universal Unique Identifier (UUID).

The "Programming with Allaire Spectra Manual" (1999) says, "A UUID is formatted as 'XXXXXXXX-XXXX-XXXX-XXXXXXXXXXXXXXXX' where *X* stands for a hexadecimal digit (0–9 or A–F). UUIDs guarantee uniqueness…"

```
currentsite = "79267565-2234-4DAF-
9F9F08AB04E758A6";

siteModelXML.sendAndLoad("http://127.0.0.1/f
lash/controller.cfm?Model=SLM&View=XML&Model
Key="+currentsite+"",SiteElementXMLReply);
```

This data will load asynchronously. When the data does arrive, the function on the receiving XML object onSiteElementXMLReply() will be executed.

Controller

The controller (controller.cfm file) is a simple bit of code, which determines the URL parameters, then calls the required model and view code.

It calls the model custom tag with the UUID parameter of "siteuuid", the rootelement that the spectra tag cfa_sitemodeltraverse requires to retrieve the descendant site model elements.

```
  <cf_SLMModel siteuuid="#url.modelkey#"
r_aSiteElements="aElements">
```

Once the controller receives the returned array from the model, it calls the view customtag with the array parameter of "elements" that's parsed into XML format.

```
  <cf_SLMXMLView Elements="#aElements#">
```

Model

The model (SLMModel.cfm file) contains the logic of accessing the data and returning it in a format that's used by the view interface (see Listing 2).

This file calls the Spectra custom tag cfa_sitemodeltraverse and processes its result into an array that's then returned back to the calling page (controller.cfm).

This data is very similar to the data returned from the cfa_sitemodeltraverse tag, but it can have the ability to perform future business logic that is not required at this point (i.e., returning security info, caching info, etc.).

View

In this scenario the view interface (SLMXMLView.cfm file) is also a simple piece of code that receives the array produced by the model and then loops over it. It creates the XML from the structure held within the array element, and outputs the XML in the format shown in Listing 1.

The Server's Reply

In the onSiteElementXMLReply function I've created a global array that holds the objects of the site elements that I'll parse from the XML (see Listing 3).

```
SLMElementArray = new Array();
```

Also created is a new XML object that will copy the XML element loaded by the sendAndLoad method.

```
 var sitecopy = new XML();
sitecopy = this;
```

- ***Node:*** XML simply marks up the data with tags. The data that are marked up are referred to as nodes. So XML transforms a set of data to a set of nodes. In this example <SLMELEMENT is the node. It has an opening and closing tag.
- ***Attribute:*** In this scenario an attribute is a name-value pair associated with the node (i.e., PARENTCOMPOSITIONID="" etc.).

Within the XML object there is a firstChild property that we are able to access. This property evaluates the specified XML object and references the first child in the parent node's children list.

```
 var sitecopychildren =
sitecopy.firstChild;
```

Next we call the childNodes method which returns an array containing references to the child nodes of the specified node:

```
var asitecopychildren =
sitecopy.childNodes;
```

Then we loop through this array until we find the SLMELEMENT node:

```
var asitecopychildrenLength = asitecopy-
children.length;

for (var countr = 0;
countr<=asitecopychildrenLength; countr++)
{
if (sitecopychildren.nodeName == "SLMELEMENT")
{
```

Once it's found, we call a function that adds the node and its attributes to the SLMElementArray array:

```
   addElementToArray(sitecopychildren);
```

In the function addElementToArray, a new object is created and the attributes of the node passed into the function are used to populate the object's properties.

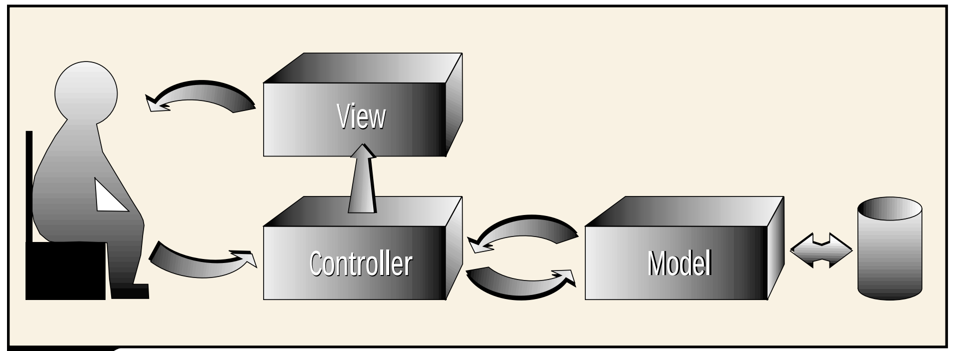


FIGURE 2: MVC pattern summary

```
var elementObject = new Object();

 elementObject.PARENTCOMPOSITIONID = ele-
ment.attributes.PARENTCOMPOSITIONID;
 elementObject.DEPTH =
element.attributes.DEPTH;
 elementObject.ELEMENTPOSITIONINDEX = ele-
ment.attributes.ELEMENTPOSITIONINDEX;
 elementObject.COMPOSITIONTYPE =
element.attributes.COMPOSITIONTYPE;
 elementObject.ELEMENTNAME =
element.attributes.ELEMENTNAME;
 elementObject.COMPOSITIONID =
element.attributes.COMPOSITIONID;
 elementObject.TARGET = element.attribut-
es.TARGET;
```

Once the object is populated, the push method of the array object is used to add the object element to the end of an array:

```
 SLMElementArray.push(elementObject);
 trace(SLMElementArray.length+"="+ ele-
ment.attributes.ELEMENTNAME);
```

Note: The use of the trace function is for debugging purposes and can be commented out.

Once it's added to the array we then proceed to the next node within this loop:

```
        }
  sitecopychildren.removeNode();
  sitecopychildren = sitecopy.firstChild;
 }
```

This process receives and parses an XML object into a Flash array where processing and displaying of the information can be performed.

Once deserialized, the processing and displaying of the XML by Flash is limited only by your imagination.

The aim of this article has been to expose you to the MVC pattern, Spectra's new way of storing the SLM and some of Flash's XML object features.

The MVC pattern is shown in Figure 2.

A request is issued to a controller that interprets it and controls the flow of information determined by predefined business logic. The controller then calls the data model which modifies, processes, forwards, and/or returns the data. The view interface then displays the data.

In the example given, a ColdFusion page is the controller, the model uses Spectra, and the view interface is in Flash. The received data could be any XML formatted data from the CODB or any other data model.

There's a growing need for content delivery and management systems such as Spectra. Once you have fully experienced the raw power of Spectra, your perception of approaching Web-based application development will be altered forever. The experience you gain from using and understanding the Spectra perspective may enhance and even clarify the knowledge you've already attained in this ever-changing environment.

As Flash continues to "push the envelope" on how multimedia can be used on the Web, Spectra will in turn allow the exploration of intelligent options for transmitting, receiving, processing, storing, and displaying media. As of now its rich vector-based media provides unlimited programming options in the process of delivering Web-based multimedia.

Using Spectra/ColdFusion and Flash together in a client/server marriage allows developers to attain a better understanding of both technologies. Together they'll be an extremely powerful and potentially incontestable tool in the field of Web development. The sky's the limit once the two technologies begin to merge and evolve into something hopefully even more powerful.

About the Author
Matt Tatam has been developing with ColdFusion since 2.0 and is an Allaire-certified Spectra and JRun instructor. He is currently working as an instructor with Highlander UK (www.highlander.co.uk) and as an R&D consultant in a number of different Internet technologies.

matt@tatam.com.au

Listing 1 (XML)

```
<SLMELEMENT
  PARENTCOMPOSITIONID=""
  ELEMENTPOSITIONINDEX=""
  DEPTH=""
  COMPOSITIONTYPE=""
  COMPOSITIONID=""
  ELEMENTNAME=""
  TARGET="">
</SLMELEMENT>
```

Listing 2 (SPECTRA)

```
Description:
 This custom tag outputs the Site layout XML data
Parameters:
 Elements = an array of structs with
--->
<cfif isdefined("attributes.Elements") and isArray
 (attributes.Elements)>
  <cfloop from="1" to="#Arraylen(attributes.Elements)#"
     index="countrSLMXMLElement">
   <cfif structkeyexists(attributes.Elements
     [countrSLMXMLElement],
"COMPOSITIONID")>
  <!--- Outputs the XML to the page --->
    <cfoutput>
<SLMELEMENT
PARENTCOMPOSITIONID="#attributes.Elements
 [countrSLMXMLElement].PARENTCOMPOSITIONID#"
ELEMENTPOSITIONINDEX="#attributes.Elements
 [countrSLMXMLElement].ELEMENTPOSITIONINDEX#"
DEPTH="#attributes.Elements[countrSLMXMLElement].DEPTH#"
COMPOSITIONTYPE ="#attributes.Elements[countrSLMXMLElement]
 .COMPOSITIONTYPE#"
COMPOSITIONID="#attributes.Elements[countrSLMXMLElement]
 .COMPOSITIONID#"
ELEMENTNAME="#attributes.Elements[countrSLMXMLElement]
 .ELEMENTNAME#"
TARGET="#attributes.Elements[countrSLMXMLElement].TARGET#" >
</SLMELEMENT>
#chr(10)#
 </cfoutput>
     </cfif>
  </cfloop>
</cfif>

Description:
This is a controller page which directs requests to
relevent Model and View pages
Parameters:
 Model = Business logic and/or Data Access Model to use
 ModelKey = Data access key used by the model
 View = Display logic for Data From Model
--->
<cfif isdefined("url.model") AND isdefined("url.View")
 and isdefined("url.modelkey")>
 <cfif url.model eq "invoke" AND len("#url.view#") GT 1>
<!---  Invokes an Object with method = url.view --->
       <cfa_contentobject datasource="#request.cfa
        .objectstore.dsn#" method="#url.view#"
        objectid="#url.modelkey#">
 <cfelseif url.model eq "SLM"  AND url.View eq "XML">
<!--- This if section displays the SLM XML Data --->
<!--- Calls the Model --->
  <cf_SLMModel siteuuid="#url.modelkey#"
   r_aSiteElements="aElements">
<!--- Calls the View --->
  <cf_SLMXMLView Elements="#aElements#">
 </cfif>
</cfif>

Description:
 This custom tag retrieves the SLM and returns an array

Parameters:
 siteUUID = UUID for site that is passed to the
 cfa_sitemodeltraverse   Tag
   i.e.  <cfa_sitemodeltraverse rootElementID =
  siteUUID>
Return variable
 r_aSiteElements - return parameter name
--->
<CFTRY>
 <cfsetting enablecfoutputonly="Yes" SHOWDEBUGOUTPUT="No" >
<!--- attributes.siteUUID is a required parameter to the
cfa_sitemodeltraverse (rootElementID parameter) --->
 <cfparam name="attributes.siteUUID" type="UUID">
<!--- This assumes all pages in SLM are for public viewing
and that we do not want Containers --->
 <cfa_sitemodeltraverse
     rootElementID = "#attributes.siteUUID#"
     datasource = "#request.cfa.objectstore.dsn#"
     bIncludeSecurity = "no"
     bIncludeContainers = "no"
     bIncludePages = "yes"
     r_aTraverse = "aSortedSLMPageDetails">
 <cfset aElementXML  = arraynew(1)>
 <cfset setArrayTempVar =
ArraySet(aElementXML,1,Arraylen(aSortedSLMPageDetails),
 structnew())>
 <cfloop from="1" to="#Arraylen(aSortedSLMPageDetails)#"
 index="countrSiteElement">
<!--- Loops thru the array of structs returned from the
cfa_sitemodeltraverse tag to gather Section and Page info--->
  <cfif aSortedSLMPageDetails[countrSiteElement]
   .COMPOSITIONTYPE neq "SITE">
<!--- As the COMPOSITIONLABEL is composed of page name as
well as Section and site labels this just gets the name of
the element --->
    <cfset siteElementName= listgetat
     ("#aSortedSLMPageDetails[countrSiteElement].
 COMPOSITIONLABEL#",1,"_____")>
    <cfset stElement = structnew()>
<!--- Creates  struct that will be used to produce the
 XML output --->
<!--- This loop can be used to add additional properties
that may be needed for the client Flash --->

    <cfset stElement.COMPOSITIONID ="#asortedslmpagedetails
      [countrsiteelement].COMPOSITIONID#">
    <cfset stElement.COMPOSITIONTYPE =
"#aSortedSLMPageDetails[countrSiteElement]
  .COMPOSITIONTYPE#">
    <cfset stElement.ELEMENTNAME ="#siteElementName#" >
    <cfset stElement.DEPTH  ="#asortedslmpagedetails
     [countrsiteelement].DEPTH#" >
    <cfset stElement.TARGET = "#asortedslmpagedetails
     [countrsiteelement].TARGET#">
    <cfset stElement.ELEMENTPOSITIONINDEX ="
      #asortedslmpagedetails[countrsiteelement]
      .ELEMENTPOSITIONINDEX#" >
    <cfset stElement.PARENTCOMPOSITIONID ="#asortedslmpagedetails
     [countrsiteelement].PARENTCOMPOSITIONID#" >
<!--- Populates an array of structs used to produce
      the XML output                      --->
    <cfset aElementXML[countrSiteElement] = stElement>

   </cfif>
  </cfloop>
<!--- returns the array   --->
  <cfset "caller.#attributes.r_aSiteElements#" =
   aElementXML>
<CFCATCH>
 <cfoutput>
<!--- Outputs the ERROR XML to the page --->
  <ERROR MESSAGE="#cfcatch.message#" ></ERROR>
 </cfoutput>
</CFCATCH>
</CFTRY>
```

Listing 3 (FLASH)

```
function getsitemodel()
{
 siteModelXML = new XML();
 SiteElementXMLReply = new XML();
 SiteElementXMLReply.onLoad = onSiteElementXMLReply;
 currentsite = "79267565-2234-4DAF-9F9F08AB04E758A6";
 siteModelXML.sendAndLoad("http://127.0.0.1/flash/
 controller.cfm?Model=SLM&View=XML&ModelKey="
 +currentsite+"",SiteElementXMLReply);
}
function onSiteElementXMLReply() {
 SLMElementArray = new Array();
 var sitecopy = new XML();
 sitecopy = this;
 var sitecopychildren = sitecopy.firstChild;
 var asitecopychildren = sitecopy.childNodes;
 var asitecopychildrenLength = asitecopychildren.length;

 for (var countr = 0; countr<=asitecopychildrenLength;
 countr++)
{
  if (sitecopychildren.nodeName == "SLMELEMENT")
{
   addElementToArray(sitecopychildren);
  }
  sitecopychildren.removeNode();
  sitecopychildren = sitecopy.firstChild;
 }

}

function addElementToArray(element) {
 var elementObject = new Object();

 elementObject.PARENTCOMPOSITIONID =
 element.attributes.PARENTCOMPOSITIONID;
 elementObject.DEPTH = element.attributes.DEPTH;
 elementObject.ELEMENTPOSITIONINDEX =
 element.attributes.ELEMENTPOSITIONINDEX;
 elementObject.COMPOSITIONTYPE = element.attributes
 .COMPOSITIONTYPE;
 elementObject.ELEMENTNAME = element.attributes
 .ELEMENTNAME;
 elementObject.COMPOSITIONID = element.attributes
 .COMPOSITIONID;
 elementObject.TARGET = element.attributes.TARGET;

 SLMElementArray.push(elementObject);
 trace(SLMElementArray.length+"="+ element.attributes
 .ELEMENTNAME);
}
```

CODE
LISTING

The code listing for
this article can also be located at
www.ColdFusionJournal.com

Pssst! Wanna Hot Tip?

Thoughts on buying low and selling high

BY HAL HELMS

There ain't no secret to gettin' rich in the stock market, just buy a good stock; when it goes up sell it. If it don't go up... don't buy it. — Will Rogers

Let's face it: stocks haven't exactly been the best place to invest this year. As I write this, the Dow has just suffered its worst-ever week, the once-mighty NASDAQ is humbled, and tech investors are running from the reach of the bear, surly and hungry after long years of confinement.

We programmers were considered by many to be immune from the normal business cycle, but we've seen our salaries slip and our prospects dim somewhat. Remember the good old days when headhunters were a daily nuisance? Before you discovered that stock options make ideal phone message logs?

Well, I've got great news for you! I have the lowdown on an investment vehicle that can't miss. I'm not normally much of a fan of hot tips, but this one is too good to hold onto and so I present you, faithful reader, with the scoop on the best investment you're likely to see for a long time. Now this is not an investment you want to day trade, and it may not provide a quick killing, but it has explosive upside potential and goes by the symbol YOU.

Viewing Yourself as an Investment

Wait! I'm not joking! I think there's much to be learned by treating yourself as an investment. Peter Lynch, the legendary investor who so successfully piloted Fidelity Magellan for years, advises investors to invest in what they know. You certainly know about YOU.

You also have one of the most venerable techniques of investing on your side: buy low and sell high. Most of us don't have a product to sell. We have a service – we have, in fact, our time. When the market for YOU wanes, it means your time is less commercially valuable. Usually, we don't like that, but it can work for YOU. In the same way that companies can buy back their own stock when it's low, you can buy back your own time: it's less expensive than it was 18 months ago.

You can make this work is by investing in YOU, buying back some of those hours and using them to retool the product and make it better. The product, in this case, is you – your knowledge, your skills, your experience – and making it better means upgrading those things, knowing that when the market roars back to life, YOU will be more valuable than IBM or MSFT. It's a simple plan: buy low and sell high.

But will it roar back? Or was the huge demand for programmers a mere side effect of dot-com–mania? After all, history has a series of unhappy tales of bubbles that expanded until the breaking point and dissipated overnight. For instance, in the seventeenth century, Dutch tulip bulbs were the rage – so much so that the market for these bulbs went up 6000% to the point where a single bulb cost as much as a large house. Then, overnight, the bubble burst, bankrupting many. Are we simply seeing the end of a bubble for programmers?

I don't think so. Bubbles are built on an artificial demand – they owe their existence to the Bigger Fool theory. There's no such thing as a price that's "too high," according to this theory, for there's always some bigger fool to come along and pay you more.

However, the need for programmers is based on the fundamental consequence of an information-based economy. Simply stated: without software, we have no information, and without programmers, we have no software. That equation isn't about to change anytime soon. Which makes your investment in YOU the best one you can make.

But what about ColdFusion specifically? Is ColdFusion still viable or has it run its course? Will all development soon be done in Java? I've certainly heard this suggested by some very smart people, but as I look at the fundamentals behind ColdFusion's enormous growth, it seems to me that CF, too, remains sound. I don't see these two languages in competition at all.

Now that the hype behind Java has thankfully died down, we can better assess its true value, which seems to me to be enormous. By providing a lingua franca for computers, Java has assured itself a long reign as the successor to C++, a language on which to build large-scale enterprise-wide core applications.

But Java development is slow and expensive, fitting for the large-scale problems it can tackle, but ill-suited to the much greater number of Web Services that are quietly growing in demand. For these ColdFusion provides a much better *value proposition*, a term much loved by those pointy-haired bosses who are the ones you'll need to green-light your project.

Much of the perceived competition between Java and ColdFusion will vanish with the upcoming version 6 of ColdFusion, in which Java becomes integral to ColdFusion. This should make it an even more natural choice for integrating with enterprise Java systems.

So, what about it? Are you ready to invest in YOU? What would it mean to do so? First, it means a commitment to setting time aside to improve your skills. Even something as modest as 30 minutes a day working on new things, trying out features, methods, and techniques can make a huge difference in the value of your product.

You might want to take an approach known as learning by teaching. Surprisingly, this can be one of the best ways to learn something, as it forces you to understand the topic deeply. Share your newfound knowledge with others and you're giving back to the ColdFusion community and helping to ensure the sustained viability of CFML. You can write an article (I happen to know a magazine that just might print it), create a viewlet with the excellent (and free) viewlet maker available at www.qarbon.com, or write a tip for inclusion by online zines. If you decide to adopt this approach, make sure you tackle something you don't yet know.

I recently gave a Programming Foundations class for beginning programmers (www.halhelms.com). My classes are heavily based on learning by doing rather than by watching PowerPoint slides. As the class was ending, one of the students had the suggestion, "Wouldn't it be great to have a series of workbooks – just guided exercises to different aspects of programming to continue the learning?"

Wow, that would be tremendous. If you're wondering how to get started, here's a great thought for you: Why not create a workbook on a single topic that uses examples to help people come to their own discoveries? These could be easy to make and use.

What about your local CFUG? Do you regularly attend? I know very well the enormous demands on all our schedules, but this is one of those very rare communities that exist for the mutual support, learning, and encouragement of its members – us. The people who run these CFUGs are true heroes. I've observed close-up the incredible amount of work involved, all of it unpaid and most of it unnoticed by others. And who knows, in a little bit of time, you with your newfound knowledge may be the speaker at one.

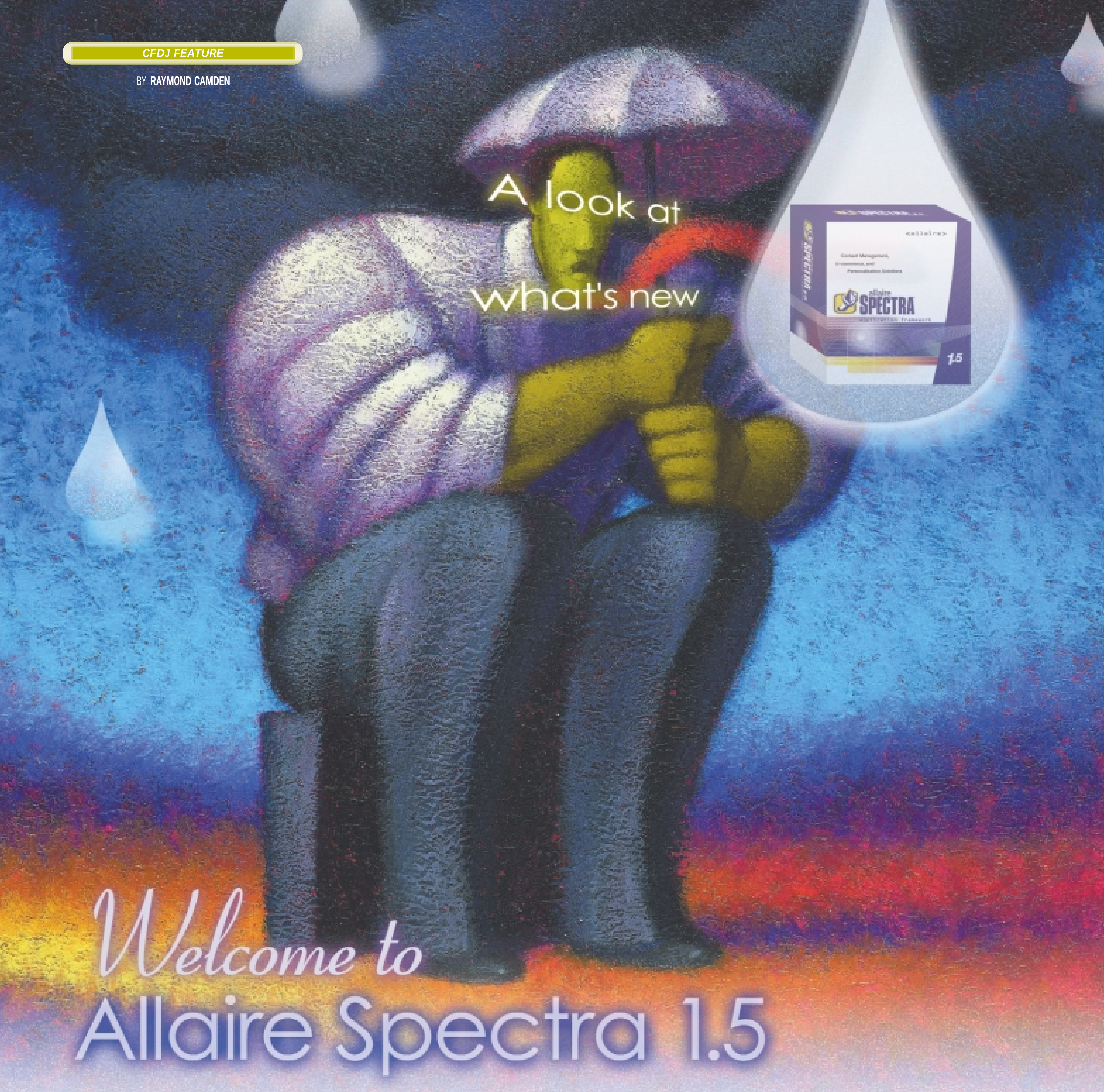
Obviously, there are books, Web sites, instructor-led training (I happen to know a trainer that just might be a fit for you) – a whole host of resources available, but only those interested in investing in YOU will really use them.

Programming isn't about learning rules and syntax, not really. It's about creating systems that work elegantly to solve problems and create new capabilities. Programmers who see themselves as craftsmen of the new technology study what works and what doesn't. They evaluate trends and technologies from the point of view of someone who is a perpetual student, always alert for new ways to master their craft. They're motivated and encouraged by writing code that's beautiful and elegant, and that works. And that, as it turns out, is the perfect profile for an investor in YOU.

ABOUT THE AUTHOR
Hal Helms (www.halhelms.com) is a Team Allaire member who provides both on-site and remote training in ColdFusion and Fusebox.

HAL.HELMS@TEAMALLAIRE.COM

BY RAYMOND CAMDEN



A look at what's new

Welcome to Allaire Spectra 1.5

At the end of 2000, Allaire released version 1.5 of Spectra, a major update of the product. This article briefly covers the major changes in version 1.5 as well as details about some of the cooler aspects. We'll also look at what future versions Spectra has in store.

What's New in Spectra 1.5?

Spectra 1.5 includes a large suite of new features:

- ***Content Versioning:*** This allows you to create multiple versions of content. It's useful for tracking the history of a document or rolling back to a previous document if a mistake is found. One of the most requested features for Spectra, it's now out of the box.
- ***Productivity Wizards:*** These wizards aid in reporting, security, and application design.
- ***Caching:*** Improved caching has greatly increased the speed of Spectra in both application code and the Webtop. A new feature, Intellicache, has been added to containers making caching even easier to use.
- ***Content Authoring:*** The HTML Editor has been upgraded to allow for spell checking and image uploading. Most important, it now works in both Netscape and Internet Explorer.
- ***Webtop Revision:*** The Webtop has had a major makeover – it's faster and allows for application switching or logging off from any page. It's also completely extensible, which means you can add your own menu items and reorder the menu anyway you see fit.

The New Webtop

The new Webtop rocks, to put it mildly. Of course I'm probably a bit biased, but it not only looks 100% better, it also runs quite a bit faster. Let's examine the new look and feel. The major change is the addition of frames. Figure 1 shows what the Webtop looks like after you've logged into the system.

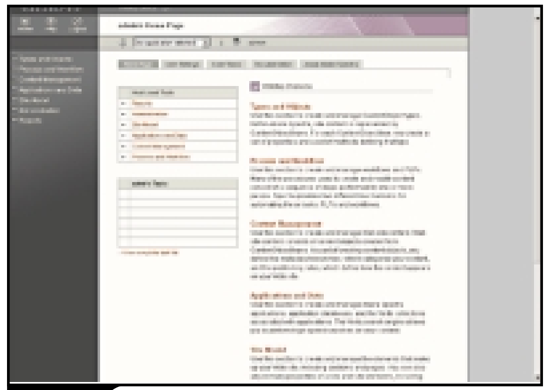


FIGURE 1: The Spectra 1.5 Webtop

The left-hand side of the Webtop now consists of a menu frame, which makes going back and forth between various tasks much quicker than the 1.01 version. The top frame has a status bar that updates when tasks are running. You can also switch between applications and log out from any part of the Webtop. Previously, these features were available only with a custom tag from the Developers Exchange.

Along with frames you also have access to your most recent tools as well as any tasks that have been assigned to your queue, as they're all available on the home page of the Webtop.

Everything has a new, sleeker design. Figure 2 shows how types are displayed in Spectra 1.5. You have immediate access to all properties and methods as well as information about which methods are default. Notice that the default methods are all faded out.

Figure 3 displays an object being viewed from the Webtop. *Note:* The tabs at the top allow quick access to various tasks within the object. You can work with version control (more on that later), edit and view the object, and perform other updates.

Of course, there's a lot more to the new Webtop than just better looks and speed.

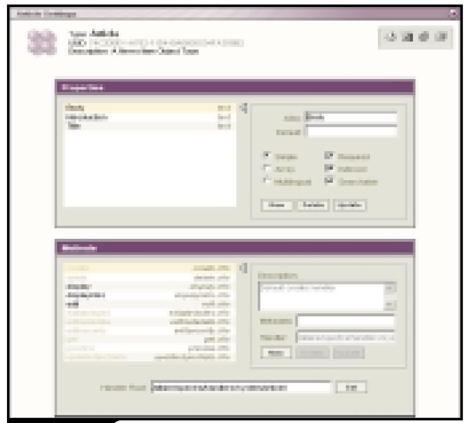


FIGURE 2: Editing the article content object type

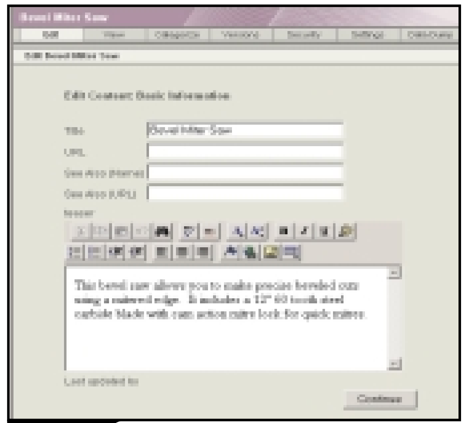


FIGURE 3: Using the default object editor

Things get really interesting when you add new items to the Webtop. Imagine you have a template that displays all users who have registered on your system over the past 24 hours”

One of the nicest new features is the ability to manage the Webtop navigation system – you can reorder items according to whatever system works best for you. The Webtop Manager lets you edit the Webtop and its appearance when using Spectra applications (see Figure 4). To reorder Webtop navigation items, a simple DHTML tree lets you select items and move them about. If I want the first item under Types and Objects to be Types instead of Objects, simply select Types and click the up arrow.

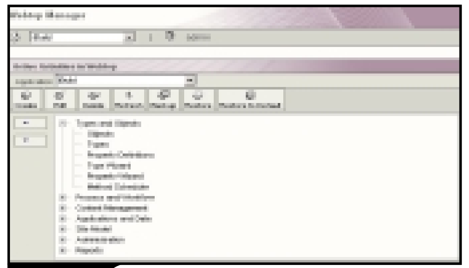
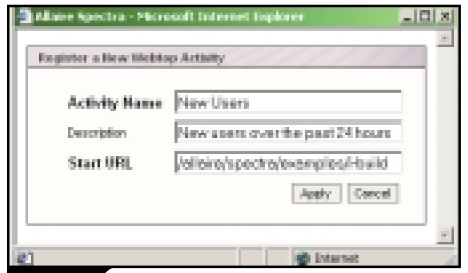


FIGURE 4: The Webtop Manager allows you to modify the Webtop navigation

Things get really interesting when you add new items to the Webtop. Imagine you have a template that displays all users who have registered on your system over the past 24 hours. To add this new activity simply use the Create button. For this example I'll select Reports before hitting Create so my new activity will be placed in that particular section. Figure 5 displays the simple form used to add a new activity.


Allaire Spectra - Microsoft Internet Explorer
Register a New Webtop Activity
Apply
Cancel
Internet

Activity Name: New Users
Description: New users over the past 24 hours
Start URL: /allaire/spectra/examples/build

FIGURE 5: Using the default object editor

Once you've added the activity, use the Refresh button on the Webtop Manager to reload the menu frame. Figure 6 shows our new menu item under the Reports section.

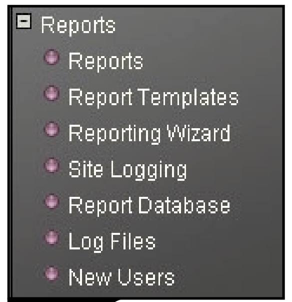


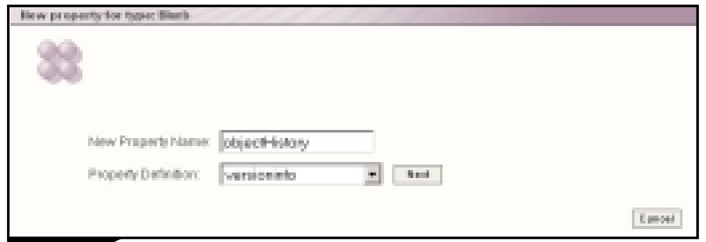
FIGURE 6: The activity has been added to the desktop

If you want you can come back later and move the activity higher within the Reports section, or possibly even add security to the activity so only certain people see it in their menu.

Version Control

Another cool feature of Spectra 1.5 is version control. Out of the box you get a system that enables version control for any content-object type. You can track changes to content, roll back to earlier versions, and even have your version data stored in a separate database.

How do you use this new feature? It comes down to two simple steps. The first involves preparing the content-object type for versioning. Like most things in Spectra, you can either use the Webtop or implement the change via a script. To add version control to a content-object type via the Webtop, go to your type and add a new property called objectHistory of type versioninfo. In Figure 7 we see what this new property should look like when added to the "Blurb" content-object type.



New Property Name: objectHistory
Property Definition: versioninfo

FIGURE 7: Adding the "objectHistory" property to the content object type

Once we've done that any content object of the type we modified can be versioned. Normally you'd want to do this when you create the object. Listing 1 provides a simple script that creates an object (using our "Blurb" type again) and tells Spectra to version control it.

The first line of Listing 1, <CFA_GlobalSettings>, sets up default settings that Spectra needs to run. To create our new object we build a structure to store the object data. Since Blurb is a simple object, it needs only two properties, Title and Body. Next we call <CFA_ContentObjectCreate>. We pass in all the relevant information, ask it to return the object ID in a variable called newID, then output this ID to the screen. Now comes the fun part. All we need to do to enable versioning on this object is run <CFA_ContentObjectVersionCreate>. The first argument to the tag is your normal data source argument. The second argument is very important. Spectra 1.5 allows you to store version information away from your application data, which enables you to store your archived information in a separate data source (which also makes things a bit safer as well). Next we tell the tag which object ID to use and pass it the ID of the newly created object. Last, we provide a label for the version, in this case, "First Version." If we go to the Webtop, find our object, launch the editor, and select Versions, we can see our first version now exists (see Figure 8).

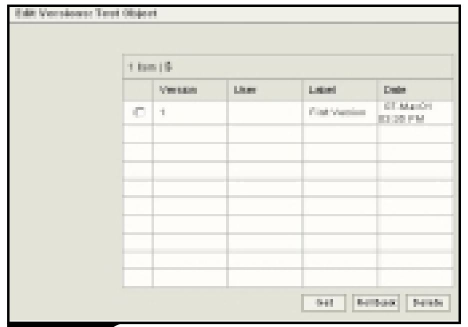


FIGURE 8: The version history for our content object

What do the three options Get, Rollback, and Delete do? Get allows you to retrieve an earlier version of the object and make it the current version. Rollback acts like get, making an earlier version the "current" version, but it also deletes any versions that were "above" that one. Delete simply removes a version from the object history.

How do we create more versions? All we need to do is run the <CFA_ContentObjectVersionCreate> tag again. If we use this tag in the Edit handler, we can ensure that every edit of the article creates new versions.

There's a lot more to version control than we mentioned here. For example, you can also limit the number of versions Spectra keeps in storage. All the version control options available via the Webtop (get, rollback, etc.) are also available to you via custom tags.

What's Coming

Allaire isn't resting now that Spectra 1.5 is released. Two big things are on their way. First, you can expect a maintenance release that will tackle issues that have already been covered with hot fixes as well as taking care of other issues. (This brings up an important point: if you want to ensure your Spectra server is running all the latest hot fixes, whether you're running 1.01 or 1.5, go to the hot fix list at www.allaire.com/Handlers/index.cfm?ID=16833" or www.allaire.com/Handlers/index.cfm?ID=16833.)

Second, in the new few months Allaire will release a site dedicated to the open source nature of Spectra. Since its inception, Spectra has been almost 100% open source. This meant that developers could easily "get under the hood" and modify the code. They could fix bugs when they found them and add new features – all without waiting for Allaire. This new Web site will allow developers to submit their modifications and bug fixes to Allaire. These modifications will go through an approval process and when approved, they'll become part of the source product. This will also be the main place to update your Spectra server. You can download individual bug fixes and modifications or simply ask for a "complete" package and not worry about the details (www.allaire.com/Handlers/index.cfm?ID=16833).

About the Author

Raymond Camden is the principal Spectra compliance engineer for Macromedia. He is coauthor of the Allaire Spectra e-Business Construction Kit *and* Mastering ColdFusion 4.5. *He helps manage the Hampton Roads ColdFusion User Group*

rcamden@macromedia.com

Listing 1

```
<CFA_GlobalSettings>

<CFSET Props = StructNew()>
<CFSET Props.Title = "Title Version 1">
<CFSET Props.Body = "Body Version 1">

<CFA_ContentObjectCreate
 DataSource="cfaibuild"
 TypeID="74C2DBD0-A782-11D4-BA0600C04FA358B2"
 bActivate=1
 label="Test Object"
 stProperties="#Props#"
 R_ID="newID"
>

<CFOUTPUT>New object created, #newID#<P></CFOUTPUT>

<!--- Enable Version Control --->
<CFA_ContentObjectVersionCreate
 DataSource="cfaibuild"
 VersionDataSource="cfaibuild"
 ObjectID="#NewID#"
 label="First Version">

<CFOUTPUT>Done enabling versioning for #newID#<P></CFOUTPUT>
```

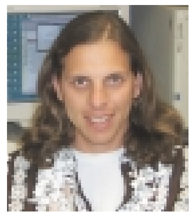
CODE LISTING

The code listing for this article can also be located at www.ColdFusionJournal.com

A Display Method for **Your Site**

Implement a framework that features a display template

BY ANGELO ALVAREZ



Saving development costs by reusing code has been a goal of the computing industry for a long time. With a little forethought you can save time and money developing Web systems using ColdFusion by encapsulating functionality in custom tags.

As a Web developer I'm constantly concerned about maintaining a consistent look and feel throughout the sites I create. However, I don't always modularize certain components of the site, which I should do.

Modularization of site components is important not only in terms of consistent site display, but also in terms of site modification. In fact, site modification can become a major issue after you've finished building a site for a client and the client asks, "Could we move that button over there?" or even "Can we put the navigation bar down the right side instead of the left?" What about giving your site a brand-new look and feel? This is not such a major task if your site is only a few pages, but it can be very tedious if you have numerous pages or templates. How do you deal with this problem of site display control? Wouldn't it be nice to have a framework that allows for easy site modification?

Possible Solutions

Use Frames

The easiest and quickest solution is to use frames since they can control certain key areas of the site display such as the top banner, footer, and navigation bars. This solution works unless your client doesn't like frames (believe it or not, there are those who don't like frames). What do you do then?

Use CFINCLUDE for Key Items

CFINCLUDE can be used on every page for key items such as the top banner, navigation bar, footer, and even background. If you need to change a component, you only need to modify the one file that controls it. For example, this method is an effective way to change the navigation buttons. One drawback, however, is that it's not effective if you need to change the location of the navigation buttons on every page throughout the site.

```
<cfinclude template="top_banner.cfm">
```

Single Display Template

Use a single display template throughout the site with a display variable (e.g., display.cfm?function=about). In this approach the display template uses CFINCLUDE, as in the previous method, to display the top banner, navigation, content, and footer sections. In the content section a CFINCLUDE statement is used to insert the content pages that are referred to by the "function" variable. Listing 1 provides an example of the display template.

One drawback to this approach is that in order to use preexisting templates in this manner, all links and form actions need to be changed to call the display template (e.g., <form name="test" method= "post" action="display_template. cfm? function=test">). In fact, modifying your preexisting templates for this approach can turn out to be quite a bit of work and not worth the trouble.

The Solution

Implement an application framework that calls the display template. This solution is a combination of the last two methods within an application framework.

Implementation

This method is very easy to implement and uses the application.cfm, which is processed first, to intercept requests and send them to the display template for processing:

```
<----Application.cfm----->
<cfapplication name="mysite">
<cfinclude template="display.cfm">
<cfabort><!---stop processing the rest--->
```

The display template setup is similar to the previous approach except the content is dynamically displayed by using a cfinclude with the #cgi.script_name# variable:

```
<!—content section --- >
 <cfinclude
template="#cgi.script_name#">
```

CFABORT is used in the application.cfm to stop the second display of the file referenced by the cgi.script_name variable. Without the CFABORT the resulting page would be a combination of the display template with the embedded content page information and the content page, one after the other (very messy). (For code including this template, see Listing 2.)

Bypassing the Display Template

To bypass the display template for certain files (e.g., JavaScript popup windows), create a list of the templates in the application.cfm file you want to bypass, then loop through to see if any one of them is being requested (see Listing 3). Everything put together in the application.cfm can be seen in Listing 4.

HTML Tags in Content Pages

If there are HTML HEAD and BODY start and end tags in the preexisting content files, the resulting page may not display correctly since there'll be multiple occurrences of these tags in the processed document, which can pose a problem. To deal with this issue you can download or create custom tags that will check for these tags and grab only what's between the body tags. Another solution is to remove those HTML tags from the content templates that will be used by the display template.

Advantages

Easy Implementation

The major advantage of this method over the previous solutions is that any preexisting templates can be easily implemented with little or no modification. An important note is that all links and form actions on these pages will need to reference the site root for files that are located outside the current directory instead of using paths relative to the current directory (e.g., don't use "../page.cfm", use "/siteroot/directory/page.cfm").

Leverage Site Framework for Other Projects

Use this method to build sites that can be easily modified for use in other projects.

Scalability

If you want to take this approach one step further you can; for example, you can change the display based on the target directory of the requested file. This can be easily accomplished by placing logic in the application.cfm file, which can detect the target directory, and then selecting which display template to use:

```
 <cfif cgi.script_name contains
"/resources">
 <cfinclude
template="display_recources.cfm">
 <cfelse>
 <cfinclude template="display_stan-
dard.cfm">
 </cfif>
```

In addition this approach can work for delivering customized content to customers based on certain criteria.

Performance

Because this method is based largely on the use of CFINCLUDE in an application framework, it'll have very minimal impact on the overall performance of the application.

Summary

When building Web sites, developers will concern themselves with maintaining a consistent look and feel throughout their site, but they may not use an approach that modularizes their site components. This doesn't become an issue until they need to make modifications to their site that deal with the display of the navigation bar, for example. It's not such a major task if the site consists of a few pages, but it can be a problem if there are many pages or templates. By using a method that implements a display template in an application framework with CFINCLUDE, you can easily deal with this problem and make site modification very easy to accomplish. And by using this method you can readily modify a site for use in other projects.

• • •

Acknowledgments

I'd like to thank Mark Ching for putting up with my ranting and raving about the need for this type of framework. I'd also like to thank Darnell Griffin for providing feedback while testing this method in some of his projects.

ABOUT THE AUTHOR
Angelo Alvarez is a consultant with Booz Allen & Hamilton. He uses ColdFusion to build dynamic database-driven applications used in telemedicine projects managed by the Pacific e-Health Innovation Center.

ANGELO@HAWAII.RR.COM

Listing 1

```
<!—this is the display template-—>
 <table>
 <tr>
 <td colspan="2">
 <!----top banner--- >
 <cfinclude template="topbanner.cfm">
</td>
 </tr>
 <tr>
 <td ><!----left navigation bar--- >
<cfinclude template="navbar.cfm">
</td>
 <td>
 <!----content section--- >
 <cfif url.function eq "about">
 <cfinclude template="about.cfm">
 <cfelse>
 <cfinclude template="home.cfm">
 </cfif>
 </td>
</tr>
<tr>
<td>&nbsp;</td>
<td>
<!----footer--- >
<cfinclude template="footer.cfm">
</td>
</tr>
</table>
```

Listing 2

```
<!-----Display_template.cfm Listing----->
<!---this is the display template--->
<table>
<tr>
<td colspan="2">
<!----top banner--->
<cfinclude template="topbanner.cfm">
</td>
</tr>
<tr>
<td >
<!----left navigation bar--->
<cfinclude template="navbar.cfm">
</td>
<td>
<!----content section--->
<!---make sure they don't call display template directly--->
<cfif cgi.script_name contains "display_template.cfm">
  <cfinclude template="home.cfm">
<cfelse>
   <cfinclude template="#cgi.script_name#">
</cfif>
</td>
</tr>
<tr>
<td>&nbsp;</td>
<td>
<cfinclude template="footer.cfm">
</td>
</tr>
</table>
```

Listing 3

```
<!---nofiles is a list of files that bypass the display
    template b/c they are popup windows, etc.-------->
<cfset nofiles="main.cfm,….">
<cfset found=0>
<cfloop index="listitem" list="#nofiles#" Delimiters=",">
<cfif cgi.script_name contains #listitem#>
<cfset found=1>
<cfbreak>
</cfif>
</cfloop>
<cfif found gt 0>
<cfelse>
<cfinclude template="main.cfm">
<cfabort>
</cfif>
```

Listing 4

```
<!---Application.cfm listing----->
<cfapplication name="myapp">
<!---nofiles is a list of files that bypass the display
template b/c they are popup windows, etc.------>
<cfset nofiles="main.cfm,...">
<cfset found=0>
<cfloop index="listitem" list="#nofiles#" Delimiters=",">
<cfif cgi.script_name contains #listitem#>
<cfset found=1>
<cfbreak>
</cfif>
</cfloop>
<cfif found gt 0>
<cfelse>
<cfinclude template="main.cfm">
<cfabort>
</cfif>
```

CODE LISTING

The code listing for this article can also be located at
www.ColdFusionJournal.com

JAVAEDGE 2001 INTERNATIONAL JAVA DEVELOPER CONFERENCE & EXPO

LEADING EDGE JAVA TECHNOLOGIES FOR THE NETWORK ECONOMY

JAVAEDGE conference&expo

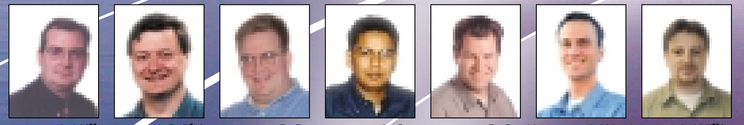
The ONLY Java Event Backed by the Power of SYS-CON MEDIA and JAVA DEVELOPER'S JOURNAL

CONFERENCE	EXPO	HILTON NEW YORK
September 23–26, 2001	September 24–25, 2001	New York, NY

Plan to Attend...

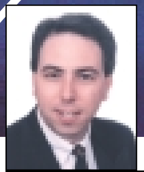
This 4-Day Cutting-Edge Conference

SAVE THE DATES!


J. Milbery S. Phipps R. Ross A. Sagar B. Scott J. Westra A. Williamson

Providing the depth and breadth of education needed to stay on the "edge."

THE PREMIER EAST COAST CONFERENCE OF THE YEAR FOR HIGHLY QUALIFIED JAVA PROFESSIONALS. SESSIONS ARE DESIGNED FOR BEGINNER, INTERMEDIATE AND ADVANCED DEVELOPERS, AS WELL AS ENGINEERS AND PROGRAMMERS. NO MATTER WHAT YOUR LEVEL OF EXPERTISE, JAVAEDGE WILL ADVANCE YOUR PROFESSIONAL KNOWLEDGE.


Conference Tech Chair Sean Rhody, Founding Editor, *Java Developer's Journal*

Conference Advisory Commitee

Calvin Austin, Lead Software Engineer, J2SE Linux Project, Sun Microsystems, Inc.
James Duncan Davidson, Java Servlet API and Java XMP API, Sun Microsystems, Inc.
Jim Driscoll, J2EE Development Team Manager, Sun Microsystems, Inc.
Jason Hunter, Senior Technologist, CollabNet
Doug McMahon, Head of Java Strategy, Oracle Corporation
Jim Milbery, Applications Editor, *Wireless Business & Technology*
Simon Phipps, Chief Software Evangelist, Sun Microsystems, Inc.
Rick Ross, President, UserMagnet, and Founder of the JavaLobby
Bill Roth, Group Product Manager, iPlanet, Sun Microsystems, Inc.
Ajit Sagar, Software Architect, VerticalNet
Bruce Scott, President & CEO, PointBase, Inc.
Eric Stahl, Product Manager, BEA WebLogic
Jon S. Stevens, Cofounder, Clear Ink Corporation
Jason Westra, CEO, Verge Technologies
Alan Williamson, Editor-in-Chief, *Java Developer's Journal*
Blair Wyman, Chief Software Architect, IBM Rochester
Peter Zadrozny, Chief Technologist, Europe, Middle East & Africa, BEA Systems Ltd.

Owned and Produced by
SYS-CON EVENTS
135 Chestnut Ridge Road,
Montvale, NJ 07645
201 802-3069 • Fax: 201 782-9601

Who Should Attend

Developers, Programmers, Engineers • i-Technology Professionals
Senior Business Management • Senior IT/IS Management • Analysts, Consultants

from these industries...

Architecture	Entertainment/Media	Manufacturing
Construction	Finance/Insurance	Nonprofit
Communications	Government	Printing
Computing & Networking	Health Care	Publishing
Consulting	Hospitality/Travel	Transportation
Education/Learning	Internet & Web	Utilities
	Legal/Real Estate	Wholesale

SYS-CON MEDIA • JAVA DEVELOPER'S JOURNAL • XML JOURNAL • wireless • LINUX BUSINESS WEEK • COLDFUSION Developer's Journal • WebSphere DEVELOPER'S JOURNAL

DON'T MISS THESE FOUR CRITICAL INFORMATION-PACKED DAYS!

Conference Highlights

- **Leading keynote speakers** will provide "Big Picture" perspectives.
- **Developer Contests** will be held to challenge attendees to put their expertise to the test.
- **Java Developer's Journal Readers' Choice Awards** and other industry awards will be presented at the leading East Coast Java Developers' Conference of 2001.
- **Birds of a Feather Sessions** will provide the venue for an exchange of ideas focusing on: WebLogic, WebSphere, iPlanet, Personalization, JDBC, and Swing, among others.
- **Book signings and a bookstore** will enable delegates to meet the authors and peruse the most pertinent assortment of Java books assembled anywhere!

Here is a preliminary glance at some of the sessions from which you will be able to select. Visit www.javaedge2001.com for frequent program updates. Program subject to change.

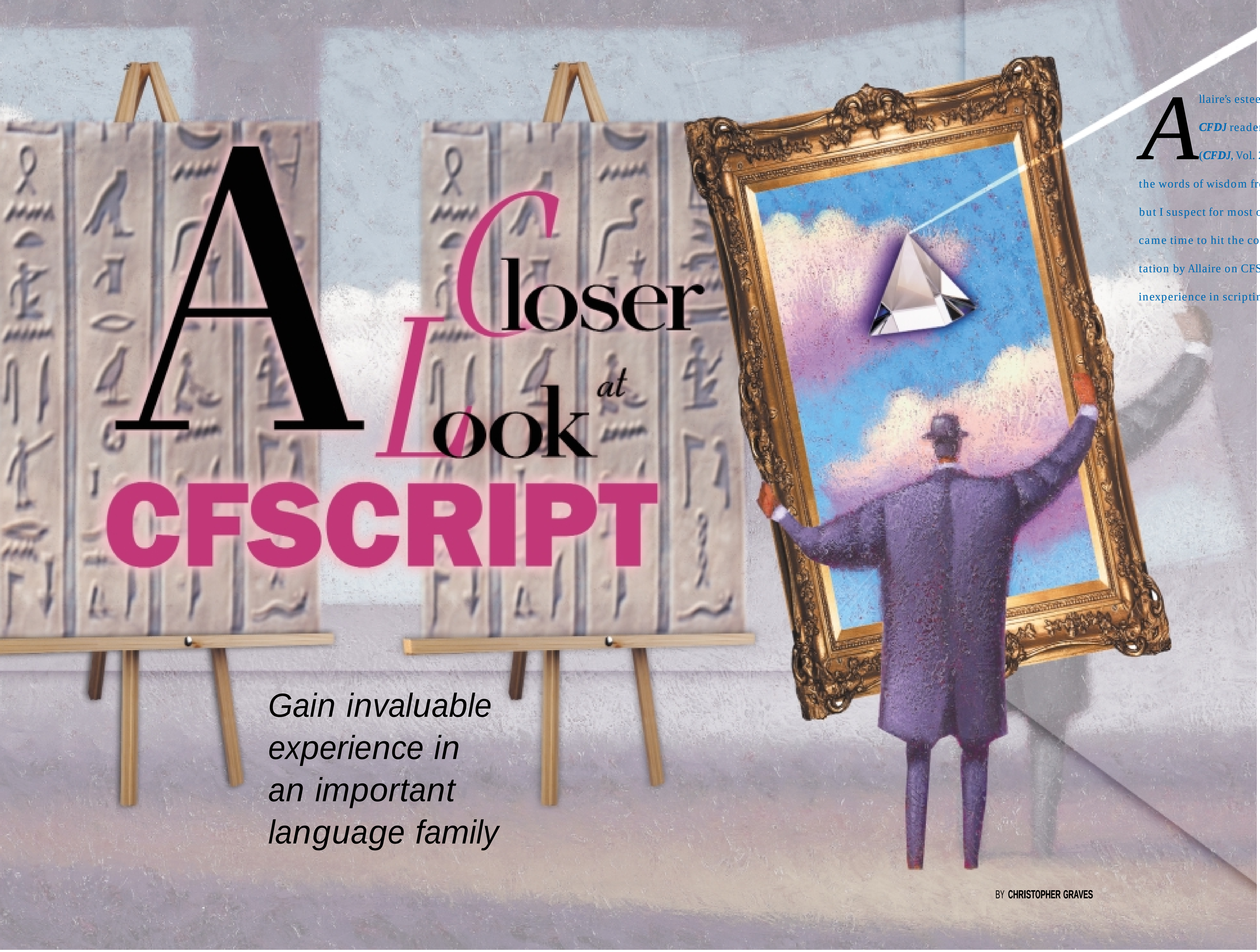
PRELIMINARY PROGRAM-AT-A-GLANCE

Sunday, September 23, 2001
Tutorials & Night School
Tutorials and Night School will kick off on Sunday, September 23, and continue through September 26, 2001.

Monday, Tuesday, Wednesday, September 24-26, 2001

Track 1: J2ME - Micro Edition and Wireless	Track 2: J2SE - Standard Edition	Track 3: J2EE - Enterprise Edition	Track 4: Working with i-Technology	Track 5: Practical Business Solutions & Best Java Tools Hands On...
Cutting-edge sessions for software engineers and hardware specialists working on wireless solutions.	General Java programming, and corporate programmers developing full Java applications including several introductory sessions.	Advanced sessions for software architects, Web programmers, corporate developers, and consultants developing server-based applications.	Technical and management sessions for business analysts, corporate systems managers, architects, project managers and CIOs.	Leading Java tools and solutions. Many classes taught by the actual creators of these tools.
Serving Up Java Technology for Wireless Devices	Threads and Java Technology	Implementing the Application Server Contracts for Java Connectors	A Java Framework for Managing Dynamic Stored Procedure Calls	Content Management
Coding for Mobile Devices with Java	Transport Independence for XML	Use JMS Point to Point Model to Achieve Load Balancing	Simplifying Complex Multi-Platform Software Deployment	Personalization
Developing Wireless Applications in Java Using CLDC/MIDP APIs	Implementing Strong Cryptography in Java	Bean There, Done That – Deployed Large EJB Systems Using Rull-Based Development	Autonomous Intelligent Agents	Commerce Servers
Using Java on a Set-top Box	An Extended JDK Environment for Software Reuse	Using Jasmine ii with JSP and JavaBeans	Component Frameworks	Exchange Engines
Key Dynamic Deployment	Using Swing Editor Kits	Web Services Using Java	Keys to Enterprise Application Integration	CRM Solutions
Using Java Cards to Encrypt/-Digitally Sign Data	Programming Aggregators, Bots and Spiders in Java	Open Course J2EE in a Services Architecture	Achieving Enterprise Level XML and e-business Integration with Java	Sales Force Automation
Storing Preferences with Javacard	Using Design Patterns to Write Better Java	EJB Anti-Patterns: Bad EJBs and the Programmers Who Love Them	Fundamental OO: A Java Perspective	
J2ME MIDP Programming	Core XML Programming I	Web Services & J2EE in The Trenches – Ready for Prime Time?	Encrypting XML Data	
Source Enhydra Java/XML Applications	Core XML Programming II	Put Your J2EE Application on a Diet	Connecting to IMS	
E-Mail Based Personal Computer Accessories	XML Parser Infrastructures	Practical Guidelines for EJB Design: Lessons Learned in the Trenches	Fighting II Entropy: Optimizing EAI to Control Application Intercommunication	
Evolutionary Development and Delivery Platform	Programming DOM Level II and Xerces	Centralized Application Logging Using JMS		
Providing Debugging Services Using the Java Debugger Platform Architecture	J2EE-based Syndication Server	Services-based Approach for Building Server Side Applications		
Java Technologies for Mobile Devices and Services		Providing Debugging Services Using the Java Debugger Platform Architecture		
		Component-Based Separation of PA		

CLICK ON www.javaedge2001.com TO REGISTER TODAY!



A Closer Look at CFSCRIPT

Gain invaluable experience in an important language family

BY **CHRISTOPHER GRAVES**

Allaire's esteemed guru Ben Forta introduced CFSCRIPT to ***CFDJ*** readers last year with his article "Stick to the Script" (***CFDJ***, Vol. 2, issue 7). Hopefully, some of you absorbed the words of wisdom from our evangelist and gave CFSCRIPT a shot, but I suspect for most of you the lesson fell to the wayside when it came time to hit the code again. I can't blame you. Paltry documentation by Allaire on CFSCRIPT coupled with most CF developers' inexperience in scripting languages makes it easy to ignore.

Most ColdFusion developers don't have a scripting background. So why take up CFSCRIPT? It can't replace conventional CF tag structure. Although a modest performance gain can be seen in some circumstances, its exclusion of CF tags significantly limits its potential functionality. What does it offer most developers who are not seasoned scripters?

I believe it can make three compelling contributions to your repertoire. First, CFSCRIPT simplifies concatenation of strings and basic mathematical manipulation, even if you know little about scripting.

Second, and perhaps most important, CFSCRIPT offers an easy way to encapsulate business logic in a single location at the top of your page. Studio's coloring of scripting helps differentiate business logic from output, making it easy to find and address. It also rests in a more native state, devoid of some of the visual overhead tags create.

The third and rarely discussed reason, at least in my opinion, is that it serves as a great introduction to scripting languages. While most of us enjoy the ease of ColdFusion, we limit ourselves by not employing JavaScript, JSP, or other scripting languages. Indeed, next year Allaire will allow the cohabitation of JSP and ColdFusion in their NEO release (sixth generation ColdFusion). Future ColdFusion developers will really need to know JSP to extract the full potential of future CF applications, and what better way to cut their teeth than inside ColdFusion.

I must confess, jumping into CFSCRIPT may be a bit intimidating. Gone are meaningful plain English tags (CFSET, CFLOOP, etc.) that made developing so easy. Also, as I mentioned earlier, Allaire's normally strong documentation provides woeful coverage of CFSCRIPT.

To help you along your daunting journey into the scary forest of scripting, I've prepared a large block of CFSCRIPT employing many of the basic constructs you'll want to use. I've chosen a credit card validation routine for a shopping cart. This doesn't expose all of CFSCRIPT's operations, but should provide enough examples to give you a jump start on your own projects. (Source code for this article can be found on the ***CFDJ*** Web site, www.coldfusionjournal.com.)

To start the process I open with a CFSCRIPT tag. There are no attributes for the tag, but I don't leave the Script block empty (ColdFusion doesn't like this and returns an error). Before I even get to the credit card validation, I'm taking advantage of the fact that I've started a CFSCRIPT block and set defaults for a page that would normally have been accomplished with CFSET or CFPARAM. By moving these sets into the CFSCRIPT, I'm reducing some visual and performance overhead. I identify the section with a comment that I start with two back slashes. These comment markers only comment out a line at a time and have no ending markers to stop the commenting (a bit weird for CF developers, but old hat for scripters).

The first three events are simple sets:

```
j=0; local.errM = ""; check  = "";
```

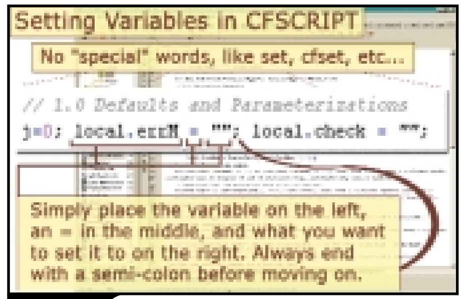
(see Figure 1).

Note: Other than comment lines, CFSCRIPT ignores carriage returns. After making each assignment you must place a semicolon before addressing another operation. CFSCRIPT doesn't require any special word to make an assignment (like var or set); simply place a variable on the left, some expression on the right, and end it with a semicolon. Missing semicolons will be the primary cause of errors when you begin CFSCRIPTing, and they're generally not obvious. This single line of code replaces the following three conventional ColdFusion tags:

```
<CFSET j=0>
<CFSET local.errM = "">
<CFSET check = "">
```

Hopefully, you can recognize the "visual" advantage of making these settings inside of CFSCRIPT in conventional tags.

The next block of code drives navigation links on the page, and again bears no relation to the credit card routine. As I mentioned, you can't use ColdFusion tags inside CFSCRIPT. We can, however, recreate the functionality of some tags using ColdFusion functions. In this case I need to parameterize a form variable, which may or may not have been passed. Since I can't use CFPARAM, I'll have to mimic it with the IsDefined function and a simple assignment if the formfield doesn't exist:

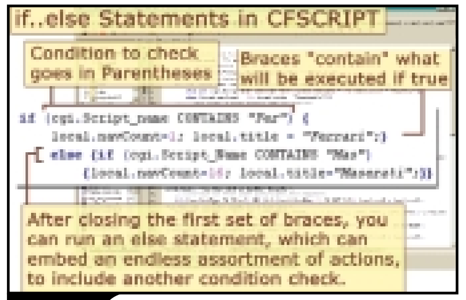

Setting Variables in CFSCRIPT

No "special" words, like set, cfset, etc...

```
// 1.0 Defaults and Parameterizations
j=0; local.errM = ""; local.check = "";
```

Simply place the variable on the left, an = in the middle, and what you want to set it to on the right. Always end with a semi-colon before moving on.

FIGURE 1: Setting variables in CFSCRIPT


if..else Statements in CFSCRIPT

Condition to check goes in Parentheses

Braces "contain" what will be executed if true

```
if (cgi.Script_name CONTAINS "Fer") {
   local.navCount=1; local.title = "Ferrari";}
 else {if (cgi.Script_Name CONTAINS "Mas")
     {local.navCount=16; local.title="Maserati";}}
```

After closing the first set of braces, you can run an else statement, which can embed an endless assortment of actions, to include another condition check.

FIGURE 2: if..else statements in CFSCRIPT

```
if (Not IsDefined("form.carType"))
form.carType = "Mondial t";
```

This introduces our first conditional statement, a simple if. JavaScripters take note, your favorite operators (!, ==, >=, etc…) are not available. You have to use ColdFusion operators (such as "NOT," "IS," "GTE," etc.), which can be a bit frustrating but are necessary nonetheless. To begin the "CFIF" simply type if, followed by the condition you wish to check enclosed in parentheses. In a simple conditional check, place the assignment or action right after the check and end it with a semicolon. In this case, if the condition is met (i.e., there's no form variable), I'd like to set it to a default of "Mondial t". We'll cover the if statements in more detail later, but I wanted to demonstrate that there are workarounds for some ColdFusion tags (although in this case, CFPARAM would have been easier).

Next, I moved some of the logic I employed in the navigation scheme into the CFSCRIPT block. This specific chunk of code occupied eight lines of code in my original template, so it was a prime candidate for moving into this block of CFSCRIPT. Again, we haven't hit the credit card validation yet, I'm just moving logic normally found throughout my page into a clean and more concise environment. This particular snippet peeks at the path to see if we're looking at Ferraris, Maseratis, or Ducatis (see Figure 2). For whichever one the script will set a nav count that will position one of those silly arrows the client absolutely must have:

```
if (cgi.Script_name CONTAINS "Fer")
{local.navCount = 1; local.title =
"Ferrari";}
else {if (cgi.Script_Name CONTAINS "Mas")
{local.navCount = 16; local.title =
"Maserati";}
else {local.navCount = 32; local.title =
"Ducati";}}
```

In CFSCRIPT the braces allow you to perform multiple actions instead of a single set (much the same way parentheses work in ColdFusion or mathematics). Without the braces the if statement would end with the first semicolon. After the first set of braces, we have an else statement (just like CFELSE). I actually needed a CFELSEIF, but that's not directly available in CFSCRIPT so I simply embed another if statement inside the else. This block could have been accomplished with a switch statement (very similar to CFSWITCH), which Ben Forta covered in his CFSCRIPT article.

The final block of code I want to discuss prior to launching into the credit card validation is a simple banner rotator. Again, this logic would normally appear elsewhere on my template, but I'm moving it into the CFSCRIPT block to clean up my page. This gives us an opportunity to witness a simple concatenation and arithmetic operation, which CFSCRIPT performs so well. The snippet takes the title I set in the last section of code and appends a number (from 1 to 6) based on what minute it is. This essentially rotates through six different banners (named, for example, Ferrari1.gif, Ferrari2. gif, etc.) and will be used in the content after the CFSCRIPT block.

```
variables.mod = local.title &
(Minute(now()) MOD 6) & ".gif";
```

You could extend this to create a string with the link and other HTML features as well. Whenever you have to assemble long strings (such as interaction with a COM object or other third-party software), always consider CFSCRIPT as an alternative to conventional CFSETs. JavaScripters should notice that ampersands, not the addition sign, concatenate strings.

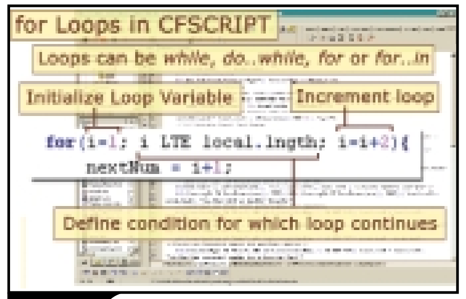
Finally we're ready to play with the credit card validation. The first thing I must do is set some defaults that I'll manipulate further downstream. I could have separated these assignments with a carriage return, but I prefer keeping the simple ones in a single line for cleanliness. Next we see a new comment tag, this time in a multiline format. This comment resembles traditional ColdFusion REM statements in that content contained between the symbols is not executed (versus the single-line comments). To begin a multiline comment we employ a backslash and an asterisk /*; to end we reverse the order and employ an asterisk and a backslash */.

```
/* Required: local.CardNum,
local.expYear, local.expMonth. PASSED:
local.errM on exception */
```

As the comment suggests, we're now going to step through the card number and strip out any nonintegers the client may have provided us. This event gives us our first loop. Scripters will welcome CFSCRIPT's loops, but regular ColdFusion developers will probably find these a tad confusing. The looping syntax in CFSCRIPT bears no resemblance to its big brother, CFLOOP, and due to the requirement that CFSCRIPT employs ColdFusion operators, it's not a direct match with JavaScript or Java loops (but is familiarly close).

Loops come in a variety of flavors in CFSCRIPT; the while loop is the first one we'll discuss. It's not the most common loop (which is the for loop), but it's the first we see in this application. The while loop inspects the conditions specified in its parentheses and executes the operations inside its braces as long as the condition is met. This is very similar, yet distinct from a do..while loop, which checks for the condition only after executing each loop, allowing a minimum of one run through the loop. In this scenario we're going to progress through each element of the credit card string the client submitted and discard any characters that are not integers.

```
while (Len(trim(local.CardNum)) GT 0) {
if(IsNumeric(mid(local.CardNum, 1, 1)))
check = check & mid(local.CardNum, 1, 1);
local.CardNum =
RemoveChars(local.CardNum, 1, 1);}
```


for Loops in CFSCRIPT
Loops can be while, do..while, for or for..in
Initialize Loop Variable
Increment loop
Define condition for which loop continues

```
for(i=1; i LTE local.lngth; i=i+2){
    nextNum = i+1;
```

FIGURE 3: for loops in CFSCRIPT

The condition in this situation is the length of the variable we're manipulating. It'll shrink by one position each time through the loop. The trim function discards accidental spaces the client may have provided. As mentioned earlier, the operator must be a ColdFusion operator (in this case "GT") instead of a mathematical or JavaScript one. I have a simple if inquiry that, when true, appends that integer to a new string, cryptically named *check*. Subsequently, the RemoveChars function strips out the character we had just checked. The same snippet could have been accomplished with a conventional for loop, which I'll discuss soon, or through an REReplace() with regular expressions (a better idea, but it doesn't afford me the opportunity to talk about loops).

The next lines of script build error messages if the card number fails to meet certain basic criteria, specifically length, expiration, and whether the first digit is appropriate for the card type specified. The only thing worth noting here is the compound conditions for the first if statement.

```
if((Len(local.CardNum) LT 13) OR
(Len(local.CardNum) GT 16)) local.errM =
local.errM & "<li>Your credit card number
must be between 13 and 16 characters
long, and should only contain numbers.";
```

As with its elder brother the CFIF tag, if statements can check for multiple conditions (ranges of values, exclusion of ranges, or different strings).

Before the template evaluates the specifics of the number, it offers an opportunity to short-circuit if any of the earlier conditions uncovers an error. This prevents unnecessary processing of a card we won't accept anyway. To do this I simply wrap the remaining logic in an if statement.

First I perform some quick math and set a couple of defaults. These numbers will be used in the loop that follows.

```
local.lngth = Len(local.CardNum) - 1;
tempCard =
Left(local.CardNum,local.lngth);
newNum = "";
```

The local.lngth is one less than the number of integers in the card, and tempCard is the card number stripped of the last digit. CFSCRIPT's variables are case insensitive, but minding case only helps build the coding discipline required in JSP or Java. Many developers prefer to employ more descriptive variables, but I've chosen to err on the side of brevity, due to the math I want to perform. For the same reason I've left the scope off tempCard, newNum, and Step1. In production I'd scope these variables to prevent possible problems.

Next I have the first for loop. The for loop is CFSCRIPT's version of a conventional CFLOOP. After the command *for*, parentheses contain a set of conditions that will determine how many loops to execute.

```
for(i=1; i LTE local.lngth; i=i+2){
```

The i=1; establishes the variable "i" as the index or incrementing variable. The second expression defines the condition under which the loop should continue (see Figure 3). In this case, as long as "i" is less than or equal to the value of local.lngth, the loop will process its contents. The last expression increments the variable. I need to step by two through the loop, thus I set "i" equal to itself plus 2. If I didn't want the step, I'd use the expression i=i+1. The CFLOOP equivalent of this for loop would be:

```
<CFLOOP from="1" to="#local.lngth#"
step="2">
```

Since some credit cards are 16 digits while others are 15, I need two different operations with an if..else operation to direct each loop iteration. In the case of Visa, Discover, and MasterCard, I need to double the digit I'm currently viewing (specifically in the "i" position) and concatenate it with the next digit in line.

```
newNum = newNum & (2 * Mid(tempCard,i,1)) &
(Mid(tempCard,nextNum,1));
```

For the American Express card the next digit in line must be doubled and concatenated to the first digit:

```
newNum = newNum & Mid(tempCard,i,1) & (2 *
Mid(tempCard,nextNum,1));
```

Once I've built the new number I must sum all of its digits. For this a simple for loop can iterate through each number and add it to a variable I'll call sumNum:

```
for(j=1; j LTE Len(newNum); j=j+1) sumNum =
sumNum + Int(Mid(newNum,j,1));
```

Given the simplicity of the operation, I've left the braces out of this for loop. If you perform more than one operation, however, you'll need braces (just as with if..else statements).

The final step compares the last digit of the credit card against the difference of the right-most digit of sumNum from 10. If it's not a match, I create an error message to display in the content.

In a real application this template would allow the client an opportunity to correct a mistaken entry before sending the card to a financial institution for verification. This template doesn't guarantee that the card submitted is valid, but that it conforms to industry standards. Performing such checks before interacting with third-party agents not only en-hances server performance (by preventing HTTP interaction for an event that was bound to fail), it also improves your chances of completing the transaction by providing the client with an opportunity to immediately correct a mistake. For best results this should be combined with a similar routine in JavaScript, providing instant feedback (if the client has JavaScript turned on) when the client has made a faulty entry.

This same application logic in traditional ColdFusion tags occupies more than twice the number of lines as its CFSCRIPT counterpart and is more difficult to read. By translating the functionality into CFSCRIPT I've made the business logic more transparent, transportable, and refined. Indeed, when I translated this logic, I uncovered several unnecessary redundancies and found a better progression for the logic simply because the operations were unencumbered by tags.

For those ColdFusion developers without scripting experience, I'd highly recommend some experimentation with CFSCRIPT in your next application. As a rule of thumb, anytime more than three consecutive CFSETs lie in a row, you'll benefit from moving them into a CFSCRIPT block. Start slow, making simple assignments in your templates. Next, for giggles, move a loop into CFSCRIPT. Finally, go back to some old business logic module and translate it into CFSCRIPT. Not only will your code be improved, you'll have gained invaluable experience in an important language family: scripting.

Performance and CFSCRIPT

In some instances our testing has demonstrated a twofold or better performance improvement with CFSCRIPT compared to an identical block in straight CFML (try the code provided with this article against the same in conventional CF and you'll see a noticeable difference). The larger the block of CFSCRIPT, the more gain.

On CFDJList, regular contributor and answer man Simon Horwith of FigLeaf software provided some insight into this phenomena. Essentially, when ColdFusion receives a request from the server, it distills down templates to find all cases of "<CF". Each of these "CFs" that CFAS finds is passed separately to the CF JIT engine for processing. With CFSCRIPT the block of code is passed in its entirety, eliminating the overhead of instantiating for each tag. Thus you'll find minimal gains for small blocks of code, but greater gains for large blocks (such as a large concatenation routine). This also is why CFAS sometimes throws the entire CFSCRIPT block in error messages.

"As a rule of thumb, anytime more than three consecutive CFSETs lie in a row, you'll benefit from moving them into a CFSCRIPT block. Start slow, making simple assignments in your templates. Next, for giggles, move a loop into CFSCRIPT. Finally, go back to some old business logic module and translate it into CFSCRIPT"

Resources

- Forta, B., et al. (1998). *Advanced ColdFusion 4 Application Development*. MacMillan.
- Allaire's *Developing Web Applications with ColdFusion*, Chapter 20: www-.infoboard.com/packagedoc/cfdocs41/main.htm

About the Author

Christopher Graves is president of RapidCF, a ColdFusion development shop in Connecticut. In his prior "life" he was a Marine Corps officer and graduate of the U.S. Naval Academy.

graves@rapidcf.com

Introducing User Defined Functions

The most requested ColdFusion feature, finally here

BY
BEN
FORTA

Support for user-defined functions is easily the most requested and anticipated enhancement to the ColdFusion Markup Language (and the announcement of this feature at last year's Developer Conference was met with an applause that can only be called thunderous).

In my last column I mentioned in passing that ColdFusion 5 would support user-defined functions (UDFs for short). With ColdFusion 5 just about out the door, I thought I'd take this opportunity to introduce you to UDFs – what they are, and how to write and use them.

Why UDFs?

CFML is made up of two types of instructions:

- *Tags* (like <CFQUERY> and <CFMAIL>)
- *Functions* (like Now(), and StructNew())

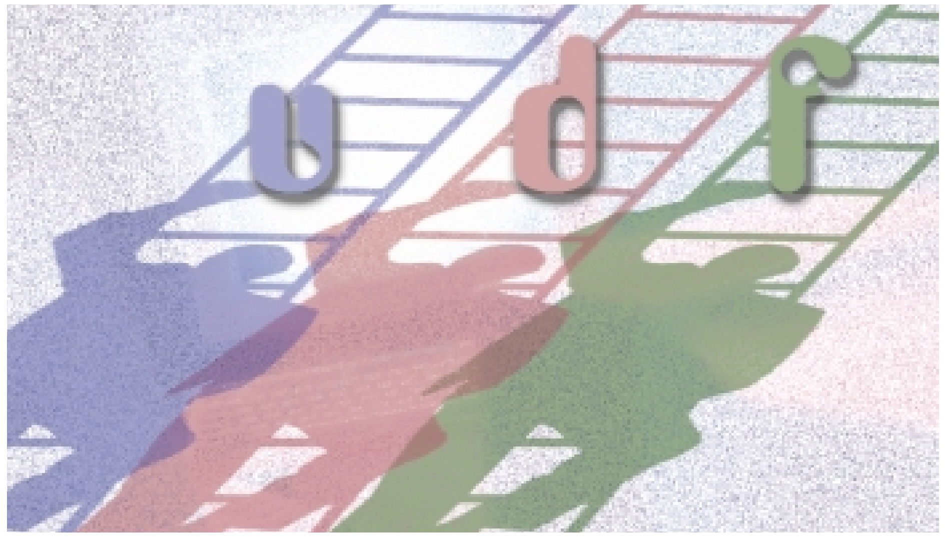
Ever since ColdFusion version 2, developers have had ways to write their own tags, and many have done so. After all, writing your own tags allows you to write maintainable and reusable code.

But we've never had a way to write our own functions. For example, ColdFusion provides a whole range of list manipulation functions (ListFirst(), ListGetAt(), ListSort(), etc.), but there's no function to get the greatest or smallest value in a list (ListMin() and ListMax()). To get these values you'd need to loop through the list doing the comparisons yourself, and there'd be two ways to do it:

- Inline, right in the middle of your code
- By creating a Custom Tag

Neither option is ideal. The former is not reusable; each time you'd want to obtain these values you'd need to copy and paste the block of code. The latter is inefficient (Custom Tags execute far slower than do functions) and clumsy because Custom Tags have no mechanism by which to return data. The right solution is a user-defined function – which is now possible.

It's worth noting that many developers treat UDFs and Custom Tags as if they were interchangeable. The truth is, while many functions could be written as tags (and vice versa), the two have very different purposes. Custom Tags are best suited for operations and processes, complete blocks of functionality. UDFs are best suited for data manipulation, small and finite in scope, returning information to the caller.



The Basics

To demonstrate how UDFs can be used, here's a simple example. Suppose you need to use yesterday's or today's date repeatedly within your code. The CFML DateAdd() function can be used to add or subtract a day to today's date, so to refer to yesterday you could use DateAdd("d", -1, Now()), and to refer to tomorrow you could use DateAdd("d", 1, Now()). And you'd have to repeat these function calls every time they're needed.

Or you could create a couple of UDFs. CFML already has a function named Now() that returns today's date (and time). Why not Yesterday() and Tomorrow() functions to complement it?

Here's the code that creates these two new functions (this, like all code in this column, is fully usable, but you must be running ColdFusion 5):

```
<CFSCRIPT>
// Get yesterday's date
function Yesterday()
{
 return DateAdd("d", -1, Now());
}

// Get tomorrow's date
function Tomorrow()
{
 return DateAdd("d", 1, Now());
}
</CFSCRIPT>
```

As you can see, UDFs are created using CFML scripting between <CFSCRIPT> and </CFSCRIPT> tags.

<CFSCRIPT> provides access to variable assignments, if statement processing, loops, and more, using script-style syntax similar to JavaScript. (For more information on using <CFSCRIPT>, see my column entitled "Stick to the Script" in ***CFDJ***, Vol 2, issue 7).

This <CFSCRIPT> block defines two functions, one named *Yesterday*, the other, *Tomorrow*. Each is preceded by the keyword *function* so that <CFSCRIPT> knows we're about to define a function.

The function code itself is specified between curly braces (the { and } characters). Any valid <CFSCRIPT> code may be used within your UDF.

These functions are simple; all they do is return the result of a DateAdd() function. One adds 1 to Now(), the other subtracts 1 from Now(). That's it. This is about as simple as UDFs get.

The value returned must be preceded by the keyword *return*, and whatever follows return is returned by the function to the caller. You may return functions (as we did here), variables, or any other valid CFML expression.

Incidentally, you'll notice that my comments began with // (instead of being placed between <!--- and --->). <CFSCRIPT> uses the JavaScript syntax for comments (instead of CFMLs) so // is required before comments.

Using User-Defined Functions

Now that two new functions have been defined, how can they be used? Just like any other CFML functions. As long as the UDFs appear in the same CFM page or are included in your code via <CFINCLUDE>, we can use them like this:

```
<CFOUTPUT>
Today is #DateFormat(Now())#<BR>
Yesterday was
#DateFormat(Yesterday())#<BR>
Tomorrow will be
#DateFormat(Tomorrow())#<BR>
</CFOUTPUT>
```

As you can see, the new Yesterday() and Tomorrow() functions are used just like the built-in Now() function, and all three functions are passed to DateFormat(). Once defined, your own UDFs can be used like any other ColdFusion functions – there's no difference whatsoever.

> It's worth noting that many developers treat UDFs and Custom Tags as if they were interchangeable. The truth is, while many functions could be written as tags (and vice versa), the two have very different purposes. Custom Tags are best suited for operations and processes, complete blocks of functionality. UDFs are best suited for data manipulation, small and finite in scope, returning information to the caller.

Accepting Parameters

The functions created above are atypical in that they accept no parameters, something most functions do. For example, UCase() takes the string to be converted, Min() takes the two values to be checked, and DateFormat() takes a date and an optional formatting specification.

UDFs can accept parameters – all you need to do is name the parameter in your function definition. Here's an example of this:

```
<CFSCRIPT>
// Escape a string to make it
// safe for use on WAP devices
function WAPSafe(string)
{
 return Replace(string, "$", "$$",
"ALL");
}
</CFSCRIPT>
```

WAPSafe() is a function that takes a string and escapes it (replacing all dollar signs with double dollar signs) so that it's safe to send to a WAP device (in WML $ is used to prefix variables and thus can't be used in plain text).

The value passed to WAPSafe(), the string to be processed, is made available within the UDF code using the variable name specified – here, "string". The contents of string will be whatever was passed to WAPSafe(), and may be literal text, the results of other functions, or any other expressions. Regardless of what the string is and what it contains, it is accessible with our UDF as such, it can be passed to a Replace() function that simply replaces all occurrences of $ with $$, and the UDF returns the converted string.

To call WAPSafe() I could do something like this:

```
<CFOUTPUT>
#WAPSafe(var)#
</CFOUTPUT>
```

WAPSafe() accepts a single parameter, but UDFs may accept as many (or as few) parameters as needed. The only rule is that when defining multiple parameters, parameter names must be separated by commas.

Using Local Function Variables

We're not done yet. The next thing to learn is how to use variables within your UDFs. Any and all variables and scopes are visible within your UDF, but to prevent possible conflicts, variables created within a UDF are visible only within the UDF itself.

Here's a complete example (the one I demonstrated at the Developer's Conference). It's a function named CapsFirst(), which takes a string and returns a modified version of it with the first letter of each word in uppercase (capitalized) and all other letters in lowercase:

```
<CFSCRIPT>
// Capitalize first char of each word
function CapsFirst(string)
{
 // Define local variables
 VAR outstring="";
 VAR c1=0;
 VAR c2=0;

 // Loop through string
 for (i=1; i LTE Len(string); i=i+1)
 {
  // Is this the first char in
string?
  if (i IS 1)
  {
   // First is always upper case
   c1=UCase(Mid(string, i, 1));
  }
  else
  {
   // Get char and previous char
   c1=Mid(string, i, 1);
   c2=Mid(string, i-1, 1);

   // If Previous char is . or space
   if ((c2 IS ".") OR (c2 IS ' '))
    // then upper case
    c1=UCase(c1);
   else
    // else lower case
    c1=LCase(c1);
  }

  // And put the string back together
  outstring=outstring & c1;
 }

 return outstring;
}
</CFSCRIPT>
```

CapsFirst() takes a single parameter – the string to be processed. Within the UDF three local variables are defined using the "var" keyword. var creates local variables and assigns values to them. This UDF needs two counter variables (c1 and c2) and one variable to contain the converted string as it's being generated (outstring).

Next, the passed string is looped over, one character at a time. The first character in the string is converted to uppercase using the UCase() function, as is any character following a space or a period; any other characters are converted to lowercase using LCase(). As each character is converted, it's appended to the end of outstring, so by the time the loop has completed, outstring contains the entire converted string, which is then returned by the UDF.

As you can see, aside from the use of var to create and initialize variables, UDF code is all straight <CFSCRIPT>.

Variable Parameter Lists

The UDFs we've looked at thus far accept a fixed number of parameters. But many functions take optional parameters. For example, as mentioned earlier, DateFormat() takes two parameters – the first required and the second optional.

ColdFusion UDFs support variable parameter lists by providing access to all passed parameters via an array named *Arguments* (which contains one element for each parameter passed). Because Arguments is a standard ColdFusion array, all the standard array manipulation functions may be used to access it – for example, the number of parameters passed can then be accessed as ArrayLen(Arguments). The first passed parameter would be accessible as Arguments[1] and the fifth as Arguments[5].

Optional parameters shouldn't be declared when the UDF is defined – any parameters named between (and) are required parameters (an error will be thrown if they're not present). All required parameters, however, should be defined (so as to make them required).

Things to Keep in Mind

Now that we've covered the basics of ColdFusion UDFs, here are some important points to keep in mind:

- UDFs are fast, much faster than Custom Tags (yes, it may pay to rewrite some of your Custom Tags as UDFs).
- UDFs can be written only in <CFSCRIPT>, the same <CFSCRIPT> that you can use to write CFML code (with the exception of the var and return statements mentioned above).
- To execute a UDF, it must be in the same CFM file, or included via <CFINCLUDE>, before it's used.
- UDF code can call any function, both CFML functions and UDFs.
- UDF code can't call tags.
- Within a UDF all variables and scopes are visible.
- By default, variables created within a UDF aren't visible outside the UDF.
- Recursion is fully supported within UDFs.
- You may not create a UDF with the same name as a built-in CFML function.

UDFs can be written when needed (right within your CFM page), or in libraries (files containing nothing but lists of UDFs) when they can be included as needed. There's no limit to the number of UDFs that may be written and used, and no limit to how many UDF libraries you include in your page.

Note: Visit www.hrcfug.org/cflib to see (and download) a fast-growing public library of ColdFusion UDFs.

Summary

There you have it – ColdFusion user-defined functions. They're useful, they're fast, they're fun, and (as you've come to expect of ColdFusion) they're really easy to write and even easier to use. Best of all, they're finally here.

So, if you haven't already done so, plan on installing ColdFusion 5 immediately. This addition alone makes the upgrade worthwhile (and there's plenty more, too).

ABOUT THE AUTHOR
Ben Forta is Allaire Corporation's product evangelist for the ColdFusion product line. He is the author of the best-selling ColdFusion 4.0 Web Application Construction Kit *and its sequel,* Advanced ColdFusion 4.0 Development, *as well as* Allaire Spectra E-Business Construction Kit *and* Sams Teach Yourself SQL in 10 Minutes. *He recently released* WAP Development with WML and WMLScript.

BEN@FORTA.COM

Who's Using **ColdFusion?**

You'd be surprised

BY
CHARLES
AREHART

As an instructor I'm often asked by newcomers to ColdFusion, "So, what big companies are using ColdFusion?"

AutoByTel:
(autobytel.com)

Bank of America
(bankofamerica.com)

BMW USA
(Bmwusa.com)

Campbell Soup
(campbellsoup.com)

Casio USA
(www.casio-usa.com)

Catholic University of America
(www.cua.edu)

Computer Warehouse
(computerwarehouse.com)

Crayola
(crayola.com)

Deluxe Check
(deluxe.com)

Dirt Devil
(dirtdevil.com)

FDIC *(fdic.gov)*

Foot Locker
(footlocker.com)

Haagen-Dazs
(haagen-dazs.com)

Half.com
(half.com)

Hertz
(hertz.com)

Hofstra University
(www.hofstra.edu)

Mail Boxes Etc.
(mbe.com)

Logitech
(Logitech.com)

MacMillan Computer Publishing
(mcp.com)

Motorola
(motorola.com)

NetGrocer
(netgrocer.com)

PeachTree Software
(peachtree.com)

Pepperidge Farm
(pepperidgefarm.com) **–no "s"**

Sams Publishing
(www.sampublishing.com)

San Diego Chargers
(chargers.com)

Simon and Schuster
(simonsays.com)

SmartMoney — Wall St. Journal
(smartmoney.com)

Swiss Army Knife
(swissarmy.com)

Symantec
(symantec.com)

TAG Heuer Watches
(www.tagheuer.com)

The McLaughlin Group
(www.mclaughlin.com)

US Bank
(usbank.com)

USDA Graduate School
(grad.usda.gov)

US Mint
(usmint.gov)

VoiceStream
(voicestream.com)

Walter Reed Army Medical Center
(www.wramc.amedd.army)mil)

I actually look forward to the question, because I enjoy seeing their surprise when I list some of them. I don't have a definitive list, but it's enough to impress most who ask.

How Can You Tell It's a CF Site?

Have you ever noticed (or wondered) if a favorite site uses CF? Actually, you might have a bit of trouble trying to tell. Why? Well, perhaps they use frames. If so, you won't see any .cfm URLs at the top of the page, since such a site nearly always shows just the main site URL in the browser address window.

Still you can get around that problem by rolling your mouse over a hyperlink to see the page it would go to, shown at the bottom of your browser in the status bar. See if you find any CFM file links. Another resource: if you look at the source code for a form submission page, the form's ACTION attribute may refer to a CF file.

Even then many high-volume sites are now using static HTML pages for much of their more frequently visited static pages (though they might get equally good performance and still benefit from CFINCLUDE, etc., if they used CFCACHE, but that's for another article).

Hertz's site is an example of this. Most links on the front page are to .htm files, but there are some to .cfm files if you look closely. Same too for VoiceStream's site. You have to dig even deeper into Motorola's site. Bank of America's front page shows a few cgi files, but keep looking. They're all using CF if they're on my list at left.

Is Anyone Keeping a List of CF Sites?

Actually, yes. Ben Forta keeps a list of CF sites, to which you may freely add any you find, at www.forta.com/cf/using/. It's quite an amazing list, broken down into nearly 30 categories.

One problem I find is that it doesn't break out the "big name" sites from mom and pop or hobbyist ones. If you're trying to impress your board or IS department with a "who's who" list, you need to know the big players (CattleSale.com really is a lovely and well-done site for its market, but doesn't have quite the prestige of BMW USA or TAG Heuer).

Also, perusing the list is a little challenging because some sites have listed themselves multiple times to gain extra coverage. For instance, ForTheFarm.com is listed in numerous categories such as automotive, careers, education, health, and recreation. That may make sense to them but with many sites cross-posting themselves this way, it becomes hard to separate the wheat from the chaff, if you'll pardon the pun.

In essence, I've done the hard work for you here, and also added a few not listed at the time of this writing on Ben's otherwise deep and much-appreciated resource (Bank of America, Campbell Soup, FDIC, Hertz, Logitech, Motorola, Swiss Army, Symantec, U.S. Mint, and VoiceStream).

Allaire also offers a small list at www.allaire.com/products/coldfusion/CustomerSuccesses/eCommerce/, as well as a few case studies elsewhere on their site.

The List, 03/2001

This is by no means an exhaustive list. I've concentrated only on names that I think most people (in the U.S.) would recognize. Even then, I also tried to limit it to sites that demonstrate large-scale use (sure, some sites may be about familiar subjects like Tolkien-Online.com, but it's probably not a highly trafficked site. No offense to fans and the site designers!). I also haven't listed some that are identified as using CF (on Ben's site or elsewhere).

If I couldn't quickly find a Web site's use of CF (even if someone said the site uses it), and instead saw the site using some other approach for the pages that should be dynamic, I dropped it from my list, figuring they'd probably converted to another server technology since being identified. I've tried to focus on sites that are committed to ColdFusion in a big way.

Also, while there are some organizations using CF for internal sites, I've chosen not to list them. I figure if we can't verify that they're using it and observe it for ourselves, then it's best not to list them.

As such, it's an arbitrary list, and I may have left off something that's quite noteworthy. I welcome feedback. I've also confirmed that each site is still running under CF. (Some noteworthy ones from the past, such as ToysRUs, are no longer CF sites.)

Finally, I've also dropped the http://www from each, since we know it's expected, except in cases where visiting the site without it would fail (such as www.casio-usa.com and www.mclaughlin.com: yes, *that* McLaughlin Group).

In a category of sites not listed are those using JSP: AT&T Wireless and FunCoLand. I should clarify. These may indeed be Allaire customers running JRun, but there's no way to know that, and anyway this is "Who's using CF." If they're still using CF on some part of their site, it wasn't apparent.

My Favorite to Share

Finally, I'll close with one of my favorite sites to share with students and others asking "Who uses CF?" It's often asked in a defensive way, asserting they've not heard of any, or wondering why a site wouldn't use some other technology (sad, really, because it seems that the question of CF's productivity and enterprise scalability has been answered by now).

So I love to point out a pair of sites (related to each other) that you'd think, if the site owners wanted to, could easily choose "another technology" for their Web application server. But they didn't. Check out sqlmag.com (*SQL Server Magazine*) and its sister publication win2000mag.net (*Windows 2000 Magazine*). Hmm. Using CF instead of ASP! That always sparks some chortles.

Oh, and while you may not be surprised to learn that our very own ColdFusionJournal.com is a CF site, you may be intrigued to learn that the same goes for its sister publications: ***Java Developer's Journal***, ***XML-Journal***, and ***Wireless Business & Technology***.

I hope this list proves useful to you. Please do send me (and Ben) any significant sites to add.

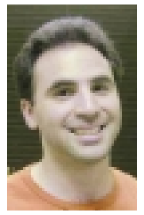
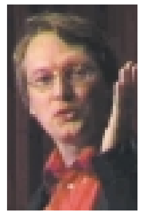
ABOUT THE AUTHOR
Charles Arehart is a certified Allaire trainer/developer and CTO of SysteManage, an Allaire partner. He contributes to several CF resources, provides on-site coaching and consultation, and is a frequent speaker at user groups throughout the country.

CAREHART@SYSTEMANAGE.COM

Upsize from MSAccess **to SQL Server**

Part 2 of 2

It's a big step, but the payoffs are well worth it

BY **ERON COHEN** AND **MICHAEL SMITH**

In March (***CFDJ***, Vol.3, issue 5) we covered reasons to upsize your Access database to SQL Server and the tools you'll use to do it. Here we explain in detail how to copy data to SQL Server and troubleshoot your upsizing project.

The IMPORT AND EXPORT data wizard (aka the DTS Wizard – Data Transformation Services) certainly helps get you going with building tables in SQL Server. It takes an Access database (as well as many other database types) to import its table structures and data to a Microsoft SQL Server. Since it uses a wizard format, it's simple to use as long as you know what data you want to transform. To start the wizard choose START > MSDE > IMPORT AND EXPORT DATA or START > Microsoft SQL Server 7.0 > IMPORT AND EXPORT DATA.

On the first data-gathering screen (see Figure 1) you'll be prompted for information about the data source. Choose Microsoft Access from the drop-down menu for the data source (or whatever is appropriate for the database you actually intend to import to SQL Server. You'll find quite a long list of possibilities). Then select the appropriate .MDB file and type your username and password if necessary. When you're finished, click the NEXT > button to continue on to set up the destination database.

On this page you'll most likely leave the destination database on the default Microsoft OLE Provider for SQL Server. This is the selection you'd normally use to import the MS Access database you chose on the previous page of the wizard into a SQL Server database. Next you'll need to provide the name of the server and the security information. Typically this information is provided to you by your system administrator or by the Web-hosting company that houses your SQL Server. Once this information is correct, you'll be able to select the actual database you wish to import the tables into.

If you haven't already created a database to house your new data, choose <new> from the database drop-down menu. Avoid making the typical mistake of importing the tables into the MASTER database or one of the example databases Microsoft provides. You'll want to start off fresh if you can. A dialog box will open (see Figure 2), prompting you for the name and size of your database. Type the name and accept the defaults. When you press OK, the new database will be created. After this is complete, press Next again. You'll be shown a lovely pictorial of the action that is about to be taken – namely, importing the MS Access database into the SQL Server database. Choose the "Copy Tables from the source database" radio button and push Next.

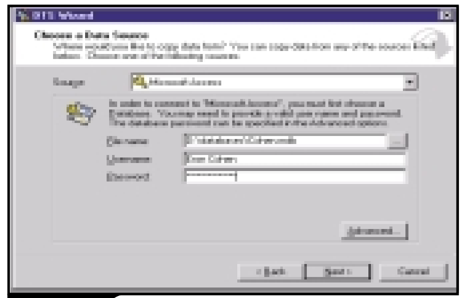


FIGURE 1: Pick your Access database.

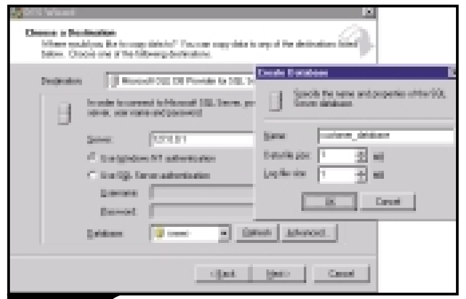


FIGURE 2: Create a new SQL Server database.

Transformation

Now you'll be given a list of the tables in your database. Notice that you won't be given the names of any queries or reports. Choose the individual tables you wish to import or just press the SELECT ALL button. As you select the tables, a field activates that will allow you to change the name of the destination table. Do so if necessary. There's also a Transform button that will allow you to make changes to the table's structure as it is being brought into the new SQL Server table. If you transform anything, you'll want to be aware of the new data types available in SQL Server but not in MS Access. There are some confusing things. For instance, a Text field in Access is similar to an NVARCHAR or VARCHAR data type in SQL Server. On the other hand, a Text field in SQL Server is like a memo field in MS Access. Please check your SQL Server references for more specifics on these data types and their exact characteristics.

On the last page of the wizard you'll be able to select when the import should be executed. You'll probably want to choose "Run Immediately" and just choose NEXT>. Finally, press FINISH and watch the action. If anything goes wrong on the data import, the wizard will let you know what the problems were. You'll need to fix them in the Transformations and then retry the tables that failed. Referential integrity can complicate the process of retrying individual tables. If life becomes too complicated, consider dropping the referential constraints for the tables in question and re-creating the constraints after the upgrade. When you're done we suggest you review the table structure in the design table view. You'll probably want to at least add the primary key constraint by clicking on the key symbol to your key field and look at which fields allow nulls. You might also check the Identity box if your key field was AutoNumber in Access and pick some fields to index. It's a good idea to look at the data by right-clicking on the table and doing the action "open all records" to make sure nothing went wrong.

Finding and Troubleshooting Upgrade Problems

After you've finished importing your data, you'll need to test your queries thoroughly. There are a few differences between MS Access's version of SQL and the implementation that SQL Server uses.

Field and Table Names

- Be careful what you name your fields and tables in MS Access: use only letters A–Z, digits 0–9, and underscores. No spaces or other funny characters are allowed unless you want to "delimit your identifiers with square brackets []" (you don't).
- Avoid reserved words for field names or you'll get errors when you use them in SQL queries – for example, DESC, AND, OR.

Field Types

- Logical fields in Access can be True, False, or Null – the default equivalent bit field in SQL Server can be only 0 or 1. For this reason you may prefer to transform logical fields to integers instead. Also, if you have any queries that compare a bit field to TRUE or –1, you'll have problems.
- AutoNumber fields are equivalent in SQL Server to an Identity field (a checkbox that you turn on when you set up a new field). Be aware that an Identity field in SQL Server 6.5 had some problems and can be replaced by a long and your own next ID code if you prefer.
- Memo fields are converted to TEXT or NTEXT fields.
- Access will auto trim text for fields that are too long, while SQL Server will give an error if you try to insert too long a piece of text. Check the length of the fields and use the left function to trim to correct length; for example, if the firstname field is varchar 30, in your insert statement put VALUES('#left(firstname,30)#') instead of VALUES('#firstname#').

Access	SQL Server	Notes
SELECT DISTINCTROW	SELECT DISTINCT	No duplicate rows
LIKE "cat*"	LIKE "%cat%"	Wildcard is * or %
DELETE *FROM tablename	DELETE *FROM tablename	Leave out the * when you are deleting rows from SQL Server
Now()	GetDate()	Current date/time
Instr()	CharIndex()	Searching a string
Ucase()	Upper()	Upper case
Lcase()	Lower()	Lower case
Trim()	RTrim(ltrim())	Removing extra spaces
int(x)	convert(int, x)	Convert to integer
A&B	A&B	String concatenation operator

TABLE 1 SQL and function differences

SQL and Function Differences

The syntax for some SQL statements, functions, and operators is different between Access and SQL Server (see Table 1).

Changing SQL and Access Queries

If you have ColdFusion Studio, use the extended search to find where these problems may occur and deal with them. For instance, the above-mentioned problem with DELETE * could be dealt with by simply doing an extended search for DELETE * and replacing it with DELETE. However, be careful with function names that are in both Access SQL and ColdFusion, such as ucase() and #ucase()#, because you only want to change the Access SQL function to upper()! We suggest you carefully watch your ColdFusion application server log after switching over to find any issues you may have missed.

Access Queries become Views in SQL Server. To convert a query to a view, cut and paste from the Access SQL view into SQL Server's View creation designer. Just be aware that, unlike Access queries, they can't use an ORDER BY clause – you'll need to do the ordering when you run the view SQL in ColdFusion. The view designer can also be useful when you want to test some SQL from ColdFusion to find out why it has a syntax error. Alternatively, you may choose to create a stored procedure for a query, especially if you need to parameterize the WHERE clause. Stored procedures also let you code multiple queries and procedural code such as IF statements inside SQL Server.

Summary

Upgrading to SQL Server may sound intimidating, but Microsoft has provided tools to ease your way. Many SQL Server interfaces either look similar to Access or can be administered using Access as a front end. Upgrading to SQL Server is a big step, but the payoffs are well worth it for ColdFusion programmers. Plus, the MSDE version of SQL Server is free with Office! As you get used to SQL Server, you'll find it superior to Microsoft Access in many ways, and you'll have an excellent addition to your résumé!

Resources

1. *SQL Server data types:* www.pin-pub.com/foxtalk/ft9902i/ft99b20.html
2. *SQL tutorial:* http://w3.one.net/~jhoffman/sqltut.htm
3. *Microsoft's SQL Server site:* www.microsoft.com/sql/
4. *Whitepaper on upsizing:* www.microsoft.com/accessdev/prod-info/upsize97.htm
5. *Creating and deploying with the MSDE:* www.microsoft.com/sql/techinfo/msde.htm
6. *Reasons to upsize:* www.sys-con.com/coldfusion/archives/0102/schulze/index.html
7. Sams Teach Yourself Sql in 10 Minutes, *by Ben Forta:* www.amazon.com/exec/obidos/ASIN/0672316641/

ABOUT THE AUTHORS

Eron Cohen is a ColdFusion programmer, MDCFUG speaker, and author.

Michael Smith is president of TeraTech (www.teratech.com/), an 11-year-old Rockville, Maryland–based consulting company that specializes in ColdFusion, database, and Visual Basic development. Michael runs the MDCFUG and CFUN-2k conference that attracted more than 750 participants.

ERON_COHEN@YAHOO.COM

MICHAEL@TERATECH.COM

Spectra Calamari

Squid's Web proxy cache

BY DAVID ANDERSON

Allaire's Spectra provides a terrific solution to the ever-growing problem of content management. It boasts a number of other features as well, but its content management features are what sold us.

The solution comes at a price, though, both in monetary and resource terms. Even though it's less expensive than its competitors, it's not cheap. Allaire's white paper on server capacity recommends at least two dual-processor front-end servers, with the appropriate licensing, of course.

Working on a very tight budget forces me to find alternative solutions to daily challenges. When I was faced with the sluggish performance of my Spectra server under full load, I immediately began to cast about for any solution. I quickly stumbled across Squid (www.squid-cache.org), a "Web proxy cache." A quick scan through the documentation revealed that it might be what I needed.

What is a Web proxy cache? Simply put, it surfs the Web and saves the results in a local file. Whenever a browser asks for a file that it has, the file is served from Squid rather than the Web server. Squid is designed to serve as a proxy for the Internet, caching the most popular pages requested by its users, usually within a company. Properly configured, it can reduce the traffic on your company's Internet connections.

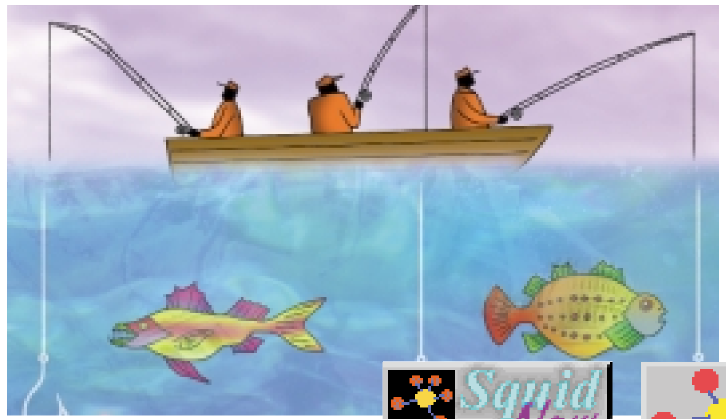
But Squid can also work the other way around, proxying all requests for a specific server no matter where they come from. This is called *httpd-accelerator mode* and it's how I use Squid to make my Spectra server perform better.

Why Cache?

Caching has one primary purpose: to lighten the load on your Web server. Squid has other useful features such as the ability to map multiple servers to one URL, but its raison d'etre in httpd-accelerator mode is to increase the apparent speed of your Web server.

As an accelerator, Squid hits your Web server and caches the results. If the machine you're using has lots of RAM, it'll be stored in memory. Eventually, a copy of the page will be written to the hard disk. When a client hits your Web server, Squid serves the page rather than your server. And like any other server Squid logs the hit in a file.

Using an httpd accelerator such as Squid has other advantages. Hardly a day goes by without someone announcing a new security hole. As a result most organizations have implemented firewalls. The problem, though, is that Web traffic needs to get to your Web server. Thus you have to expose a system to the Internet and every script kiddie out there. With Squid you can place your Web server inside your firewall. It can also actively protect your system against many known attacks simply by listening on all ports.



Why Not Cache?

Caching is not a solution for everyone. Spectra has extensive personalization features that require a live connection between the browser and the server. Caching by its very nature disrupts that connection. Since I'm using Spectra primarily as a content management system, I'd like the files served as fast as possible.

If you have a mix of files, caching will allow Squid to serve the static files, letting your server focus on handling the "live" requests.

How It's Done

Setting up a Squid server can be quite an adventure for the uninitiated. It's designed as a proxy cache; the documentation on the httpd-accelerator mode is relatively sparse. Before you begin you'll need a UNIX box of some sort to run Squid. I've used RedHat 6.2 Linux successfully. Squid is developed on Compaq Tru64 UNIX (formerly Dec OSF/1) and is ported to a number of other Unixen.

Ideally, you'll want to dedicate a box to Squid. I'd suggest a custom install of RedHat 6.2 or 7.0; custom installs as of 6.2 don't set up inetd. This makes your box moderately secure by default since it won't launch all those default servers (telnetd, ftpd, finger, etc). When you do the install, make sure you include the development packages as you'll need the include files and libraries.

Once you've got a box set up, download a copy of Squid. You'll want to get the latest stable release (2.3 as of this writing), since they're meant for production environments. Older releases are maintained for compatibility purposes, so there's no real reason to download them. Development releases are for the hardiest experimenters.

Untar the file; the Squid docs recommend the /usr/local/squid/src directory. If you don't have root access to the server, you'll probably want to untar Squid into another directory.

Compilation is relatively straightforward – simply cd to the appropriate directory and type *./configure*. When it's done, you'll want to *make*, then *make install*. Squid should now be built and installed on your server.

Now for the fun part. By default Squid is configured to be a proxy cache. Save the existing squid.conf file and open a new one in your favorite editor. The most basic squid.conf file is shown in Listing 1.

Squid uses access control lists (ACLs) to determine who gets what. In the example config file, the ACLs grant everyone access to my Web server, which is restricted to port 80, a security measure that tightens up my server. This prevents people from using our Web server to generate spam on port 25, for instance.

The ACLs also grant access on port 3128 on the Squid server so I can delete individual files using the client program. The last ACL rule ("http_access deny all") prevents any kind of connection that didn't fall into one of the previous rules.

There are a few other important settings near the bottom of the listing. Since I want to be able to track usage of content objects, I've set strip_query_terms to off. This ensures that everything after the question mark is logged. Squid has its own format for logging requests that records more information than I really need and is slightly nonstandard. Setting emulate_httpd_log makes the access.log file look like a standard Web log.

Finally, I'm not interested in what happens to the cache store, so I turn that log off.

Expanding the Config File

Depending on how you've laid files out on your server, you'll probably want to add some exceptions to the no_cache ACL. The ACL I've defined uses regular expressions (regex) to match the listed patterns in the URL (urlpath). This minimal ACL won't cache anything that has cgi-bin or errors in the path. You can add more exceptions at the end of the ACL, and add more ACLs for specific file extensions, destination servers, time of day, and more.

You can also prevent users from seeing specific directories, files, and more on your servers with a simple deny ACL. For instance, I regularly use a test directory to work on projects. I also don't want folks getting at my cfide and cfdocs directories. I don't want others to see the contents of the directory so I've added an ACL that looks like the following:

```
acl noSeeUm urlpath_regex test cfide
cfdocs
http_access deny noSeeUm
```

The Squid documentation contains lots of helpful hints about ACLs and are well worth reading. Always make sure you have a backup of a working squid.conf before you crash your server.

"But wait, we're not done yet. At this point, the cache-control header will be issued for all files in all directories. At the very least you should remove the header from the cfide, allaire, and any other directories you don't want cached in the browser. I was puzzled by bizarre behavior from the server before it dawned on me that I was constantly getting cached pages where I wasn't expecting them"

Running Squid

Once you have your squid.conf file ready you can start Squid. Before you do, make sure the Squid user owns the directories and files it resides in. Simply type *chown -R nobody:nobody squid* in the /usr/local directory, then create the cache directories. Assuming you're in the /usr/local/squid directory, type *bin/squid -z* at the prompt. At the next prompt type *bin/squid -NCd1* and the cache server should start. You should see "Ready to serve requests" after several messages from the server. Try connecting to your Squid server with a browser; you should see a series of messages on the console.

To start Squid as a daemon, simply type *bin/squid*. To start Squid automatically when your server boots, add */usr/local/squid/bin/squid* to the end of your rc.local file. Squid has a built-in restart feature so you don't need to put it in inittab.

You don't need to kill the Squid process to implement changes in the squid.conf file. Simply type *bin/squid -k reconfigure* to tell it to reread its config file.

At this point you'll want to test your Squid server so try browsing your site through it. To make sure it's actually caching your pages, comment out the cache_store_log directive in the squid.conf file and restart the server. Look for SWAPOUT entries in the store.log file. There'll be a number after the SWAPOUT; if it's FFFFFFFF, there's a problem. You'll probably see this for your ColdFusion files.

It turns out the problem is how your Web server handles files. The cure is relatively painless – simply issue a cache-control header directive. With IIS you can set the server to issue cache-control directives at the site, directory, and file levels. Bring up the properties for the Web site in the management console and click on the Document Headers tab. You'll want to add a new header called *cache-control*. The length of time pages are cached for is specified in seconds; I use a value of max-age=3600 as a default.

But wait, we're not done yet. At this point, the cache-control header will be issued for all files in all directories. At the very least you should remove the header from the cfide, allaire, and any other directories you don't want cached in the browser. I was puzzled by bizarre behavior from the server before it dawned on me that I was constantly getting cached pages where I wasn't expecting them.

If you don't want to issue the cache-control header site-wide, you can do the same thing with the CFHEADER tag. You'll need to include it in every page you want Squid to cache.

Tuning Squid

Depending on how busy your site is, you'll probably need to tune the Squid server. The first problem you'll experience is a lack of file descriptors. Most Linux systems are designed for average users and need some tweaking to work well as servers. The two salient parameters you need to change are /proc/sys/fs/file-max and /proc/sys/fs/inode-max. These control the maximum number of active file and inode descriptors available, which you can change in the following way:

```
echo 8192 > /proc/sys/fs/file-max
echo 32768 > /proc/sys/fs/inode-max
```

In most instances these settings should do the trick. To make them "permanent," put them in your rc.local before the line that starts Squid.

Since Squid likes to use all the RAM it can, try to reduce the number of other processes running on the server. I set up my Squid server as a custom RedHat install and didn't install *X* or any other noncritical services. I have 512MB on my machine and it works rather well. Naturally, Squid will use any RAM you're willing to throw at it.

Daily Maintenance

Squid is similar to any other Web server: it generates log files that grow with usage. You'll probably want to rotate the log files on a regular basis. With RedHat Linux you can put the log-rotation command in several places. The traditional place is the root crontab. Decide how often you want the logs rotated and put */usr/local/squid/bin/squid -krotate* in the crontab.

As an alternative, you can put the same line at the end of the /etc/logrotate.conf file or in a special file in one of the cron.* directories under /etc.

If you're curious about the inner workings of your Squid server, set up the Cache Manager. It's a simple cgi that prints out loads of interesting (and sometimes mind-boggling) information. You'll need to install and set up a conventional Web server such as Apache and map /usr/local/squid/bin/ as a cgi directory.

Eventually you'll come to a point where you'll want to flush a file from the cache. Assuming you've included the ACLs to accept purge commands, you can use the client program (in the /usr/local/squid/bin directory) to delete individual files. You need to know the full URL of the file you want to flush. If I want to flush the home page, for instance, I'd type *client -m PURGE http://www.plattsburgh.edu/* at the command line. Squid will return a 200 if it found the page and 404 if it didn't.

The client program is useful for clearing individual files from the cache. Deleting the entire cache is another issue entirely. If you want to clear the cache without disrupting the server, edit your squid.conf and change the cache_dir directive (if you don't have one, simply add it). The new cache_dir should point to a directory that's not the one currently in use. If the new directory doesn't have the Squid cache structure, create it by typing *squid -z*. Next, shut down the existing Squid with *squid -k shutdown*, then restart it with *squid*. Now you can delete the old cache directory with *rm -Rf dirname*.

Squid and Spectra

Allaire recommends a minimum of two front-end servers for Spectra. The goal is to balance the load of incoming requests between the servers. At approximately $20,000/server, it's a relatively expensive solution. In our environment we needed to find another solution. Squid gives us the benefit of a "production" server (Squid) and a "development" server (Spectra) at a much lower cost.

"If you're familiar with Spectra you know that many of its objects can be displayed independent of a fixed page via the invoke.cfm script. Since we're not using any data that changes in less than a few hours, I'm perfectly comfortable with letting Squid cache those requests. If your ColdFusion pages are more dynamic, you might want to add ACLs to the squid.conf file to exclude the dynamic code"

If you're familiar with Spectra you know that many of its objects can be displayed independent of a fixed page via the invoke.cfm script. Since we're not using any data that changes in less than a few hours, I'm perfectly comfortable with letting Squid cache those requests. If your ColdFusion pages are more dynamic, you might want to add ACLs to the squid.conf file to exclude the dynamic code.

The other major consideration is just how many of the Spectra features you plan to use. Using Squid as your primary server means that the logging features built into Spectra won't accurately reflect Web usage. On the other hand, since the Squid file is in a standard format, you could potentially write code to import it into the Spectra logs. Personalization is similarly affected since all requests appear to come from the same client.

The single most challenging part of getting our site to work with Spectra was browser sniffing. I had implemented minimal sniffing in my application.cfm file. Because the results were cached by Squid, I had to implement sniffing in JavaScript instead.

All in all the advantages outweighed the disadvantages for us. Our site is mostly static, so in essence we gained an additional Spectra server. Even though we enabled page caching on the Spectra server, each request had to go through ColdFusion. By moving the majority of those requests to another server, I freed up our Spectra server for our content developers.

Code appears on next page

ABOUT THE AUTHOR
David Anderson is the Web site administrator at Plattsburgh State University. He's been working with ColdFusion for several years and Spectra for over a year.

DAVID.ANDERSON@PLATTSBURGH.EDU

Listing 1

```
# http_port defines where Squid should listen
http_port 80 3128

# Defines an ACL for patterns that should not be cached
acl QUERY urlpath_regex cgi-bin errors
no_cache deny QUERY

# And an ACL for purging entries from the cache
acl PURGE method purge
acl localhost src 127.0.0.1

# These ACLs define the main behavior of Squid
# Replace yourhost.name.here with the actual webserver name
httpd_accel_host yourhost.name.here

# And the port it accepts connections on
httpd_accel_port 80

# This will accept connections from all computers
acl all src 0/0

# Define valid destination address and port
acl acceleratedHost dst yourhost.name.here
acl acceleratedPort port 80

# And define a port for the management tools
acl managerPort port 3128

# Allow connections to the right machine and port number
http_access allow acceleratedHost acceleratedPort

# Allow PURGE requests from this machine on the manager port
http_access allow PURGE localhost managerPort

# Deny all other requests from anywhere on the manager port
http_access deny all managerPort

# Finally, deny all requests that don't meet one of the
above conditions
http_access deny all

# these need to be set to accounts on the system. nobody
is traditionally
# used for webservers
cache_effective_user nobody
cache_effective_group nobody

# logs the full url instead of stripping everything after
the ? in a request
strip_query_terms off

# use standard web logs instead of Squid-style logs
emulate_httpd_log on

# don't log info about the cache store
cache_store_log none
```

CODE LISTING

The code listing for this article can also be located at www.ColdFusionJournal.com

SYS-CON INTERACTIVE

The World's Leading ColdFusion Resource

BPA INTERNATIONAL

ColdFusion Developer's Journal .com

SUBSCRIBE | COLDFUSION FORUM | DIGITAL EDITION | SOURCE CODE
ADVERTISING | PRODUCT REVIEWS | SEARCH

BUYER'S GUIDE
READER SERVICE
FEATURE STORIES
EDITORIAL COLUMNS
DEVELOPER STORE
COLDFUSION JOBS
AUTHOR GUIDELINES
EDITORIAL BOARD
CUSTOMER SERVICE

Search CFDJ

May 2001

What's Online This Month...

CFDJ Online

For up-to-the-minute news, events, and developments in the ColdFusion industry, visit www.sys-con.com/coldfusion. Check in often to read about the latest developments, new product releases, software updates, trends, and more.

Product Reviews

Considering a product upgrade? Want to know the ins and outs of a new product before you purchase it? Make sure you read our in-depth product reviews.

Our writers get behind the hype and give you the facts. All products are tested by experts and leaders in the information technology industry.

ColdFusion Archives

With your FREE access to our *CFDJ* archives, you can find every article published starting with our premier issue. Didn't get a copy of ***ColdFusion Developer's Journal*** last month? Not to worry, just visit us online.

Search ColdFusion Jobs

ColdFusion Developer's Journal is pleased to offer our employment portal for IT professionals. Get direct access to the best companies in the nation. If you're an IT pro and are curious about the job market, demand privacy, and don't want to waste time, you've found the right site.

Simply type in the keyword, job title, and location and get instant results. You can search by salary, company, or industry.

Need more help? Our experts can assist you with retirement planning, putting together a résumé, immigration issues, and more.

ColdFusion Forum

Join ColdFusion Forum, the new ColdFusion mailing list community. Join other IT professionals and industry gurus for ColdFusion discussions, FAQs, and more. Voice your opinions and assessments on topical issues, or hear what others have to say.

Join ColdFusionList now to monitor the pulse of the ColdFusion industry.

ColdFusion Buyer's Guide

Visit CFBuyer's Guide, your online source for ColdFusion-related software and products, including books, consulting services, custom tags, testing tools, Web hosting, and much more. Check out the Buyer's Guide for the best ColdFusion technologies available today.

Ben Forta's ColdFusion Tip-of-the-Day

Click here for ColdFusion tips, links, tags, and resources from Allaire's CF evangelist. A new tip every day!

CFDJnews

Macromedia Embraces InCert Software's Application

(San Jose, CA) – InCert Software Corporation has announced that Macromedia has chosen TraceBack for Windows to maximize application availability for the ColdFusion Application Server.

TraceBack provides advanced warning and facilitates the rapid diagnosis and recovery from application faults that occur in live production systems. Using patented instrumentation technology to capture detailed program execution and system information, it links failures back to the code that produced them.

A 30-day trial version of TraceBack can be downloaded free at www.incert.com.

Learning Tree Releases New Course

(Reston, VA) – Learning Tree International, Inc., has introduced a new Hands-On IT course, ColdFusion Web Application Development.

During the four-day course, participants learn how to create interactive Web pages that are fully integrated with existing databases; manipulate, retrieve, and update database information; add search capabilities to a Web site; store visitor information with cookies and session variables; validate and process user input; and exchange data using XML technology. Participants also learn to use custom tags to create reusable code and other productivity-boosting techniques.

For further information and course dates visit www.learningtree.com.

SYS-CON Announces ColdFusion Edge 2001

A Fast Track for ColdFusion Certification at International Web Services Conference & Expo East

(Montvale, NJ) – **SYS-CON Media** (www.sys-con.com) has announced the expansion of its Web Services Edge 2001/East: International Web Services Conference & Expo technical conference and exhibition program to be held September 24–26 in New York City (www.sys-con.com/webservicesedge). The conference program, produced and presented by **SYS-CON Events**, will include the ColdFusion Edge 2001 Certification Track.

"The ColdFusion Edge 2001 Fast Track for ColdFusion Certification will include supersessions presented by the biggest names in ColdFusion technology, such as Ben Forta," said Robert Diamond, editor-in-chief of ***ColdFusion Developer's Journal***. "We're bringing together the best and brightest faculty in the ColdFusion world to prepare developers for the ColdFusion Certification Exam."

Macromedia president Kevin Lynch will give one of the keynote speeches at Web Services Edge 2001/East, and Jeremy Allaire, CTO of Macromedia and the creator of ColdFusion, will open ColdFusion Edge 2001 with his keynote on Monday, September 24.

Daily updates and news about the ColdFusion Edge 2001 Certification Track can be found at www.sys-con.com/coldfusionedge.

ADVERTISER INDEX

ADVERTISER	URL	PH	PG
ABLECOMMERCE	WWW.ABLECOMMERCE.COM	360.253.4142	2
ACTIVEPDF	WWW.ACTIVEPDF.COM	888.389.1653	25
ALLAIRE CERTIFIED PROGRAM	WWW.VUE.COM/ALLAIRE	877.460.8679	29
ALLAIRE CORPORATION	WWW.ALLAIRE.COM/DEVELOPER.TRAINING.CFM	888.939.2545	17
ALLAIRE CORPORATION	WWW.ALLAIRE.COM/USERGROUPS		23
AMERICAN CYBERNETICS	WWW.MULTIEDIT.COM	800.899.0100	27
CAREER OPPORTUNITIES	CAREERS.SYS-CON.COM	201.802.3028	56
CFDYNAMICS	WWW.CFDYNAMICS.COM	800.422.7957	51
CFXHOSTING	WWW.CFXHOSTING.COM	866.CFX-HOST	47
CFXTRAS	WWW.CFXTRAS.COM	704.408.6186	39
CODECHARGE	WWW.CODECHARGE.COM	650.754.9810	4
CONCEPTWARE AG	WWW.CONCEPTWARE.COM/EYE	011.781.275.6171	49
CORDA TECHNOLOGIES	WWW.CORDA.COM	801.805.9400	9
ECONOMIST.COM	WWW.ECONOMIST.COM	44(0)20 7436 4424	57
EMPRIX	WWW.EMPRIX.COM	781.993.8500	7
EVOLUTION B	WWW.EVOLUTIONB.COM/SYNERGYFORFREE	877.622.7551	59
HOSTCENTRIC	WWW.HOSTCENTRIC.COM/CFDJ	888.932.9376	15
HOSTMYSITE.COM	WWW.HOSTMYSITE.COM	877.215.HOST	38
INTERMEDIA.NET	WWW.INTERMEDIA.NET	800.379.7729	60
JAVAEDGE CALL FOR PAPERS	WWW.SYS-CON.COM/JAVAEDGE	201.802.3069	45
JAVAEDGE CONFERENCE & EXPO	WWW.SYS-CON.COM/JAVAEDGE	201.802.3069	32-33
JDJ STORE	WWW.JDJSTORE.COM	888.303.JAVA	46
PACIFIC ONLINE	WWW.PACONLINE.NET	877.503.9870	39
PAPERTHIN	WWW.PAPERTHIN.COM	800.940.3087	21
SYS-CON CUSTOM MEDIA	WWW.SYS-CON.COM	925.244.9109	43
TECHIEGOLD	WWW.TECHIEGOLD.COM	617.267.4700	56
TOPHOSTS.COM	WWW.TOPHOSTS.COM	306.546.3778	3
WEB SERVICES EDGE 2001	WWW.WEBSERVICESEDGE2001.COM	617.267.4700	56
WIRELESS BUSINESS & TECH	WWW.WIRELESS-MAGAZINE.COM.	201.802.3069	53
WOODBOURNE SOLUTIONS	WWW.WOODBOURNESOLUTIONS.COM	877.668.8575	51
XMLEDGE CONFERENCE & EXPO	WWW.SYS-CON.COM/XMLEDGE	201.802.3069	40-41

Next Month...

Here's a sneak peek...

Handling Credit Card Transactions Using CyberCash by David Keener

Setting Up Shop: Adding a Cart with ColdFusion by Kevin Schmidt

ColdFusion & Oracle: Putting Oracle SQL to Work in Your ColdFusion Applications Part 1 by Kailasnath Awati and Mario Techera

Caching Redundant Dynamic Content with Custom Tags: Creating Partially Static Pages by Jon Block

Product Review: cfx_kmSuite from Hopkins Technology, LLC by Carey Lilly

ColdFusion Developer's Journal

Don't miss the June issue!